AN1873

Host API Design for MM7150 Motion Module

Author: *Tom Tse / Arjun M. Das*
Microchip Technology Inc.

INTRODUCTION

This application note provides detailed design information regarding how to write application firmware to access the MM7150 motion module's data by using the defined API (application programming interface) and virtual registers.

This specification and design is also used in the MM7150 with Explorer 16 (PIC24FJ) development board's Sensor Fusion Sample Code, which is available to customers.

Audience

This document is written for developers who have a background in embedded controller firmware development, as well as basic knowledge of the HID over I^2C protocol.

References

The following documents should be referenced when using this application note. Please contact your Microchip representative for availability.

- Microsoft HID over I^2C protocol specification, v1.0 on April 24, 2012 (available on msdn.microsoft.com)
- HID Sensor Usage Tables (Request #: HUTRR39, available on usb.org)
- Microchip MM7150 Motion Module/PICTail on Explorer 16 Development Board User's Manual, Rev. A
- Microchip MM7150 with Explorer 16 (PIC24) Sensor Fusion Sample Code v1.3.1 or later

Glossary of Terms and AcronymsHost API Design for MM7150 Motion Module

This section describes glossary terms and acronyms used in this document.

TABLE 1-1: GLOSSARY

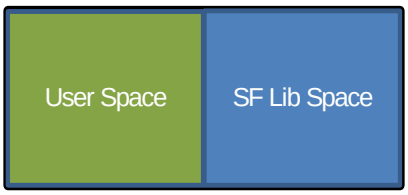
Term	Definition
HID	Human Interface Device
I^2C	Inter-Integrated Circuit
HOST	Refers to the application processor in the embedded systems that is used to communicate to the MM7150, which is PIC24FJ128GA010 in the Exp 16 sample code
Device / EC	Refers to the MM7150 Motion Module or the SSC7150 Sensor Hub Device
VREG	Virtual Registers
Acc, Mag, Gyro	Accelerometer, Magnetometer, Gyroscope (physical sensors)
Ori, Incl, Cmp	Orientation, Inclinometer, Compass (virtual sensors)
SF	Sensor Fusion

1.0 SYSTEM OVERVIEW

Host space will be divided into two separate entities:

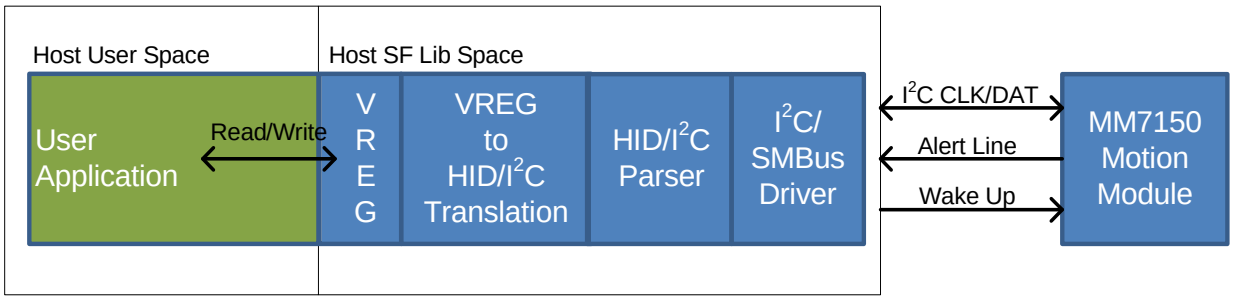
- Host user space
- Host SF lib (library) space

FIGURE 1-1: HOST LAYOUT



Host applications should reside in the Host user space, and the SF related infrastructure should reside in the SF lib space. Any host application can make use of the host SF lib through read/write of virtual registers. The virtual registers are the uppermost layer of Host SF lib that is exposed to the user application, which eventually drives down through all the layers of host SF lib.

FIGURE 1-2: HOST DEVICE LAYOUT



2.0 HOST SF LIB FEATURES

This section provides the summary table to describe all the features that are currently supported or will be supported in the Host SF Library.

TABLE 2-1: HOST SF LIB FEATURES

Features	Descriptions
Reset	Supports HID/I^{2}C reset command of MM7150 module, also resets the virtual registers.
Sensor Hub Enable	Enables the sensor hub.
Sensor Hub Sleep	When enabled, enters D3 deepest sleep state.
Sensor List	Lists the sensors supported in the MM7150 motion module.
Sensor Sensitivity	Sensitivity of sensors.
Sensor Data Rate/SPS	Rate at which the sensor can update data (Report Interval).
Sensor Enable	Enables reporting from sensor.
Sensor Data	Read data from an enabled sensor.
PID,VID,DID	Read the various IDs.
Vendor Collection	Reads MM7150 Firmware ID, update sensor coefficients, update configuration file, and supports flash update.

3.0 HOST SF LIB

Host SF lib space is comprised of the following four layers:

- Virtual Register layer
- VREG to HID translation layer
- HID/I^2C parser layer
- I^2C/SMBus driver layer

The various layers are depicted below.

3.1 Virtual Registers

Virtual registers are 16-bit registers implemented in RAM which can be accessed by the User application to configure and read/write the MM7150 motion module.

TABLE 3-1: DEFINITION TERMS USED IN VIRTUAL REGISTERS

Term / Acronym	Definition
R	“R” is an access type. It is used to indicate if a register or bit is read-only. Bits and registers that are read-only return the current value in the register when read. Reading or writing the bits or registers has no effect on the value.
W	“W” is an access type. It is used to indicate if a register or bit is write-only. Bits and registers that are write-only can only be written to. Reading the bits or registers returns 0.
R/W	“R/W” is an access type. It is used to indicate if a register or bit is read-write. Bits and registers that are read-write return the current value in the register when read. Writing a value will cause the register or bit to be updated. Reading the bits or registers has no effect on the value.

Some characteristics of the virtual registers are:

- Writing the same value again to a VREG has no effect.
- Some VREGs have auto-clearing bits i.e., the bits are automatically cleared after the operation is completed.
- VREG strictly follows the access type.
- Many of the VREGs have corresponding bits in an associated status registers to indicate if the operation corresponding to a VREG write has been successful.
- Some VREGs are valid only if that particular sensor is available on the module (Sensor List register); otherwise those related registers are reserved.

Please see the Appendix B: "Virtual registers map" for details information.

3.1.1 REGISTER DESCRIPTION

3.1.1.1 Register 00h: Sensor Hub Configuration (SHC)

- Access Type: R/W
- POR Default: 0x0001

TABLE 3-2: VREG - 00H

BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8
RES	RES	Raw Gyro EN	Raw Mag EN	Raw Acc EN	RES	RES	Incl EN

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Ori EN	Cmp EN	Gyro EN	Acc EN	MM_Reset	MM_Start	MM_Sleep	MM_Wake

