

# **Polyspace Bug Finder**

**Detailed Report for Project: mpu6500**

**Report Author: LibDriver**

**Polyspace Bug Finder: Detailed Report for Project: mpu6500**

by Report Author: LibDriver

**Published 01-Jul-2024 15:18:08**

Analysis Author(s): LibDriver

Polyspace Version(s): Polyspace Bug Finder 3.2 (R2020a)

Project Version(s): 1.0

Result Folder(s):

E:\Polyspace\mpu6500\Module\BF\_Result

## Table of Contents

Chapter 1. Polyspace Bug Finder Summary.....	1
Chapter 2. MISRA C:2012 Guidelines.....	3
<b>MISRA C:2012 Guidelines Summary - Violations by File .....</b>	<b>3</b>
<b>MISRA C:2012 Guidelines Violations .....</b>	<b>3</b>
Chapter 3. Defects .....	361
<b>Defects.....</b>	<b>361</b>
Chapter 4. Appendix 1 - Configuration Settings.....	362
<b>Polyspace Settings.....</b>	<b>362</b>
<b>Coding Standard Configuration.....</b>	<b>363</b>
Chapter 5. Appendix 2 - Definitions.....	371

# Chapter 1. Polyspace Bug Finder Summary

**Table 1.1. Project Summary**

	Count	Reviewed	Unreviewed	Pass/Fail
MISRA C:2012 Guidelines	3737	3737	0	Pass
Defects	0	0	0	Pass
<b>Total</b>	<b>3737</b>	<b>3737</b>	<b>0</b>	<b>Pass</b>

**Table 1.2. Summary By File**

File	Defects (Reviewed)	MISRA C:2012 Guidelines (Reviewed)
E:\Github\mpu6500\example\driver_mpu6500_basic.c	0 (0)	56 (56)
E:\Github\mpu6500\example\driver_mpu6500_basic.h	0 (0)	0 (0)
E:\Github\mpu6500\example\driver_mpu6500_dmp.c	0 (0)	96 (96)
E:\Github\mpu6500\example\driver_mpu6500_dmp.h	0 (0)	1 (1)
E:\Github\mpu6500\example\driver_mpu6500_fifo.c	0 (0)	55 (55)
E:\Github\mpu6500\example\driver_mpu6500_fifo.h	0 (0)	0 (0)
E:\Github\mpu6500\interface\driver_mpu6500_interface.h	0 (0)	0 (0)
E:\Github\mpu6500\interface\driver_mpu6500_interface_template.c	0 (0)	33 (33)
E:\Github\mpu6500\src\driver_mpu6500.c	0 (0)	1477 (1477)
E:\Github\mpu6500\src\driver_mpu6500.h	0 (0)	5 (5)
E:\Github\mpu6500\src\driver_mpu6500_code.h	0 (0)	0 (0)
E:\Github\mpu6500\test\driver_mpu6500_dmp_pedometer_test.c	0 (0)	105 (105)
E:\Github\mpu6500\test\driver_mpu6500_dmp_pedometer_test.h	0 (0)	0 (0)

E:\Github\mpu6500\test\driver_mpu6500_dmp_read_test.c	0 (0)	219 (219)
E:\Github\mpu6500\test\driver_mpu6500_dmp_read_test.h	0 (0)	0 (0)
E:\Github\mpu6500\test\driver_mpu6500_dmp_tap_orient_motion_test.c	0 (0)	276 (276)
E:\Github\mpu6500\test\driver_mpu6500_dmp_tap_orient_motion_test.h	0 (0)	0 (0)
E:\Github\mpu6500\test\driver_mpu6500_fifo_test.c	0 (0)	78 (78)
E:\Github\mpu6500\test\driver_mpu6500_fifo_test.h	0 (0)	0 (0)
E:\Github\mpu6500\test\driver_mpu6500_read_test.c	0 (0)	133 (133)
E:\Github\mpu6500\test\driver_mpu6500_read_test.h	0 (0)	0 (0)
E:\Github\mpu6500\test\driver_mpu6500_register_test.c	0 (0)	1203 (1203)
E:\Github\mpu6500\test\driver_mpu6500_register_test.h	0 (0)	0 (0)

# Chapter 2. MISRA C:2012 Guidelines

## MISRA C:2012 Guidelines Summary - Violations by File

File	Total
E:\Github\mpu6500\example\driver_mpu6500_basic.c	56
E:\Github\mpu6500\example\driver_mpu6500_dmp.c	96
E:\Github\mpu6500\example\driver_mpu6500_dmp.h	1
E:\Github\mpu6500\example\driver_mpu6500_fifo.c	55
E:\Github\mpu6500\interface\driver_mpu6500_interface_template.c	33
E:\Github\mpu6500\src\driver_mpu6500.c	1477
E:\Github\mpu6500\src\driver_mpu6500.h	5
E:\Github\mpu6500\test\driver_mpu6500_dmp_pedometer_test.c	105
E:\Github\mpu6500\test\driver_mpu6500_dmp_read_test.c	219
E:\Github\mpu6500\test\driver_mpu6500_dmp_tap_orient_motion_test.c	276
E:\Github\mpu6500\test\driver_mpu6500_fifo_test.c	78
E:\Github\mpu6500\test\driver_mpu6500_read_test.c	133
E:\Github\mpu6500\test\driver_mpu6500_register_test.c	1203
<b>Total</b>	<b>3737</b>

## MISRA C:2012 Guidelines Violations

Table 2.1. E:\Github\mpu6500\example\driver\_mpu6500\_basic.c

ID	Guideline	Message	Function	Severity	Status	Comment
1802	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1860	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1730	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1709	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1748	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1931	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2011	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1740	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3321	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1819	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3604	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1739	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1744	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1812	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2123	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3265	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2293	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3307	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2755	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1800	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3217	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1722	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3651	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1795	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2079	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3615	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1792	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1726	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1791	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3177	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1773	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3327	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1834	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3520	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2551	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2468	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1784	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1794	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1771	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3180	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2508	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3008	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2221	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2302	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2211	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1605	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1660	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3502	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1838	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1807	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1765	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3134	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3370	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3446	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3726	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.	mpu6500_basic_read()	Low	Justified	(handle == NULL)checked.
3727	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.	mpu6500_basic_read()	Low	Justified	(handle == NULL)checked.

**Table 2.2. E:\Github\mpu6500\example\driver\_mpu6500\_dmp.c**

ID	Guideline	Message	Function	Severity	Status	Comment
2866	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1782	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3658	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1774	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2206	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1758	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2637	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3309	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1763	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2787	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2505	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1953	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1697	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1761	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1752	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2929	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3101	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1622	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1715	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1737	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2246	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1743	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2373	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2827	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3089	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2842	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1682	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3436	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2023	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1631	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1630	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2567	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1767	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1714	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3541	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3021	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2189	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2197	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2635	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3479	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3693	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2553	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2939	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3627	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3642	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1684	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1707	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1624	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1688	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1671	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2527	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1927	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3169	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3550	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1716	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3453	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1941	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1691	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1675	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3384	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2018	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1656	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1672	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3712	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2108	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1689	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
6	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
11	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
4	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee

		right operand has essentially enum type.				the safety of the operation.
5	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
9	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
8	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
10	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
7	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
12	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3406	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3190	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1818	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1665	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2960	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3084	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1653	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1732	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3565	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1642	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1678	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1690	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2416	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3225	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2218	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1614	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1754	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3119	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2426	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

Table 2.3. E:\Github\mpu6500\example\driver\_mpu6500\_dmp.h

ID	Guideline	Message	Function	Severity	Status	Comment
----	-----------	---------	----------	----------	--------	---------

1	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (floating)	File Scope	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
---	------	--	------------	-----	--------------	--

**Table 2.4. E:\Github\mpu6500\example\driver\_mpu6500\_fifo.c**

ID	Guideline	Message	Function	Severity	Status	Comment
2014	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2225	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2821	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3716	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2168	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2170	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2157	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1854	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1729	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2167	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2171	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3433	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3488	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2839	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1862	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3138	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2029	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1906	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2802	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1972	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3046	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2163	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1872	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2161	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2177	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1789	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2648	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3569	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1943	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2823	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2633	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2153	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2148	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3708	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1801	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2160	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2711	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3471	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2451	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3630	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2174	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2183	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1853	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2179	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1659	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3206	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2142	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1772	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3117	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2106	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1786	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2720	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1997	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3387	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2208	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

**Table 2.5. E:\Github\mpu6500\interface\driver\_mpu6500\_interface\_template.c**

ID	Guideline	Message	Function	Severity	Status	Comment
1562	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3374	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1555	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2076	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1556	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2448	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1564	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2432	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1557	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3437	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2115	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1565	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2441	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1559	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1910	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1560	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1825	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1563	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2445	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1566	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2636	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1567	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2890	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2443	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1558	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the

						operation.
2440	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1561	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3295	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1568	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2685	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1569	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_interface_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2828	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3656	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

Table 2.6. E:\Github\mpu6500\src\driver\_mpu6500.c

ID	Guideline	Message	Function	Severity	Status	Comment
962	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	File Scope	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
502	10.1	Operands shall not be of an inappropriate essential type.	File Scope	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1099	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	File Scope	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
326	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.	a_mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
17	10.1	Operands shall not be of an inappropriate essential type. The right operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
832	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially unsigned type while the right operand has essentially signed type.	a_mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
359	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.	a_mpu6500_write()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
38	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially signed type.	a_mpu6500_write()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
173	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_write()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
33	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	a_mpu6500_write()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1013	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
22	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
117	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1439	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
654	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1440	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed.  The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1330	10.1	Operands shall not be of an inappropriate essential type.	a_mpu6500_write_mem()	Low	Not a	Embedded drivers need

		The operand of the ~ operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
529	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
60	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	a_mpu6500_write_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
71	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
37	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
57	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
76	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
679	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the == operator has essentially unsigned type	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially enum type.				safety of the operation.
28	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
79	10.1	Operands shall not be of an inappropriate essential type. The ~ operand of the ~ operator is of an inappropriate essential type category signed.	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
287	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_read_mem()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
88	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
958	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1104	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
913	10.1	Operands shall not be of an inappropriate essential type. The ~ operand of the ~ operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
317	10.1	Operands shall not be of an inappropriate essential type.	a_mpu6500_reset_fifo()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
64	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
806	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1365	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
92	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
82	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
110	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
684	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
1427	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
116	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1078	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
552	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1181	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
306	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
131	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
212	10.4	Both operands of an operator in which the usual arithmetic	a_mpu6500_reset_fifo()	Low	Not a	Embedded drivers need

		<p>conversions are performed shall have the same essential type category.</p> <p>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.</p>			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
557	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
41	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the   operator is of an inappropriate essential type category signed.</p> <p>The right operand of the   operator is of an inappropriate essential type category signed.</p>	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
428	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
48	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The right operand of the  = operator is of an inappropriate essential type category signed.</p>	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
453	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.</p>	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
364	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_reset_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
851	10.3	<p>The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.</p> <p>The expression (of essential type category signed) is assigned to</p>	a_mpu6500_dmp_decode_gesture()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		an object with a different essential type category (unsigned)				safety of the operation.
725	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially signed type.	a_mpu6500_dmp_decode_gesture()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
773	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_dmp_decode_gesture()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
145	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_dmp_decode_gesture()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
181	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_dmp_decode_gesture()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1131	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	a_mpu6500_dmp_decode_gesture()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
111	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	a_mpu6500_accel_self_test()	Low	Justified	Can't be.
392	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating.	a_mpu6500_accel_self_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of

						the operation.
115	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	a_mpu6500_accel_self_test()	Low	Justified	Can't be.
122	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating.	a_mpu6500_accel_self_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
374	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	a_mpu6500_accel_self_test()	Low	Justified	Can't be.
45	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (floating)	a_mpu6500_gyro_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
47	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating.	a_mpu6500_gyro_self_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
99	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	a_mpu6500_gyro_self_test()	Low	Justified	Can't be.
152	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an	a_mpu6500_gyro_self_test()	Low	Justified	Can't be.

