

# **Polyspace Bug Finder**

**Detailed Report for Project: max31865**

**Report Author: LibDriver**

**Polyspace Bug Finder: Detailed Report for Project: max31865**

by Report Author: LibDriver

**Published 10-Apr-2022 20:15:39**

Analysis Author(s): LibDriver

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

Project Version(s): 1.0

Result Folder(s):

E:\Polyspace\max31865\Module\BF\_Result

## Table of Contents

|   |    |
|---|----|
| Chapter 1. Polyspace Bug Finder Summary .....                     | 1  |
| Chapter 2. MISRA C:2012 Guidelines .....                          | 2  |
| <b>MISRA C:2012 Guidelines Summary - Violations by File</b> ..... | 2  |
| <b>MISRA C:2012 Guidelines Violations</b> .....                   | 2  |
| Chapter 3. Defects .....  | 32 |
| <b>Defects</b> .....  | 32 |
| Chapter 4. Appendix 1 - Configuration Settings .....              | 33 |
| <b>Polyspace Settings</b> .....                                   | 33 |
| <b>Coding Standard Configuration</b> .....                        | 34 |
| Chapter 5. Appendix 2 - Definitions .....                         | 42 |

# Chapter 1. Polyspace Bug Finder Summary

Table 1.1. Project Summary

|                         | Count | Reviewed | Unreviewed | Pass/Fail |
|-------------------------|-------|----------|------------|-----------|
| MISRA C:2012 Guidelines | 285   | 285      | 0          | Pass      |
| Defects                 | 0     | 0        | 0          | Pass      |
| Total                   | 285   | 285      | 0          | Pass      |

Table 1.2. Summary By File

| File  | Defects (Reviewed) | MISRA C:2012 Guidelines (Reviewed) |
|---|--------------------|------------------------------------|
| E:\Github\max31865\example\driver_max31865_basic.c                | 0 (0)              | 10 (10)                            |
| E:\Github\max31865\example\driver_max31865_basic.h                | 0 (0)              | 0 (0)                              |
| E:\Github\max31865\example\driver_max31865_shot.c                 | 0 (0)              | 8 (8)                              |
| E:\Github\max31865\example\driver_max31865_shot.h                 | 0 (0)              | 0 (0)                              |
| E:\Github\max31865\interface\driver_max31865_interface.h          | 0 (0)              | 0 (0)                              |
| E:\Github\max31865\interface\driver_max31865_interface_template.c | 0 (0)              | 0 (0)                              |
| E:\Github\max31865\src\driver_max31865.c                          | 0 (0)              | 149 (149)                          |
| E:\Github\max31865\src\driver_max31865.h                          | 0 (0)              | 0 (0)                              |
| E:\Github\max31865\test\driver_max31865_read_test.c               | 0 (0)              | 31 (31)                            |
| E:\Github\max31865\test\driver_max31865_read_test.h               | 0 (0)              | 0 (0)                              |
| E:\Github\max31865\test\driver_max31865_register_test.c           | 0 (0)              | 87 (87)                            |
| E:\Github\max31865\test\driver_max31865_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\max31865\example\driver_max31865_basic.c      | 10         |
| E:\Github\max31865\example\driver_max31865_shot.c       | 8          |
| E:\Github\max31865\src\driver_max31865.c                | 149        |
| E:\Github\max31865\test\driver_max31865_read_test.c     | 31         |
| E:\Github\max31865\test\driver_max31865_register_test.c | 87         |
| <b>Total</b>  | <b>285</b> |

## MISRA C:2012 Guidelines Violations

Table 2.1. E:\Github\max31865\example\driver\_max31865\_basic.c

| ID  | Guideline | Message  | Function   | Severity | Status    | Comment         |
|-----|-----------|--|------------|----------|-----------|-----------------|
| 161 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 160 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 159 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 276 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 157 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 257 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 149 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |

|     |     |  |            |     |           |                 |
|-----|-----|--|------------|-----|-----------|-----------------|
| 150 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 178 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 148 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |

**Table 2.2. E:\Github\max31865\example\driver\_max31865\_shot.c**

| ID  | Guideline | Message  | Function   | Severity | Status    | Comment         |
|-----|-----------|--|------------|----------|-----------|-----------------|
| 270 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 152 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 267 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 255 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 278 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 153 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 186 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 254 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |

**Table 2.3. E:\Github\max31865\src\driver\_max31865.c**

| ID | Guideline | Message   | Function                            | Severity | Status       | Comment  |
|----|-----------|---|-------------------------------------|----------|--------------|--|
| 6  | 10.4      | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the /= operator has essentially floating type while the right operand has essentially signed type. | a_max31865_temperature_conversion() | Low      | Not a defect | We use this function to convert driver data and drivers guarantee the safety of the operation. |

|    |      |  |                                     |     |           |  |
|----|------|--|-------------------------------------|-----|-----------|--|
| 29 | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (floating) | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 44 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.<br>Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.                                  | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 65 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the * operator has essentially signed type while the right operand has essentially floating type.                 | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 36 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the * operator has essentially signed type while the right operand has essentially floating type.                 | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 9  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the * operator has essentially signed type while the right operand has essentially floating type.                 | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 25 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the >= operator has essentially floating type while the right operand has essentially signed type.                | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 14 | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (floating) | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 90 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the *= operator has essentially floating type while the right operand has essentially signed type.                | a_max31865_temperature_conversion() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |

|    |      |  |                              |     |              |  |
|----|------|--|------------------------------|-----|--------------|--|
| 4  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_deinit()            | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 19 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_deinit()            | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 54 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.                | max31865_deinit()            | 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.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_deinit()            | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_deinit()            | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 10 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.                       | max31865_deinit()            | 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.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_set_filter_select() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 26 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type  | max31865_set_filter_select() | Low | Not a defect | Embedded drivers need this method to set or clear  |



|     |      |  |                              |     |              |  |
|-----|------|--|------------------------------|-----|--------------|--|
|     |      | category.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.  |                              |     |              | some bits and drivers guarantee the safety of the operation.   |
| 83  | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_set_filter_select() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 24  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_set_filter_select() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_set_filter_select() | 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.<br>The right operand of the  = operator is of an inappropriate essential type category enum.   | max31865_set_filter_select() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 28  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.                  | max31865_set_filter_select() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 133 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.<br>The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)   | max31865_set_filter_select() | 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.<br>The left operand of the << operator is of an inappropriate essential type category enum.  | max31865_set_filter_select() | 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.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.                  | max31865_set_filter_select() | 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.<br>The right operand of the &= operator is of an inappropriate essential type category signed.  | max31865_get_filter_select() | Low | Not a defect | Embedded drivers need 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.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.           | max31865_get_filter_select() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.  |
| 128 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_get_filter_select() | 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.<br>The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum. | max31865_get_filter_select() | 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. |
| 1   | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.           | max31865_set_wire()          | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.  |
| 38  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the &= operator is of an inappropriate essential type category signed.  | max31865_set_wire()          | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of   |

|     |      |  |                     |     |              |  |
|-----|------|--|---------------------|-----|--------------|--|
|     |      |  |                     |     |              | the operation.   |
| 129 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_set_wire() | 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.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_set_wire() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 31  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_set_wire() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 40  | 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.<br>The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)   | max31865_set_wire() | 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 | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category enum.   | max31865_set_wire() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 49  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.                  | max31865_set_wire() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 42  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category enum.  | max31865_set_wire() | 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.  | max31865_set_wire() | Low | Not a defect | Embedded drivers need  |

|    |      |  |                      |     |              |  |
|----|------|--|----------------------|-----|--------------|--|
|    |      | <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>   |                      |     |              | <p>this method to set or clear some bits and drivers guarantee the safety of the operation.</p>  |
| 30 | 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>  | max31865_get_wire()  | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p>  |
| 33 | 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 &amp;= operator has essentially unsigned type while the right operand has essentially signed type.</p>       | max31865_get_wire()  | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p>  |
| 74 | 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>   | max31865_get_wire()  | Low | Justified    | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p>  |
| 50 | 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 unsigned shall not be cast to the different essential type category enum.</p> | max31865_get_wire()  | Low | Not a defect | <p>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.</p> |
| 5  | 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 &amp;= operator has essentially unsigned type while the right operand has essentially signed type.</p>       | max31865_set_vbias() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p>  |
| 52 | 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>  | max31865_set_vbias() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p>  |