TABLE 3-3: VREG - 00H BITS DESCRIPTION

Bits	Description
0 - MM_Wake	This bit is set by default. It is cleared automatically when sleep bit is set. It has to be set to wake up the host from sleep.
1 - MM_Sleep	By setting this bit, the MM7150 motion module enters D3 deepest sleep state. To wake from this sleep, the HOST SF Lib must issue a wake signal by setting MM_Wake bit 0. This is a self-clearing bit. Any other combination of bits other than what is given in the below table, do not set or clear this register. • MM_Wake = 1, MM_Sleep = 0, MM7150 is Active • MM_Wake = 0, MM_Sleep = 1, MM7150 is Sleep This bit is effective, only if all sensors are disabled. If any sensor is enabled, this bit cannot be set. If this bit is set and the operation is successful (status register), then the next VREG write must be to set MM_Wake. If any other VREG write is issued instead of setting MM_Wake after MM_Sleep, then that VREG write is ignored.
2 - MM_Start	This bit must be set before any sensor is enabled. It can be set only once, after POR. This bit must be set in HOST SF lib initialization. It is only cleared on POR. This bit is required to start the sensor hub. If this bit is not enabled, the sensor enable bits cannot be set. This bit should not be set by user application and should always read as “1” for the user application.
3 - MM_Reset	On setting this bit, the HOST SF lib issues a HID/I^2C reset. This also resets the SHC (except MM_Start), Sensitivity, data rate and data virtual registers to POR values. This is an auto clearing bit.
4 - Acc EN	On setting this bit, the Acc is enabled. Its data can be read from the Acc data register.
5 - Gyro EN	On setting this bit, the Gyro is enabled. Its data can be read from the Gyro data register.
6 - Cmp EN	On setting this bit, the Cmp is enabled. Its data can be read from the Cmp data register.
7 - Ori EN	On setting this bit, the Ori is enabled. Its data can be read from the Ori data register.
8 - Incl EN	On setting this bit, the Incl is enabled. Its data can be read from the Incl data register.
9, 10 - RES	Reserved bits for future use.
11 - Raw Acc EN	On setting this bit, the user is enabled to read raw data of Acc from the raw Acc data register
12 - Raw Gyro EN	On setting this bit, the user is enabled to read raw data of Gyro from the raw Gyro data register
13 - Raw Mag EN	On setting this bit, the user is enabled to read raw data of Mag from the raw Mag data register.
14, 15- RES	Reserved bits for future use

Note: Raw sensors require their corresponding physical / virtual sensors to be enabled. Please refer to **Section 3.2.2.1 “Register 00h: Sensor Hub Config (SHC)”** for more information.

3.1.1.2 Register 01h: Sensor List (SL)

- Access Type: R
- POR Default: 0x0000

TABLE 3-4: VREG - 01H

BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8
RES	RES	RES	RES	RES	RES	RES	RES

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
RES	RES	Cmp	Ori	Incli	Gyro	Mag	Acc

This SL Register is used to track all the sensors (physical and virtual) available in the sensor hub solution. For instance if BIT0 is set, then Acc sensor is available in the sensor Hub solution. If a sensor's corresponding bit is not set in the SL register, it indicates that the particular sensor is not present on the Sensor Hub and all the registers and bits related to that sensor becomes reserved and not accessible by user.

Note: The SL register is populated after the MM_Start bit is set in SHC register.

3.1.1.3 Register 02h-0Bh: xxxx Sensitivity

- Access Type: R/W
- POR Default: 0x0000

Registers from offset 02h-0Bh are used to set the sensitivity of each respective sensor. For instance ACSEN (Acc Sensitivity) register is used to set the sensitivity of the Acc sensor. This register can be read to identify the current Sensitivity of the sensor.

3.1.1.4 Register 0Ch-15h: xxxx Data Rate

- Access Type: R/W
- POR Default: 0x0000

Registers from offset 0Ch-15h are used to set the Data Rate of each respective sensor in milliseconds. The minimum report interval supported is 10 milliseconds. For instance, the ACDR Register is used to set the data rate or report interval for the Accelerometer sensor. In order to get a data sample every 30 milliseconds, this register must be set to 0x001E. This register can be read to identify the current Data Rate /Report interval of the desired sensor.

3.1.1.5 Register 16h-35h: xxxx Data

- Access Type: R
- POR Default: 0x0000

Registers from offset 16h-35h are used to get the Data of each respective sensor. Sensor Data can be read only if the sensor is enabled in the SHC register. Sensor data is updated based on the default report interval supported by the sensor or the user modified report interval; the report interval will be available in the data rate registers. It should be noted that the user application must read the data of a sensor in ascending order of the register offsets. For instance Accelerometer has 3 data registers – AccX, AccY,AccZ, these registers must be read in the order AccX, AccY and then AccZ. If the sensors are disabled, these registers are cleared.

Note: Mag FluxX Data, Mag FluxY Data, Mag FluxZ Data registers are updated when compass is enabled.

3.1.1.6 Register 36h: Product ID

- Access Type: R
- POR Default: 0x0000

This register provides the product ID of the device. This register is populated after the MM_Start bit is set in SHC register and is cleared only on POR.

3.1.1.7 Register 37h: Vendor ID

- Access Type: R
- POR Default: 0x0000

This register provides the product ID of the device. This register is populated after the MM_Start bit is set in SHC register and is cleared only on POR.

3.1.1.8 Register 38h: Device ID

- Access Type: R
- POR Default: 0x0000

This register provides the product ID of the device. This register is populated after the MM_Start bit is set in SHC register and is cleared only on POR.

3.1.1.9 Register 39-3Eh: Exponents

- Access Type: R
- POR Default: 0x0000

Registers 0x39-0x3B gives the unit exponent value of each sensor's data register.

Registers 0x3A-0x3E gives the unit exponent value of each sensor's sensitivity register.

TABLE 3-5: EXPONENTS REGISTER BITS DEFINITION

Registers	Bits	Sensors
Exponent1(0x39)	0-3	Acc data exponent value
	4-7	Gyro data exponent value
	8-11	Cmp data exponent value
	12-15	Ori data exponent value
Exponent2(0x3A)	0-3	Incl data exponent value
	4-11	Reserved
	12-15	Raw Acc data exponent value
Exponent3(0x3B)	0-3	Raw Mag data exponent value
	4-7	Raw Gyro data exponent value
	8-15	Reserved
ExponentCS1(0x3C)	0-3	Acc sensitivity exponent value
	4-7	Gyro sensitivity exponent value
	8-11	Cmp sensitivity exponent value
	12-15	Ori sensitivity exponent value
ExponentCS2(0x3D)	0-3	Incl sensitivity exponent value
	4-11	Reserved
	12-15	Raw Acc sensitivity exponent value
ExponentCS3(0x3E)	0-3	Raw Mag sensitivity exponent value
	4-7	Raw Gyro sensitivity exponent value
	8-15	Reserved

TABLE 3-6: UNIT EXPONENT ENCODING AND MEANINGS

Value	Exponent	Power of Ten
0x00	1x10E0	1
0x01	1x10E1	10
0x02	1x10E2	100
0x03	1x10E3	1 000
0x04	1x10E4	10 000
0x05	1x10E5	100 000
0x06	1x10E6	1 000 000
0x07	1x10E7	10 000 000
0x08	1x10E-8	0.00 000 001
0x09	1x10E-7	0.0 000 001
0x0A	1x10E-6	0.000 001
0x0B	1x10E-5	0.00 001
0x0C	1x10E-4	0.0 001
0x0D	1x10E-3	0.001
0x0E	1x10E-2	0.01
0x0F	1x10E-1	0.1

3.1.1.10 Register 3F-42h: xxxx Status

- Access Type: R/W
- POR Default: 0x0000

These registers are used to give the status of operation supported in SHC/xxxx sensitivity/xxxx Data rate registers. It will indicate whether the intended operation has been completed or if there has been an error. There are 4 status registers in all. All the status registers have the same sequence to clear the status. Once the status is cleared, the VREG operation is repeated again.