		implementation-defined direction of rounding in some cases.				
554	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating.	a_mpu6500_gyro_self_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
113	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.  Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	a_mpu6500_gyro_self_test()	Low	Justified	Can't be.
1252	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
46	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.  The right operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
155	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
27	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.  The right operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
158	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.  The right operand of the   operator is of an inappropriate essential	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		type category signed.				safety of the operation.
98	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1406	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
162	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1190	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
120	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
169	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
817	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
174	10.1	Operands shall not be of an inappropriate essential type.	a_mpu6500_get_st_biases()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
182	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
204	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
542	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
413	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
816	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
383	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
121	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		the right operand has essentially unsigned type.				safety of the operation.
585	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1091	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
161	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1044	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
185	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
629	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the / operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
345	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
148	10.4	Both operands of an operator in which the usual arithmetic	a_mpu6500_get_st_biases()	Low	Not a	Embedded drivers need

		<p>conversions are performed shall have the same essential type category.</p> <p>The left operand of the / operator has essentially signed type while the right operand has essentially unsigned type.</p>			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
336	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1068	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the / operator has essentially signed type while the right operand has essentially unsigned type.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
78	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1117	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the / operator has essentially signed type while the right operand has essentially unsigned type.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
119	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
199	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the / operator has essentially signed type while the right operand has essentially unsigned type.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
206	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The left operand of the &lt;&lt; operator is of an inappropriate essential type category signed.</p>	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
125	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially signed type while the right operand has essentially unsigned type.	a_mpu6500_get_st_biases()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
136	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.	a_mpu6500_deinit()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
207	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_load_firmware()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
101	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_load_firmware()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
143	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_load_firmware()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
134	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_load_firmware()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1000	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_walk_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
55	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_set_pedometer_walk_time()	Low	Not a	Embedded drivers need

		The right operand of the & operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1141	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_walk_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
209	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_walk_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
230	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_step_count()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
216	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_step_count()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
747	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_step_count()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1242	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_pedometer_step_count()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3735	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source.	mpu6500_dmp_get_pedometer_step_count()	Low	Justified	(handle == NULL)checked.

		Pointer may be NULL or may point to unknown memory.				
89	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_shake_reject_timeout()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
132	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_timeout()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
197	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_shake_reject_timeout()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1468	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_timeout()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2310	5.1	External identifiers shall be distinct. External function mpu6500_dmp_set_shake_reject_time conflicts with the external identifier mpu6500_dmp_set_shake_reject_timeout (driver_mpu6500.c line 1269).	File Scope	Low	Justified	Be distinct.
218	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_shake_reject_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
217	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
214	10.3	The value of an expression shall not be assigned to an object with	mpu6500_dmp_set_shake_reject_time()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
221	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2006	5.1	External identifiers shall be distinct. External function mpu6500_dmp_get_shake_reject_time conflicts with the external identifier mpu6500_dmp_get_shake_reject_timeout (driver_mpu6500.c line 1316).	File Scope	Low	Justified	Be distinct.
409	10.6	The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 16 bits) is assigned to an object with a wider essential type (unsigned on 32 bits)	mpu6500_dmp_set_shake_reject_thresh()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
224	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
523	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
167	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
225	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_shake_reject_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
231	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_get_shake_reject_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
892	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_get_shake_reject_thresh()	Low	Justified	Can't be.
1387	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_get_shake_reject_thresh()	Low	Justified	Can't be.
902	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_tap_time_multi()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
234	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_time_multi()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
647	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_tap_time_multi()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
238	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_time_multi()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

789	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_tap_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
239	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
242	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_tap_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
30	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_time()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
252	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the != operator has essentially enum type while the right operand has essentially signed type.	mpu6500_dmp_set_gyro_calibrate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
163	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_gyro_calibrate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1206	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_gyro_calibrate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
246	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate	mpu6500_dmp_set_gyro_calibrate()	Low	Not a defect	Embedded drivers need this method to set or

		essential type category enum.				clear some bits and drivers guarantee the safety of the operation.
250	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_gyro_calibrate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
229	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_gyro_calibrate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
598	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the != operator has essentially enum type while the right operand has essentially signed type.	mpu6500_dmp_set_3x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
256	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_3x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
258	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_3x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
259	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_3x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
262	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_3x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
164	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_3x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
669	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the != operator has essentially enum type while the right operand has essentially signed type.	mpu6500_dmp_set_6x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
32	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_6x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
796	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_6x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
85	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_6x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
641	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_6x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
226	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_6x_quaternion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
271	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_set_gyro_bias()	Low	Not a	Embedded drivers need

		The right operand of the & operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1084	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1060	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
273	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
275	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
278	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1281	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
284	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
605	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1050	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
237	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
828	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
385	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
96	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1338	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
673	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1325	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
389	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
632	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
293	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
177	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver

		The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.				configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
474	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
527	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1444	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
297	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
971	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
441	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming

		unsigned.					method and should be accepted and drivers guarantee the safety of the operation.
512	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
394	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1231	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
299	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
310	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
304	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and	

						drivers guarantee the safety of the operation.
308	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
829	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
366	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
338	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
157	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
514	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

290	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
321	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
320	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_gyro_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
298	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
53	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
86	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
133	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

956	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
322	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1248	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1430	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
324	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1114	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1093	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
325	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or

		type category signed.				clear some bits and drivers guarantee the safety of the operation.
859	10.1	Operands shall not be of an inappropriate essential type. The left operand of the <code>&gt;&gt;</code> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
327	10.1	Operands shall not be of an inappropriate essential type. The left operand of the <code>&gt;&gt;</code> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
592	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1219	10.1	Operands shall not be of an inappropriate essential type. The left operand of the <code>&gt;&gt;</code> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
548	10.1	Operands shall not be of an inappropriate essential type. The left operand of the <code>&amp;</code> operator is of an inappropriate essential type category signed. The right operand of the <code>&amp;</code> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1247	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.

							the operation.
1189	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
344	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
203	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
328	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
29	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
683	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	

681	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
191	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
178	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1022	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
302	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1223	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
253	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or

		type category signed. The right operand of the & operator is of an inappropriate essential type category signed.				clear some bits and drivers guarantee the safety of the operation.
518	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
505	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
127	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
334	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
333	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
997	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming

		unsigned.					method and should be accepted and drivers guarantee the safety of the operation.
492	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
137	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
929	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
424	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
839	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
288	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers	

						guarantee the safety of the operation.
277	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1149	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
144	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
35	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_accel_bias()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
863	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1419	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
867	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and

		type category signed.				drivers guarantee the safety of the operation.
1351	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
490	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
874	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
337	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
343	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
462	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
348	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

943	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
289	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1232	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
50	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
429	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_orientation()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
435	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
83	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
985	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or

		type category signed.				clear some bits and drivers guarantee the safety of the operation.
1414	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
672	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1303	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
661	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
459	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
589	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
354	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
696	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
819	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
180	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1126	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
360	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
154	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
879	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
405	10.4	Both operands of an operator in which the usual arithmetic	mpu6500_dmp_set_feature()	Low	Not a	Embedded drivers need

		<p>conversions are performed shall have the same essential type category.</p> <p>The left operand of the &amp; operator has essentially unsigned type while the right operand has essentially enum type.</p>			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1012	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The right operand of the &amp; operator is of an inappropriate essential type category enum.</p>	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
323	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the / operator has essentially floating type while the right operand has essentially signed type.</p>	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
170	10.1	<p>Operands shall not be of an inappropriate essential type.</p> <p>The right operand of the &amp; operator is of an inappropriate essential type category signed.</p>	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
583	10.8	<p>The value of a composite expression shall not be cast to a different essential type category or a wider essential type.</p> <p>The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.</p>	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
372	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the * operator has essentially floating type while the right operand has essentially signed type.</p>	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
697	10.8	<p>The value of a composite expression shall not be cast to a different essential type category or a wider essential type.</p> <p>The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.</p>	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers

						guarantee the safety of the operation.
362	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
565	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
365	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
367	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
196	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
368	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be

						accepted and drivers guarantee the safety of the operation.
820	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
481	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1021	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
380	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
961	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
655	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category	mpu6500_dmp_set_feature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming

		unsigned.				method and should be accepted and drivers guarantee the safety of the operation.
200	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1369	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
340	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1473	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.  The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1259	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1345	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.  The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
838	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
390	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
384	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
391	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
480	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
398	10.6	The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 16 bits) is assigned to an object with a wider essential type (unsigned on 32 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
848	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
417	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
142	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_set_feature()	Low	Not a	Embedded drivers need

		The right operand of the & operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
249	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
461	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
25	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
400	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
91	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
176	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1076	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
401	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
909	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
408	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
860	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
410	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
628	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
97	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
414	10.4	Both operands of an operator in which the usual arithmetic	mpu6500_dmp_set_feature()	Low	Not a	Embedded drivers need

		conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
67	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
244	10.1	Operands shall not be of an inappropriate essential type. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
767	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_feature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
254	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_fifo_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
418	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_fifo_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
425	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.

420	10.1	Operands shall not be of an inappropriate essential type. The left operand of the - operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
421	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the - operator has essentially enum type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
431	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
483	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1165	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
427	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
843	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1176	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or

		category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.				clear some bits and drivers guarantee the safety of the operation.
1394	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
883	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1140	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
799	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_dmp_get_tap_axes()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
433	10.1	Operands shall not be of an inappropriate essential type. The left operand of the - operator is of an inappropriate essential type category enum.	mpu6500_dmp_get_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
434	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the - operator has essentially enum type while the right operand has essentially signed type.	mpu6500_dmp_get_tap_axes()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
436	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential	mpu6500_dmp_get_tap_axes()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
437	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1435	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
439	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
974	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
445	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
836	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
319	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
456	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
94	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
403	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1125	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
983	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and

		The left operand of the * operator has essentially floating type while the right operand has essentially signed type.				drivers guarantee the safety of the operation.
465	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
862	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
443	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1020	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
475	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
531	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or

		category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.				clear some bits and drivers guarantee the safety of the operation.
232	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
444	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
604	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
594	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially unsigned type while the right operand has essentially floating type.	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1296	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially unsigned type while the right operand has essentially floating type.	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
245	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially unsigned type while the right operand has essentially floating type.	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
735	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially unsigned type	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially floating type.				safety of the operation.
452	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1200	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially floating type while the right operand has essentially signed type.	mpu6500_dmp_get_tap_thresh()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1377	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
699	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
928	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1263	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
676	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
187	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1201	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
455	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
396	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1220	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
451	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
967	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
485	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_read()	Low	Not a	Embedded drivers need

		The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1373	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
458	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
703	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
351	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
411	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
460	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
466	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		type category signed.				safety of the operation.
464	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1040	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
658	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1100	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1158	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1154	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
469	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
468	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_read()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
381	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
393	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
741	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
923	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
205	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
572	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1386	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
19	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
446	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
476	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
559	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
388	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
477	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
478	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1215	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_read()	Low	Not a	Embedded drivers need

		The left operand of the >> operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
600	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_read()	Low	Justified	Can't be.
58	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
591	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_read()	Low	Justified	Can't be.
482	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1337	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_read()	Low	Justified	Can't be.
266	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1412	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_read()	Low	Justified	Can't be.
541	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or

		category. The left operand of the / operator has essentially signed type while the right operand has essentially floating type.				clear some bits and drivers guarantee the safety of the operation.
528	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
484	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
486	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1405	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
282	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
422	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
263	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the + operator has essentially floating type	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
564	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
43	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
660	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1399	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
579	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
544	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
711	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

692	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
489	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
309	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
168	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
495	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1061	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
449	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.

		bits) which is a composite expression				
582	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
261	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
488	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
811	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type.  The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
1327	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
355	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
80	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type.  The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.

		bits) which is a composite expression				
370	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1062	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1217	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
500	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
350	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
506	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
95	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		the right operand has essentially unsigned type.				safety of the operation.
447	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1122	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
159	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8 bits) which is a composite expression	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.
508	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
36	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
329	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
707	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The right operand of the + operator shall not have wider essential type (unsigned on 16 bits) than the left operand (unsigned on 8	mpu6500_dmp_read()	Low	Not a defect	We use this function to convert driver data and drivers guarantee the safety of the operation.

		bits) which is a composite expression				
1033	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
510	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
646	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
272	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
279	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1384	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
375	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
842	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_set_enable()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
267	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
664	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1030	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
586	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_dmp_set_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
511	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
39	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
515	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming

		signed.				method and should be accepted and drivers guarantee the safety of the operation.
341	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
192	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1088	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
769	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
21	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
521	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.