|     |      |  |                      |     |              |  |
|-----|------|--|----------------------|-----|--------------|--|
| 53  | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_set_vbias() | 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.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_set_vbias() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_set_vbias() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 7   | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category enum.   | max31865_set_vbias() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 16  | 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.<br>The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)   | max31865_set_vbias() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 39  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.                  | max31865_set_vbias() | 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.<br>The left operand of the << operator is of an inappropriate essential type category enum.  | max31865_set_vbias() | 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.<br>The left operand of the   operator is of an inappropriate essential type  | max31865_set_vbias() | Low | Not a defect | Embedded drivers need this method to set or clear  |

|     |      |  |                               |     |              |   |
|-----|------|--|-------------------------------|-----|--------------|---|
|     |      | category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.   |                               |     |              | some bits and drivers guarantee the safety of the operation.  |
| 45  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.                | max31865_get_vbias()          | 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.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_get_vbias()          | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_get_vbias()          | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.  |
| 80  | 10.8 | The value of a composite expression shall not be cast to a different essential type category or a wider essential type.<br>The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum.      | max31865_get_vbias()          | 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. |
| 2   | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_clear_fault_status() | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_clear_fault_status() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.  |
| 103 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type  | max31865_clear_fault_status() | Low | Not a defect | Embedded drivers need this method to set or clear   |

|     |       |   |                               |     |              |  |
|-----|-------|---|-------------------------------|-----|--------------|--|
|     |       | category.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.   |                               |     |              | some bits and drivers guarantee the safety of the operation.   |
| 61  | 10.1  | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.   | max31865_clear_fault_status() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_clear_fault_status() | 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.<br>The right operand of the  = operator is of an inappropriate essential type category signed.  | max31865_clear_fault_status() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 63  | 10.4  | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type. | max31865_clear_fault_status() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 89  | 10.1  | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_clear_fault_status() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 67  | 10.1  | Operands shall not be of an inappropriate essential type.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.        | max31865_clear_fault_status() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 280 | D4.14 | The validity of values received from external sources shall be checked.<br>Dereferenced pointer is from an unsecure source.<br>Pointer may be NULL or may point to unknown memory.  | max31865_get_fault_status()   | Low | Justified    | (handle == NULL)checked.   |

|     |      |  |  |     |              |  |
|-----|------|--|--|-----|--------------|--|
| 283 | D4.1 | Run-time failures shall be minimized.<br>Operation << overflows.<br>Valid range: [0 .. 65535]  | max31865_set_high_fault_threshold()          | Low | Justified    | safe driver.   |
| 72  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the & operator is of an inappropriate essential type category signed.  | max31865_set_high_fault_threshold()          | 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 | Operands shall not be of an inappropriate essential type.<br>The right operand of the & operator is of an inappropriate essential type category signed.  | max31865_set_high_fault_threshold()          | 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.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed. | max31865_set_high_fault_threshold()          | 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.<br>The right operand of the & operator is of an inappropriate essential type category signed.  | max31865_set_low_fault_threshold()           | 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.<br>The right operand of the & operator is of an inappropriate essential type category signed.  | max31865_set_low_fault_threshold()           | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 95  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed. | max31865_set_low_fault_threshold()           | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 13  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the &= operator has essentially unsigned type   | max31865_set_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of                |



|     |      |  |  |     |              |  |
|-----|------|--|--|-----|--------------|--|
|     |      | while the right operand has essentially signed type.   |  |     |              | the operation.   |
| 73  | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_set_fault_detection_cycle_control() | 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.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_set_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 69  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_set_fault_detection_cycle_control() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_set_fault_detection_cycle_control() | 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 | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category enum.   | max31865_set_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 71  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially enum type.                  | max31865_set_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 78  | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.<br>The expression (of essential type category enum) is assigned to an object with a different essential type category (unsigned)   | max31865_set_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 109 | 10.1 | Operands shall not be of an inappropriate essential type.  | max31865_set_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need  |