3.1.1.11 Register 3F: Status 1, Register 40: Status 2 Bit0-Bit3

These bits hold the status for sensor enable bits in SHC register. 2 Bits are allocated for every sensor enable bit. The status register can be cleared only when they hold a sensor enable/disable failed status or after a reset. The 2 bits indicate the following status:

TABLE 3-7: STATUS OF SENSORS ENABLE

xxxx status1	xxxx status0	Description
0	0	Clear status register
0	1	Sensor enable successful
1	0	Sensor disable successful
1	1	Sensor enable/disable failed

3.1.1.12 Register 40: Status 2 Bit4-Bit15, Register 41-42: Status 3, 4

The status 2, 3, 4 registers also hold the status bits for Sensor sensitivity change and sensor data rate VREG. 3 bits are allocated for every sensor‘s sensitivity and data rate register. The 3 bits indicate the following status:

TABLE 3-8: STATUS OF SENSORS DATA RATE AND SENSITIVITY CHANGES

xxxx CS/Data2 (Error bit)	xxxx CS/Data1 (Data rate bit)	xxxx CS/Data0 (Sensitivity bit)	Description
0	0	0	Clear Data rate status bits.
0	1	0	Update Data rate successful
0	0	1	Update sensitivity successful
0	1	1	Both data rate and sensitivity update successful
1	0	0	Clear Sensitivity status bits.
1	1	0	Update Data rate error
1	0	1	Update sensitivity error
1	1	1	Both data rate and sensitivity error

3.1.1.13 Register 42: Bit 3-6

The bits 3 and 4 have status for the MM_Start bit in the SHC register. The following table indicates the status:

TABLE 3-9: STATUS OF MM_START BIT

MM_Start status1	MM_Start status0	Description
0	0	Clear status register
0	1	MM_Start successful
1	0	Don’t care
1	1	MM_Start error

Similarly, bits 5 and 6 have status for the reset bit in the SHC register. The following table indicates the status:

TABLE 3-10: STATUS OF MM_RESET BIT

MM_Reset status1	MM_Reset status0	Description
0	0	Clear status register
0	1	MM_Reset successful
1	0	Don’t care
1	1	MM_Reset error

3.1.1.14 Register 42: Bit 7-9

3 bits are allocated for MM_Sleep and MM_Wake up. The 3 bits indicate the following status:

TABLE 3-11: STATUS OF MM_SLEEP & MM_WAKE BITS

xxxx CS/Data2 (Error bit)	xxxx CS/Data1 (SH wake bit)	xxxx CS/Data0 (SH sleep bit)	Description
0	0	0	Clear MM_Wake status bits
0	1	0	MM_Wake successful
0	0	1	MM_Sleep successful
0	1	1	Don’t care
1	0	0	Clear MM_Sleep status bits.
1	1	0	MM_Wake error
1	0	1	MM_Sleep error
1	1	1	Don’t care

3.1.2 VREG ACCESS

Two APIs will be available to the user for all the VREGs in the VREG map. These APIs are:

- HOST_SF_Lib_VREG_write
- HOST_SF_Lib_VREG_read

3.1.2.1 Function Prototype

```
UINT8 HOST_SF_LIB_VREG_read (UINT8 offset, UINT16* data)
```

Function to read the content of virtual registers

Where

offset → *offset is the offset/register address of the virtual register*

data* → *data is the pointer to which the VREG data is read into.*

return → *Returns 0 if success, else returns Error code*

```
UINT8 HOST_SF_LIB_VREG_write (UINT8 offset, UINT16 data)
```

Function to queue the write request to virtual registers

Where

offset → *offset is the offset/register address of the virtual register*

data → *data is the content that has to be written to the VREG*

return → *Returns 0 if success, else returns Error code*

Note:	Both VREG_write and VREG_read are non-blocking calls.

3.1.2.2 VREG Write mechanism

Any user application can issue a VREG write with a desired 2 byte value. On success, the 2 byte value is written to the selected VREG. If the write is not successful, an error code is returned. However a VREG write success does not indicate that the desired operation is successful. The status bits of the status VREG give the status of the VREG related operations.

3.1.2.3 VREG Read Mechanism

Every time an application issues a VREG read, it is immediately serviced by HOST SF Lib and the registers two byte value is returned. Only for the data registers, the register needs to be read in ascending order of their offsets for a particular sensor.

3.1.2.4 VREG Write/Read Error Codes

TABLE 3-12: VREG ERROR CODE

Error Codes	Description
0x00	Success
0x01	Error in access type
0x02	Unrecognized VREG offset

3.1.3 VREG INITIALIZATION (HOST SF LIB INITIALIZATION)

The following sequence of operations must be done during initialization of SF lib:

- Configure the I^2C CLK and Data lines
- Configure the GPIO interrupt attention line
- Configure the GPIO wake up line
- Initialize all the virtual registers
- Set MM_Start bit in SHC register (this bit must be set by Host SF lib init)

Note:	Initialization process will require up to ~5s since HID/I^2C reset of the device can take up to 5s as per HID/I^2C spec.

3.2 VREG to HID translation

Every virtual register setting needs to be handled and most of them translate directly or indirectly into a HID/I^2C packet. This section gives an overview on how to access these registers. It also provides information on the HID translation required for each Bit/Registers provided in the VREG map.

3.2.1 TRANSLATION OF VIRTUAL REGISTERS TO HID/I^2C PROTOCOL

The translation of HID/I^2C protocol with respect to virtual register content is carried out under this topic. The virtual registers can be divided into two categories based on HID/I^2C back-end support:

- Command request based
- Device data reading based

3.2.2 COMMAND REQUEST BASED

The Virtual Registers that come under this category are based on Class specific requests (as per HID/I^2C spec). The registers are:

- Sensor Hub Configuration Register
- Sensor Sensitivity Registers
- Sensor Data Rate Registers

3.2.2.1 Register 00h: Sensor Hub Config (SHC)

TABLE 3-13: SENSOR HUB CONFIGURATION OPERATIONS

Bits	HID/I^2C requests sequence	Action to be taken before/after the HID requests
MM_Wake	When this bit is set, the HOST SF LIB needs to issue wake signal, and **wait for 11ms** and send the following HID/I^2C request sequence: 1. SET POWER (ON) **Note:** **Wait for another 30ms** before sending other command to Motion Module	The following actions are to be taken: 1. After successful wake up, BIT1 (MM_Sleep bit) of this register must be cleared.
MM_Sleep	When this bit is set, HOST SF LIB must send the following HID/I^2C request: 1. SET POWER (SLEEP) **Note:** **Wait for 70ms** after sleep command is sent before issuing the wake signal.	This bit is effective, only if all sensors are disabled. If any sensor is enabled, then this bit is not valid and will abort the operation that is trying to set it.

TABLE 3-13: SENSOR HUB CONFIGURATION OPERATIONS (CONTINUED)

Bits	HID/I^{2}C requests sequence	Action to be taken before/after the HID requests
MM_Start	HID/I^{2}C request sequences: 1. Get HID descriptor 2. SET POWER (ON) 3. RESET 4. Get Report descriptor 5. Get Feature Report for all available sensors If step 1 fails, the operation is aborted immediately with an error. If step 4 fails, the operation is returned with an error.	The following actions are to be taken: 1. Parse HID descriptor • Retrieve max input report length • Retrieve register addresses of report descriptor, request register and data register • Retrieve length of report descriptor 2. Parse Report descriptor • Retrieve the sensors available and populate sensor list (VREG) accordingly • Identify the report IDs of various sensors • Retrieve the sizes of all feature reports • Identify the offset of sensor data field in input report • Identify the offset of default sensitivity and data rate in the feature reports 3. Parse Feature Reports • Retrieve the default sensitivity and data rate of all sensors and populate the appropriate sensor's data rate and sensitivity virtual registers with the retrieved values • Get the exponent values of all sensors and populate the sensor exponent virtual registers
MM_Reset	When this bit is set, HOST SF LIB must send the following HID/I^{2}C request: 1. RESET If step 1 fails, the operation is aborted with an error.	The following actions are to be taken: 1. On a successful reset, this bit is cleared 2. The Sensitivity, data rate and status virtual registers need to be reset to POR values except MM_Start status bits 3. All the bits in the SHC must be cleared, since on reset all sensors are disabled except MM_Start bit

TABLE 3-13: SENSOR HUB CONFIGURATION OPERATIONS (CONTINUED)

Bits	HID/I^{2}C requests sequence	Action to be taken before/after the HID requests
Acc EN Gyro EN Cmp EN Ori EN Incl EN Raw Acc EN Raw Gyro EN Raw Mag EN	When any of these bits are set, HOST SF LIB must send the following HID/I^{2}C request: 1. GET FEATURE REPORT 2. SET FEATURE REPORT • Set sensor power state to *HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D0_FULL_POWER_ENUM* • Set sensor reporting state to *HID_USAGE_SENSOR_PROPERTY_REPORTING_STATE_ALL_EVENTS_ENUM* 3. SET FEATURE REPORT If step 1,2 or 3 fails, the operation is aborted with an error. Feature report set in step 2, must be same as what is received in step 3, else steps 2 and 3 needs to be repeated upto 3 times. If the command sequence still fails, return with an error. If this bit is cleared, do the same operation as above but in step 2 set sensor power state to *HID_USAGE_SENSOR_PROPERTY_POWER_STATE_D1_LOW_POWER_ENUM*	The following actions are to be taken: 1. Issue the get feature report with the appropriate report ID for the chosen sensor 2. The feature report length must also be obtained from the report descriptor 3. Only if this bit is set, the sensor data virtual register must be readable
RES	Reserved bits for future use	

In addition to the above table, for any raw sensor to be enabled properly, the corresponding sensors given below must also be enabled.