440	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
44	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
62	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1275	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
395	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
442	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1241	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.

525	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
749	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
139	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
905	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
184	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
973	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
526	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		the right operand has essentially floating type.				safety of the operation.
620	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1137	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.  Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
536	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
281	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
87	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.  Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
126	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
270	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a

		floating shall not be cast to the different essential type category signed.				friendly programming method and should be accepted and drivers guarantee the safety of the operation.
965	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
530	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
537	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
981	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
533	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
969	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of

							the operation.
519	D1.1	<p>Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.</p> <p>Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.	
147	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the * operator has essentially signed type while the right operand has essentially floating type.</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
202	10.8	<p>The value of a composite expression shall not be cast to a different essential type category or a wider essential type.</p> <p>The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
540	D1.1	<p>Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.</p> <p>Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.	
1342	10.4	<p>Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.</p> <p>The left operand of the * operator has essentially signed type while the right operand has essentially floating type.</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
608	10.8	<p>The value of a composite expression shall not be cast to a different essential type category or a wider essential type.</p> <p>The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
128	D1.1	<p>Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.</p> <p>Conversion of integer to floating-point number uses an</p>	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.	

		implementation-defined direction of rounding in some cases.				
543	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
785	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
546	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
549	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1464	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
555	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
172	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and

		The left operand of the * operator has essentially signed type while the right operand has essentially floating type.				drivers guarantee the safety of the operation.
852	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
556	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
756	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
65	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
846	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
40	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
471	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a	We use enumeration to define driver

		The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.			defect	configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
903	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
562	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
854	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
369	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
1418	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
567	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers

						guarantee the safety of the operation.
563	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
566	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1474	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1333	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
1370	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
493	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
352	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.

		Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.				
1307	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
573	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
570	D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Justified	Can't be.
1080	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the * operator has essentially signed type while the right operand has essentially floating type.	mpu6500_dmp_gyro_accel_raw_offset_convert()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
42	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
934	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
358	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the & operator has essentially unsigned type	mpu6500_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
575	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
574	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_init()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
810	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_deinit()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1413	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_deinit()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
877	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_deinit()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
107	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category signed.	mpu6500_deinit()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
576	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_deinit()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3736	D4.14	The validity of values received from external sources shall be	mpu6500_read()	Low	Justified	(handle ==

		checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.				NULL)checked.
577	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
580	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
621	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
81	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1015	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
581	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
639	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_read()	Low	Not a	Embedded drivers need this method to set or

		category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.			defect	clear some bits and drivers guarantee the safety of the operation.
1054	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
112	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
736	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
727	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1228	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1466	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		the right operand has essentially unsigned type.				safety of the operation.
1347	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
124	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
569	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
285	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
593	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
596	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

597	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()		Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
248	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
601	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1235	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()		Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
18	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
603	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1335	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	mpu6500_read()		Low	Not a defect	We use enumeration to define driver

		The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.				configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
387	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
571	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
211	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
606	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1283	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
611	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category	mpu6500_read()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming

		signed.					method and should be accepted and drivers guarantee the safety of the operation.
612	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
613	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
614	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()		Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
617	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
619	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()		Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1172	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read()		Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers

						guarantee the safety of the operation.
522	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1409	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
622	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_read_temperature()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
419	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_read_temperature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
494	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the   operator has essentially signed type while the right operand has essentially unsigned type.	mpu6500_read_temperature()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3737	D4.14	The validity of values received from external sources shall be checked.  Dereferenced pointer is from an unsecure source.  Pointer may be NULL or may point to unknown memory.	mpu6500_read_temperature()	Low	Justified	(handle == NULL)checked.
624	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
623	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
950	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
106	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1179	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
517	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1055	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
227	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1047	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_irq_handler()	Low	Not a	Embedded drivers need

		The right operand of the & operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
627	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_irq_handler()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
470	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
595	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
631	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1192	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
630	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
233	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		an object with a different essential type category (unsigned)				safety of the operation.
633	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
635	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
499	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1174	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_fifo()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
976	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_fifo()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
90	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
599	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1123	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1476	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
802	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
763	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
776	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1204	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
643	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
650	10.8	The value of a composite expression shall not be cast to a	mpu6500_get_iic_master()	Low	Not a	We use enumeration to

		different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.			defect	define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1426	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
651	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
652	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
657	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1017	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
102	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

560	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
653	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1233	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
463	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
778	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_disable_iic_slave()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
292	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_disable_iic_slave()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
535	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

607	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
656	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
123	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1262	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
280	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
662	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1073	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
945	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	mpu6500_get_fifo_reset()	Low	Not a defect	We use enumeration to define driver

		The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.				configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1212	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_fifo_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
491	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
667	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
871	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
215	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
666	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
307	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_iic_master_reset()	Low	Not a	Embedded drivers need this method to set or

		category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.			defect	clear some bits and drivers guarantee the safety of the operation.
670	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
114	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
671	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_master_reset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1119	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_master_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
75	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
558	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
744	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_sensor_reset()	Low	Not a	Embedded drivers need this method to set or

		The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	clear some bits and drivers guarantee the safety of the operation.
294	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
953	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
675	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
680	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1442	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_sensor_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
363	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_sensor_reset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1177	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential	mpu6500_get_sensor_reset()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
171	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
407	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1188	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
682	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1273	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
700	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1225	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1286	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
264	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_device_reset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
301	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_device_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
952	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1007	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1306	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
509	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
361	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
193	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
739	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
760	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
686	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
687	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_clock_source()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
638	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_clock_source()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
296	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
691	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1052	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
649	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1450	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
977	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1053	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1277	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_ptat()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
873	10.1	Operands shall not be of an inappropriate essential type. The operand of the ! operator is of an inappropriate essential type category enum.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
698	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category Boolean.	mpu6500_set_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1446	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category Boolean shall not be cast to the different essential type category enum.	mpu6500_get_ptat()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1401	10.1	Operands shall not be of an inappropriate essential type. The operand of the ! operator is of an inappropriate essential type category unsigned.	mpu6500_get_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1371	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_ptat()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
52	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1086	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1309	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
710	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1226	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
553	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
713	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
986	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
51	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential	mpu6500_set_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or

		type category enum.				clear some bits and drivers guarantee the safety of the operation.
412	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_cycle_wake_up()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
714	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_cycle_wake_up()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
821	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1110	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1202	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1094	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
865	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential	mpu6500_set_sleep()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
634	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
715	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
963	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
316	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1156	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_sleep()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
195	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_sleep()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
295	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_set_gyro_standby()	Low	Not a	Embedded drivers need this method to set or

		The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	clear some bits and drivers guarantee the safety of the operation.
717	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
718	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1195	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
318	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
236	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
951	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
968	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially enum type.				safety of the operation.
371	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
719	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_gyro_standby()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
826	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_gyro_standby()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
118	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
251	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
809	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1005	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1292	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
432	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
932	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1098	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
723	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1011	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_standby_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
228	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
720	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_standby_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
659	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
724	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1267	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1036	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1432	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
726	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
989	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_signal_path_reset()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1193	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1462	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_signal_path_reset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
283	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
377	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
695	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
704	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
732	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
907	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1291	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1383	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
503	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
939	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_extern_sync()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
179	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_extern_sync()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
782	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
1155	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1389	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
331	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
937	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
104	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
738	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
869	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
737	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_set_low_pass_filter()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category enum.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
702	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_low_pass_filter()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1208	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
740	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1107	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1216	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
751	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1448	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
743	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
766	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1300	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
587	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
750	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_fifo_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1431	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_fifo_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

141	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
274	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
520	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
588	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
276	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
220	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
265	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
752	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or

		The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)				clear some bits and drivers guarantee the safety of the operation.
748	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
149	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_gyroscope_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
753	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
757	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_gyroscope_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
524	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
837	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1260	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_set_gyroscope_range()	Low	Not a	Embedded drivers need this method to set or

		category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.			defect	clear some bits and drivers guarantee the safety of the operation.
933	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
551	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
378	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
872	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1472	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
818	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
787	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category	mpu6500_get_gyroscope_range()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be

		enum.				accepted and drivers guarantee the safety of the operation.
844	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_gyroscope_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
423	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_gyroscope_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
764	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
765	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyroscope_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
584	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1355	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_gyroscope_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
988	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_gyroscope_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
59	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
793	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1028	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
550	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
685	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
438	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
677	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
770	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_accelerometer_test()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
590	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
870	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_accelerometer_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
772	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
774	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_accelerometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
382	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
779	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1315	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1010	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1422	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
788	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
876	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1045	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
190	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
804	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	mpu6500_get_accelerometer_range()	Low	Not a defect	We use enumeration to define driver

		The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.				configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
223	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_accelerometer_range()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
31	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
625	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
978	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
784	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
858	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
23	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate	mpu6500_set_fifo_1024kb()	Low	Not a	Embedded drivers need this method to set or

		essential type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
1182	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
791	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_1024kb()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
166	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
353	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_accelerometer_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
415	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
795	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
406	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1227	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_accelerometer_choice()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
74	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
160	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1304	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
153	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1105	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
701	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
798	10.4	Both operands of an operator in which the usual arithmetic	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a	Embedded drivers need

		conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
800	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
893	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1001	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_accelerometer_low_pass_filter()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
803	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_accelerometer_low_pass_filter()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
349	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
357	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1138	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1407	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
797	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
100	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
347	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
545	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1085	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
781	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	mpu6500_get_low_power_accel_output_rate()	Low	Not a defect	We use enumeration to define driver

		The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.				configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
568	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_low_power_accel_output_rate()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
286	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
761	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1095	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
448	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
34	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
745	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate	mpu6500_set_wake_on_motion()	Low	Not a	Embedded drivers need this method to set or

		essential type category enum.			defect	clear some bits and drivers guarantee the safety of the operation.
812	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1410	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
538	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1029	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_wake_on_motion()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
151	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_wake_on_motion()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
948	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1004	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a	Embedded drivers need this method to set or

		essential type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
1210	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
668	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
825	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
257	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
457	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
794	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
66	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
833	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_accel_compare_with_previous_sample()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1433	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_accel_compare_with_previous_sample()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
886	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
947	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1167	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1451	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
807	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		essential type category enum.				safety of the operation.
189	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
618	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1366	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
674	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
830	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_fifo_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
386	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
642	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
733	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
849	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
850	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
996	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
834	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
831	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
853	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1332	10.4	Both operands of an operator in which the usual arithmetic	mpu6500_set_interrupt_level()	Low	Not a	Embedded drivers need

		conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
130	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
314	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_interrupt_level()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
547	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
501	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
709	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
855	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1298	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1326	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
644	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
742	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
775	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
790	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
496	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_interrupt_pin_type()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.