|    |      |   |  |     |              |   |
|----|------|---|--|-----|--------------|---|
|    |      | The left 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.  |
| 98 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.                  | max31865_set_fault_detection_cycle_control() | 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.<br>The right operand of the &= operator is of an inappropriate essential type category signed.  | max31865_get_fault_detection_cycle_control() | 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.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.           | max31865_get_fault_detection_cycle_control() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_get_fault_detection_cycle_control() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.  |
| 99 | 10.8 | The value of a composite expression shall not be cast to a different essential type category or a wider essential type.<br>The value of the composite expression of essential type category unsigned shall not be cast to the different essential type category enum. | max31865_get_fault_detection_cycle_control() | 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. |
| 84 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.           | max31865_single_read()                       | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.  |

|     |      |   |                        |     |              |  |
|-----|------|---|------------------------|-----|--------------|--|
| 115 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category signed.  | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 47  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 3   | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type. | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 87  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category signed.  | max31865_single_read() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 91  | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type. | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 108 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the &= operator is of an inappropriate essential type category signed.  | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 139 | 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.   | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear  |

|     |      |   |                        |     |              |  |
|-----|------|---|------------------------|-----|--------------|--|
|     |      | The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned)   |                        |     |              | some bits and drivers guarantee the safety of the operation.   |
| 116 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.   | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 124 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 104 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type. | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 136 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category signed.  | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 94  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_single_read() | 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.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.        | max31865_single_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 18  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the & operator is of an inappropriate essential type category signed.   | max31865_single_read() | Low | Not a defect | Embedded drivers need 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_single_read()           | 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.<br>The right operand of the & operator is of an inappropriate essential type category signed.   | max31865_single_read()           | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.    |
| 58  | 10.4  | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.   | max31865_single_read()           | Low | Not a defect | We use this function to convert driver settings. Developers need to refer to the data manual to convert settings. |
| 284 | D4.14 | The validity of values received from external sources shall be checked.<br>Dereferenced pointer is from an unsecure source.<br>Pointer may be NULL or may point to unknown memory.  | max31865_single_read()           | Low | Justified    | (handle == NULL)checked.  |
| 32  | 10.1  | Operands shall not be of an inappropriate essential type.<br>The right operand of the  = operator is of an inappropriate essential type category signed.  | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.    |
| 134 | 10.4  | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type. | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.    |
| 141 | 10.1  | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_start_continuous_read() | 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.<br>The right operand of the  = operator is of an inappropriate essential  | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear   |

|     |      |  |                                  |     |              |  |
|-----|------|--|----------------------------------|-----|--------------|--|
|     |      | type category signed.  |                                  |     |              | some bits and drivers guarantee the safety of the operation.   |
| 132 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.                | max31865_start_continuous_read() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_start_continuous_read() | 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.<br>The right operand of the  = operator is of an inappropriate essential type category signed.   | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 122 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type.                | max31865_start_continuous_read() | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 37  | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_start_continuous_read() | 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.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_start_continuous_read() | 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.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type. | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 126 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.   | max31865_start_continuous_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.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_start_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 100 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.        | max31865_start_continuous_read() | 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.<br>The right operand of the  = operator is of an inappropriate essential type category signed.  | max31865_stop_continuous_read()  | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 123 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the  = operator has essentially unsigned type while the right operand has essentially signed type. | max31865_stop_continuous_read()  | 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.<br>The left operand of the << operator is of an inappropriate essential type category signed.   | max31865_stop_continuous_read()  | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 96  | 10.4 | Both operands of an operator in which the usual arithmetic  | max31865_stop_continuous_read()  | Low | Not a defect | 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 signed type.</p>   |                                 |     |              | <p>this method to set or clear some bits and drivers guarantee the safety of the operation.</p>                       |
| 110 | 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 an object with a different essential type category (unsigned)</p> | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p> |
| 113 | 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>   | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p> |
| 23  | 10.1 | <p>Operands shall not be of an inappropriate essential type.</p> <p>The operand of the ~ operator is of an inappropriate essential type category signed.</p>  | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p> |
| 60  | 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>  | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p> |
| 112 | 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>   | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p> |
| 117 | 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 an object with a different essential type category (unsigned)</p> | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.</p> |
| 118 | 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 &amp;= operator has essentially unsigned type</p>   | max31865_stop_continuous_read() | Low | Not a defect | <p>Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of</p>                |