Note:	It is advised to set the sensitivity registers of both – Raw sensor and Corresponding sensor to Zero to achieve the desired data rates for raw sensors. Non-zero values for sensitivity for either the raw sensor or the corresponding sensor might not guarantee data rate (especially if desired data rate is more than 60ms) in the data rate register for raw sensors.

If multiple bits are set, then the following priority table is followed. If MM_Reset bit is set, it will override the other bits and will reset the device and also all the other bits in this register. MM_Wake and MM_Sleep are always complimentary, if they are given the same values, then both of them are not modified. MM_Sleep is only set if all the other bits (except MM_Start) are cleared.

TABLE 3-14: SENSOR HUB CONFIGURATION PRIORITY

Bits	Priorities
MM_Wake	-
MM_Sleep	-
MM_Start	-
MM_Reset	High
Acc EN	Medium
Gyro EN	Medium
Cmp EN	Medium
Ori EN	Medium
Incl EN	Medium
Raw Acc EN	Medium
Raw Gyro EN	Medium
Raw Mag EN	Medium

Note: **It is always advised to enable accelerometer, before enabling any other sensor. This will ensure appropriate data rate by detecting any movements of the device.**

3.2.2.2 Register 02h-0Bh: xxxx Sensitivity

When this Register is written, the following HID/I^{2}C requests must be issued:

1. GET FEATURE REPORT
2. SET FEATURE REPORT
3. GET FEATURE REPORT

The following actions are to be taken:

1. Issue the set feature report with all the fields same except the sensitivity field which will have the new value now.
2. Verify the SET FEATURE REPORT with a GET FEATURE REPORT and if the sensitivity has not been changed, re-issue the SET FEATURE report. This sequence has to be repeated three times. If sensitivity is still not set, then return with an error.

3.2.2.3 Register 0Ch-15h: xxxx DataRate

When this Register is written, the following HID/I^{2}C requests must be issued:

1. GET FEATURE REPORT
2. SET FEATURE REPORT
3. GET FEATURE REPORT

The following actions are to be taken:

1. Issue the set feature report with all the fields same except the report interval field which will have the new value now.
2. Verify the SET FEATURE REPORT with a GET FEATURE REPORT and if the report interval has not been changed, re-issue the SET FEATURE report. This sequence has to be repeated three times. If report interval is still not set, then return with an error.

Note: Set sensitivity to zero, to achieve correct Data Rate without any movement.

3.2.3 DEVICE DATA READING BASED

The Virtual Registers that come under this category are based on the data read from MM7150:

- Sensor Data Registers

3.2.3.1 Register 16h-35h: xxxx Data

When the alert line is pulled low by MM7150, the most recent data is read from the host. On reading the input report, the following actions are to be taken:

1. Whenever an Input Report is read, the Report must be parsed to identify the report ID. After identifying the sensor related to the Report ID, the data must be parsed and filled into the appropriate sensor data registers. It should be taken care that the sensor data registers must be locked when being written to. However, before locking the sensor data registers, the previous data in sensor data registers must be stored into another shadow registers.
2. The user must be able to read the data register at all times when the sensor is enabled. When the user tries to read data when the sensor data registers are being updated (or locked), then user must be given the previous data. Once the user completes reading the last data register of a particular sensor then the previous data registers are cleared.

If the sensor is disabled, the data register of that sensor must read 0x0000.

3.2.4 OTHER REGISTERS AND FUNCTIONALITIES

3.2.4.1 Register 01h: Sensor List (SL)

This Register is not directly related to HID/I^{2}C. This register holds data based on the sensors available on the hardware. This register is to be populated whenever the MM_Start bit in SHC is set. This register is written only once and will be cleared only on POR.

3.2.4.2 PID, VID, DID Registers

These registers are to be populated whenever the MM_Start bit in SHC is set. These registers are written only once and will be cleared only on POR.

3.2.4.3 Register 39-3Eh: Exponent Registers

These registers are to be populated whenever the MM_Start bit in SHC is set. These registers are written only once and will be cleared only on POR.

3.2.4.4 Status registers

All the write accessible bits in registers have corresponding bits in the status registers. These status bits needs to be populated based on success/failure of a VREG write on a particular VREG.

3.2.5 VREG IMPLEMENTATION BRIEF

Every VREG bit has a corresponding operation related to it. Based on the VREG bits, the VREG to HID translation must take place and the status registers must be updated with the appropriate status. If the user needs to re-issue any operation based on VREG bit, then the corresponding status bit must be cleared. Host SF lib must start the VREG to HID translation on the following conditions:

- Check if Bit is set in VREG
- Check if the corresponding status bits are in cleared state

3.3 HID/I^{2}C Parser

The HID/I^{2}C parser is the layer that provides the HID APIs. The whole HID I^{2}C protocol is carried out under this topic, the I^{2}C request from HOST SF LIB request control flow is sub divided into two parts:

- Descriptor request sequence
- Device Command request sequence

3.3.1 DESCRIPTOR REQUEST SEQUENCE

3.3.1.1 HID Descriptor

The HOST SF LIB must perform I^{2}C combined read for the device descriptor reading sequence and must read 0x1E bytes.

FIGURE 3-1: EXAMPLE FOR DEVICE DESCRIPTOR RETRIEVAL

Address	Write	
Data	Write	01 00
Address	Read	
Data	Read	1E 00 00 01 E7 06 02 00 03 00 0D 00 00 00 00 00 05 00 06 00 24 04 22 13 01 00 00 00 00 00

Note: Device Descriptor address is 0x0001 in this case. Device descriptor Register needs to be available in a specified location in flash.

3.3.1.2 Report Descriptor

The HOST SF LIB requests for the Report descriptor on successful read of the Device descriptor, HOST SF LIB issues the report address specified in the device descriptor for the report descriptor retrieval. I^2C combined write read is used for this reading sequence.

FIGURE 3-2: EXAMPLE FOR REPORT DESCRIPTOR RETRIEVAL

Address	Write	
Data	Write	02 00
Address	Read	
Data	Read	05 20 09 01 A1 00 85 01 05 20 09 73 A1 00 05 20 0A 09 03 15 00 25 02 75 08 95 01 A1 02 0A 30

Note: Report Descriptor Address is 0x0002 in this case.

3.3.2 DEVICE COMMAND REQUEST SEQUENCE

The various commands which must be taken care in the HOST SF LIB are as follows:

- Reset
- Get Report
- Set Report
- Set Power

3.3.2.1 Reset

The reset command is a request that the HOST SF LIB can issue to the DEVICE at any time.

3.3.2.1.1 Reset Request

Reset request type will be as follows:

TABLE 3-15: RESET REQUEST TYPE

Data	Value	Remarks
High Byte		
RESERVED	0000b	This value is reserved and must be set to 0000b
OpCode	0001b	This value is reserved for the RESET Command
Low Byte		
RESERVED	00b	This value is reserved and must be set to 00b
Report Type	00b	The HOST SF LIB shall always set this to the specified value. The DEVICE shall ignore this value and treat it as null.
Report ID	0000b	The HOST SF LIB shall always set this to the specified value. The DEVICE shall ignore this value and treat it as null.