473	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_interrupt_pin_type()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
315	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
813	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
857	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
801	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1382	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
165	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
534	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or

		The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)				clear some bits and drivers guarantee the safety of the operation.
861	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1166	10.1	Operands shall not be of an inappropriate essential type. The operand of the ! operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
222	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category Boolean.	mpu6500_set_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
243	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category Boolean shall not be cast to the different essential type category enum.	mpu6500_get_interrupt_latch()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1428	10.1	Operands shall not be of an inappropriate essential type. The operand of the ! operator is of an inappropriate essential type category unsigned.	mpu6500_get_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
260	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_interrupt_latch()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
54	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_set_interrupt_read_clear()	Low	Not a	Embedded drivers need this method to set or

		The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	clear some bits and drivers guarantee the safety of the operation.
780	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
864	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1280	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1035	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
497	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
866	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1134	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1184	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
69	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_interrupt_read_clear()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1019	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_interrupt_read_clear()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
240	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
665	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1363	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
868	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1256	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
291	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
467	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
875	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1186	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1074	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_fsync_interrupt_level()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
878	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_fsync_interrupt_level()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
208	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
884	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1070	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
882	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1458	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
880	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
917	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1391	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_fsync_interrupt()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
426	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
847	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_fsync_interrupt()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
689	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_fsync_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
762	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
840	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
889	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

888	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
186	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
894	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
896	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
925	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1441	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
898	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_bypass()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.

1253	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_bypass()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
822	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
899	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1459	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
373	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
915	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
70	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
824	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or

		category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.				clear some bits and drivers guarantee the safety of the operation.
1222	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
210	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
610	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_interrupt()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1322	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
49	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
138	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
219	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential	mpu6500_set_accelerometer_x_offset()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
690	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
561	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1016	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
900	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
887	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_get_accelerometer_x_offset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
910	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
906	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential	mpu6500_set_accelerometer_y_offset()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
908	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
912	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
911	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
783	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
891	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_get_accelerometer_y_offset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
916	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
479	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential	mpu6500_set_accelerometer_z_offset()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
914	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
103	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_accelerometer_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1218	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
68	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_accelerometer_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
920	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_get_accelerometer_z_offset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
924	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_accelerometer_offset_convert_to_register()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of

							the operation.
3688	5.1	External identifiers shall be distinct. External function mpu6500_accelerometer_offset_convert_to_data conflicts with the external identifier mpu6500_accelerometer_offset_convert_to_register (driver_mpu6500.c line 8367).	File Scope	Low	Justified	Be distinct.	
1127	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
20	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
926	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
927	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1128	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1278	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_x_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1397	10.8	The value of a composite expression shall not be cast to a	mpu6500_get_gyro_x_offset()	Low	Not a	We use enumeration to	

		different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.			defect	define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1404	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1162	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
935	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1385	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1120	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
663	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_y_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

936	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_get_gyro_y_offset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
109	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
814	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
402	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
940	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_gyro_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1290	10.1	Operands shall not be of an inappropriate essential type. The left operand of the >> operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1191	10.1	Operands shall not be of an inappropriate essential type. The left operand of the & operator is of an inappropriate essential type category signed. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_gyro_z_offset()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

942	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category signed.	mpu6500_get_gyro_z_offset()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
944	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category signed.	mpu6500_gyro_offset_convert_to_register()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
2832	5.1	External identifiers shall be distinct. External function mpu6500_gyro_offset_convert_to_data conflicts with the external identifier mpu6500_gyro_offset_convert_to_register (driver_mpu6500.c line 8646).	File Scope	Low	Justified	Be distinct.
1006	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category floating shall not be cast to the different essential type category unsigned.	mpu6500_motion_threshold_convert_to_register()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
3715	5.1	External identifiers shall be distinct. External function mpu6500_motion_threshold_convert_to_data conflicts with the external identifier mpu6500_motion_threshold_convert_to_register (driver_mpu6500.c line 8770).	File Scope	Low	Justified	Be distinct.
1009	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
949	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1027	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1118	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1270	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
827	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_self_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
269	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
835	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1065	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_iic_clock()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
955	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
841	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
957	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1319	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1024	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_clock()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
636	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_clock()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

342	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
960	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1108	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1146	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1103	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
72	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
376	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
845	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or

		essential type category enum.				clear some bits and drivers guarantee the safety of the operation.
964	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1362	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_multi_master()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
970	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_multi_master()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
93	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1343	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1443	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1102	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
1173	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
966	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1203	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1378	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
648	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_wait_for_external_sensor()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1115	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_wait_for_external_sensor()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
895	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential	mpu6500_get_iic_wait_for_external_sensor()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
140	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
975	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
979	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
194	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
972	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
930	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1331	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially enum type.				safety of the operation.
1360	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1274	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1257	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_read_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
931	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_read_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
808	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1042	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1308	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
980	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
105	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
982	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
984	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1207	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
921	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
993	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
994	10.4	Both operands of an operator in which the usual arithmetic	mpu6500_set_iic_fifo_enable()	Low	Not a	Embedded drivers need

		conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
998	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
990	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
146	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
999	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1143	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1359	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1163	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1072	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_fifo_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
705	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1003	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1014	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_fifo_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1051	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_fifo_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
332	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1023	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1034	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1018	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
454	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
312	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1026	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1436	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1025	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or

		type category enum.				clear some bits and drivers guarantee the safety of the operation.
1148	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1251	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1420	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1031	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
728	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1032	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1449	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1455	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
416	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
815	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1037	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1038	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1067	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
399	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
63	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_set_iic_mode()	Low	Not a	Embedded drivers need

		The right operand of the  = operator is of an inappropriate essential type category enum.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1039	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1318	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.  The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1429	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1048	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1049	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1358	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.  The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1043	10.1	Operands shall not be of an inappropriate essential type.  The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
311	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
247	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
507	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
786	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
938	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
339	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1130	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1452	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_iic_mode()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1372	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
904	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
129	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
992	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1059	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1364	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1109	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category	mpu6500_get_iic_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming

		enum.					method and should be accepted and drivers guarantee the safety of the operation.
1063	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1064	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
235	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1069	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
513	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1071	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category	mpu6500_get_iic_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a	

		unsigned shall not be cast to the different essential type category enum.				friendly programming method and should be accepted and drivers guarantee the safety of the operation.
716	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1244	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
954	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1077	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1089	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1353	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1075	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
539	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
754	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
777	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
881	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1079	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1083	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
84	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or

		essential type category signed.				clear some bits and drivers guarantee the safety of the operation.
1124	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1132	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1087	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1092	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
300	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1066	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1348	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
73	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
616	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1113	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1340	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1457	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1475	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1096	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_set_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1097	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_get_iic_address()	Low	Not a	Embedded drivers need

		The right operand of the & operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1196	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
135	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
734	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1106	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_address()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
397	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1112	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1151	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
1324	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1111	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
771	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1081	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1460	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1116	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1129	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1135	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_iic_enable()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)				defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1285	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1121	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1295	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
213	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
959	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1416	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1380	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the	

						safety of the operation.
746	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1160	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1454	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1145	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
183	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1142	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1170	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1368	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_iic_enable()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)				defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1139	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
626	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1180	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1294	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1144	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1301	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
897	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the	

		an object with a different essential type category (unsigned)				safety of the operation.
1398	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1461	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1411	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
721	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1147	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1152	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.

1150	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1159	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1438	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1268	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1272	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
335	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
640	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or

		essential type category signed.				clear some bits and drivers guarantee the safety of the operation.
995	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
731	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1424	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
61	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
305	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1153	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1336	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
759	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1175	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1254	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
516	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1194	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1157	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1161	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1467	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_set_iic_byte_swap()	Low	Not a	Embedded drivers need

		The right operand of the  = operator is of an inappropriate essential type category enum.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1239	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
823	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1311	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1329	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1437	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1002	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
885	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially enum type.				safety of the operation.
1164	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1168	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1349	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
609	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1041	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1171	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1169	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1249	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_set_iic_byte_swap()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
722	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1183	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1238	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
241	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1187	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_byte_swap()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1185	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1265	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_byte_swap()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1321	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1197	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_byte_swap()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
108	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1198	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_byte_swap()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1058	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_byte_swap()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1209	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1211	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1213	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1199	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
987	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
56	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1221	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1314	10.3	The value of an expression shall not be assigned to an object with	mpu6500_set_iic_transaction_mode()	Low	Not a	Embedded drivers need

		a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)				defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1214	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
901	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1056	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1350	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
532	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1224	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
688	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the	

		while the right operand has essentially enum type.				safety of the operation.
1230	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1234	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1229	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1236	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1243	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1271	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1008	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1279	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_set_iic_transaction_mode()	Low	Not a	Embedded drivers need

		The left operand of the << operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
77	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1082	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1479	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.  The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
268	10.1	Operands shall not be of an inappropriate essential type.  The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
472	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1237	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.  The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1266	10.1	Operands shall not be of an inappropriate essential type.  The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
303	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1245	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
313	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1246	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1456	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1240	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
805	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.  The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of

							the operation.
1250	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
645	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
1323	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
1434	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.	
1334	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.	
379	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers	

						guarantee the safety of the operation.
1481	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
694	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
712	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1046	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1379	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1255	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1261	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1339	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1465	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1258	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
255	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
602	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1390	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1352	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1346	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or

		type category signed.				clear some bits and drivers guarantee the safety of the operation.
946	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1264	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1276	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1470	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
615	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1393	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1421	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
1395	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
693	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
26	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1282	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1408	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1403	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
24	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
919	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_set_iic_group_order()	Low	Not a	Embedded drivers need

		The right operand of the &= operator is of an inappropriate essential type category signed.			defect	this method to set or clear some bits and drivers guarantee the safety of the operation.
1469	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1288	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
450	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
637	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
678	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1284	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1312	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1293	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_group_order()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1287	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1344	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_group_order()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1297	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
922	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_group_order()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1299	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and

						drivers guarantee the safety of the operation.
150	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_group_order()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1302	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_group_order()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
498	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
504	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
768	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1269	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
404	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and

		The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.				drivers guarantee the safety of the operation.
1313	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1357	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1396	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1310	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1316	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1375	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1305	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

430	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
729	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1328	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1320	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
201	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1388	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1341	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1090	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential	mpu6500_get_iic_transferred_len()	Low	Not a defect	Embedded drivers need this method to set or

		type category signed.				clear some bits and drivers guarantee the safety of the operation.
918	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1133	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1356	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
758	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1374	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
330	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1361	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

						safety of the operation.
1367	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1400	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum. The right operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
755	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic_delay_enable()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
890	10.1	Operands shall not be of an inappropriate essential type. The right operand of the >> operator is of an inappropriate essential type category enum.	mpu6500_get_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1376	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic_delay_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
156	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
188	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the

		while the right operand has essentially signed type.				safety of the operation.
706	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
941	10.1	Operands shall not be of an inappropriate essential type. The ~ operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
991	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
487	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1101	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1381	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
578	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
346	10.8	The value of a composite expression shall not be cast to a	mpu6500_get_iic4_enable()	Low	Not a	We use enumeration to

		different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.			defect	define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1205	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic4_enable()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
792	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1392	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1453	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1417	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
356	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

