|  |  |  |
| --- | --- | --- |
|  | **InvenSense Inc.**  1745 Technology Drive, San Jose, CA 95110 U.S.A. Tel: +1 (408) 988-7339 Fax: +1 (408) 988-8104  Website: [www.invensense.com](http://www.invensense.com/) | Document Number: RM-MPU-9250A-00 Revision: 1.4  Release Date: 9/9/2013 |

MPU-9250

Register Map and Descriptions Revision 1.4

|  |  |  |
| --- | --- | --- |
|  | **MPU-9250 Register Map and Descriptions** | Document Number: RM-MPU-9250A-00 Revision: 1.4  Release Date: 9/9/2013 |

**CONTENTS**

## REVISION HISTORY 5

## PURPOSE AND SCOPE 6

## REGISTER MAP FOR GYROSCOPE AND ACCELEROMETER 7

## REGISTER DESCRIPTIONS 10

* 1. REGISTERS 0 TO 2 – GYROSCOPE SELF-TEST REGISTERS 10
  2. REGISTERS 13 TO 15 – ACCELEROMETER SELF-TEST REGISTERS 10
  3. REGISTERS 19 TO 24 – GYRO OFFSET REGISTERS 11
  4. REGISTER 25 – SAMPLE RATE DIVIDER 12
  5. REGISTER 26 – CONFIGURATION 12
  6. REGISTER 27 – GYROSCOPE CONFIGURATION 13
  7. REGISTER 28 – ACCELEROMETER CONFIGURATION 14
  8. REGISTER 29 – ACCELEROMETER CONFIGURATION 2 14
  9. REGISTER 30 – LOW POWER ACCELEROMETER ODR CONTROL 16
  10. REGISTER 31 – WAKE-ON MOTION THRESHOLD 17
  11. REGISTER 35 – FIFO ENABLE 17
  12. REGISTER 36 – I2C MASTER CONTROL 18
  13. REGISTERS 37 TO 39 – I2C SLAVE 0 CONTROL 20

REGISTER 37 - I2C\_SLV0\_ADDR 20

REGISTER 38 - I2C\_SLV0\_REG 20

REGISTER 39 - I2C\_SLV0\_CTRL 20

* 1. REGISTERS 40 TO 42 – I2C SLAVE 1 CONTROL 21

REGISTER 40 - I2C\_SLV1\_ADDR 21

REGISTER 41 - I2C\_SLV1\_REG 22

REGISTER 42 - I2C\_SLV1\_CTRL 22

* 1. REGISTERS 43 TO 45 – I2C SLAVE 2 CONTROL 23

REGISTER 43 - I2C\_SLV2\_ADDR 23

REGISTER 44 - I2C\_SLV2\_REG 24

REGISTER 45 - I2C\_SLV2\_CTRL 24

* 1. REGISTERS 46 TO 48 – I2C SLAVE 3 CONTROL 25

REGISTER 46 - I2C\_SLV3\_ADDR 25

REGISTER 47 - I2C\_SLV3\_REG 25

REGISTER 48 - I2C\_SLV3\_CTRL 25

* 1. REGISTERS 49 TO 53 – I2C SLAVE 4 CONTROL 26

|  |  |  |
| --- | --- | --- |
|  | **MPU-9250 Register Map and Descriptions** | Document Number: RM-MPU-9250A-00 Revision: 1.4  Release Date: 9/9/2013 |

REGISTER 49 - I2C\_SLV4\_ADDR 26

REGISTER 50 - I2C\_SLV4\_REG 26

[REGISTER 51 - I2C\_SLV4\_DO 27](#_TOC_250030)

[REGISTER 52 - I2C\_SLV4\_CTRL 27](#_TOC_250029)

[REGISTER 53 - I2C\_SLV4\_DI 27](#_TOC_250028)

* 1. [REGISTER 54 – I2C MASTER STATUS 28](#_TOC_250027)
  2. [REGISTER 55 – INT PIN / BYPASS ENABLE CONFIGURATION 29](#_TOC_250026)
  3. [REGISTER 56 – INTERRUPT ENABLE 29](#_TOC_250025)
  4. [REGISTER 58 – INTERRUPT STATUS 30](#_TOC_250024)
  5. [REGISTERS 59 TO 64 – ACCELEROMETER MEASUREMENTS 31](#_TOC_250023)
  6. [REGISTERS 65 AND 66 – TEMPERATURE MEASUREMENT 33](#_TOC_250022)
  7. REGISTERS 67 TO 72 – GYROSCOPE MEASUREMENTS 33
  8. [REGISTERS 73 TO 96 – EXTERNAL SENSOR DATA 35](#_TOC_250021)
  9. [REGISTER 99 – I2C SLAVE 0 DATA OUT 37](#_TOC_250020)
  10. REGISTER 100 – I2C SLAVE 1 DATA OUT 37
  11. REGISTER 101 – I2C SLAVE 2 DATA OUT 37
  12. [REGISTER 102 – I2C SLAVE 3 DATA OUT 37](#_TOC_250019)
  13. REGISTER 103 – I2C MASTER DELAY CONTROL 38
  14. REGISTER 104 – SIGNAL PATH RESET 39
  15. [REGISTER 105 – ACCELEROMETER INTERRUPT CONTROL 39](#_TOC_250018)
  16. REGISTER 106 – USER CONTROL 39
  17. [REGISTER 107 – POWER MANAGEMENT 1 40](#_TOC_250017)
  18. [REGISTER 108 – POWER MANAGEMENT 2 41](#_TOC_250016)
  19. REGISTER 114 AND 115 – FIFO COUNT REGISTERS 42
  20. [REGISTER 116 – FIFO READ WRITE 43](#_TOC_250015)
  21. REGISTER 117 – WHO AM I 44
  22. REGISTERS 119, 120, 122, 123, 125, 126 ACCELEROMETER OFFSET REGISTERS 44

1. [REGISTER MAP FOR MAGNETOMETER 47](#_TOC_250014)
   1. [REGISTER MAP DESCRIPTION 48](#_TOC_250013)
   2. [DETAILED DESCRIPTIONS FOR MAGNETOMETER REGISTERS 49](#_TOC_250012)
   3. [WIA: DEVICE ID 49](#_TOC_250011)
   4. [INFO: INFORMATION 49](#_TOC_250010)
   5. [ST1: STATUS 1 49](#_TOC_250009)
   6. [HXL TO HZH: MEASUREMENT DATA 50](#_TOC_250008)
   7. [ST2: STATUS 2 51](#_TOC_250007)
   8. [CNTL1: CONTROL 1 51](#_TOC_250006)
   9. [CNTL2: CONTROL 2 52](#_TOC_250005)
   10. [ASTC: SELF-TEST CONTROL 52](#_TOC_250004)

[5.11 TS1, TS2: TEST 1, 2 52](#_TOC_250003)

* 1. [I2CDIS: I2C DISABLE 52](#_TOC_250002)
  2. [ASAX, ASAY, ASAZ: SENSITIVITY ADJUSTMENT VALUES 53](#_TOC_250001)

1. [ADVANCED HARDWARE FEATURES 54](#_TOC_250000)

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision Date** | **Revision** | **Description** |
| 9/9/2013 | 1.4 | Initial release |

1. **Purpose and Scope**

This document provides preliminary information regarding the register map and descriptions for the Motion Processing Unit™ MPU-9250™. This document should be used in conjunction with the MPU-9250 Product Specification (PS-MPU-9250A-00) for detailed features, specifications, and other product information.