3.3.2.1.2 Reset Response

After the HOST SF LIB sends the RESET command to the Command Register, the DEVICE shall RESET itself back to the initialized state. At the end of the reset, the DEVICE also writes a 2 Byte value to the Input Register with the sentinel value of 0x0000 (2 Bytes containing 0) and asserts the Interrupt to indicate that it has been initialized. The HOST SF LIB must read the input register on reset and interpret the device has been reset if these values have been written.

Note: Device needs to respond within 5 seconds for this request – otherwise HOST SF LIB must consider the device as inoperable.

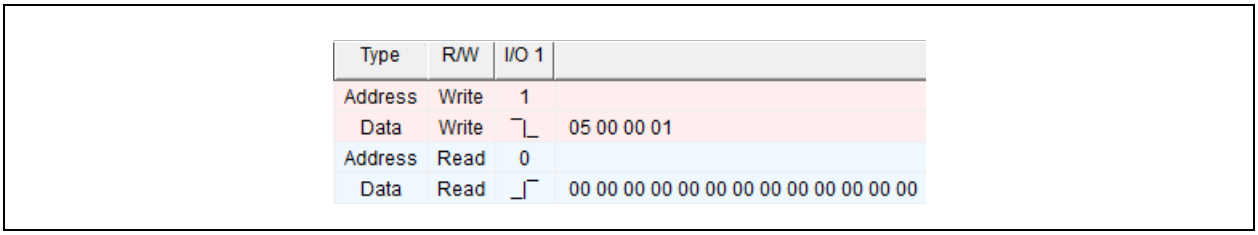
3.3.2.1.3 Reset Command Sequence

The following is the sequence of operations on the HOST SF LIB end and the DEVICE end for this specific request:

TABLE 3-16: RESET COMMAND OPERATIONS

	Host SF LIB Side	Device (MM7150) Side
Step 1	HOST SF LIB sends RESET to Command Register	
Step 2		DEVICE reinitializes itself completely, updates the input register data length field with 0x0000 and then triggers the interrupt.

FIGURE 3-3: EXAMPLE OF RESET COMMAND



3.3.2.2 Get Report

The Get report command is a request (Input or Feature reports) that the HOST SF LIB can issue to the DEVICE at any time after initialization to get a singular report from the DEVICE.

3.3.2.2.1 Get report Request

Get report request type will be as follows:

TABLE 3-17: GET REPORT REQUEST TYPE

Data	Value	Remarks
High Byte		
RESERVED	0000b	This value is reserved and must be set to 0000b
OpCode	0001b	This value is reserved for the GET_REPORT Command
Low Byte		
RESERVED	00b	This value is reserved and must be set to 00b
Report Type	{Input (01) | Feature (11)}	The HOST SF LIB shall always set this to the specified value based on the report type the HOST SF LIB is trying to get. The DEVICE shall honor this value and return data only if supported in the Report Descriptor for the specified report.
Report ID	xxxxb	The HOST SF LIB shall always set this to the specified value based on the TLC specific report the HOST SF LIB is trying to get. The DEVICE shall honor this value per the rules outlined in the notes section below

3.3.2.2.2 Get Report Response

After the HOST SF LIB sends the GET_REPORT command to the Command Register, the DEVICE will fill the report in to the DATA Register. A DEVICE may optionally stretch the clock for the subsequent HOST SF LIB read per repeated start. The Data shall be packaged as follows:

- Length of Report (2 Bytes)
- Report including Report ID

The HOST SF LIB does not issue a new command to the Device (except the RESET Command) until this command has been completed and data has been sent back to the HOST SF LIB. If the DEVICE fails to respond back within a HOST SF LIB defined period of time (generally longer than 5 seconds) the HOST SF LIB is allowed to RESET the DEVICE.

3.3.2.2.3 Get Report Command Sequence

The following is the sequence of operations on the HOST SF LIB end and the DEVICE end for this specific request:

TABLE 3-18: GET REPORT COMMAND OPERATIONS

	Host SF LIB Side	Device (MM7150) Side
Step 1	HOST SF LIB sends GET_REPORT to Command Register	
Step 2		DEVICE fills the Data Register (with length of report and data) to report back to HOST SF LIB.
Step 3	HOST SF LIB reads the first 2 Bytes of DATA register to identify length of report and then reads remainder of data (length identified above) from DATA Register	

FIGURE 3-4: EXAMPLE OF GET REPORT COMMAND

Address	Write	
Data	Write	05 00 31 02 06 00
Address	Read	
Data	Read	13 00 01 01 02 02 02 7D 00 04 00 04 00 14 00 D0 07 30 F8

3.3.2.3 Set Report

The Set Report command is a specific request that the HOST SF LIB may issue to the DEVICE at any time after initialization to set a singular report on the DEVICE.

3.3.2.3.1 Set report Request

The HOST SF LIB shall issue this to the command register and then fill the data register with the report being sent to the DEVICE. The Data shall be packaged as follows:

- Length of Report (2 Bytes)
- Report including Report ID

TABLE 3-19: SET REPORT REQUEST TYPE

Data	Value	Remarks
High Byte		
RESERVED	0000b	This value is reserved and must be set to 0000b
OpCode	0011b	This value is reserved for the SET_REPORT Command
Low Byte		
RESERVED	00b	This value is reserved and must be set to 00b

TABLE 3-19: SET REPORT REQUEST TYPE (CONTINUED)

Data	Value	Remarks
Report Type	{Output (10) | Feature (11)}	The HOST SF LIB shall always set this to the specified value based on the report type the HOST SF LIB is trying to set. The DEVICE shall honor and accept this report if the report type is supported in the Report Descriptor for the specified TLC.
Report ID	xxxxb	The HOST SF LIB shall always set this to the specified value based on the TLC specific report the HOST SF LIB is trying to set. The DEVICE shall honor this value per the rules outlined in the notes section below

3.3.2.3.2 Set Report Response

The DEVICE shall not need to respond back after receiving the data on the device register.

3.3.2.3.3 Set Report Command Sequence

The following is the sequence of operations on the HOST SF LIB end and the DEVICE end for this specific request:

TABLE 3-20: SET REPORT COMMAND OPERATIONS

	Host SF LIB Side	Device (MM7150) Side
Step 1	HOST SF LIB sends SET_REPORT to Command Register	
Step 2	HOST SF LIB fills the Data Register with length and report and writes to the DEVICE.	
Step 3		DEVICE interprets the command and performs the necessary actions. The DEVICE does not need to respond back to HOST SF LIB.

FIGURE 3-5: EXAMPLE OF SET REPORT COMMAND

Address	Write	
Data	Write	05 00 31 03 06 00 13 00 01 01 01 06 02 7D 00 04 00 04 00 14 00 D0 07 30 F8

3.3.2.4 Set Power

The Set Power command is a specific request that the HOST SF LIB may issue to the DEVICE to identify the Power State that the DEVICE should Transition to.

3.3.2.4.1 Set Power Request

The Low Byte uses a special format in the case of SET_POWER Request. The Low byte contains the Power State.

The following are the defined Power State values:

00 = ON

01 = SLEEP

1x = RESERVED

TABLE 3-21: SET POWER REQUEST TYPE

Data	Value	Remarks
High Byte		
RESERVED	0000b	This value is reserved and must be set to 0000b
OpCode	1000b	This value is reserved for the SET_POWER Command
Low Byte		
RESERVED	000000b	This value is reserved and must be set to 000000b
Power State	{On (00) | SLEEP (01)}	The HOST SF LIB shall always set this to the Power State that the DEVICE is required to transition to.

3.3.2.4.2 Set Power Response

The DEVICE shall not respond back after receiving the command. The DEVICE is mandated to enter that power state imminently.