708	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1136	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1178	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1402	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category enum.	mpu6500_set_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
730	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic4_interrupt()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
1289	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic4_interrupt()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1415	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1447	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1471	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1057	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1423	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1354	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category enum.	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1425	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.  The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1445	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
856	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential	mpu6500_set_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or

		type category enum.				clear some bits and drivers guarantee the safety of the operation.
175	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.	mpu6500_get_iic4_transaction_mode()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
198	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_get_iic4_transaction_mode()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1317	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_set_iic_delay()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1463	10.1	Operands shall not be of an inappropriate essential type. The right operand of the &= operator is of an inappropriate essential type category signed.	mpu6500_set_iic_delay()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1478	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.	mpu6500_set_iic_delay()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1477	10.1	Operands shall not be of an inappropriate essential type. The operand of the ~ operator is of an inappropriate essential type category signed.	mpu6500_set_iic_delay()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1480	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential	mpu6500_get_iic_delay()	Low	Not a	Embedded drivers need this method to set or

		type category signed.			defect	clear some bits and drivers guarantee the safety of the operation.
3731	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.	mpu6500_get_iic4_data_in()	Low	Justified	(handle == NULL)checked.
3732	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.	mpu6500_read_extern_sensor_data()	Low	Justified	(handle == NULL)checked.
3733	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.	mpu6500_set_reg()	Low	Justified	(handle == NULL)checked.
3734	D4.14	The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory.	mpu6500_get_reg()	Low	Justified	(handle == NULL)checked.

Table 2.7. E:\Github\mpu6500\src\driver\_mpu6500.h

ID	Guideline	Message	Function	Severity	Status	Comment
3669	5.1	External identifiers shall be distinct. External function mpu6500_accelerometer_offset_convert_to_data conflicts with the external identifier mpu6500_accelerometer_offset_convert_to_register (driver_mpu6500.c line 8367).	File Scope	Low	Justified	Be distinct.
3659	5.1	External identifiers shall be distinct. External function mpu6500_gyro_offset_convert_to_data conflicts with the external identifier mpu6500_gyro_offset_convert_to_register (driver_mpu6500.c line 8646).	File Scope	Low	Justified	Be distinct.
1655	5.1	External identifiers shall be distinct. External function mpu6500_motion_threshold_convert_to_data conflicts with the external identifier mpu6500_motion_threshold_convert_to_register (driver_mpu6500.c line 8770).	File Scope	Low	Justified	Be distinct.

3633	5.1	External identifiers shall be distinct. External function mpu6500_dmp_set_shake_reject_time conflicts with the external identifier mpu6500_dmp_set_shake_reject_timeout (driver_mpu6500.c line 1269).	File Scope	Low	Justified	Be distinct.
2627	5.1	External identifiers shall be distinct. External function mpu6500_dmp_get_shake_reject_time conflicts with the external identifier mpu6500_dmp_get_shake_reject_timeout (driver_mpu6500.c line 1316).	File Scope	Low	Justified	Be distinct.

Table 2.8. E:\Github\mpu6500\test\driver\_mpu6500\_dmp\_pedometer\_test.c

ID	Guideline	Message	Function	Severity	Status	Comment
1698	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1883	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2384	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1939	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3573	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3469	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2021	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1922	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1983	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3299	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2729	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3682	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3149	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1952	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1897	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1658	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1930	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2369	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2073	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2710	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3188	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2618	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2433	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1775	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2703	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2403	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1903	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2114	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1918	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1757	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2762	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3111	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1899	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2175	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3109	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1902	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2059	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3415	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1876	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1895	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3512	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1647	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3212	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2928	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1925	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1888	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3097	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1628	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2287	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2130	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1998	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2088	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3619	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2329	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1869	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1868	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2461	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2886	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3596	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2210	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2289	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2095	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1945	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2566	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2888	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2357	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1856	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2695	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1703	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2408	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2285	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1848	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2383	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2954	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2144	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1914	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
13	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_dmp_pedometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some

		The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)				bits and drivers guarantee the safety of the operation.
2699	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1831	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2349	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1710	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1873	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2162	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2394	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2139	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3029	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1904	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1846	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2092	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
14	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_pedometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
15	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed.	mpu6500_dmp_pedometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee

		The right operand of the   operator is of an inappropriate essential type category enum.				the safety of the operation.
16	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_pedometer_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2779	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1837	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1769	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1830	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2535	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1843	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1632	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2752	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
3729	D4.14	The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite.	mpu6500_dmp_pedometer_test()	Low	Justified	Loop can't be infinite.
1833	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3430	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2003	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
3371	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

**Table 2.9. E:\Github\mpu6500\test\driver\_mpu6500\_dmp\_read\_test.c**

ID	Guideline	Message	Function	Severity	Status	Comment
1667	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1919	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3634	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2017	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2110	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2587	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2147	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1696	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2984	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1894	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2126	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1966	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1677	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2122	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1779	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.

2255	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2604	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1979	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1974	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2412	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2140	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3657	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3566	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2120	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1877	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3720	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1666	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2116	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2873	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1778	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2094	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2093	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2482	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2901	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2942	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1749	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1824	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3654	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2085	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2855	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3570	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2077	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2074	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2071	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2028	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3449	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1679	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2109	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2630	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2060	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1700	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2066	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2326	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2241	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1994	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1611	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1991	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2117	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2063	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2593	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3559	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1965	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3290	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2951	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3182	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2442	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2137	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1756	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2599	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2049	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2010	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2119	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2064	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2047	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2365	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1645	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3360	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1482	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1796	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1995	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2485	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3280	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2749	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2039	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2033	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2026	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1849	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1712	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2977	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1606	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2020	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2417	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2019	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2055	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3689	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2052	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3553	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1633	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2133	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2993	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2489	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2215	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2256	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1993	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3648	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1821	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2494	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3350	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2410	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2922	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2040	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1806	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1706	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2937	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1646	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2000	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1760	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2107	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3164	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2517	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2087	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2054	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2100	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1776	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2103	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3163	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2185	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3107	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1725	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1949	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3168	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2297	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1988	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2454	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2111	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1783	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1990	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1683	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1996	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2640	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2152	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3124	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1851	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1852	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2118	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2002	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1733	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1936	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3375	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2051	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1736	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1805	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1962	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1607	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1984	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1980	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3039	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1828	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1977	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1976	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2090	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2651	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1982	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1626	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2121	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3668	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2738	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1975	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3405	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1971	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1970	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2048	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1968	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2272	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3054	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2865	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3523	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1685	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3071	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1484	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1485	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1487	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1488	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1490	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1483	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1491	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1486	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1492	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1489	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1493	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_read_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1985	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1686	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2715	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2638	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1652	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1617	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2718	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1932	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1661	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2187	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1911	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1957	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2965	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2091	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2430	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3718	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
3730	D4.14	The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite.	mpu6500_dmp_read_test()	Low	Justified	Loop can't be infinite.
3025	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2141	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2102	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2050	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1946	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3207	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2358	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1886	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3617	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3646	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2692	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2946	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

**Table 2.10. E:\Github\mpu6500\test\driver\_mpu6500\_dmp\_tap\_orient\_motion\_test.c**

ID	Guideline	Message	Function	Severity	Status	Comment
1510	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1496	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1509	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1506	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2368	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1501	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2981	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1520	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2499	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1513	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1539	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_receive_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2364	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1508	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1499	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1500	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1498	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear

		category signed.				some bits and drivers guarantee the safety of the operation.
2362	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1517	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1538	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1548	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1552	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2317	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1495	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1507	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

1536	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1502	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1882	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1519	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1512	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1522	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1532	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1704	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1521	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers

		object with a different essential type category (unsigned)				guarantee the safety of the operation.
1511	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1515	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1514	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3126	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1523	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1541	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1546	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1531	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_tap_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers

						guarantee the safety of the operation.
3532	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1884	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1494	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1497	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1537	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1524	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2352	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1529	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1527	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the

		the right operand has essentially signed type.				operation.
1542	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1516	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3090	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1544	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1528	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1534	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1525	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2351	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1530	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear

		The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)				some bits and drivers guarantee the safety of the operation.
1533	10.1	Operands shall not be of an inappropriate essential type. The right operand of the  = operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1545	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1518	10.1	Operands shall not be of an inappropriate essential type. The left operand of the << operator is of an inappropriate essential type category signed.	a_dmp_orient_callback()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2681	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2067	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3717	5.1	External identifiers shall be distinct. External function mpu6500_dmp_tap_orient_motion_test conflicts with the external identifier mpu6500_dmp_tap_orient_motion_test_irq_handler (driver_mpu6500_dmp_tap_orient_motion_test.c line 58).	File Scope	Low	Justified	Be distinct.
1637	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2864	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2830	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2372	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2437	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2158	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2096	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1861	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2273	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2025	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3424	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2268	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2554	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2333	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2330	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2679	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3102	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2098	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3030	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2012	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2356	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2439	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1924	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2105	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2462	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2355	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3281	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1874	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1845	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2671	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2339	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2906	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3558	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3647	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2367	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3187	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2777	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2469	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2375	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3675	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1827	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3003	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2300	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2191	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1865	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3314	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2229	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3527	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3139	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2616	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2259	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2557	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2374	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2298	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2920	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3722	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2284	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3304	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2283	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1727	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2282	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2134	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1781	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2346	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3093	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3452	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3420	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2280	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2336	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1615	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3372	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1859	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3115	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3010	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2058	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3495	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2320	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2780	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1550	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2913	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3679	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1751	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2744	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2660	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2311	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2275	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2623	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2005	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2513	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3343	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2265	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1644	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2086	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2321	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2262	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2731	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2278	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2292	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2322	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3660	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2972	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2202	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2307	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2258	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3362	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2294	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3409	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2250	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2249	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2306	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2661	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1668	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2245	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2771	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1954	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2360	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2301	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2483	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2238	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2276	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2270	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2232	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2234	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1938	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2877	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2860	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2227	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1820	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2264	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2406	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3378	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1881	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2222	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3612	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2223	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2143	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1803	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1799	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2571	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2217	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2216	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2022	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1960	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2969	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2288	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3555	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2212	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2382	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2209	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1629	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1933	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1836	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2243	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2083	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2207	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2239	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2325	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3635	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3585	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2847	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2344	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2203	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1826	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2201	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2242	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2540	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2224	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2291	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1958	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2474	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2722	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3398	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2359	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2733	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1705	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3369	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2296	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2129	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2348	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1847	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1503	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category enum. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1526	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the

						operation.
1551	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1504	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1553	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1535	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1540	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1543	10.1	Operands shall not be of an inappropriate essential type. The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1549	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1505	10.1	Operands shall not be of an inappropriate essential type.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need

		The left operand of the   operator is of an inappropriate essential type category signed. The right operand of the   operator is of an inappropriate essential type category enum.				this method to set or clear some bits and drivers guarantee the safety of the operation.
1547	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the   operator has essentially signed type while the right operand has essentially enum type.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2743	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1616	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1745	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2524	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1905	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2380	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2195	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2905	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2193	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2042	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2789	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2970	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3249	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3704	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2188	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3439	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2343	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3125	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1554	10.1	Operands shall not be of an inappropriate essential type. The right operand of the & operator is of an inappropriate essential type category signed.	mpu6500_dmp_tap_orient_motion_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2235	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