# Register Map for Gyroscope and Accelerometer

The following table lists the register map for the gyroscope and accelerometer in the MPU-9250 MotionTracking device.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr (Hex)** | **Addr (Dec.)** | **Register Name** | **Serial I/F** | **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| 00 | 0 | SELF\_TEST\_X\_GYRO | R/W | xg\_st\_data [7:0] | | | | | | | |
| 01 | 1 | SELF\_TEST\_Y\_GYRO | R/W | yg\_st\_data [7:0] | | | | | | | |
| 02 | 2 | SELF\_TEST\_Z\_GYRO | R/W | zg\_st\_data [7:0] | | | | | | | |
| 0D | 13 | SELF\_TEST\_X\_ACCEL | R/W | XA\_ST\_DATA [7:0] | | | | | | | |
| 0E | 14 | SELF\_TEST\_Y\_ACCEL | R/W | YA\_ST\_DATA [7:0] | | | | | | | |
| 0F | 15 | SELF\_TEST\_Z\_ACCEL | R/W | ZA\_ST\_DATA [7:0] | | | | | | | |
| 13 | 19 | XG\_OFFSET\_H | R/W | X\_OFFS\_USR [15:8] | | | | | | | |
| 14 | 20 | XG\_OFFSET\_L | R/W | X\_OFFS\_USR [7:0] | | | | | | | |
| 15 | 21 | YG\_OFFSET\_H | R/W | Y\_OFFS\_USR [15:8] | | | | | | | |
| 16 | 22 | YG\_OFFSET\_L | R/W | Y\_OFFS\_USR [7:0] | | | | | | | |
| 17 | 23 | ZG\_OFFSET\_H | R/W | Z\_OFFS\_USR [15:8] | | | | | | | |
| 18 | 24 | ZG\_OFFSET\_L | R/W | Z\_OFFS\_USR [7:0] | | | | | | | |
| 19 | 25 | SMPLRT\_DIV | R/W | SMPLRT\_DIV[7:0] | | | | | | | |
| 1A | 26 | CONFIG | R/W | - | FIFO\_ MODE | EXT\_SYNC\_SET[2:0] | | | DLPF\_CFG[2:0] | | |
| 1B | 27 | GYRO\_CONFIG | R/W | XGYRO\_Ct  en | YGYRO\_Ct  en | ZGYRO\_Ct  en | GYRO\_FS\_SEL [1:0] | | - | FCHOICE\_B[1:0] | |
| 1C | 28 | ACCEL\_CONFIG | R/W | ax\_st\_en | ay\_st\_en | az\_st\_en | ACCEL\_FS\_SEL[1:0] | | - | | |
| 1D | 29 | ACCEL\_CONFIG 2 | R/W | - | | | | ACCEL\_FCHOICE\_B | | A\_DLPF\_CFG | |
| 1E | 30 | LP\_ACCEL\_ODR | R/W | - | | | | Lposc\_clksel [3:0] | | | |
| 1F | 31 | WOM\_THR | R/W | WOM\_Threshold [7:0] | | | | | | | |
| 23 | 35 | FIFO\_EN | R/W | TEMP  \_FIFO\_EN | GYRO\_XO UT | GYRO\_YO UT | GYRO\_ZO UT | ACCEL | SLV2 | SLV1 | SLV0 |
| 24 | 36 | I2C\_MST\_CTRL | R/W | MULT  \_MST\_EN | WAIT  \_FOR\_ES | SLV\_3  \_FIFO\_EN | I2C\_MST  \_P\_NSR | I2C\_MST\_CLK[3:0] | | | |
| 25 | 37 | I2C\_SLV0\_ADDR | R/W | I2C\_SLV0  \_RNW | I2C\_ID\_0 [6:0] | | | | | | |
| 26 | 38 | I2C\_SLV0\_REG | R/W | I2C\_SLV0\_REG[7:0] | | | | | | | |
| 27 | 39 | I2C\_SLV0\_CTRL | R/W | I2C\_SLV0  \_EN | I2C\_SLV0  \_BYTE\_SW | I2C\_SLV0  \_REG\_DIS | I2C\_SLV0  \_GRP | I2C\_SLV0\_LENG[3:0] | | | |
| 28 | 40 | I2C\_SLV1\_ADDR | R/W | I2C\_SLV1  \_RNW | I2C\_ID\_1 [6:0] | | | | | | |
| 29 | 41 | I2C\_SLV1\_REG | R/W | I2C\_SLV1\_REG[7:0] | | | | | | | |
| 2A | 42 | I2C\_SLV1\_CTRL | R/W | I2C\_SLV1  \_EN | I2C\_SLV1  \_BYTE\_SW | I2C\_SLV1  \_REG\_DIS | I2C\_SLV1  \_GRP | I2C\_SLV1\_LENG[3:0] | | | |
| 2B | 43 | I2C\_SLV2\_ADDR | R/W | I2C\_SLV2  \_RNW | I2C\_ID\_2 [6:0] | | | | | | |
| 2C | 44 | I2C\_SLV2\_REG | R/W | I2C\_SLV2\_REG[7:0] | | | | | | | |
| 2D | 45 | I2C\_SLV2\_CTRL | R/W | I2C\_SLV2  \_EN | I2C\_SLV2  \_BYTE\_SW | I2C\_SLV2  \_REG\_DIS | I2C\_SLV2  \_GRP | I2C\_SLV2\_LENG[3:0] | | | |
| 2E | 46 | I2C\_SLV3\_ADDR | R/W | I2C\_SLV3  \_RNW | I2C\_ID\_3 [6:0] | | | | | | |
| 2F | 47 | I2C\_SLV3\_REG | R/W | I2C\_SLV3\_REG[7:0] | | | | | | | |
| 30 | 48 | I2C\_SLV3\_CTRL | R/W | I2C\_SLV3  \_EN | I2C\_SLV3  \_BYTE\_SW | I2C\_SLV3  \_REG\_DIS | I2C\_SLV3  \_GRP | I2C\_SLV3\_LENG [3:0] | | | |
| 31 | 49 | I2C\_SLV4\_ADDR | R/W | I2C\_SLV4  \_RNW | I2C\_ID\_4 [6:0] | | | | | | |
| 32 | 50 | I2C\_SLV4\_REG | R/W | I2C\_SLV4\_REG[7:0] | | | | | | | |
| 33 | 51 | I2C\_SLV4\_DO | R/W | I2C\_SLV4\_DO[7:0] | | | | | | | |
| 34 | 52 | I2C\_SLV4\_CTRL | R/W | I2C\_SLV4  \_EN | SLV4\_DON E\_INT\_EN | I2C\_SLV4  \_REG\_DIS | I2C\_MST\_DLY[4:0] | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr (Hex)** | **Addr (Dec.)** | **Register Name** | **Serial I/F** | **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| 35 | 53 | I2C\_SLV4\_DI | R | I2C\_SLV4\_DI[7:0] | | | | | | | |
| 36 | 54 | I2C\_MST\_STATUS | R | PASS\_ THROUGH | I2C\_SLV4  \_DONE | I2C\_LOST  \_ARB | I2C\_SLV4  \_NACK | I2C\_SLV3  \_NACK | I2C\_SLV2  \_NACK | I2C\_SLV1  \_NACK | I2C\_SLV0  \_NACK |
| 37 | 55 | INT\_PIN\_CFG | R/W | ACTL | OPEN | LATCH  \_INT\_EN | INT\_ANYR D  \_2CLEAR | ACTL\_FSY NC | FSYNC  \_INT\_MOD E\_EN | BYPASS  \_EN | - |
| 38 | 56 | INT\_ENABLE | R/W | - | WOM\_EN | - | FIFO  \_OFLOW  \_EN | FSYNC\_INT  \_EN | - | - | RAW\_RDY\_ EN |
| 3A | 58 | INT\_STATUS | R | - | WOM\_INT | - | FIFO  \_OFLOW  \_INT | FSYNC  \_INT | - | - | RAW\_DATA  \_RDY\_INT |
| 3B | 59 | ACCEL\_XOUT\_H | R | ACCEL\_XOUT\_H[15:8] | | | | | | | |
| 3C | 60 | ACCEL\_XOUT\_L | R | ACCEL\_XOUT\_L[7:0] | | | | | | | |
| 3D | 61 | ACCEL\_YOUT\_H | R | ACCEL\_YOUT\_H[15:8] | | | | | | | |
| 3E | 62 | ACCEL\_YOUT\_L | R | ACCEL\_YOUT\_L[7:0] | | | | | | | |
| 3F | 63 | ACCEL\_ZOUT\_H | R | ACCEL\_ZOUT\_H[15:8] | | | | | | | |
| 40 | 64 | ACCEL\_ZOUT\_L | R | ACCEL\_ZOUT\_L[7:0] | | | | | | | |
| 41 | 65 | TEMP\_OUT\_H | R | TEMP\_OUT\_H[15:8] | | | | | | | |
| 42 | 66 | TEMP\_OUT\_L | R | TEMP\_OUT\_L[7:0] | | | | | | | |
| 43 | 67 | GYRO\_XOUT\_H | R | GYRO\_XOUT\_H[15:8] | | | | | | | |
| 44 | 68 | GYRO\_XOUT\_L | R | GYRO\_XOUT\_L[7:0] | | | | | | | |
| 45 | 69 | GYRO\_YOUT\_H | R | GYRO\_YOUT\_H[15:8] | | | | | | | |
| 46 | 70 | GYRO\_YOUT\_L | R | GYRO\_YOUT\_L[7:0] | | | | | | | |
| 47 | 71 | GYRO\_ZOUT\_H | R | GYRO\_ZOUT\_H[15:8] | | | | | | | |
| 48 | 72 | GYRO\_ZOUT\_L | R | GYRO\_ZOUT\_L[7:0] | | | | | | | |
| 49 | 73 | EXT\_SENS\_DATA\_00 | R | EXT\_SENS\_DATA\_00[7:0] | | | | | | | |
| 4A | 74 | EXT\_SENS\_DATA\_01 | R | EXT\_SENS\_DATA\_01[7:0] | | | | | | | |
| 4B | 75 | EXT\_SENS\_DATA\_02 | R | EXT\_SENS\_DATA\_02[7:0] | | | | | | | |
| 4C | 76 | EXT\_SENS\_DATA\_03 | R | EXT\_SENS\_DATA\_03[7:0] | | | | | | | |
| 4D | 77 | EXT\_SENS\_DATA\_04 | R | EXT\_SENS\_DATA\_04[7:0] | | | | | | | |
| 4E | 78 | EXT\_SENS\_DATA\_05 | R | EXT\_SENS\_DATA\_05[7:0] | | | | | | | |
| 4F | 79 | EXT\_SENS\_DATA\_06 | R | EXT\_SENS\_DATA\_06[7:0] | | | | | | | |
| 50 | 80 | EXT\_SENS\_DATA\_07 | R | EXT\_SENS\_DATA\_07[7:0] | | | | | | | |
| 51 | 81 | EXT\_SENS\_DATA\_08 | R | EXT\_SENS\_DATA\_08[7:0] | | | | | | | |
| 52 | 82 | EXT\_SENS\_DATA\_09 | R | EXT\_SENS\_DATA\_09[7:0] | | | | | | | |
| 53 | 83 | EXT\_SENS\_DATA\_10 | R | EXT\_SENS\_DATA\_10[7:0] | | | | | | | |
| 54 | 84 | EXT\_SENS\_DATA\_11 | R | EXT\_SENS\_DATA\_11[7:0] | | | | | | | |
| 55 | 85 | EXT\_SENS\_DATA\_12 | R | EXT\_SENS\_DATA\_12[7:0] | | | | | | | |
| 56 | 86 | EXT\_SENS\_DATA\_13 | R | EXT\_SENS\_DATA\_13[7:0] | | | | | | | |
| 57 | 87 | EXT\_SENS\_DATA\_14 | R | EXT\_SENS\_DATA\_14[7:0] | | | | | | | |
| 58 | 88 | EXT\_SENS\_DATA\_15 | R | EXT\_SENS\_DATA\_15[7:0] | | | | | | | |
| 59 | 89 | EXT\_SENS\_DATA\_16 | R | EXT\_SENS\_DATA\_16[7:0] | | | | | | | |
| 5A | 90 | EXT\_SENS\_DATA\_17 | R | EXT\_SENS\_DATA\_17[7:0] | | | | | | | |
| 5B | 91 | EXT\_SENS\_DATA\_18 | R | EXT\_SENS\_DATA\_18[7:0] | | | | | | | |
| 5C | 92 | EXT\_SENS\_DATA\_19 | R | EXT\_SENS\_DATA\_19[7:0] | | | | | | | |
| 5D | 93 | EXT\_SENS\_DATA\_20 | R | EXT\_SENS\_DATA\_20[7:0] | | | | | | | |
| 5E | 94 | EXT\_SENS\_DATA\_21 | R | EXT\_SENS\_DATA\_21[7:0] | | | | | | | |
| 5F | 95 | EXT\_SENS\_DATA\_22 | R | EXT\_SENS\_DATA\_22[7:0] | | | | | | | |
| 60 | 96 | EXT\_SENS\_DATA\_23 | R | EXT\_SENS\_DATA\_23[7:0] | | | | | | | |
| 63 | 99 | I2C\_SLV0\_DO | R/W | I2C\_SLV0\_DO[7:0] | | | | | | | |
| 64 | 100 | I2C\_SLV1\_DO | R/W | I2C\_SLV1\_DO[7:0] | | | | | | | |
| 65 | 101 | I2C\_SLV2\_DO | R/W | I2C\_SLV2\_DO[7:0] | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr (Hex)** | **Addr (Dec.)** | **Register Name** | **Serial I/F** | **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| 66 | 102 | I2C\_SLV3\_DO | R/W | I2C\_SLV3\_DO[7:0] | | | | | | | |
| 67 | 103 | I2C\_MST\_DELAY\_CTRL | R/W | DELAY\_ES  \_SHADOW | - | - | I2C\_SLV4  \_DLY\_EN | I2C\_SLV3  \_DLY\_EN | I2C\_SLV2  \_DLY\_EN | I2C\_SLV1  \_DLY\_EN | I2C\_SLV0  \_DLY\_EN |
| 68 | 104 | SIGNAL\_PATH\_RESET | R/W | - | - | - | - | - | GYRO  \_RST | ACCEL  \_RST | TEMP  \_RST |
| 69 | 105 | MOT\_DETECT\_CTRL | R/W | ACCEL\_INT EL\_EN | ACCEL\_INT EL\_MODE | - | | - | | - | |
| 6A | 106 | USER\_CTRL | R/W | - | FIFO\_EN | I2C\_MST  \_EN | I2C\_IF  \_DIS | - | FIFO  \_RST | I2C\_MST  \_RST | SIG\_COND  \_RST |
| 6B | 107 | PWR\_MGMT\_1 | R/W | H\_RESET | SLEEP | CYCLE | GYRO\_ STANDBY | PD\_PTAT | CLKSEL[2:0] | | |
| 6C | 108 | PWR\_MGMT\_2 | R/W | - | | DIS\_XA | DIS\_YA | DIS\_ZA | DIS\_XG | DIS\_YG | DIS\_ZG |
| 72 | 114 | FIFO\_COUNTH | R/W | - | | | FIFO\_CNT[12:8] | | | | |
| 73 | 115 | FIFO\_COUNTL | R/W | FIFO\_CNT[7:0] | | | | | | | |
| 74 | 116 | FIFO\_R\_W | R/W | D[7:0] | | | | | | | |
| 75 | 117 | WHO\_AM\_I | R | WHOAMI[7:0] | | | | | | | |
| 77 | 119 | XA\_OFFSET\_H | R/W | XA\_OFFS [14:7] | | | | | | | |
| 78 | 120 | XA\_OFFSET\_L | R/W | XA\_OFFS [6:0] | | | | | | | - |
| 7A | 122 | YA\_OFFSET\_H | R/W | YA\_OFFS [14:7] | | | | | | | |
| 7B | 123 | YA\_OFFSET\_L | R/W | YA\_OFFS [6:0] | | | | | | | - |
| 7D | 125 | ZA\_OFFSET\_H | R/W | ZA\_OFFS [14:7] | | | | | | | |
| 7E | 126 | ZA\_OFFSET\_L | R/W | ZA\_OFFS [6:0] | | | | | | | - |