3.3.2.4.3 Set Power Sequence

The following is the sequence of operations on the HOST SF LIB end and the DEVICE end for this specific request:

TABLE 3-22: SET POWER COMMAND OPERATIONS

	Host SF LIB Side	Device (MM7150) Side
Step 1	HOST SF LIB sends SET_POWER to Command Register	
Step 2		DEVICE interprets the command and transitions to the appropriate power state. The DEVICE does not need to respond back to HOST SF LIB.

FIGURE 3-6: SET POWER FOR DEVICE ON

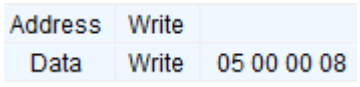
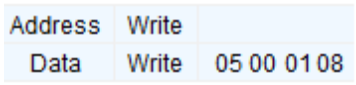


FIGURE 3-7: SET POWER FOR DEVICE SLEEP



3.3.3 FUNCTION PROTOTYPE:

3.3.3.1 HID I2C Descriptor handler

```
VOID hid_i2c_descriptor_handler (UINT8 cmd_req);
```

Where

cmd_reg → *Command request from HOST SF LIB*

Return → *returns success/error code*

The various HOST SF LIB command requests are as follows.

```
GET_HID_DESC        01
GET_RPT_DESC        02
```

The pseudo code for the function is as follows:

```
UINT8 hid_i2c_descriptor_handler (UINT8 cmd_req) {

    UINT8 status;
    Switch (cmd_req) {
        Case GET_HID_DESC:
            Issue the Get HID descriptor request
            Read the HID descriptor and store it.
            status = success/error code
            break;
        Case GET_RPT_DESC:
            Issue the Get Report descriptor request
            Read the Report descriptor and store it.
            status = success/error code
            break;
    }
    Return status
}
```

3.3.3.2 HID I2C command handler

```
UINT8 hid_i2c_cmd_process (UINT8 * buffer, UINT8 cmd_req, UINT8 report_id);
```

Where

****Buffer*** → *Data Buffer pointer for the IN/OUT packet control*

cmd_reg → *Command request from HOST SF LIB*

report_id → *Report ID for the command request from HOST SF LIB if applicable*

Return → *Returns success/error code*

The various HOST SF LIB command requests are as follows.

```
RESET                01
POWER_ON             02
SLEEP                03
HID_GET_RPT_INPT     04
HID_GET_RPT_FEAT     05
HID_SET_RPT_OUTP     06
HID_SET_RPT_FEAT     07
```

The pseudo code for the function is as follows:

```
UINT8 hid_i2c_cmd_process (UINT8 * buffer, UINT8 cmd_req, UINT8 report_id) {

    UINT8 status;
    Switch (cmd_req) {
        Case RESET:
            Issue the Reset command
```

```
            Read input register to verify if device has been reset
            status = success/error code
            break;
        Case POWER_ON:
            Issue the wake signal and wait for 11ms
            Issue the Power On command
            status = success/error code
            break;
        Case SLEEP:
            Issue the Sleep command
            status = success/error code
            break;
        Case HID_GET_RPT_INPT:
            Issue command for HID get report input for the particular report_id
            Fill in the * buffer with the data packet that is read back
            status = success/error code
            break;
        Case HID_GET_RPT_FEAT:
            Issue command for HID get report feature for the particular report_id
            Fill in the buffer with the Feature report data that is read back
            status = success/error code
            break;
        Case HID_SET_RPT_OUTP:
            Issue command for HID set report output for the particular report_id
            Set the output report data into the buffer for transfer
            status = success/error code
            break;
        Case HID_SET_RPT_FEAT:
            Issue command for HID set report Feature for the particular report_id
            Set the feature report data in the buffer that needs to be send
            status = success/error code
            break;
        default:
            status = error code
            break;
    }
    return status;
}
```

3.3.3.3 Error codes

Following Error codes are returned as status in the HID/I^{2}C parser layer:

TABLE 3-23: ERROR CODE FOR HID/I^{2}C PARSER LAYER

Error Code	Description
00	Success
01	HID descriptor retrieval failed: No response from device, wrong HID descriptor register address
02	HID descriptor retrieval failed: VID, PID is zero
03	Report descriptor retrieval failed: No response from device, wrong report descriptor register address
04	Report descriptor retrieval failed: Did not receive the complete report descriptor, error in Report descriptor length that's given in the HID descriptor
05	Report descriptor retrieval failed: Invalid Report Descriptor, end of collection is not found
06	Get feature report Failed: Report id field is null
07	Set Feature Report Failed
08	Reset failed, input register not having 00 00 values

3.4 I²C/SMBus driver layer

This is the physical layer that directly interacts with the MM7150 motion module. The physical layer might be an I²C stack or a SMBus stack based on the hardware support in the MM7150 HOST SoC. The functionality of this layer can be split into 3 main components:

- Master read
- Master write
- Alert line handling

Apart from supporting read/write, this layer also needs to support alert line handling.

3.4.1 ALERT LINE HANDLING

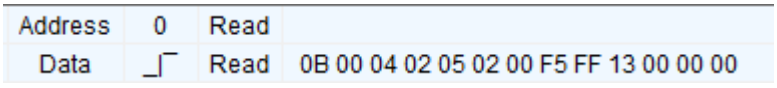
Alert Pin is a HOST SF LIB GPIO pin configured as per HID/I²C alert pin specification. The corresponding GPIO interrupt must also be enabled. When the alert line is pulled low, the HOST SF Lib must issue an I²C read for MAX INPUT REPORT LENGTH of bytes.

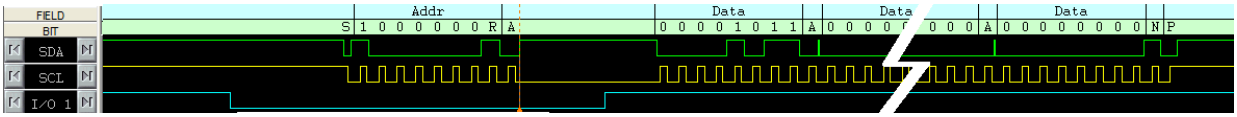
3.4.1.1 Data reading Sequence

The following sequence of operations explains the process of retrieval of the Input Report:

TABLE 3-24: INPUT REPORT OPERATIONS

Sequence	Host SF LIb Side	Device (MM7150) Side
Step 1		DEVICE asserts the Interrupt indicating that it has an Input Report to send to HOST SF LIB.
Step 2	HOST SF LIB issues a READ request over the I²C protocol, after it receives the interrupt.	
Step 3		DEVICE returns the length (2 Bytes) and the entire Input Report.
Step 4		If the DEVICE has no more Input Reports to send, it de-asserts the interrupt line.





3.5 Vendor Collection

Host SF lib supports sensor collection with the VREGs. To support Vendor collections, Host SF lib will give direct access to the HID/I^{2}C layer within the HOST SF lib, thus giving access to issue HID get report, set report commands.

3.5.1 USER APIS

Two APIs will be available to the user for implementing applications based on vendor collections:

- HOST_SF_Lib_HID_Get_Feature
- HOST_SF_Lib_HID_Set_Feature

3.5.1.1 Function Prototype

```
UINT8 HOST_SF_LIB_ HID_Get_Report (UINT8 type, VOID* ReportBuffer, UINT8 size )
```

Function to read the content of virtual registers

Where

type → *type refers to feature or input report; type = 3 – feature, 1 - input*

ReportBuffer * → *ReportBuffer is the pointer to which the feature report is read into*

size → *Specifies the size, in bytes, of the report buffer*

return → *Returns 0 if success, else returns Error code*

```
UINT8 HOST_SF_LIB_ HID_Set_Feature (UINT8 type, VOID* ReportBuffer, UINT8 size)
```

Function to queue the write request to virtual registers

Where

type → *type refers to feature or output report; type = 3 – feature, 2 - output*

ReportBuffer* → *ReportBuffer is the pointer data that needs to be sent to the device*

size → *Specifies the size, in bytes, of the report buffer*

return → *Returns 0 if success, else returns Error code*

Note: The caller must set the first byte of the *ReportBuffer* parameter to a nonzero report ID.