Table 2.11. E:\Github\mpu6500\test\driver\_mpu6500\_fifo\_test.c

ID	Guideline	Message	Function	Severity	Status	Comment
2555	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2611	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3441	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2458	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2987	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2350	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2415	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2413	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3200	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2405	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3085	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2590	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2404	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3258	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2414	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1989	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3324	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3551	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2397	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1871	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2399	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2750	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2097	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2392	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2251	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1651	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2390	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2156	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2915	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2503	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1900	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2596	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2402	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3145	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2409	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2418	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2395	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3285	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2263	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1613	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2389	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3466	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2387	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3013	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2863	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2975	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1662	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2816	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2398	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3061	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2814	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2770	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2274	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2388	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2385	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2558	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3064	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2481	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2379	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3588	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2378	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3075	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2728	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2525	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2927	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1625	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2434	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
3728	D4.14	The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite.	mpu6500_fifo_test()	Low	Justified	Loop can't be infinite.
2376	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3515	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2577	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2446	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2247	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2621	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2831	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3385	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2145	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2419	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

**Table 2.12. E:\Github\mpu6500\test\driver\_mpu6500\_read\_test.c**

ID	Guideline	Message	Function	Severity	Status	Comment
3499	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1747	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3096	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1643	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2269	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3521	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3489	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1670	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2528	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2545	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2882	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2747	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2948	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2518	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3081	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2694	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2450	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2497	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3713	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3326	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3496	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1634	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1816	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2447	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2204	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1780	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2532	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2062	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2164	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2520	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3007	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2495	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3724	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3211	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2501	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3407	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2035	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2260	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2312	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2500	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3536	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2248	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3698	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2381	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2220	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2531	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3670	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2363	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2938	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1920	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2492	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2529	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2754	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2690	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3592	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2303	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2717	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1790	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2345	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1955	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2490	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1864	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2478	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2475	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2597	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1978	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3725	D4.14	The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite.	mpu6500_read_test()	Low	Justified	Loop can't be infinite.
2480	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1746	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3296	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3062	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2581	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2146	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2479	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.

3284	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2507	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3399	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1692	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2502	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2626	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2693	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2422	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2008	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1721	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2498	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1694	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1840	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1649	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2456	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1950	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
3667	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2429	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2473	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2435	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2594	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2034	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2150	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2745	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1935	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3344	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1623	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2736	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3038	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2467	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2605	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2979	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2257	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2065	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2124	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3251	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2464	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2277	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2875	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2463	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	print function.
2460	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3267	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3638	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2472	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2523	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2457	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1959	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3310	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
1896	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3154	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2423	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2194	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2072	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1610	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2453	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1913	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3639	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2030	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2666	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

Table 2.13. E:\Github\mpu6500\test\driver\_mpu6500\_register\_test.c

ID	Guideline	Message	Function	Severity	Status	Comment
3533	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3509	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2031	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3506	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3377	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3681	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3215	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3486	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2713	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3625	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2849	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2515	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3470	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2613	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2237	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2181	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3359	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3451	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3445	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2335	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3432	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2037	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3426	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3529	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3421	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3055	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2559	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1641	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3501	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3018	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2135	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3186	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3367	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
2751	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2958	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3113	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2778	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2549	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3476	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3580	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3305	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2001	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2491	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3482	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3600	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2043	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3438	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3252	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2016	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2233	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3091	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1627	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3419	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2354	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2080	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2340	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3485	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3311	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3629	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2477	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3254	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3279	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3412	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3292	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3183	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3271	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2236	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3428	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3579	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3465	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1621	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2876	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3129	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3404	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1879	2.2	There shall be no dead code. The call to function mpu6500_interface_delay_ms has no effect.	File Scope	Low	Justified	delay function.
3255	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1999	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3491	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3240	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3454	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2955	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3297	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3392	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3498	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2075	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2684	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2516	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2366	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3238	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3322	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2980	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3233	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3472	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3455	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2254	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3691	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3231	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2323	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3414	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1951	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3224	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2199	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3418	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2182	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3219	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3073	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2332	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3171	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3210	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2176	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3650	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3601	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2961	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3435	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3508	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2907	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3519	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3461	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3208	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3023	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2166	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3248	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2316	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3226	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3160	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2809	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3590	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3678	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3209	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2342	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3333	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2898	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1880	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3175	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3684	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1832	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3230	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3623	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2053	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2759	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2856	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3192	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3323	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3185	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2045	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1901	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2449	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2667	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3413	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3293	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3162	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3308	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3721	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1889	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3158	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2568	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3457	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3329	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1940	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2534	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3616	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2617	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3463	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3410	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2769	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3388	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3317	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1571	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_register_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
3286	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3711	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2165	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3621	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3542	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3450	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3640	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3140	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3599	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3589	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3133	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2988	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3556	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3170	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3127	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3123	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3380	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2101	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1580	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned.	mpu6500_register_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
3686	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3383	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3641	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3195	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3653	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3583	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3456	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2421	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3330	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3607	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2835	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2687	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3361	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3173	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3120	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2155	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2266	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2112	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2792	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3036	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3068	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3116	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3514	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3507	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2631	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3400	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2214	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2420	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3099	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2544	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3155	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1731	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2934	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3403	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3662	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3561	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2803	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3357	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3632	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2758	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3530	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2943	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3094	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1699	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3546	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3118	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3504	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1944	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2824	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3710	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2424	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2861	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3397	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2438	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3176	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1734	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2542	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3518	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2286	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3078	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1759	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2781	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3545	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3356	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2180	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1907	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3095	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2068	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1650	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3320	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1770	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3100	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2724	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3593	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3105	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3332	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3458	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1893	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3065	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2069	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2228	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3448	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3283	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3056	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3047	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3346	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1693	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3575	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2240	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2709	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3037	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3035	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3031	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3086	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1687	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1891	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2205	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2522	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3319	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3666	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3166	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3016	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2496	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2914	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3136	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3012	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3005	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1947	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3157	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3665	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3467	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1648	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2768	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3147	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3179	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3534	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2634	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2172	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3676	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3277	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2471	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3256	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2911	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3586	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3243	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2664	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2991	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1798	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2843	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3548	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3401	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1981	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3237	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3335	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3494	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2081	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3080	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2763	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2983	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1973	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2889	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3161	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2757	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2794	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1842	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1823	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1890	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1942	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1639	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2990	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2986	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2526	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1809	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1967	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3492	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3269	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2104	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2997	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3700	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3024	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2976	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3703	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3114	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2963	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3298	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2511	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3340	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3382	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3443	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2200	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3402	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3191	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2953	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3554	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3373	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1813	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2688	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3181	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1835	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2945	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2959	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2253	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3535	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2753	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3706	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2655	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2149	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1619	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3002	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3341	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3334	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3306	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2840	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3525	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3348	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2935	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1724	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3687	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2947	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2931	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3043	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3048	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2926	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2192	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2159	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2801	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2996	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3702	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3049	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1620	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1793	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3709	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2308	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2917	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3677	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2912	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3442	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2910	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1701	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2701	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2219	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3683	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3218	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1867	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3131	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3578	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1850	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2444	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3263	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2902	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3019	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2279	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3006	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2892	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1723	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3543	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2662	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3257	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3167	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3246	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2598	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2560	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3022	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3493	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3602	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3723	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2887	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3396	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1681	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2541	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2564	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2885	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2884	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2925	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2401	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2878	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3714	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3098	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3316	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2476	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1785	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2514	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3070	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3262	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3294	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2625	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1926	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3540	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3464	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2921	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2550	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2895	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1719	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2978	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3645	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2455	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1717	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3282	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2400	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3033	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3079	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3146	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3204	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3152	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3562	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3044	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2867	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3027	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1912	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2338	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2998	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2841	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3159	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2061	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3447	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3366	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2891	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2761	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3597	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1593	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3422	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3144	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2190	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2985	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2916	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1570	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1728	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2683	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2642	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2552	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3705	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1598	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1870	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2370	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3128	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2309	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2136	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1573	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1969	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3707	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2775	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2844	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2561	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1579	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3474	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3050	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2933	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1673	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2231	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1575	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3576	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3261	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2950	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3636	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2919	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1576	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2851	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3484	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3663	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3531	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1875	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1601	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the / operator has essentially signed type while the right operand has essentially floating type.	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1898	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2903	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3524	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3444	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1808	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2425	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3594	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3609	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3069	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2893	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2848	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2989	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3268	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1702	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3685	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2057	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2883	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2837	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2836	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3088	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2391	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3389	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3696	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2834	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2347	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2131	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2601	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3483	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3313	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1916	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3312	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2900	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3468	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1657	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3063	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3245	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3331	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3083	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3606	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1635	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3184	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3067	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3577	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3661	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1766	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3205	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3434	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2819	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2838	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1669	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2818	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3626	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2854	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3221	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3178	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3526	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3719	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3053	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2198	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2924	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3315	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2533	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2465	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3390	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3354	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3017	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3287	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2154	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2428	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3058	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2881	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2619	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3355	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2810	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2252	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2393	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3624	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2936	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3628	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2999	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3363	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3460	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1878	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3538	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3274	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3001	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2299	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3539	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2804	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3234	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3655	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2800	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1986	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3045	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3135	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3503	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2487	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2371	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2314	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1768	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1956	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3106	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3220	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2796	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1934	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2113	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3411	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2793	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3201	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3198	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3266	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3595	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2089	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2411	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2788	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2974	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3620	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3481	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3537	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2315	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2813	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1753	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2896	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2874	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2178	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3480	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2436	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2624	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2452	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2774	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1654	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2992	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2795	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3560	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2573	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2044	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2966	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2334	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3151	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2725	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2261	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2689	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2772	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3032	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2470	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2870	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3273	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2967	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3618	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3112	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2007	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3672	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2853	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1608	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3522	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2726	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3568	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2678	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1917	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1866	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3417	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3622	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3487	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3338	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1887	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2760	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2169	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3130	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3041	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2862	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2852	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1581	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2226	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3121	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2918	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2151	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1572	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1858	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2746	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2742	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3552	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1574	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2719	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2846	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3694	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3223	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1591	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3150	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3174	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2740	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3572	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1602	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3353	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3040	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3582	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2036	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3216	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1583	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2646	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3153	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2739	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2826	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1597	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee

		object with a different essential type category (unsigned)				the safety of the operation.
3057	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2313	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1992	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2305	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1584	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3342	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2732	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2647	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2940	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1586	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3364	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3598	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1708	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2727	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1589	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2643	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2971	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2894	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2319	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2723	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1578	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2721	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2132	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2173	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2798	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1594	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3066	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2817	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1738	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1787	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1585	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3386	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3699	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3701	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2230	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1587	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2820	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2714	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3199	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3692	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2396	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3581	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1928	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3259	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3202	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1735	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3505	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3228	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2957	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2674	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2791	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3143	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2570	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3490	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3214	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3137	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2569	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3250	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3244	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3302	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2712	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3072	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2765	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1741	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2486	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1636	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2748	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3165	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2032	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2707	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1762	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3528	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2782	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2509	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3605	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2641	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3227	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2706	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3649	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2845	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2585	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3358	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2353	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3020	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3637	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3497	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3015	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3189	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1885	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1640	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2956	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2730	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3103	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3122	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1987	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2704	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3408	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2337	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2459	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1804	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2304	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3549	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2546	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1841	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3253	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3076	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2909	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3303	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2868	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2702	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2767	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3291	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2785	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3156	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3052	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1909	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2776	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1929	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3196	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2591	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2700	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2138	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1961	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2244	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2084	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1664	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3477	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2698	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1695	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3235	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3697	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1921	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3511	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2691	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3680	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1844	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2696	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3429	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1810	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3584	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3431	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2669	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2686	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3563	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3092	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2822	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2962	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2600	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3232	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3337	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2995	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3229	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3425	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3339	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2859	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3395	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2506	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2013	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3643	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3148	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3009	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3345	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1777	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1817	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1680	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2850	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2682	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2676	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3213	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3059	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2923	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2672	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2734	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1582	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.