## Table 1 MPU-9250 mode register map for Gyroscope and Accelerometer

Note: Register Names ending in \_H and \_L contain the high and low bytes, respectively, of an internal register value.

In the detailed register tables that follow, register names are in capital letters, while register values are in capital letters and italicized. For example, the ACCEL\_XOUT\_H register (Register 59) contains the 8 most significant bits, *ACCEL\_XOUT*[15:8], of the 16-bit X-Axis accelerometer measurement, *ACCEL\_XOUT*.

The reset value is 0x00 for all registers other than the registers below.

* + Register 107 (0x01) Power Management 1
  + Register 117 (0x71) WHO\_AM\_I

# Register Descriptions

This section describes the function and contents of each register within the MPU-9250. All the descriptions relate to the default MPU-9250 mode of operation.

## Registers 0 to 2 – Gyroscope Self-Test Registers Serial IF: R/W

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **REGISTER** | **BITS** | **FUNCTION** |
| SELF\_TEST\_X\_GYRO | XG\_ST\_DATA[7:0] | The value in this register indicates the self test output generated during manufacturing tests. This value is to be used to check against subsequent self test outputs performed by the end user. |
| SELF\_TEST\_Y\_GYRO | YG\_ST\_DATA[7:0] | The value in this register indicates the self test output generated during manufacturing tests. This value is to be used to check against subsequent self test outputs performed by the end user. |
| SELF\_TEST\_Z\_GYRO | ZG\_ST\_DATA[7:0] | The value in this register indicates the self test output generated during manufacturing tests. This value is to be used to check against subsequent self test outputs performed by the end user. |

For details of the MPU-9250 self-test implementation, please refer to the following document: AN- MPU-9250A-03, MPU-9250 Accelerometer, Gyroscope and Compass Self-Test Implementation.

## Registers 13 to 15 – Accelerometer Self-Test Registers

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **REGISTER** | **BITS** | **FUNCTION** |
| SELF\_TEST\_X\_ACCEL | XA\_ST\_DATA[7:0] | The value in this register indicates the self test output generated during manufacturing tests. This value is to be used to check against subsequent self test outputs performed by the end user. |
| SELF\_TEST\_Y\_ACCEL | YA\_ST\_DATA[7:0] | The value in this register indicates the self test output generated during manufacturing tests. This value is to be used to check against subsequent self test outputs performed by the end user. |
| SELF\_TEST\_Z\_ACCEL | ZA\_ST\_DATA[7:0] | The value in this register indicates the self test output generated during manufacturing tests. This value is to be used to check against subsequent self test outputs performed by the end user. |

For details of the MPU-9250 self-test implementation, please refer to the following document: AN-MPU- 9250A-03, MPU-9250 Accelerometer, Gyroscope and Compass Self-Test Implementation.

## Registers 19 to 24 – Gyro Offset Registers

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| [7:0] | X\_OFFS\_USR[15:8] | High byte, Low byte in USR register (14h)  **OffsetLSB=** X\_OFFS\_USR \* 4 / 2^FS\_SEL  **OffsetDPS=** X\_OFFS\_USR \* 4 / 2^FS\_SEL / Gyro\_Sensitivity Nominal FS\_SEL = 0  Conditions Gyro\_Sensitivity = 2^16 LSB / 500dps Max 999.969 dps  Min -1000 dps  Step 0.0305 dps |
| [7:0] | X\_OFFS\_USR[7:0] | Low byte, High byte in USR register (13h) |
| [7:0] | Y\_OFFS\_USR[15:8] | High byte, Low byte in USR register (16h)  **OffsetLSB=** Y\_OFFS\_USR \* 4 / 2^FS\_SEL  **OffsetDPS=** Y\_OFFS\_USR \* 4 / 2^FS\_SEL / Gyro\_Sensitivity Nominal FS\_SEL = 0  Conditions Gyro\_Sensitivity = 2^16 LSB / 500dps Max 999.969 dps  Min -1000 dps  Step 0.0305 dps |
| [7:0] | Y\_OFFS\_USR[7:0] | Low byte, High byte in USR register (15h) |
| [7:0] | Z\_OFFS\_USR[15:8] | High byte, Low byte in USR register (18h)  **OffsetLSB=** Z\_OFFS\_USR \* 4 / 2^FS\_SEL  **OffsetDPS=** Z\_OFFS\_USR \* 4 / 2^FS\_SEL / Gyro\_Sensitivity Nominal FS\_SEL = 0  Conditions Gyro\_Sensitivity = 2^16 LSB / 500dps Max 999.969 dps  Min -1000 dps  Step 0.0305 dps |
| [7:0] | Z\_OFFS\_USR[7:0] | Low byte, High byte in USR register (17h) |

These registers are used to remove DC bias from the gyro sensor data output for X, Y and Z axes. The values in these registers are subtracted from the gyro sensor values before going into the sensor registers. Please refer to registers 67 to 72 for units.

## Register 25 – Sample Rate Divider Serial IF: R/W

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| [7:0] | SMPLRT\_DIV[7:0] | Divides the internal sample rate (see register CONFIG) to generate the sample rate that controls sensor data output rate, FIFO sample rate. NOTE: This register is only effective when Fchoice = 2’b11 (fchoice\_b register bits are 2’b00), and (0 < dlpf\_cfg < 7), such that the average filter’s output is selected (see chart below).  This is the update rate of sensor register.  **SAMPLE\_RATE=** Internal\_Sample\_Rate / (1 + SMPLRT\_DIV) |

Data should be sampled at or above sample rate; SMPLRT\_DIV is only used for1kHz internal sampling.

## Register 26 – Configuration

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| [7] | - | Reserved |
| [6] | FIFO\_MODE | When set to ‘1’, when the fifo is full, additional writes will not be written to fifo. When set to ‘0’, when the fifo is full, additional writes will be written to the fifo, replacing the oldest data. |

|  |  |
| --- | --- |
| **EXT\_SYNC\_SET** | **FSYNC bit location** |
| 0 | function disabled |
| 1 | TEMP\_OUT\_L[0] |
| 2 | GYRO\_XOUT\_L[0] |
| 3 | GYRO\_YOUT\_L[0] |
| 4 | GYRO\_ZOUT\_L[0] |
| 5 | ACCEL\_XOUT\_L[0] |
| 6 | ACCEL\_YOUT\_L[0] |
| 7 | ACCEL\_ZOUT\_L[0] |

|  |  |  |
| --- | --- | --- |
| [5:3] | EXT\_SYNC\_SET[2:0] | Enables the FSYNC pin data to be sampled.  Fsync will be latched to capture short strobes. This will be done such that if Fsync toggles, the latched value toggles, but won’t toggle again until the new latched value is captured by the sample rate strobe. This is a requirement for working with some 3rd party devices that have fsync strobes shorter than our sample rate. |
| [2:0] | DLPF\_CFG[2:0] | For the DLPF to be used, fchoice[1:0] must be set to 2’b11, fchoice\_b[1:0] is 2’b00.  See table 3 below. |

The DLPF is configured by *DLPF\_CFG,* when *FCHOICE\_B* [1:0] = 2b’00. The gyroscope and temperature sensor are filtered according to the value of *DLPF\_CFG* and *FCHOICE\_B* as shown in the table below. Note that FCHOICE mentioned in the table below is the inverted value of *FCHOICE\_B* (e.g. FCHOICE=2b’00 is same as FCHOICE\_B=2b’11).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **FCHOICE** | | **DLPF\_CFG** | **Gyroscope** | | | **Temperature Sensor** | |
| **<1>** | **<0>** | **Bandwidth (Hz)** | **Delay (ms)** | **Fs (kHz)** | **Bandwidth (Hz)** | **Delay (ms)** |
| x | 0 | x | 8800 | 0.064 | 32 | 4000 | 0.04 |
| 0 | 1 | x | 3600 | 0.11 | 32 | 4000 | 0.04 |
| 1 | 1 | 0 | 250 | 0.97 | 8 | 4000 | 0.04 |
| 1 | 1 | 1 | 184 | 2.9 | 1 | 188 | 1.9 |
| 1 | 1 | 2 | 92 | 3.9 | 1 | 98 | 2.8 |
| 1 | 1 | 3 | 41 | 5.9 | 1 | 42 | 4.8 |
| 1 | 1 | 4 | 20 | 9.9 | 1 | 20 | 8.3 |
| 1 | 1 | 5 | 10 | 17.85 | 1 | 10 | 13.4 |
| 1 | 1 | 6 | 5 | 33.48 | 1 | 5 | 18.6 |
| 1 | 1 | 7 | 3600 | 0.17 | 8 | 4000 | 0.04 |

## Register 27 – Gyroscope Configuration

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | XGYRO\_Cten | X Gyro self-test |
| **[6]** | YGYRO\_Cten | Y Gyro self-test |
| **[5]** | ZGYRO\_Cten | Z Gyro self-test |
| **[4:3]** | GYRO\_FS\_SEL[1:0] | Gyro Full Scale Select: 00 = +250dps  01= +500 dps  10 = +1000 dps  11 = +2000 dps |
| **[2]** | - | Reserved |
| **[1:0]** | Fchoice\_b[1:0] | Used to bypass DLPF as shown in table 1 above. NOTE: Register is Fchoice\_b (inverted version of Fchoice), table 1 uses Fchoice (which is the inverted version of this register). |

* 1. **Register 28 – Accelerometer Configuration Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | ax\_st\_en | X Accel self-test |
| **[6]** | ay\_st\_en | Y Accel self-test |
| **[5]** | az\_st\_en | Z Accel self-test |
| **[4:3]** | ACCEL\_FS\_SEL[1:0] | Accel Full Scale Select:  ±2g (00), ±4g (01), ±8g (10), ±16g (11) |
| **[2:0]** | - | Reserved |