|     |      |  |                                 |     |              |  |
|-----|------|--|---------------------------------|-----|--------------|--|
|     |      | while the right operand has essentially signed type.   |                                 |     |              | the operation.   |
| 138 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 111 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 21  | 10.1 | Operands shall not be of an inappropriate essential type.<br>The right operand of the &= operator is of an inappropriate essential type category signed.   | max31865_stop_continuous_read() | 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.<br>The left operand of the &= operator has essentially unsigned type while the right operand has essentially signed type.                | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 127 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 121 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The operand of the ~ operator is of an inappropriate essential type category signed.  | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation. |
| 140 | 10.1 | Operands shall not be of an inappropriate essential type.<br>The left operand of the << operator is of an inappropriate essential type category signed.  | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need 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.  | max31865_stop_continuous_read() | Low | Not a defect | Embedded drivers need  |

|     |       |  |                            |     |              |   |
|-----|-------|--|----------------------------|-----|--------------|---|
|     |       | The left operand of the   operator is of an inappropriate essential type category signed.<br>The right operand of the   operator is of an inappropriate essential type category signed.  |                            |     |              | this method to set or clear some bits and drivers guarantee the safety of the operation.                          |
| 85  | 10.1  | Operands shall not be of an inappropriate essential type.<br>The right operand of the & operator is of an inappropriate essential type category signed.  | max31865_continuous_read() | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.    |
| 35  | 10.4  | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the == operator has essentially unsigned type while the right operand has essentially enum type.  | max31865_continuous_read() | Low | Not a defect | We use this function to convert driver settings. Developers need to refer to the data manual to convert settings. |
| 285 | D4.14 | The validity of values received from external sources shall be checked.<br>Dereferenced pointer is from an unsecure source.<br>Pointer may be NULL or may point to unknown memory.   | max31865_continuous_read() | Low | Justified    | (handle == NULL)checked.  |
| 281 | D4.14 | The validity of values received from external sources shall be checked.<br>Dereferenced pointer is from an unsecure source.<br>Pointer may be NULL or may point to unknown memory.   | max31865_set_reg()         | Low | Justified    | (handle == NULL)checked.  |
| 135 | 10.1  | Operands shall not be of an inappropriate essential type.<br>The right operand of the   operator is of an inappropriate essential type category signed.  | max31865_set_reg()         | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.    |
| 143 | 10.4  | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.<br>The left operand of the   operator has essentially unsigned type while the right operand has essentially signed type. | max31865_set_reg()         | Low | Not a defect | Embedded drivers need this method to set or clear some bits and drivers guarantee the safety of the operation.    |
| 282 | D4.14 | The validity of values received from external sources shall be checked.<br>Dereferenced pointer is from an unsecure source.<br>Pointer may be NULL or may point to unknown memory.   | max31865_get_reg()         | Low | Justified    | (handle == NULL)checked.  |

**Table 2.4. E:\Github\max31865\test\driver\_max31865\_read\_test.c**

| ID  | Guideline | Message  | Function   | Severity | Status    | Comment         |
|-----|-----------|--|------------|----------|-----------|-----------------|
| 151 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 258 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 182 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 162 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 173 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 205 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 241 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 213 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 197 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 218 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 194 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 275 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 236 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 195 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 253 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |

|     |       |  |                      |     |           |                         |
|-----|-------|--|----------------------|-----|-----------|-------------------------|
| 251 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 187 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 183 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 214 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 172 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 174 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 279 | D4.14 | The validity of values received from external sources shall be checked.<br>Loop is controlled by a value from an unsecure source.<br>Loop may be infinite. | max31865_read_test() | Low | Justified | Loop can't be infinite. |
| 193 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_delay_ms has no effect.  | File Scope           | Low | Justified | delay function.         |
| 167 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 202 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 168 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 184 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 171 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_delay_ms has no effect.  | File Scope           | Low | Justified | delay function.         |
| 170 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 154 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |
| 164 | 2.2   | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope           | Low | Justified | print function.         |