3110	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2799	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3393	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2668	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1599	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
3325	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2665	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1829	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3247	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1588	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2645	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2663	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3108	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1948	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1590	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some

		The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)				bits and drivers guarantee the safety of the operation.
2622	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2659	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2735	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2658	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3611	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2806	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2657	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3264	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3513	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3365	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2099	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1618	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2427	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1612	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2386	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2327	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2879	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3336	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3203	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2213	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2519	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2716	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2324	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1663	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2908	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2196	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3603	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2973	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2656	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2186	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3351	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3391	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2784	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2653	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1814	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2652	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2331	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3587	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3132	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2650	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3440	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3272	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2504	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3544	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2644	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2548	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2825	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2639	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2632	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3462	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3004	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3077	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3194	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2510	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3242	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3564	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3197	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2574	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1676	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2944	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3347	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3014	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2812	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2786	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3028	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2538	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2649	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3427	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2038	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3516	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2620	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1755	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2805	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3193	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2670	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2982	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3652	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1718	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2125	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2615	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3368	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2654	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1892	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2614	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1592	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2361	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2610	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2790	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2811	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1822	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1595	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2609	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3500	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2858	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2941	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2741	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1577	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1750	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3074	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3300	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2341	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2815	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2705	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1713	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2608	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1638	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2606	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3275	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2530	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3222	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2603	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3051	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
2602	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3574	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3517	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2024	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2833	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2930	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3270	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2184	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2697	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2271	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3673	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3060	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2041	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3478	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2737	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3674	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

1603	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
1742	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2629	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3631	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3318	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3352	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1596	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)	mpu6500_register_test()	Low	Not a defect	Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.
2595	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3475	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2009	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1720	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2493	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2899	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3376	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2046	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3087	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3423	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2829	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2082	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3349	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3082	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2628	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1937	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1908	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1963	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2290	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2897	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2078	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2677	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3608	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3416	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3026	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3042	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2592	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2968	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3381	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2295	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1855	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3379	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1915	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2589	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2872	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2588	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3278	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2673	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2027	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1811	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2586	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2584	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3664	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2871	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2004	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2807	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2583	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2904	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2783	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3610	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2579	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2766	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3260	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1711	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3695	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3671	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3011	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2267	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
3614	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2857	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2578	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2127	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2576	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2575	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2952	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1609	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3510	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3557	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2994	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1815	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2572	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2431	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3328	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3239	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

3034	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3276	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2880	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3644	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2488	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2773	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2536	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1857	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3241	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3547	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3690	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2565	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2563	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3301	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2070	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2808	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3288	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1788	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2466	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3394	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2964	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3172	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3289	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3142	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2949	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2582	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2318	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3141	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2512	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2407	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2612	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2797	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2556	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2708	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2675	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1863	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2128	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3591	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2015	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3613	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2056	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2764	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3571	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2547	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2281	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1839	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1600	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating.	mpu6500_register_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
2377	2.2	There shall be no dead code.	File Scope	Low	Justified	print function.

		The call to function mpu6500_interface_debug_print has no effect.				
1674	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1764	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2543	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2484	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3567	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2328	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3236	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2539	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2756	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3473	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2580	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2869	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3000	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2562	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2521	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3459	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

2537	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1797	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1923	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1604	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating.	mpu6500_register_test()	Low	Not a defect	We use enumeration to define driver configuration, which is a friendly programming method and should be accepted and drivers guarantee the safety of the operation.
2607	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
3104	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2932	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
2680	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.
1964	2.2	There shall be no dead code. The call to function mpu6500_interface_debug_print has no effect.	File Scope	Low	Justified	print function.

# **Chapter 3. Defects**

## **Defects**

No defects were found.

# Chapter 4. Appendix 1 - Configuration Settings

## Polyspace Settings

Option	Value
-author	LibDriver
-bug-finder	true
-checkers	ALIGNMENT_CHANGE, ASSERT, ATOMIC_VAR_ACCESS_TWICE, ATOMIC_VAR_SEQUENCE_NOT_ATOMIC, BAD_EQUAL_EQUAL_USE, BAD_EQUAL_USE, BAD_FREE, BAD_LOCK, BAD_PTR_SCALING, BAD_UNLOCK, CHARACTER_MISUSE, CHAR_EOF_CONFUSED, CLOSED_RESOURCE_USE, CONSTANT_OBJECT_WRITE, DATA_RACE, DATA_RACE_STD_LIB, DEADLOCK, DECL_MISMATCH, DOUBLE_DEALLOCATION, DOUBLE_LOCK, DOUBLE_RESOURCE_CLOSE, DOUBLE_RESOURCE_OPEN, DOUBLE_UNLOCK, ERRNO_MISUSE, FILE_OBJECT_MISUSE, FLEXIBLE_ARRAY_MEMBER_STRUCT_MISUSE, FLOAT_ABSORPTION, FLOAT_CONV_OVFL, FLOAT_STD_LIB, FLOAT_ZERO_DIV, FREED_PTR, FUNC_CAST, IMPROPER_ARRAY_INIT, INLINE_CONSTRAINT_NOT.Respected, INT_CONV_OVFL, INT_STD_LIB, INT_ZERO_DIV, INVALID_ENV_POINTER, INVALID_MEMORY_ASSUMPTION, INVALID_VA_LIST_ARG, IO_INTERLEAVING, LOCAL_ADDR_ESCAPE, MACRO_USED_AS_OBJECT, MEMCMP_PADDING_DATA, MEMCMP_STRINGS, MEM_STD_LIB, MISSING_ERRNO_RESET, MISSING_NULL_CHAR, MISSING_RETURN, NON_INIT_PTR, NON_INIT_VAR, NON_POSITIVE_VLA_SIZE, NULL_PTR, OPERATOR_PRECEDENCE, OTHER_STD_LIB, OUT_BOUND_ARRAY, OUT_BOUND_PTR, PARTIALLY_ACCESSED_ARRAY, PRE_DIRECTIVE_MACRO_ARG, PRE_UCNAME_JOIN_TOKENS, PTR_CAST, PTR_SIZEOF_MISMATCH, PTR_TO_DIFF_ARRAY, PUTENV_AUTO_VAR, READ_ONLY_RESOURCE_WRITE, RESOURCE_LEAK, SIDE_EFFECT_IGNORED, SIGN_CHANGE, SIG_HANDLER_CALLING_SIGNAL, SIG_HANDLER_COMP_EXCP_RETURN, SIG_HANDLER_ERRNO_MISUSE, SIG_HANDLER_SHARED_OBJECT, SIZEOF_MISUSE, STD_FUNC_ARG_MISMATCH, STREAM_WITH_SIDE_EFFECT, STRING_FORMAT, STRLIB_BUFFER_OVERFLOW, STRLIB_BUFFER_UNDERFLOW, STR_FORMAT_BUFFER_OVERFLOW, STR_STD_LIB, TEMP_OBJECT_ACCESS, TOO_MANY_VA_ARG_CALLS, TYPEDEF_MISMATCH, UINT_CONV_OVFL, UNPROTOTYPED_FUNC_CALL, UNREACHABLE, USELESS_IF, USELESS_WRITE, VAR_SHADOWING, VA_ARG_INCORRECT_TYPE, VA_START_INCORRECT_TYPE, VA_START_MISUSE
-compiler	iar
-D	__TID__=14,__SIZE_T_TYPE__=unsigned int,__PTRDIFF_T_TYPE__=signed int,__IAR_SYSTEMS_ICC=1
-date	01/07/2024
-dos	true
-I	E:\Github\mpu6500\src,E:\Github\mpu6500\interface,E:\Github\mpu6500\example,E:\Github\mpu6500\test
-import-comments	E:\Polyspace\mpu6500\Module\BF_Result\comments_bak
-lang	C

-little-endian	true
-logical-signed-right-shift	true
-misra3	mandatory-required
-prog	mpu6500
-results-dir	E:\Polyspace\mpu6500\Module\BF_Result
-sfr-types	sfr8=8,sfr16=16,sfr32=32,sfr=8
-target	mcpu
-verif-version	1.0

## Coding Standard Configuration

Table 4.1. MISRA C:2012 Guidelines Configuration

Guideline	Description	Mode	Comment	Enabled
D1.1	Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.	required	-	yes
D2.1	All source files shall compile without any compilation errors.	required	-	yes
D3.1	All code shall be traceable to documented requirements.	required	Not enforceable	no
D4.1	Run-time failures shall be minimized.	required	-	yes
D4.2	All usage of assembly language should be documented.	advisory	Not enforceable	no
D4.3	Assembly language shall be encapsulated and isolated.	required	-	yes
D4.4	Sections of code should not be "commented out".	advisory	Not implemented	no
D4.5	Identifiers in the same name space with overlapping visibility should be typographically unambiguous.	advisory	-	no
D4.6	typedefs that indicate size and signedness should be used in place of the basic numerical types.	advisory	-	no
D4.7	If a function returns error information, then that error information shall be tested.	required	-	yes
D4.8	If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden.	advisory	-	no
D4.9	A function should be used in preference to a function-like macro where they are interchangeable.	advisory	-	no
D4.10	Precautions shall be taken in order to prevent the contents of a header file being included more than once.	required	-	yes

D4.11	The validity of values passed to library functions shall be checked.	required	-	yes
D4.12	Dynamic memory allocation shall not be used.	required	-	yes
D4.13	Functions which are designed to provide operations on a resource should be called in an appropriate sequence.	advisory	-	no
D4.14	The validity of values received from external sources shall be checked.	required	-	yes
1.1	The program shall contain no violations of the standard C syntax and constraints, and shall not exceed the implementation's translation limits.	required	-	yes
1.2	Language extensions should not be used.	advisory	-	no
1.3	There shall be no occurrence of undefined or critical unspecified behaviour.	required	-	yes
2.1	A project shall not contain unreachable code.	required	-	yes
2.2	There shall be no dead code.	required	-	yes
2.3	A project should not contain unused type declarations.	advisory	-	no
2.4	A project should not contain unused tag declarations.	advisory	-	no
2.5	A project should not contain unused macro declarations.	advisory	-	no
2.6	A function should not contain unused label declarations.	advisory	-	no
2.7	There should be no unused parameters in functions.	advisory	-	no
3.1	The character sequences /* and // shall not be used within a comment.	required	-	yes
3.2	Line-splicing shall not be used in // comments.	required	-	yes
4.1	Octal and hexadecimal escape sequences shall be terminated.	required	-	yes
4.2	Trigraphs should not be used.	advisory	-	no
5.1	External identifiers shall be distinct.	required	-	yes
5.2	Identifiers declared in the same scope and name space shall be distinct.	required	-	yes
5.3	An identifier declared in an inner scope shall not hide an identifier declared in an outer scope.	required	-	yes
5.4	Macro identifiers shall be distinct.	required	-	yes
5.5	Identifiers shall be distinct from macro names.	required	-	yes
5.6	A typedef name shall be a unique identifier.	required	-	yes
5.7	A tag name shall be a unique identifier.	required	-	yes
5.8	Identifiers that define objects or functions with external linkage shall be unique.	required	-	yes