3.5.1.2 HOST_SF_LIB_ HID_Get_Report mechanism

Any user application can issue a Get_report request, which will directly pass to the HID/I^{2}C layer of Host SF lib to retrieve the desired report. The user application needs to wait till the transaction is complete (Blocking call).

3.5.1.3 HOST_SF_LIB_ HID_Set_Report mechanism

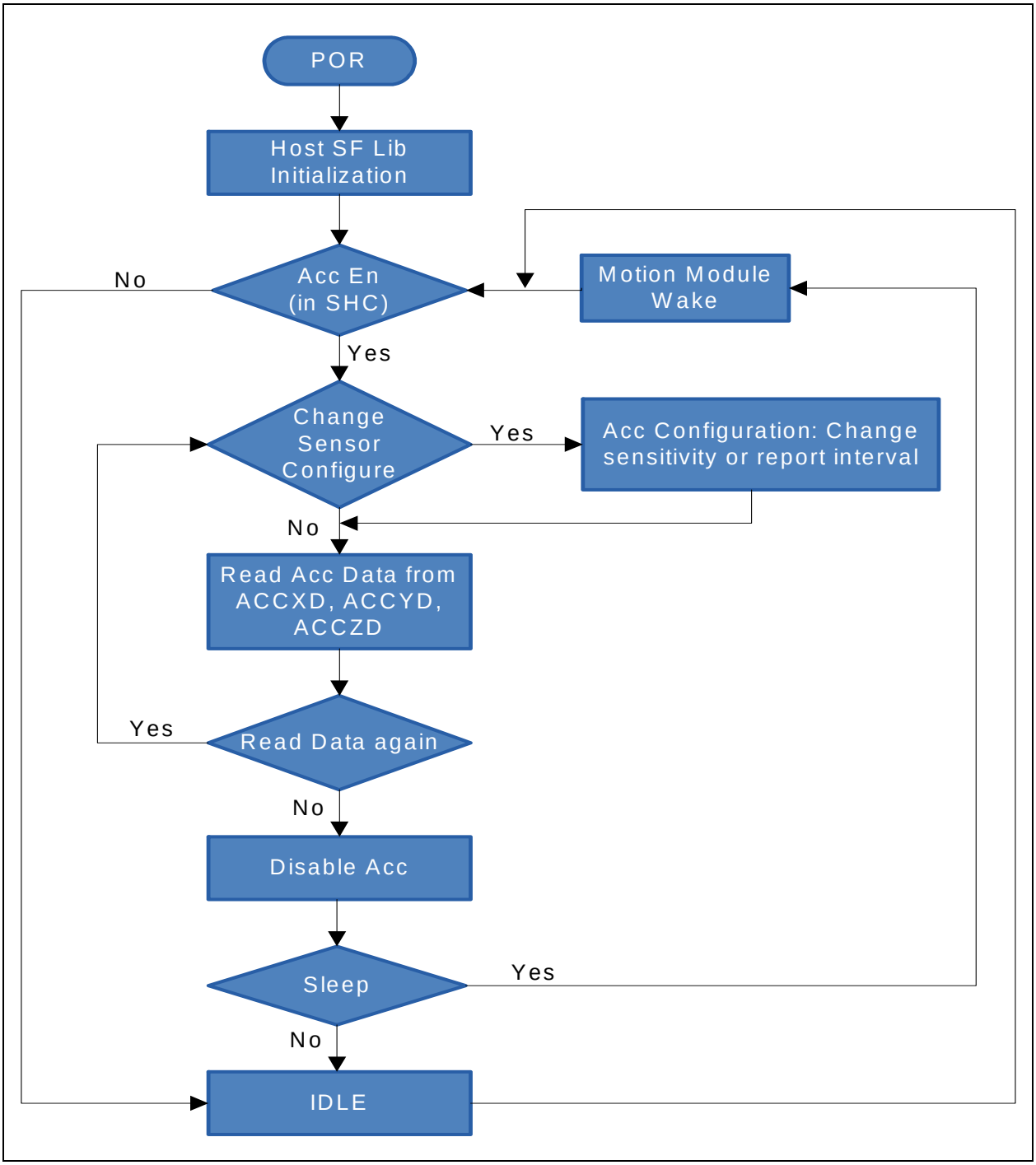
Any user application can issue a Set_report request, which will directly pass to the HID/I^{2}C layer of Host SF lib to send the desired report. The user application needs to wait till the transaction is complete (Blocking call).

3.5.1.4 Error code

The error code returned by APIs will be the same as the error codes returned by the HID I^{2}C parser layer (see Section 3.3, HID/I^{2}C Parser for more information).

Note: Any applications that use vendor collection will need to use these APIs to build their application.

4.0 SAMPLE FLOW CHART FOR ACCELEROMETER



APPENDIX A: SAMPLE CODE FOR ACCELEROMETER

Sample application code that uses HOST SF lib to enable and use accelerometer.

```
/*
App init task: the Applications initialization function.
*/
void App init task (void) {

    return code = VREG_write (SHC, 0xYY)//Enable accelerometer
    return code = VREG_write (Accel_Sensitivity,02)//set sensitivity
    return code = VREG_write (Accel_data_rate, 10)//set data rate
}
//Note: VREG writes fail if a read-only register offset is given or an unsupported offset is given

/*
App main task: the Application's main function.
*/
void  App main task (void) {
    while (1) {
        return = VREG_read (status register, status) //Check if acc enable has been successful
        if (status == operation completed) {
            VREG_read (accel data)
            /*Do the required processing */
            VREG_write (Accel_data_rate, 20) //change report interval based on some state
            return = VREG_read(status register, status)
            if (status == operation completed){
                /*Do the desired operations*/
            }
            else {
                if (status! = error)
                /*Do the task with current report interval, wait for the change report interval in
                next iteration*/
                else
                /*If error is detected and Change report interval failed, then do error handling*/
            }
        }
        else {
            if (status == error)
            //Issue Accel enable again or exit with accel enable failed
        }
    }
}
```