* 1. **Register 29 – Accelerometer Configuration 2 Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:6]** | Reserved |  |
| **[5:4]** | Reserved |  |
| **[3]** | accel\_fchoice\_b | Used to bypass DLPF as shown in table 2 below. NOTE: This register contains accel\_fchoice\_b (the inverted version of accel\_fchoice as described in the table below). |
| **[2:0]** | A\_DLPFCFG | Accelerometer low pass filter setting as shown in table 2 below. |

**Accelerometer Data Rates and Bandwidths (Normal Mode)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ACCEL\_FCHOICE** | **A\_DLPF\_CFG** | **Output** | | | |
| **Bandwidth (Hz)** | **Delay (ms)** | **Noise Density**  **(ug/rtHz)** | **Rate (kHz)** |
| 0 | X | 1.13 K | 0.75 | 250 | 4 |
| 1 | 0 | 460 | 1.94 | 250 | 1 |
| 1 | 1 | 184 | 5.80 | 250 | 1 |
| 1 | 2 | 92 | 7.80 | 250 | 1 |
| 1 | 3 | 41 | 11.80 | 250 | 1 |
| 1 | 4 | 20 | 19.80 | 250 | 1 |
| 1 | 5 | 10 | 35.70 | 250 | 1 |
| 1 | 6 | 5 | 66.96 | 250 | 1 |
| 1 | 7 | 460 | 1.94 | 250 | 1 |

The data output rate of the DLPF filter block can be further reduced by a factor of 1/(1+SMPLRT\_DIV), where SMPLRT\_DIV is an 8-bit integer. Following is a small subset of ODRs that are configurable for the accelerometer in the normal mode in this manner (Hz):

3.91, 7.81, 15.63, 31.25, 62.50, 125, 250, 500, 1K

The following table lists the approximate accelerometer filter bandwidths available in the low-power mode of operation.

In the low-power mode of operation, the accelerometer is duty-cycled. Fchoice=0 for all options.

## Accelerometer Data Rates and Bandwidths (Low-Power Mode)

|  |  |  |  |
| --- | --- | --- | --- |
| **ACCEL\_FCHOICE** | **ODR (Hz)** | **Output** | |
| **Bandwidth**  **(Hz)** | **Delay**  **(ms)** |
| 0 | 0.24 | 1.1 k | 1 |
| 0 | 0.49 | 1.1 k | 1 |
| 0 | 0.98 | 1.1 k | 1 |
| 0 | 1.95 | 1.1 k | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 3.91 | 1.1 k | 1 |
| 0 | 7.81 | 1.1 k | 1 |
| 0 | 15.63 | 1.1 k | 1 |
| 0 | 31.25 | 1.1 k | 1 |
| 0 | 62.50 | 1.1 k | 1 |
| 0 | 125 | 1.1 k | 1 |
| 0 | 250 | 1.1 k | 1 |
| 0 | 500 | 1.1 kHz | 1 |

As you can see from the tables above, some of the ODRs can be configured in the normal accelerometer mode as well as low power mode.

For further details on how to configure the individual ODRs, please refer to register 30 Low Power Accelerometer ODR Control.

## Register 30 – Low Power Accelerometer ODR Control Serial IF: R/W

**Reset value: 0x00**

|  |  |
| --- | --- |
| **Lposc\_clksel** | **Output Frequency (Hz)** |
| 0 | 0.24 |
| 1 | 0.49 |
| 2 | 0.98 |
| 3 | 1.95 |
| 4 | 3.91 |
| 5 | 7.81 |
| 6 | 15.63 |
| 7 | 31.25 |
| 8 | 62.50 |
| 9 | 125 |
| 10 | 250 |
| 11 | 500 |
| 12-15 | RESERVED |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:4]** | Reserved |  |
| **[3:0]** | lposc\_clksel[3:0] | Sets the frequency of waking up the chip to take a sample of accel data – the low power accel Output Data Rate. |

* 1. **Register 31 – Wake-on Motion Threshold**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | WOM\_Threshold | This register holds the threshold value for the Wake on Motion Interrupt for accel x/y/z axes. LSB = 4mg. Range is 0mg to 1020mg. |

For more details on how to configure the Wake-on-Motion interrupt, please refer to section 5 in the MPU-9250 Product Specification document.

## Register 35 – FIFO Enable Serial IF: R/W

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | TEMP\_OUT | 1 – Write TEMP\_OUT\_H and TEMP\_OUT\_L to the FIFO at the sample rate; If enabled, buffering of data occurs even if data path is in standby.  0 – function is disabled |
| **[6]** | GYRO\_XOUT | 1 – Write GYRO\_XOUT\_H and GYRO\_XOUT\_L to the FIFO at the sample rate; If enabled, buffering of data occurs even if data path is in standby.  0 – function is disabled |
| **[5]** | GYRO\_YOUT | 1 – Write GYRO\_YOUT\_H and GYRO\_YOUT\_L to the FIFO at the sample rate; If enabled, buffering of data occurs even if data path is in standby.  0 – function is disabled  NOTE: Enabling any one of the bits corresponding to the Gyros or Temp data paths, data is buffered into the FIFO even though that data path is not enabled. |
| **[4]** | GYRO\_ZOUT | 1 – Write GYRO\_ZOUT\_H and GYRO\_ZOUT\_L to the FIFO at the sample rate; If enabled, buffering of data occurs even if data path is in standby.  0 – function is disabled |
| **[3]** | ACCEL | 1 – write ACCEL\_XOUT\_H, ACCEL\_XOUT\_L, ACCEL\_YOUT\_H,  ACCEL\_YOUT\_L, ACCEL\_ZOUT\_H, and ACCEL\_ZOUT\_L to the FIFO at the sample rate;  0 – function is disabled |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[2]** | SLV\_2 | 1 – write EXT\_SENS\_DATA registers associated to SLV\_2 (as determined by I2C\_SLV0\_CTRL, I2C\_SLV1\_CTRL, and I2C\_SL20\_CTRL) to the FIFO at  the sample rate;  0 – function is disabled |
| **[1]** | SLV\_1 | 1 – write EXT\_SENS\_DATA registers associated to SLV\_1 (as determined by I2C\_SLV0\_CTRL and I2C\_SLV1\_CTRL) to the FIFO at the sample rate;  0 – function is disabled |
| **[0]** | SLV\_0 | 1 – write EXT\_SENS\_DATA registers associated to SLV\_0 (as determined by I2C\_SLV0\_CTRL) to the FIFO at the sample rate;  0 – function is disabled  NOTE: See I2C\_SLV3\_CTRL register to enable this feature for SLV\_3 |

Note: For further information regarding the association of EXT\_SENS\_DATA registers to particular slave devices, please refer to Registers 73 to 96.

## Register 36 – I2C Master Control

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | MULT\_MST\_EN | Enables multi-master capability. When disabled, clocking to the I2C\_MST\_IF can be disabled when not in use and the logic to detect lost arbitration is disabled. |
| **[6]** | WAIT\_FOR\_ES | Delays the data ready interrupt until external sensor data is loaded. If I2C\_MST\_IF is disabled, the interrupt will still occur. |
| **[5]** | SLV\_3\_FIFO\_EN | 1 – write EXT\_SENS\_DATA registers associated to SLV\_3 (as determined by I2C\_SLV0\_CTRL and I2C\_SLV1\_CTRL and I2C\_SLV2\_CTRL) to the FIFO at the sample rate;  0 – function is disabled |
| **[4]** | I2C\_MST\_P\_NSR | This bit controls the I2C Master’s transition from one slave read to the next slave read. If 0, there is a restart between reads. If 1, there is a stop between reads. |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[3:0]** | I2C\_MST\_CLK [3:0] | *I2C\_MST\_CLK* is a 4 bit unsigned value which configures a divider on the MPU- 9250 internal 8MHz clock. It sets the I2C master clock speed according to the following table: |

Note: For further information regarding the association of EXT\_SENS\_DATA registers to particular slave devices, please refer to Registers 73 to 96.

|  |  |  |
| --- | --- | --- |
| **I2C\_MST\_CLK** | **I2C Master Clock Speed** | **8MHz Clock Divider** |
| 0 | 348 kHz | 23 |
| 1 | 333 kHz | 24 |
| 2 | 320 kHz | 25 |
| 3 | 308 kHz | 26 |
| 4 | 296 kHz | 27 |
| 5 | 286 kHz | 28 |
| 6 | 276 kHz | 29 |
| 7 | 267 kHz | 30 |
| 8 | 258 kHz | 31 |
| 9 | 500 kHz | 16 |
| 10 | 471 kHz | 17 |
| 11 | 444 kHz | 18 |
| 12 | 421 kHz | 19 |
| 13 | 400 kHz | 20 |
| 14 | 381 kHz | 21 |
| 15 | 364 kHz | 22 |

## Registers 37 to 39 – I2C Slave 0 Control Register 37 - I2C\_SLV0\_ADDR

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV0\_RNW | 1 – Transfer is a read  0 – Transfer is a write |
| **[6:0]** | I2C\_ID\_0[6:0] | Physical address of I2C slave 0 |

**Register 38 - I2C\_SLV0\_REG**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV0\_REG[7:0] | I2C slave 0 register address from where to begin data transfer |

**Register 39 - I2C\_SLV0\_CTRL Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV0\_EN | 1 – Enable reading data from this slave at the sample rate and storing data at the first available EXT\_SENS\_DATA register, which is always EXT\_SENS\_DATA\_00 for I2C slave 0.  0 – function is disabled for this slave |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[6]** | I2C\_SLV0\_BYTE\_SW | 1 – Swap bytes when reading both the low and high byte of a word. Note there is nothing to swap after reading the first byte if I2C\_SLV0\_REG[0] = 1, or if the last byte read has a register address lsb = 0.  For example, if I2C\_SLV0\_REG = 0x1, and I2C\_SLV0\_LENG = 0x4:   1. The first byte read from address 0x1 will be stored at EXT\_SENS\_DATA\_00, 2. the second and third bytes will be read and swapped, so the data read from address 0x2 will be stored at EXT\_SENS\_DATA\_02, and the data read from address 0x3 will be stored at EXT\_SENS\_DATA\_01, 3. The last byte read from address 0x4 will be stored at EXT\_SENS\_DATA\_03   0 – no swapping occurs, bytes are written in order read. |
| **[5]** | I2C\_SLV0\_REG\_DIS | When set, the transaction does not write a register value, it will only read data, or write data |
| **[4]** | I2C\_SLV0\_GRP | External sensor data typically comes in as groups of two bytes. This bit is used to determine if the groups are from the slave’s register address 0 and 1, 2 and 3, etc.., or if the groups are address 1 and 2, 3  and 4, etc..  0 indicates slave register addresses 0 and 1 are grouped together (odd numbered register ends the group). 1 indicates slave register addresses 1 and 2 are grouped together (even numbered register ends the group). This allows byte swapping of registers that are grouped starting at any address. |
| **[3:0]** | I2C\_SLV0\_LENG[3:0] | Number of bytes to be read from I2C slave 0 |