**Table 2.5. E:\Github\max31865\test\driver\_max31865\_register\_test.c**

| ID  | Guideline | Message  | Function   | Severity | Status    | Comment         |
|-----|-----------|--|------------|----------|-----------|-----------------|
| 250 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 189 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 211 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 242 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 217 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 259 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 165 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 238 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 231 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 239 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 226 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 244 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 271 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 227 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |
| 249 | 2.2       | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low      | Justified | print function. |

|     |     |  |            |     |           |                 |
|-----|-----|--|------------|-----|-----------|-----------------|
| 235 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 243 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 224 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 277 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 221 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 247 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 228 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 219 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 232 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 206 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 225 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 261 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 215 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 207 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 274 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 272 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 256 | 2.2 | There shall be no dead code.   | File Scope | Low | Justified | print function. |

|     |      |   |                          |     |           |  |
|-----|------|---|--------------------------|-----|-----------|--|
|     |      | The call to function max31865_interface_debug_print has no effect.  |                          |     |           |  |
| 248 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 163 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 176 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 262 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 265 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 204 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 203 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 199 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 156 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 220 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 273 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 223 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.  | File Scope               | Low | Justified | print function.  |
| 144 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood.<br>Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases.                               | max31865_register_test() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 145 | 10.8 | The value of a composite expression shall not be cast to a different essential type category or a wider essential type.<br>The value of the composite expression of essential type category signed shall not be cast to the different essential type category floating. | max31865_register_test() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |

|     |     |  |            |     |           |                 |
|-----|-----|--|------------|-----|-----------|-----------------|
| 201 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 216 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 208 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 266 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 185 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 212 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 158 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 166 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 269 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 252 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 263 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 268 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 237 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 190 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 191 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 210 | 2.2 | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect. | File Scope | Low | Justified | print function. |
| 188 | 2.2 | There shall be no dead code.   | File Scope | Low | Justified | print function. |



|     |      |  |                          |     |           |  |
|-----|------|--|--------------------------|-----|-----------|--|
|     |      | The call to function max31865_interface_debug_print has no effect.   |                          |     |           |  |
| 260 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 198 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 246 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 179 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 209 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 192 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 181 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 175 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 180 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 264 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 229 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 196 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 146 | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_register_test() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 233 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 230 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |

|     |      |  |                          |     |           |  |
|-----|------|--|--------------------------|-----|-----------|--|
| 234 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 200 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 169 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 147 | 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.<br>The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | max31865_register_test() | Low | Justified | We use this function to convert driver data and drivers guarantee the safety of the operation. |
| 155 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 245 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 222 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 240 | 2.2  | There shall be no dead code.<br>The call to function max31865_interface_debug_print has no effect.   | File Scope               | Low | Justified | print function.  |
| 177 | 2.2  | There shall be no dead code.<br>The call to function max31865_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            | 10/04/2022   |
| -dos             | true   |
| -I               | E:\Github\max31865\src,E:\Github\max31865\interface,E:\Github\max31865\example,E:\Github\max31865\test   |
| -import-comments | E:\Polyspace\max31865\Module\BF_Result\comments_bak  |
| -lang            | C  |

|                             |  |
|-----------------------------|--|
| -little-endian              | true                                   |
| -logical-signed-right-shift | true                                   |
| -misra3                     | mandatory-required                     |
| -prog                       | max31865                               |
| -results-dir                | E:\Polyspace\max31865\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 memcpy, memmove and memcmp shall be pointers to qualified or unqualified versions of compatible types.   | required  | - | yes |
| 21.16 | The pointer arguments to the Standard Library function memcmp 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 <string.h> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters.   | mandatory | - | yes |
| 21.18 | The size_t argument passed to any function in <string.h> shall have an appropriate value.  | mandatory | - | yes |
| 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror 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 asctime, ctime, gmtime, localtime, localeconv, getenv, setlocale or strerror 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 FILE object shall not be dereferenced.  | mandatory | - | yes |
| 22.6  | The value of a pointer to a FILE shall not be used after the associated stream has been closed.  | mandatory | - | yes |
| 22.7  | The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF.   | required  | - | yes |
| 22.8  | The value of errno shall be set to zero prior to a call to an errno-setting-function.  | required  | - | yes |
| 22.9  | The value of errno shall be tested against zero after calling an errno-setting-function.   | required  | - | yes |
| 22.10 | The value of errno shall only be tested when the last function to be called was an errno-setting-function.   | required  | - | yes |

# Chapter 5. Appendix 2 - Definitions

Table 5.1. Abbreviations

| Abbreviation | Definition    |
|--------------|---------------|
| NA           | Not Available |