5.9	Identifiers that define objects or functions with internal linkage should be unique.	advisory	-	no
6.1	Bit-fields shall only be declared with an appropriate type.	required	-	yes
6.2	Single-bit named bit fields shall not be of a signed type.	required	-	yes
7.1	Octal constants shall not be used.	required	-	yes
7.2	A "u" or "U" suffix shall be applied to all integer constants that are represented in an unsigned type.	required	-	yes
7.3	The lowercase character "l" shall not be used in a literal suffix.	required	-	yes
7.4	A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char".	required	-	yes
8.1	Types shall be explicitly specified.	required	-	yes
8.2	Function types shall be in prototype form with named parameters.	required	-	yes
8.3	All declarations of an object or function shall use the same names and type qualifiers.	required	-	yes
8.4	A compatible declaration shall be visible when an object or function with external linkage is defined.	required	-	yes
8.5	An external object or function shall be declared once in one and only one file.	required	-	yes
8.6	An identifier with external linkage shall have exactly one external definition.	required	-	yes
8.7	Functions and objects should not be defined with external linkage if they are referenced in only one translation unit.	advisory	-	no
8.8	The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage.	required	-	yes
8.9	An object should be defined at block scope if its identifier only appears in a single function.	advisory	-	no
8.10	An inline function shall be declared with the static storage class.	required	-	yes
8.11	When an array with external linkage is declared, its size should be explicitly specified.	advisory	-	no
8.12	Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique.	required	-	yes
8.13	A pointer should point to a const-qualified type whenever possible.	advisory	-	no
8.14	The restrict type qualifier shall not be used.	required	-	yes
9.1	The value of an object with automatic storage duration shall not be read before it has been set.	mandatory	-	yes
9.2	The initializer for an aggregate or union shall be enclosed in braces.	required	-	yes
9.3	Arrays shall not be partially initialized.	required	-	yes
9.4	An element of an object shall not be initialized more than once.	required	-	yes

9.5	Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly.	required	-	yes
10.1	Operands shall not be of an inappropriate essential type.	required	-	yes
10.2	Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations.	required	-	yes
10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.	required	-	yes
10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.	required	-	yes
10.5	The value of an expression should not be cast to an inappropriate essential type.	advisory	-	no
10.6	The value of a composite expression shall not be assigned to an object with wider essential type.	required	-	yes
10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type.	required	-	yes
10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.	required	-	yes
11.1	Conversions shall not be performed between a pointer to a function and any other type.	required	-	yes
11.2	Conversions shall not be performed between a pointer to an incomplete type and any other type.	required	-	yes
11.3	A cast shall not be performed between a pointer to object type and a pointer to a different object type.	required	-	yes
11.4	A conversion should not be performed between a pointer to object and an integer type.	advisory	-	no
11.5	A conversion should not be performed from pointer to void into pointer to object.	advisory	-	no
11.6	A cast shall not be performed between pointer to void and an arithmetic type.	required	-	yes
11.7	A cast shall not be performed between pointer to object and a non-integer arithmetic type.	required	-	yes
11.8	A cast shall not remove any const or volatile qualification from the type pointed to by a pointer.	required	-	yes
11.9	The macro NULL shall be the only permitted form of integer null pointer constant.	required	-	yes
12.1	The precedence of operators within expressions should be made explicit.	advisory	-	no
12.2	The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand.	required	-	yes
12.3	The comma operator should not be used	advisory	-	no
12.4	Evaluation of constant expressions should not lead to unsigned integer wrap-around.	advisory	-	no
12.5	The sizeof operator shall not have an operand which is a function parameter declared as "array of	mandatory	-	yes

	type".			
13.1	Initializer lists shall not contain persistent side effects.	required	-	yes
13.2	The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders.	required	-	yes
13.3	A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator.	advisory	-	no
13.4	The result of an assignment operator should not be used.	advisory	-	no
13.5	The right hand operand of a logical && or    operator shall not contain persistent side effects.	required	-	yes
13.6	The operand of the sizeof operator shall not contain any expression which has potential side effects.	mandatory	-	yes
14.1	A loop counter shall not have essentially floating type.	required	-	yes
14.2	A for loop shall be well-formed.	required	-	yes
14.3	Controlling expressions shall not be invariant.	required	-	yes
14.4	The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type.	required	-	yes
15.1	The goto statement should not be used.	advisory	-	no
15.2	The goto statement shall jump to a label declared later in the same function.	required	-	yes
15.3	Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement.	required	-	yes
15.4	There should be no more than one break or goto statement used to terminate any iteration statement.	advisory	-	no
15.5	A function should have a single point of exit at the end.	advisory	-	no
15.6	The body of an iteration-statement or a selection-statement shall be a compound-statement.	required	-	yes
15.7	All if ... else if constructs shall be terminated with an else statement.	required	-	yes
16.1	All switch statements shall be well-formed.	required	-	yes
16.2	A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement.	required	-	yes
16.3	An unconditional break statement shall terminate every switch-clause.	required	-	yes
16.4	Every switch statement shall have a default label.	required	-	yes
16.5	A default label shall appear as either the first or the last switch label of a switch statement.	required	-	yes
16.6	Every switch statement shall have at least two switch-clauses.	required	-	yes

16.7	A switch-expression shall not have essentially Boolean type.	required	-	yes
17.1	The features of <stdarg.h> shall not be used.	required	-	yes
17.2	Functions shall not call themselves, either directly or indirectly.	required	-	yes
17.3	A function shall not be declared implicitly.	mandatory	-	yes
17.4	All exit paths from a function with non-void return type shall have an explicit return statement with an expression.	mandatory	-	yes
17.5	The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements.	advisory	-	no
17.6	The declaration of an array parameter shall not contain the static keyword between the [ ].	mandatory	-	yes
17.7	The value returned by a function having non-void return type shall be used.	required	-	yes
17.8	A function parameter should not be modified.	advisory	-	no
18.1	A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand.	required	-	yes
18.2	Subtraction between pointers shall only be applied to pointers that address elements of the same array.	required	-	yes
18.3	The relational operators >, >=, < and <= shall not be applied to objects of pointer type except where they point into the same object.	required	-	yes
18.4	The +, -, += and -= operators should not be applied to an expression of pointer type.	advisory	-	no
18.5	Declarations should contain no more than two levels of pointer nesting.	advisory	-	no
18.6	The address of an object with automatic storage shall not be copied to another object that persists after the first object has ceased to exist.	required	-	yes
18.7	Flexible array members shall not be declared.	required	-	yes
18.8	Variable-length array types shall not be used.	required	-	yes
19.1	An object shall not be assigned or copied to an overlapping object.	mandatory	-	yes
19.2	The union keyword should not be used.	advisory	-	no
20.1	#include directives should only be preceded by preprocessor directives or comments.	advisory	-	no
20.2	The ', " or \ characters and the /* or // character sequences shall not occur in a header file name.	required	-	yes
20.3	The #include directive shall be followed by either a <filename> or "filename" sequence.	required	-	yes
20.4	A macro shall not be defined with the same name as a keyword.	required	-	yes

20.5	#undef should not be used.	advisory	-	no
20.6	Tokens that look like a preprocessing directive shall not occur within a macro argument.	required	-	yes
20.7	Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses.	required	-	yes
20.8	The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1.	required	-	yes
20.9	All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation.	required	-	yes
20.10	The # and ## preprocessor operators should not be used.	advisory	-	no
20.11	A macro parameter immediately following a # operator shall not immediately be followed by a ## operator.	required	-	yes
20.12	A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators.	required	-	yes
20.13	A line whose first token is # shall be a valid preprocessing directive.	required	-	yes
20.14	All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related.	required	-	yes
21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name.	required	-	yes
21.2	A reserved identifier or macro name shall not be declared.	required	-	yes
21.3	The memory allocation and deallocation functions of <stdlib.h> shall not be used.	required	-	yes
21.4	The standard header file <setjmp.h> shall not be used.	required	-	yes
21.5	The standard header file <signal.h> shall not be used.	required	-	yes
21.6	The Standard Library input/output functions shall not be used.	required	-	yes
21.7	The atof, atoi, atol, and atoll functions of <stdlib.h> shall not be used.	required	-	yes
21.8	The library functions abort, exit and system of <stdlib.h> shall not be used.	required	-	yes
21.9	The library functions bsearch and qsort of <stdlib.h> shall not be used.	required	-	yes
21.10	The Standard Library time and date functions shall not be used.	required	-	yes
21.11	The standard header file <tgmath.h> shall not be used.	required	-	yes
21.12	The exception handling features of <fenv.h> should not be used.	advisory	-	no
21.13	Any value passed to a function in <ctype.h> shall be representable as an unsigned char or be the value EOF.	mandatory	-	yes
21.14	The Standard Library function memcmp shall not be used to compare null terminated strings.	required	-	yes

21.15	The pointer arguments to the Standard Library functions <code>memcpy</code> , <code>memmove</code> and <code>memcmp</code> shall be pointers to qualified or unqualified versions of compatible types.	required	-	yes
21.16	The pointer arguments to the Standard Library function <code>memcmp</code> shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type.	required	-	yes
21.17	Use of the string handling functions from <code>&lt;string.h&gt;</code> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters.	mandatory	-	yes
21.18	The <code>size_t</code> argument passed to any function in <code>&lt;string.h&gt;</code> shall have an appropriate value.	mandatory	-	yes
21.19	The pointers returned by the Standard Library functions <code>localeconv</code> , <code>getenv</code> , <code>setlocale</code> or, <code>strerror</code> shall only be used as if they have pointer to const-qualified type.	mandatory	-	yes
21.20	The pointer returned by the Standard Library functions <code>asctime</code> , <code>ctime</code> , <code>gmtime</code> , <code>localtime</code> , <code>localeconv</code> , <code>getenv</code> , <code>setlocale</code> or <code>strerror</code> shall not be used following a subsequent call to the same function.	mandatory	-	yes
22.1	All resources obtained dynamically by means of Standard Library functions shall be explicitly released.	required	-	yes
22.2	A block of memory shall only be freed if it was allocated by means of a Standard Library function.	mandatory	-	yes
22.3	The same file shall not be open for read and write access at the same time on different streams.	required	-	yes
22.4	There shall be no attempt to write to a stream which has been opened as read-only.	mandatory	-	yes
22.5	A pointer to a <code>FILE</code> object shall not be dereferenced.	mandatory	-	yes
22.6	The value of a pointer to a <code>FILE</code> shall not be used after the associated stream has been closed.	mandatory	-	yes
22.7	The macro <code>EOF</code> shall only be compared with the unmodified return value from any Standard Library function capable of returning <code>EOF</code> .	required	-	yes
22.8	The value of <code>errno</code> shall be set to zero prior to a call to an <code>errno</code> -setting-function.	required	-	yes
22.9	The value of <code>errno</code> shall be tested against zero after calling an <code>errno</code> -setting-function.	required	-	yes
22.10	The value of <code>errno</code> shall only be tested when the last function to be called was an <code>errno</code> -setting-function.	required	-	yes

## Chapter 5. Appendix 2 - Definitions

**Table 5.1. Abbreviations**

Abbreviation	Definition
NA	Not Available