* 1. **Registers 40 to 42 – I2C Slave 1 Control Register 40 - I2C\_SLV1\_ADDR**

**Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV1\_RNW | 1 – Transfer is a read  0 – Transfer is a write |
| **[6:0]** | I2C\_ID\_1[6:0] | Physical address of I2C slave 1 |

**Register 41 - I2C\_SLV1\_REG**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV1\_REG[7:0] | I2C slave 1 register address from where to begin data transfer |

**Register 42 - I2C\_SLV1\_CTRL**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV1\_EN | 1 – Enable reading data from this slave at the sample rate and storing data at the first available EXT\_SENS\_DATA register as determined by I2C\_SLV1\_EN and I2C\_SLV1\_LENG.  0 – function is disabled for this slave |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[6]** | I2C\_SLV1\_BYTE\_SW | 1 – Swap bytes when reading both the low and high byte of a word. Note there is nothing to swap after reading the first byte if I2C\_SLV1\_REG[0] = 1, or if the last byte read has a register address lsb = 0.  For example, if I2C\_SLV1\_EN = 0x1, and I2C\_SLV1\_LENG  = 0x3 (to show swap has to do with I2C slave address not EXT\_SENS\_DATA address), and if I2C\_SLV1\_REG = 0x1, and I2C\_SLV1\_LENG = 0x4:   1. The first byte read from address 0x1 will be stored at EXT\_SENS\_DATA\_03 (slave 0’s data will be in EXT\_SENS\_DATA\_00, EXT\_SENS\_DATA\_01, and EXT\_SENS\_DATA\_02), 2. the second and third bytes will be read and swapped, so the data read from address 0x2 will be stored at EXT\_SENS\_DATA\_04, and the data read from address 0x3 will be stored at EXT\_SENS\_DATA\_05, 3. The last byte read from address 0x4 will be stored at EXT\_SENS\_DATA\_06   0 – no swapping occurs, bytes are written in order read. |
| **[5]** | I2C\_SLV1\_REG\_DIS | When set, the transaction does not write a register value, it will only read data, or write data |
| **[4]** | I2C\_SLV1\_GRP | External sensor data typically comes in as groups of two bytes. This bit is used to determine if the groups are from the slave’s register address 0 and 1, 2 and 3, etc.., or if the groups are address 1 and 2, 3 and 4, etc..  0 indicates slave register addresses 0 and 1 are grouped together (odd numbered register ends the group). 1 indicates slave register addresses 1 and 2 are grouped together (even numbered register ends the group). This allows byte swapping of registers that are grouped starting at any address. |
| **[3:0]** | I2C\_SLV1\_LENG[3:0] | Number of bytes to be read from I2C slave 1 |

* 1. **Registers 43 to 45 – I2C Slave 2 Control Register 43 - I2C\_SLV2\_ADDR**

**Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV2\_RNW | 1 – Transfer is a read  0 – Transfer is a write |
| **[6:0]** | I2C\_ID\_2[6:0] | Physical address of I2C slave 2 |

**Register 44 - I2C\_SLV2\_REG**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV2\_REG[7:0] | I2C slave 2 register address from where to begin data transfer |

**Register 45 - I2C\_SLV2\_CTRL**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV2\_EN | 1 – Enable reading data from this slave at the sample rate and storing data at the first available EXT\_SENS\_DATA register as determined by I2C\_SLV0\_EN, I2C\_SLV0\_LENG, I2C\_SLV1\_EN and I2C\_SLV1\_LENG.  0 – function is disabled for this slave |
| **[6]** | I2C\_SLV2\_BYTE\_SW | 1 – Swap bytes when reading both the low and high byte of a word. Note there is nothing to swap after reading the first byte if I2C\_SLV2\_REG[0] = 1, or if the last byte read has a register address lsb = 0.  See I2C\_SLV1\_CTRL for an example.  0 – no swapping occurs, bytes are written in order read. |
| **[5]** | I2C\_SLV2\_REG\_DIS | When set, the transaction does not write a register value, it will only read data, or write data |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[4]** | I2C\_SLV2\_GRP | External sensor data typically comes in as groups of two bytes. This bit is used to determine if the groups are from the slave’s register address 0 and 1, 2 and 3, etc.., or if the groups are address 1 and 2, 3 and 4, etc..  0 indicates slave register addresses 0 and 1 are grouped together (odd numbered register ends the group). 1 indicates slave register addresses 1 and 2 are grouped together (even numbered register ends the group). This allows byte swapping of registers that are grouped starting at any address. |
| **[3:0]** | I2C\_SLV2\_LENG[3:0] | Number of bytes to be read from I2C slave 2 |

* 1. **Registers 46 to 48 – I2C Slave 3 Control Register 46 - I2C\_SLV3\_ADDR**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV3\_RNW | 1 – Transfer is a read  0 – Transfer is a write |
| **[6:0]** | I2C\_ID\_3[6:0] | Physical address of I2C slave 3 |

**Register 47 - I2C\_SLV3\_REG**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV3\_REG[7:0] | I2C slave 3 register address from where to begin data transfer |

**Register 48 - I2C\_SLV3\_CTRL**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV3\_EN | 1 – Enable reading data from this slave at the sample rate and storing data at the first available EXT\_SENS\_DATA register as determined by I2C\_SLV0\_EN, I2C\_SLV0\_LENG, I2C\_SLV1\_EN, I2C\_SLV1\_LENG, I2C\_SLV2\_EN and I2C\_SLV2\_LENG.  0 – function is disabled for this slave |
| **[6]** | I2C\_SLV3\_BYTE\_SW | 1 – Swap bytes when reading both the low and high byte of a word. Note there is nothing to swap after reading the first byte if I2C\_SLV3\_REG[0] = 1, or if the last byte read has a register address lsb = 0.  See I2C\_SLV1\_CTRL for an example.  0 – no swapping occurs, bytes are written in order read. |
| **[5]** | I2C\_SLV0\_REG\_DIS | When set, the transaction does not write a register value, it will only read data, or write data |
| **[4]** | I2C\_SLV3\_GRP | External sensor data typically comes in as groups of two bytes. This bit is used to determine if the groups are from the slave’s register address 0 and 1, 2 and 3, etc.., or if the groups are address 1 and 2, 3 and 4, etc..  0 indicates slave register addresses 0 and 1 are grouped together (odd numbered register ends the group). 1 indicates slave register addresses 1 and 2 are grouped together (even numbered register ends the group). This allows byte swapping of registers that are grouped starting at any address. |
| **[3:0]** | I2C\_SLV3\_LENG[3:0] | Number of bytes to be read from I2C slave 3 |

* 1. **Registers 49 to 53 – I2C Slave 4 Control Register 49 - I2C\_SLV4\_ADDR**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV4\_RNW | 1 – Transfer is a read  0 – Transfer is a write |
| **[6:0]** | I2C\_ID\_4[6:0] | Physical address of I2C slave 4 |

**Register 50 - I2C\_SLV4\_REG Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV4\_REG[7:0] | I2C slave 4 register address from where to begin data transfer |

## Register 51 - I2C\_SLV4\_DO

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV4\_DO[7:0] | Data to be written to I2C Slave 4 if enabled. |

## Register 52 - I2C\_SLV4\_CTRL

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | I2C\_SLV4\_EN | 1 – Enable data transfer with this slave at the sample rate. If read command, store data in I2C\_SLV4\_DI register, if write command, write data stored in I2C\_SLV4\_DO register. Bit is cleared when a single transfer is complete. Be sure to write I2C\_SLV4\_DO first  0 – function is disabled for this slave |
| **[6]** | SLV4\_DONE\_INT\_EN | 1 – Enables the completion of the I2C slave 4 data transfer to cause an interrupt.  0 – Completion of the I2C slave 4 data transfer will not cause an interrupt. |
| **[5]** | I2C\_SLV4\_REG\_DIS | When set, the transaction does not write a register value, it will only read data, or write data |
| **[4:0]** | I2C\_MST\_DLY | When enabled via the I2C\_MST\_DELAY\_CTRL, those slaves will only be enabled every (1+I2C\_MST\_DLY) samples (as determined by the SMPLRT\_DIV and DLPF\_CFG registers. |

## Register 53 - I2C\_SLV4\_DI

**Serial IF: R**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV4\_DI[7:0] | Data read from I2C Slave 4. |

## Register 54 – I2C Master Status

**Serial IF: R/C Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | PASS\_THROUGH | Status of FSYNC interrupt – used as a way to pass an external interrupt through this chip to the host. If enabled in the INT\_PIN\_CFG register by asserting bit FSYNC\_INT\_EN and if the FSYNC signal transitions from low to high, this will cause an interrupt. A read of this register clears all status bits in this register. |
| **[6]** | I2C\_SLV4\_DONE | Asserted when I2C slave 4’s transfer is complete, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted, and if the SLV4\_DONE\_INT\_EN bit is asserted in the I2C\_SLV4\_CTRL register. |
| **[5]** | I2C\_LOST\_ARB | Asserted when I2C slave looses arbitration of the I2C bus, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted. |
| **[4]** | I2C\_SLV4\_NACK | Asserted when slave 4 receives a nack, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted. |
| **[3]** | I2C\_SLV3\_NACK | Asserted when slave 3 receives a nack, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted. |
| **[2]** | I2C\_SLV2\_NACK | Asserted when slave 2 receives a nack, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted. |
| **[1]** | I2C\_SLV1\_NACK | Asserted when slave 1 receives a nack, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted. |
| **[0]** | I2C\_SLV0\_NACK | Asserted when slave 0 receives a nack, will cause an interrupt if bit I2C\_MST\_INT\_EN in the INT\_ENABLE register is asserted. |

## Register 55 – INT Pin / Bypass Enable Configuration

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | ACTL | 1 – The logic level for INT pin is active low.  0 – The logic level for INT pin is active high. |
| **[6]** | OPEN | 1 – INT pin is configured as open drain.  0 – INT pin is configured as push-pull. |
| **[5]** | LATCH\_INT\_EN | 1 – INT pin level held until interrupt status is cleared.  0 – INT pin indicates interrupt pulse’s is width 50us. |
| **[4]** | INT\_ANYRD\_2CLEAR | 1 – Interrupt status is cleared if any read operation is performed.  0 – Interrupt status is cleared only by reading INT\_STATUS register |
| **[3]** | ACTL\_FSYNC | 1 – The logic level for the FSYNC pin as an interrupt is active low.  0 – The logic level for the FSYNC pin as an interrupt is active high. |
| **[2]** | FSYNC\_INT\_MODE\_EN | 1 – This enables the FSYNC pin to be used as an interrupt. A transition to the active level described by the ACTL\_FSYNC bit will cause an interrupt. The status of the interrupt is read in the I2C Master Status register PASS\_THROUGH bit.  0 – This disables the FSYNC pin from causing an interrupt. |
| **[1]** | BYPASS\_EN | When asserted, the i2c\_master interface pins(ES\_CL and ES\_DA) will go into ‘bypass mode’ when the i2c master interface is disabled. The pins will float high due to the internal pull-up if not enabled and the i2c master interface is disabled. |
| **[0]** | RESERVED |  |

## Register 56 – Interrupt Enable

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | RESERVED |  |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[6]** | WOM\_EN | 1 – Enable interrupt for wake on motion to propagate to interrupt pin.  0 – function is disabled. |
| **[5]** | RESERVED |  |
| **[4]** | FIFO\_OVERFLOW\_EN | 1 – Enable interrupt for fifo overflow to propagate to interrupt pin.  0 – function is disabled. |
| **[3]** | FSYNC\_INT\_EN | 1 – Enable Fsync interrupt to propagate to interrupt pin.  0 – function is disabled. |
| **[2]** | RESERVED |  |
| **[1]** | RESERVED |  |
| **[0]** | RAW\_RDY\_EN | 1 – Enable Raw Sensor Data Ready interrupt to propagate to interrupt pin. The timing of the interrupt can vary depending on the setting in register 36 I2C\_MST\_CTRL, bit [6] WAIT\_FOR\_ES.  0 – function is disabled. |

## Register 58 – Interrupt Status

**Serial IF: R/C Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | Reserved |  |
| **[6]** | WOM\_INT | 1 – Wake on motion interrupt occurred. |
| **[5]** | Reserved |  |
| **[4]** | FIFO\_OVERFLOW\_INT | 1 – Fifo Overflow interrupt occurred. Note that the oldest data is has been dropped from the fifo. |
| **[3]** | FSYNC\_INT | 1 – Fsync interrupt occurred. |
| **[2]** | Reserved |  |
| **[1]** | Reserved |  |
| **[0]** | RAW\_DATA\_RDY\_INT | 1 – Sensor Register Raw Data sensors are updated and Ready to be read. The timing of the interrupt can vary depending on the setting in register 36 I2C\_MST\_CTRL, bit [6] WAIT\_FOR\_ES. |

## Registers 59 to 64 – Accelerometer Measurements

**Name: ACCEL\_XOUT\_H**

Serial IF: SyncR

## Reset value: 0x00 (if sensor disabled)

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of accelerometer x-axis data. |

Name: ACCEL\_XOUT\_L

Serial IF: SyncR

## Reset value: 0x00 (if sensor disabled)

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of accelerometer x-axis data. |

Name: ACCEL\_YOUT\_H

Serial IF: SyncR

## Reset value: 0x00 (if sensor disabled)

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of accelerometer y-axis data. |

Name: ACCEL\_YOUT\_L

Serial IF: SyncR

## Reset value: 0x00 (if sensor disabled)

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of accelerometer y-axis data. |

Name: ACCEL\_ZOUT\_H

Serial IF: SyncR

## Reset value: 0x00 (if sensor disabled)

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of accelerometer z-axis data. |

Name: ACCEL\_ZOUT\_L

Serial IF: SyncR

## Reset value: 0x00 (if sensor disabled)

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of accelerometer z-axis data. |

## Registers 65 and 66 – Temperature Measurement

**Name: TEMP\_OUT\_H Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of the temperature sensor output |

**Name: TEMP\_OUT\_L Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of the temperature sensor output:  **TEMP\_degC** = ((TEMP\_OUT –  RoomTemp\_Offset)/Temp\_Sensitivity)  + 21degC  Where Temp\_degC is the temperature in degrees C measured by the temperature sensor. TEMP\_OUT is the actual output of the temperature sensor. |

* 1. **Registers 67 to 72 – Gyroscope Measurements Name: GYRO\_XOUT\_H**

**Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of the X-Axis gyroscope output |

**Name: GYRO\_XOUT\_L Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of the X-Axis gyroscope output **GYRO\_XOUT =** Gyro\_Sensitivity \* X\_angular\_rate Nominal FS\_SEL = 0  Conditions Gyro\_Sensitivity = 131 LSB/(º/s) |

**Name: GYRO\_YOUT\_H Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of the Y-Axis gyroscope output |

**Name: GYRO\_YOUT\_L Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of the Y-Axis gyroscope output  **GYRO\_YOUT =** Gyro\_Sensitivity \* Y\_angular\_rate Nominal FS\_SEL = 0  Conditions Gyro\_Sensitivity = 131 LSB/(º/s) |

**Name: GYRO\_ZOUT\_H Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | High byte of the Z-Axis gyroscope output |

**Name: GYRO\_ZOUT\_L Serial IF: SyncR**

**Reset value: 0x00 (if sensor disabled)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Low byte of the Z-Axis gyroscope output **GYRO\_ZOUT =** Gyro\_Sensitivity \* Z\_angular\_rate Nominal FS\_SEL = 0  Conditions Gyro\_Sensitivity = 131 LSB/(º/s) |

## Registers 73 to 96 – External Sensor Data

**EXT\_SENS\_DATA\_00 – 23**

**Serial IF: SyncR Reset value: 0x00**

**24 registers with the same description as below:**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Sensor data read from external I2C devices via the I2C master interface. The data stored is controlled by the I2C\_SLV(0-4)\_ADDR, I2C\_SLV(0-4)\_REG, and  I2C\_SLV(0-4)\_CTRL registers |

**Description:**

These registers store data read from external sensors by the Slave 0, 1, 2, and 3 on the auxiliary I2C interface. Data read by Slave 4 is stored in I2C\_SLV4\_DI (Register 53).

External sensor data is written to these registers at the Sample Rate as defined in Register 25. This access rate can be reduced by using the Slave Delay Enable registers (Register 103).

Data is placed in these external sensor data registers according to I2C\_SLV0\_CTRL, I2C\_SLV1\_CTRL, I2C\_SLV2\_CTRL, and I2C\_SLV3\_CTRL (Registers 39, 42, 45, and 48). When more than zero bytes are read (*I2C\_SLVx\_LEN* > 0) from an enabled slave (*I2C\_SLVx\_EN* = 1), the slave is read at the Sample Rate (as defined in Register 25) or delayed rate (if specified in Register 52 and 103). During each sample cycle, slave reads are performed in order of Slave number. If all slaves are enabled with more than zero bytes to be read, the order will be Slave 0, followed by Slave 1, Slave 2, and Slave 3.

Each enabled slave will have EXT\_SENS\_DATA registers associated with it by number of bytes read (*I2C\_SLVx\_LEN)* in order of slave number, starting from EXT\_SENS\_DATA\_00. Note that this means enabling or disabling a slave may change the higher numbered slaves’ associated registers. Furthermore, if fewer total bytes are being read from the external sensors as a result of such a change, then the data remaining in the registers which no longer have an associated slave device (i.e. high numbered registers) will remain in these previously allocated registers unless reset.

If the sum of the read lengths of all SLVx transactions exceed the number of available EXT\_SENS\_DATA registers, the excess bytes will be dropped. There are 24 EXT\_SENS\_DATA

registers and hence the total read lengths between all the slaves cannot be greater than 24 or some bytes will be lost.

Note: Slave 4’s behavior is distinct from that of Slaves 0-3. For further information regarding the characteristics of Slave 4, please refer to Registers 49 to 53.

**Example:**

Suppose that Slave 0 is enabled with 4 bytes to be read (*I2C\_SLV0\_EN* = 1 and *I2C\_SLV0\_LEN* =

4) while Slave 1 is enabled with 2 bytes to be read, (*I2C\_SLV1\_EN=*1 and *I2C\_SLV1\_LEN* = 2). In such a situation, EXT\_SENS\_DATA \_00 through \_03 will be associated with Slave 0, while EXT\_SENS\_DATA \_04 and 05 will be associated with Slave 1.

If Slave 2 is enabled as well, registers starting from EXT\_SENS\_DATA\_06 will be allocated to Slave 2.

If Slave 2 is disabled while Slave 3 is enabled in this same situation, then registers starting from EXT\_SENS\_DATA\_06 will be allocated to Slave 3 instead.

**Register Allocation for Dynamic Disable vs. Normal Disable**

If a slave is disabled at any time, the space initially allocated to the slave in the EXT\_SENS\_DATA register, will remain associated with that slave. This is to avoid dynamic adjustment of the register allocation.

The allocation of the EXT\_SENS\_DATA registers is recomputed only when (1) all slaves are disabled, or (2) the *I2C\_MST\_RST* bit is set (Register 106).

This above is also true if one of the slaves gets NACKed and stops functioning.

## Register 99 – I2C Slave 0 Data Out

**I2C\_SLV0\_DO**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV0\_DO | Data out when slave 0 is set to write |

For further information regarding Slave 1 control, please refer to Registers 37 to 39.

## Register 100 – I2C Slave 1 Data Out I2C\_SLV1\_DO

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV1\_DO | Data out when slave 1 is set to write |

For further information regarding Slave 1 control, please refer to Registers 40 to 42.

## Register 101 – I2C Slave 2 Data Out I2C\_SLV2\_DO

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV2\_DO | Data out when slave 2 is set to write |

For further information regarding Slave 2 control, please refer to Registers 43 to 45.

## Register 102 – I2C Slave 3 Data Out

**I2C\_SLV3\_DO**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | I2C\_SLV3\_DO | Data out when slave 3 is set to write |

For further information regarding Slave 2 control, please refer to Registers 46 to 48.

## Register 103 – I2C Master Delay Control I2C\_MST\_DELAY\_CTRL

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | DELAY\_ES\_SHADOW | Delays shadowing of external sensor data until all data is received |
| **[6:5]** | Reserved |  |
| **[4]** | I2C\_SLV4\_DLY\_EN | When enabled, slave 4 will only be accessed (1+I2C\_MST\_DLY) samples as determined by SMPLRT\_DIV and DLPF\_CFG |
| **[3]** | I2C\_SLV3\_DLY\_EN | When enabled, slave 3 will only be accessed (1+I2C\_MST\_DLY) samples as determined by SMPLRT\_DIV and DLPF\_CFG |
| **[2]** | I2C\_SLV2\_DLY\_EN | When enabled, slave 2 will only be accessed 1+I2C\_MST\_DLY) samples as determined by SMPLRT\_DIV and DLPF\_CFG |
| **[1]** | I2C\_SLV1\_DLY\_EN | When enabled, slave 1 will only be accessed 1+I2C\_MST\_DLY) samples as determined by SMPLRT\_DIV and DLPF\_CFG |
| **[0]** | I2C\_SLV0\_DLY\_EN | When enabled, slave 0 will only be accessed 1+I2C\_MST\_DLY) samples as determined by SMPLRT\_DIV and DLPF\_CFG |

* 1. **Register 104 – Signal Path Reset SIGNAL\_PATH\_RESET**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:3]** | Reserved |  |
| **[2]** | GYRO\_RST | Reset gyro digital signal path. Note: Sensor registers are not cleared. Use SIG\_COND\_RST to clear sensor registers. |
| **[1]** | ACCEL\_RST | Reset accel digital signal path. Note: Sensor registers are not cleared. Use SIG\_COND\_RST to clear sensor registers. |
| **[0]** | TEMP\_RST | Reset temp digital signal path. Note: Sensor registers are not cleared. Use SIG\_COND\_RST to clear sensor registers. |

## Register 105 – Accelerometer Interrupt Control

**ACCEL\_INTEL\_CTRL**

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | ACCEL\_INTEL\_EN | This bit enables the Wake-on-Motion detection logic. |
| **[6]** | ACCEL\_INTEL\_MODE | This bit defines  1 = Compare the current sample with the previous sample. 0 = Not used. |
| **[5:0]** | Reserved |  |

Please refer to the Wake-on-Motion Interrupt section of the MPU-9250 Product Specification for additional details.

## Register 106 – User Control Name: USER\_CTRL

**Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | Reserved |  |
| **[6]** | FIFO\_EN | 1 – Enable FIFO operation mode.  0 – Disable FIFO access from serial interface. To disable FIFO writes by dma, use FIFO\_EN register. To disable possible FIFO writes from DMP, disable the DMP. |
| **[5]** | I2C\_MST\_EN | 1 – Enable the I2C Master I/F module; pins ES\_DA and ES\_SCL are isolated from pins SDA/SDI and SCL/ SCLK.  0 – Disable I2C Master I/F module; pins ES\_DA and ES\_SCL are logically driven by pins SDA/SDI and SCL/ SCLK.  NOTE: DMP will run when enabled, even if all internal sensors are disabled, except when the sample rate is set to 8Khz. |
| **[4]** | I2C\_IF\_DIS | 1 – Reset I2C Slave module and put the serial interface in SPI mode only. This bit auto clears after one clock cycle. |
| **[3]** | Reserved |  |
| **[2]** | FIFO\_RST | 1 – Reset FIFO module. Reset is asynchronous. This bit auto clears after one clock cycle. |
| **[1]** | I2C\_MST\_RST | 1 – Reset I2C Master module. Reset is asynchronous. This bit auto clears after one clock cycle.  NOTE: This bit should only be set when the I2C master has hung. If this bit is set during an active I2C master transaction, the I2C slave will hang, which will require the host to reset the slave. |
| **[0]** | SIG\_COND\_RST | 1 – Reset all gyro digital signal path, accel digital signal path, and temp digital signal path. This bit also clears all the sensor registers.  SIG\_COND\_RST is a pulse of one clk8M wide. |

## Register 107 – Power Management 1

**Name: PWR\_MGMT\_1 Serial IF: R/W**

**Reset value: (Depends on PU\_SLEEP\_MODE bit, see below)**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7]** | H\_RESET | 1 – Reset the internal registers and restores the default settings. Write a 1 to set the reset, the bit will auto clear. |
| **[6]** | SLEEP | When set, the chip is set to sleep mode (After OTP loads, the PU\_SLEEP\_MODE bit will be written here) |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[5]** | CYCLE | When set, and SLEEP and STANDBY are not set, the chip will cycle between sleep and taking a single sample at a rate determined by LP\_ACCEL\_ODR register  NOTE: When all accelerometer axis are disabled via PWR\_MGMT\_2 register bits and cycle is enabled, the chip will wake up at the rate determined by the respective registers above, but will not take any samples. |
| **[4]** | GYRO\_STANDBY | When set, the gyro drive and pll circuitry are enabled, but the sense paths are disabled. This is a low power mode that allows quick enabling of the gyros. |
| **[3]** | PD\_PTAT | Power down internal PTAT voltage generator and PTAT ADC |
| **[2:0]** | CLKSEL[2:0] | **Code Clock Source**   1. Internal 20MHz oscillator 2. Auto selects the best available clock source – PLL if ready, else use the Internal oscillator 3. Auto selects the best available clock source – PLL if ready, else use the Internal oscillator 4. Auto selects the best available clock source – PLL if ready, else use the Internal oscillator 5. Auto selects the best available clock source – PLL if ready, else use the Internal oscillator 6. Auto selects the best available clock source – PLL if ready, else use the Internal oscillator 7. Internal 20MHz oscillator 8. Stops the clock and keeps timing generator in reset   (After OTP loads, the inverse of PU\_SLEEP\_MODE bit will be written to CLKSEL[0]) |

## Register 108 – Power Management 2

**Name: PWR\_MGMT\_2 Serial IF: R/W**

**Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:6]** | Reserved |  |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[5]** | DISABLE\_XA | 1 – X accelerometer is disabled  0 – X accelerometer is on |
| **[4]** | DISABLE\_YA | 1 – Y accelerometer is disabled  0 – Y accelerometer is on |
| **[3]** | DISABLE\_ZA | 1 – Z accelerometer is disabled  0 – Z accelerometer is on |
| **[2]** | DISABLE\_XG | 1 – X gyro is disabled  0 – X gyro is on |
| **[1]** | DISABLE\_YG | 1 – Y gyro is disabled  0 – Y gyro is on |
| **[0]** | DISABLE\_ZG | 1 – Z gyro is disabled  0 – Z gyro is on |

The MPU-9250 can be put into Accelerometer Only Low Power Mode using the following steps:

* + 1. Set CYCLE bit to 1
    2. Set SLEEP bit to 0
    3. Set TEMP\_DIS bit to 1
    4. Set DIS\_XG, DIS\_YG, DIS\_ZG bits to 1

The bits mentioned in the steps (i) to (iii) can be found in Power Management 1 register (Register 107).

In this mode, the device will power off all devices except for the primary I2C interface, waking only the accelerometer at fixed intervals to take a single measurement.

## Register 114 and 115 – FIFO Count Registers Name: FIFO\_COUNTH

**Address: 114**

**Serial IF: Read Only Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:5]** | Reserved |  |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[4:0]** | FIFO\_CNT[12:8] | High Bits, count indicates the number of written bytes in the FIFO.  Reading this byte latches the data for both FIFO\_COUNTH, and FIFO\_COUNTL. |

**FIFO\_COUNTL**

**Address: 115**

**Serial IF: Read Only Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | FIFO\_CNT[7:0] | Low Bits, count indicates the number of written bytes in the FIFO. NOTE: Must read FIFO\_COUNTH to latch new data for both FIFO\_COUNTH and FIFO\_COUNTL. |

## Register 116 – FIFO Read Write

**Name: FIFO\_R\_W Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | D[7:0] | Read/Write command provides Read or Write operation for the FIFO. |

**Description:**

This register is used to read and write data from the FIFO buffer.

Data is written to the FIFO in order of register number (from lowest to highest). If all the FIFO enable flags (see below) are enabled and all External Sensor Data registers (Registers 73 to 96) are associated with a Slave device, the contents of registers 59 through 96 will be written in order at the Sample Rate.

The contents of the sensor data registers (Registers 59 to 96) are written into the FIFO buffer when their corresponding FIFO enable flags are set to 1 in FIFO\_EN (Register 35). An additional flag for the sensor data registers associated with I2C Slave 3 can be found in I2C\_MST\_CTRL (Register 36).

If the FIFO buffer has overflowed, the status bit *FIFO\_OFLOW\_INT* is automatically set to 1. This bit is located in INT\_STATUS (Register 58). When the FIFO buffer has overflowed, the oldest data will be lost and new data will be written to the FIFO unless register 26 CONFIG, bit[6] FIFO\_MODE = 1.

If the FIFO buffer is empty, reading this register will return the last byte that was previously read from the FIFO until new data is available. The user should check *FIFO\_COUNT* to ensure that the FIFO buffer is not read when empty.

## Register 117 – Who Am I Name: WHOAMI

**Serial IF: Read Only Reset value: 0x68**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | WHOAMI | Register to indicate to user which device is being accessed. |

This register is used to verify the identity of the device. The contents of *WHO\_AM\_I* is an 8-bit device ID. The default value of the register is 0x71.

## Registers 119, 120, 122, 123, 125, 126 Accelerometer Offset Registers For MPU-9250 mode:

**Name: XA\_OFFS\_H Address: 119 Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | XA\_OFFS[14:7] | Upper bits of the X accelerometer offset cancellation. +/- 16g Offset cancellation in all Full Scale modes, 15 bit 0.98-mg steps |

**Name: XA\_OFFS\_L Address: 120 Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:1]** | XA\_OFFS[6:0] | Lower bits of the X accelerometer offset cancellation. +/- 16g Offset cancellation in all Full Scale modes, 15 bit 0.98-mg steps |

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[0]** | Reserved |  |

**Name: YA\_OFFS\_H Address: 122 Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | YA\_OFFS[14:7] | Upper bits of the Y accelerometer offset cancellation. +/- 16g Offset cancellation in all Full Scale modes, 15 bit 0.98-mg steps |

**Name: YA\_OFFS\_L Address: 123 Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:1]** | YA\_OFFS[6:0] | Lower bits of the Y accelerometer offset cancellation. +/- 16g Offset cancellation in all Full Scale modes, 15 bit 0.98-mg steps |
| **[0]** | Reserved |  |

**Name: ZA\_OFFS\_H Address: 125 Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:0]** | ZA\_OFFS[14:7] | Upper bits of the Z accelerometer offset cancellation. +/- 16g Offset cancellation in all Full Scale modes, 15 bit 0.98-mg steps |

**Name: ZA\_OFFS\_L Address: 126 Serial IF: R/W Reset value: 0x00**

|  |  |  |
| --- | --- | --- |
| **BIT** | **NAME** | **FUNCTION** |
| **[7:1]** | ZA\_OFFS[6:0] | Lower bits of the Z accelerometer offset cancellation. +/- 16g Offset cancellation in all Full Scale modes, 15 bit 0.98-mg steps |
| **[0]** | Reserved |  |

# Register Map for Magnetometer

The register map for the MPU-9250’s Magnetometer (AK8963) section is listed below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Address** | **READ/**  **WRITE** | **Description** | **Bit**  **width** | **Explanation** |
| WIA | 00H | READ | Device ID | 8 |  |
| INFO | 01H | READ | Information | 8 |  |
| ST1 | 02H | READ | Status 1 | 8 | Data status |
| HXL | 03H | READ | Measurement data | 8 | X-axis data |
| HXH | 04H | 8 |
| HYL | 05H | 8 | Y-axis data |
| HYH | 06H | 8 |
| HZL | 07H | 8 | Z-axis data |
| HZH | 08H | 8 |
| ST2 | 09H | READ | Status 2 | 8 | Data status |
| CNTL | 0AH | READ/  WRITE | Control | 8 |  |
| RSV | 0BH | READ/  WRITE | Reserved | 8 | DO NOT ACCESS |
| ASTC | 0CH | READ/ WRITE | Self-test | 8 |  |
| TS1 | 0DH | READ/  WRITE | Test 1 | 8 | DO NOT ACCESS |
| TS2 | 0EH | READ/ WRITE | Test 2 | 8 | DO NOT ACCESS |
| I2CDIS | 0FH | READ/  WRITE | I2C disable | 8 |  |
| ASAX | 10H | READ | X-axis sensitivity adjustment value | 8 | Fuse ROM |
| ASAY | 11H | READ | Y-axis sensitivity adjustment value | 8 | Fuse ROM |
| ASAZ | 12H | READ | Z-axis sensitivity adjustment value | 8 | Fuse ROM |

## Table 2 Register Table

Addresses from 00H to 0CH and from 10H to 12H are compliant with automatic increment function of serial interface respectively. Values of addresses from 10H to 12H can be read only in Fuse access mode. In other modes, read data is not correct.

## Register Map Description

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register**  **Name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| **Read-only Register** | | | | | | | | | |
| 00H | WIA | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 01H | INFO | INFO7 | INFO6 | INFO5 | INFO4 | INFO3 | INFO2 | INFO1 | INFO0 |
| 02H | ST1 | 0 | 0 | 0 | 0 | 0 | 0 | DOR | DRDY |
| 03H | HXL | HX7 | HX6 | HX5 | HX4 | HX3 | HX2 | HX1 | HX0 |
| 04H | HXH | HX15 | HX14 | HX13 | HX12 | HX11 | HX10 | HX9 | HX8 |
| 05H | HYL | HY7 | HY6 | HY5 | HY4 | HY3 | HY2 | HY1 | HY0 |
| 06H | HYH | HY15 | HY14 | HY13 | HY12 | HY11 | HY10 | HY9 | HY8 |
| 07H | HZL | HZ7 | HZ6 | HZ5 | HZ4 | HZ3 | HZ2 | HZ1 | HZ0 |
| 08H | HZH | HZ15 | HZ14 | HZ13 | HZ12 | HZ11 | HZ10 | HZ9 | HZ8 |
| 09H | ST2 | 0 | 0 | 0 | BITM | HOFL | 0 | 0 | 0 |
| **Write/read Register** | | | | | | | | | |
| 0AH | CNTL1 | 0 | 0 | 0 | 0 | MODE3 | MODE2 | MODE1 | MODE0 |
| 0BH | CNTL2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SRST |
| 0CH | ASTC | - | SELF | - | - | - | - | - | - |
| 0DH | TS1 | - | - | - | - | - | - | - | - |
| 0EH | TS2 | - | - | - | - | - | - | - | - |
| 0FH | I2CDIS | I2CDIS7 | I2CDIS6 | I2CDIS5 | I2CDIS4 | I2CDIS3 | I2CDIS2 | I2CDIS1 | I2CDIS0 |
| **Read-only Register** | | | | | | | | | |
| 10H | ASAX | COEFX7 | COEFX6 | COEFX5 | COEFX4 | COEFX3 | COEFX2 | COEFX1 | COEFX0 |
| 11H | ASAY | COEFY7 | COEFY6 | COEFY5 | COEFY4 | COEFY3 | COEFY2 | COEFY1 | COEFY0 |
| 12H | ASAZ | COEFZ7 | COEFZ6 | COEFZ5 | COEFZ4 | COEFZ3 | COEFZ2 | COEFZ1 | COEFZ0 |

**Table 3 Register Map**

Note: When VDD is turned ON, POR function works and all registers of AK893 are initialized. TS1 and TS2 are test registers for shipment test. Do not use these registers.

RSV is reserved register. Do not use this register.

## Detailed Descriptions for Magnetometer Registers

This section details each register within the MPU-9250’s Magnetometer section.

## WIA: Device ID

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 00H | WIA | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

Device ID of AKM. It is described in one byte and fixed value.

48H: fixed

## INFO: Information

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 01H | INFO | INFO7 | INFO6 | INFO5 | INFO4 | INFO3 | INFO2 | INFO1 | INFO0 |

INFO[7:0]: Device information for AKM.

## ST1: Status 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 02H | ST1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DRDY |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DRDY: Data Ready "0": Normal

"1": Data is ready

DRDY bit turns to “1” when data is ready in single measurement mode or self-test mode. It returns to “0” when any one of ST2 register or measurement data register (HXL to HZH) is read.

DOR: Data Overrun "0": Normal

"1": Data overrun

DOR bit turns to “1” when data has been skipped in continuous measurement mode or external trigger measurement mode. It returns to “0” when any one of ST2 register or measurement data register (HXL~HZH) is read.

## HXL to HZH: Measurement Data

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 03H | HXL | HX7 | HX6 | HX5 | HX4 | HX3 | HX2 | HX1 | HX0 |
| 04H | HXH | HX15 | HX14 | HX13 | HX12 | HX11 | HX10 | HX9 | HX8 |
| 05H | HYL | HY7 | HY6 | HY5 | HY4 | HY3 | HY2 | HY1 | HY0 |
| 06H | HYH | HY15 | HY14 | HY13 | HY12 | HY11 | HY10 | HY9 | HY8 |
| 07H | HZL | HZ7 | HZ6 | HZ5 | HZ4 | HZ3 | HZ2 | HZ1 | HZ0 |
| 08H | HZH | HZ15 | HZ14 | HZ13 | HZ12 | HZ11 | HZ10 | HZ9 | HZ8 |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Measurement data of magnetic sensor X-axis/Y-axis/Z-axis

HXL[7:0]: X-axis measurement data lower 8bit HXH[15:8]: X-axis measurement data higher 8bit HYL[7:0]: Y-axis measurement data lower 8bit HYH[15:8]: Y-axis measurement data higher 8bit HZL[7:0]: Z-axis measurement data lower 8bit HZH[15:8]: Z-axis measurement data higher 8bit

Measurement data is stored in two’s complement and Little Endian format. Measurement range of each axis is from -32760 ~ 32760 decimal in 16-bit output.

|  |  |  |  |
| --- | --- | --- | --- |
| **Measurement data (each axis) [15:0]** | | | **Magnetic flux density [µT]** |
| **Two’s complement** | **Hex** | **Decimal** |
| 0111 1111 1111 1000 | 7FF8 | 32760 | 4912(max.) |
| | | | | | | | |
| 0000 0000 0000 0001 | 0001 | 1 | 0.15 |
| 0000 0000 0000 0000 | 0000 | 0 | 0 |
| 1111 1111 1111 1111 | FFFF | -1 | -0.15 |
| | | | | | | | |
| 1000 0000 0000 1000 | 8008 | -32760 | -4912(min.) |

## Table 4 Measurement data format

## ST2: Status 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 09H | ST2 | 0 | 0 | 0 | BITM | HOFL | 0 | 0 | 0 |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

HOFL: Magnetic sensor overflow "0": Normal

"1": Magnetic sensor overflow occurred

In single measurement mode, continuous measurement mode, external trigger measurement mode and self-test mode, magnetic sensor may overflow even though measurement data regiseter is not saturated. In this case, measurement data is not correct and HOFL bit turns to “1”. When next measurement stars, it returns to “0”.

BITM: Output bit setting (mirror) "0": 14-bit output

"1": 16-bit output

Mirror data of BIT bit of CNTL1 register.

ST2 register has a role as data reading end register, also. When any of measurement data register is read in continuous measurement mode or external trigger measurement mode, it means data reading start and taken as data reading until ST2 register is read. Therefore, when any of measurement data is read, be sure to read ST2 register at the end.

## CNTL1: Control 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 0AH | CNTL1 | 0 | 0 | 0 | BIT | MODE3 | MODE2 | MODE1 | MODE0 |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

MODE[3:0]: Operation mode setting "0000": Power-down mode

"0001": Single measurement mode "0010": Continuous measurement mode 1

"0110": Continuous measurement mode 2 "0100": External trigger measurement mode "1000": Self-test mode

"1111": Fuse ROM access mode Other code settings are prohibited

BIT: Output bit setting

"0": 14-bit output

"1": 16-bit output

When each mode is set, AK8963 transits to set mode.

When CNTL register is accessed to be written, registers from 02H to 09H are initialized.

## CNTL2: Control 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 0BH | CNTL2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SRST |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SRST: Soft reset "0": Normal

"1": Reset

When “1” is set, all registers are initialized. After reset, SRST bit turns to “0” automatically.

## ASTC: Self-Test Control

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Write/read register | | | | | | | | | |
| 0CH | ASTC | - | SELF | - | - | - | - | - | - |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SELF: Self-test control "0": Normal

"1": Generate magnetic field for self-test

Do not write “1” to any bit other than SELF bit in ASTC register. If “1” is written to any bit other than SELF bit, normal measurement cannot be done.

## TS1, TS2: Test 1, 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Write/read register | | | | | | | | | |
| 0DH | TS1 | - | - | - | - | - | - | - | - |
| 0EH | TS2 | - | - | - | - | - | - | - | - |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TS1 and TS2 registers are test registers for shipment test. Do not use these registers.

## I2CDIS: I2C Disable

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Write/read register | | | | | | | | | |
| 0FH | I2CDIS | I2CDIS7 | I2CDIS6 | I2CDIS5 | I2CDIS4 | I2CDIS3 | I2CDIS2 | I2CDIS1 | I2CDIS0 |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register disables I2C bus interface. I2C bus interface is enabled in default. To disable I2C bus interface, write “00011011” to I2CDIS register. Then I2C bus interface is disabled.

Once I2C bus interface is disabled, it is impossible to write other value to I2CDIS register. To enable I2C bus interface, reset AK8963 or input start condition 8 times continuously.

## ASAX, ASAY, ASAZ: Sensitivity Adjustment values

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Addr** | **Register name** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Read-only register | | | | | | | | | |
| 10H | ASAX | COEFX7 | COEFX6 | COEFX5 | COEFX4 | COEFX3 | COEFX2 | COEFX1 | COEFX0 |
| 11H | ASAY | COEFY7 | COEFY6 | COEFY5 | COEFY4 | COEFY3 | COEFY2 | COEFY1 | COEFY0 |
| 12H | ASAZ | COEFZ7 | COEFZ6 | COEFZ5 | COEFZ4 | COEFZ3 | COEFZ2 | COEFZ1 | COEFZ0 |
| Reset | | - | - | - | - | - | - | - | - |

Sensitivity adjustment data for each axis is stored to fuse ROM on shipment.

ASAX[7:0]: Magnetic sensor X-axis sensitivity adjustment value ASAY[7:0]: Magnetic sensor Y-axis sensitivity adjustment value ASAZ[7:0]: Magnetic sensor Z-axis sensitivity adjustment value

* + - Sensitivity Adjustment

The sensitivity adjustment is done by the equation below;

*Hadj*  *H*   *ASA*  128 0.5 



 ,

1

 128 

where *H* is the measurement data read out from the measurement data register, *ASA* is the sensitivity adjustment value, and *Hadj* is the adjusted measurement data.

# Advanced Hardware Features

The MPU-9250 includes advanced hardware features that support Android that can be enabled and disabled through simple hardware register settings. The advanced hardware features are not initially enabled after device power up, and must be individually enabled and configured. The following motion-based functions are supported and do not require an external hub or microprocessor:

* Android Orientation
* Step Count, Step Detection
* Significant Motion Detection
* Batch mode
* Low Power Quaternion (3, 6, 9 axis)

Features supported for embedded applications include:

* Pedometer, Directional Tap
* Low Power Quaternion (3, 6, 9 axis)

Features supported for Windows 8 UMDF implementation (no external hub required) include:

* Quaternion Output with CS/CSI filtering

For further details please see the Application Note “Programming Sequence for DMP Hardware Functions.”

This information furnished by InvenSense is believed to be accurate and reliable. However, no responsibility is assumed by InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.

InvenSense® is a registered trademark of InvenSense, Inc. MPUTM, MPU-9250TM, MPU-6000TM, MPU-6050TM, Digital Motion Processor™, DMP ™, Motion Processing Unit™, MotionFusion™, and MotionApps™ are trademarks of InvenSense, Inc.

©2013 InvenSense, Inc. All rights reserved.