APPENDIX B: VIRTUAL REGISTERS MAP

VREG ADDR (offset)	REG ACC TYPE	REGISTER NAME	ABBR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	POR
00	R/W	Sensor Hub Configure	SHC	RES	RES	Raw Gyro EN	Raw Mag EN	Raw Acc EN	RES	RES	Incl EN	Ori EN	Cmp EN	Gyro EN	Acc EN	MM_Reset	MM_Start	MM_Sleep	MM_Wake	0x01
01	R	Sensor List	SL	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	Incl	Ori	Cmp	Gyro	Acc	0x00
02	R/W	Acc Sensitivity	ACSEN	ACSEN15	ACSEN14	ACSEN13	ACSEN12	ACSEN11	ACSEN10	ACSEN9	ACSEN8	ACSEN7	ACSEN6	ACSEN5	ACSEN4	ACSEN3	ACSEN2	ACSEN1	ACSEN0	0x00
03	R/W	Gyro Sensitivity	GYSEN	GYSEN15	GYSEN14	GYSEN13	GYSEN12	GYSEN11	GYSEN10	GYSEN9	GYSEN8	GYSEN7	GYSEN6	GYSEN5	GYSEN4	GYSEN3	GYSEN2	GYSEN1	GYSEN0	0x00
04	R/W	Cmp Sensitivity	CMSEN	CMSEN15	CMSEN14	CMSEN13	CMSEN12	CMSEN11	CMSEN10	CMSEN9	CMSEN8	CMSEN7	CMSEN6	CMSEN5	CMSEN4	CMSEN3	CMSEN2	CMSEN1	CMSEN0	0x00
05	R/W	Ori Sensitivity	ORSEN	ORSEN15	ORSEN14	ORSEN13	ORSEN12	ORSEN11	ORSEN10	ORSEN9	ORSEN8	ORSEN7	ORSEN6	ORSEN5	ORSEN4	ORSEN3	ORSEN2	ORSEN1	ORSEN0	0x00
06	R/W	Incl Sensitivity	INSEN	INSEN15	INSEN14	INSEN13	INSEN12	INSEN11	INSEN10	INSEN9	INSEN8	INSEN7	INSEN6	INSEN5	INSEN4	INSEN3	INSEN2	INSEN1	INSEN0	0x00
07	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
08	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
09	R/W	Raw Acc Sensitivity	RACSEN	RACSEN15	RACSEN14	RACSEN13	RACSEN12	RACSEN11	RACSEN10	RACSEN9	RACSEN8	RACSEN7	RACSEN6	RACSEN5	RACSEN4	RACSEN3	RACSEN2	RACSEN1	RACSEN0	0x00
0A	R/W	Raw Mag Sensitivity	RMGSEN	RMGSEN15	RMGSEN14	RMGSEN13	RMGSEN12	RMGSEN11	RMGSEN10	RMGSEN9	RMGSEN8	RMGSEN7	RMGSEN6	RMGSEN5	RMGSEN4	RMGSEN3	RMGSEN2	RMGSEN1	RMGSEN0	0x00
0B	R/W	Raw Gyro Sensitivity	RGYSEN	RGYSEN15	RGYSEN14	RGYSEN13	RGYSEN12	RGYSEN11	RGYSEN10	RGYSEN9	RGYSEN8	RGYSEN7	RGYSEN6	RGYSEN5	RGYSEN4	RGYSEN3	RGYSEN2	RGYSEN1	RGYSEN0	0x00
0C	R/W	Acc Data Rate	ACDXR	ACDXR15	ACDXR14	ACDXR13	ACDXR12	ACDXR11	ACDXR10	ACDXR9	ACDXR8	ACDXR7	ACDXR6	ACDXR5	ACDXR4	ACDXR3	ACDXR2	ACDXR1	ACDXR0	0x00
0D	R/W	Gyro Data Rate	GYDR	GYDR15	GYDR14	GYDR13	GYDR12	GYDR11	GYDR10	GYDR9	GYDR8	GYDR7	GYDR6	GYDR5	GYDR4	GYDR3	GYDR2	GYDR1	GYDR0	0x00
0E	R/W	Cmp Data Rate	CMDR	CMDR15	CMDR14	CMDR13	CMDR12	CMDR11	CMDR10	CMDR9	CMDR8	CMDR7	CMDR6	CMDR5	CMDR4	CMDR3	CMDR2	CMDR1	CMDR0	0x00

VREG ADDR (offset)	REG ACC TYPE	REGISTER NAME	ABBR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	POR
0F	R/W	Ori Data Rate	ORDR	ORDR 15	ORDR 14	ORDR 13	ORDR 12	ORDR 11	ORDR 10	ORDR 9	ORDR 8	ORDR 7	ORDR 6	ORDR 5	ORDR 4	ORDR 3	ORDR 2	ORDR 1	ORDR 0	0x00
10	R/W	Incl Data Rate	INDR	INDR15	INDR14+F32	INDR13	INDR12	INDR11	INDR10	INDR9	INDR8	INDR7	INDR6	INDR5	INDR4	INDR3	INDR2	INDR1	INDR0	0x00
11	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
12	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
13	R/W	Raw Acc Data Rate	RACDR	RACD R15	RACD R14	RACD R13	RACD R12	RACD R11	RACD R10	RACD R9	RACD R8	RACD R7	RACD R6	RACD R5	RACD R4	RACD R3	RACD R2	RACD R1	RACD R0	0x00
14	R/W	Raw Mag Data Rate	RMGDR	RMGD R15	RMGD R14	RMGD R13	RMGD R12	RMGD R11	RMGD R10	RMGD R9	RMGD R8	RMGD R7	RMGD R6	RMGD R5	RMGD R4	RMGD R3	RMGD R2	RMGD R1	RMGD R0	0x00
15	R/W	Raw Gyro Data Rate	RGYDR	RGYD R15	RGYD R14	RGYD R13	RGYD R12	RGYD R11	RGYD R10	RGYD R9	RGYD R8	RGYD R7	RGYD R6	RGYD R5	RGYD R4	RGYD R3	RGYD R2	RGYD R1	RGYD R0	0x00
16	R/W	AccX Data	ACXD	ACXD 15	ACXD 14	ACXD 13	ACXD 12	ACXD 11	ACXD 10	ACXD 9	ACXD 8	ACXD 7	ACXD 6	ACXD 5	ACXD 4	ACXD 3	ACXD 2	ACXD 1	ACXD 0	0x00
17	R/W	AccY Data	ACYD	ACYD 15	ACYD 14	ACYD 13	ACYD 12	ACYD 11	ACYD 10	ACYD 9	ACYD 8	ACYD 7	ACYD 6	ACYD 5	ACYD 4	ACYD 3	ACYD 2	ACYD 1	ACYD 0	0x00
18	R/W	AccZ Data	ACZD	ACZD 15	ACZD 14	ACZD 13	ACZD 12	ACZD 11	ACZD 10	ACZD 9	ACZD 8	ACZD 7	ACZD 6	ACZD 5	ACZD 4	ACZD 3	ACZD 2	ACZD 1	ACZD 0	0x00
19	R/W	Raw AccX Data	RACXD	ACXD 15	ACXD 14	ACXD 13	ACXD 12	ACXD 11	ACXD 10	ACXD 9	ACXD 8	ACXD 7	ACXD 6	ACXD 5	ACXD 4	ACXD 3	ACXD 2	ACXD 1	ACXD 0	0x00
1A	R/W	Raw AccY Data	RACYD	ACYD 15	ACYD 14	ACYD 13	ACYD 12	ACYD 11	ACYD 10	ACYD 9	ACYD 8	ACYD 7	ACYD 6	ACYD 5	ACYD 4	ACYD 3	ACYD 2	ACYD 1	ACYD 0	0x00
1B	R/W	Raw AccZ Data	RACZD	ACZD 15	ACZD 14	ACZD 13	ACZD 12	ACZD 11	ACZD 10	ACZD 9	ACZD 8	ACZD 7	ACZD 6	ACZD 5	ACZD 4	ACZD 3	ACZD 2	ACZD 1	ACZD 0	0x00
1C	R/W	GyroX Data	GYXD	GYXD 15	GYXD 14	GYXD 13	GYXD 12	GYXD 11	GYXD 10	GYXD 9	GYXD 8	GYXD 7	GYXD 6	GYXD 5	GYXD 4	GYXD 3	GYXD 2	GYXD 1	GYXD 0	0x00
1D	R/W	GyroY Data	GYYD	GYYD 15	GYYD 14	GYYD 13	GYYD 12	GYYD 11	GYYD 10	GYYD 9	GYYD 8	GYYD 7	GYYD 6	GYYD 5	GYYD 4	GYYD 3	GYYD 2	GYYD 1	GYYD 0	0x00
1E	R/W	GyroZ Data	GYZD	GYZD 15	GYZD 14	GYZD 13	GYZD 12	GYZD 11	GYZD 10	GYZD 9	GYZD 8	GYZD 7	GYZD 6	GYZD 5	GYZD 4	GYZD 3	GYZD 2	GYZD 1	GYZD 0	0x00
1F	R/W	Raw GyroX Data	RGYXD	RGYX D15	RGYX D14	RGYX D13	RGYX D12	RGYX D11	RGYX D10	RGYX D9	RGYX D8	RGYX D7	RGYX D6	RGYX D5	RGYX D4	RGYX D3	RGYX D2	RGYX D1	RGYX D0	0x00

VREG ADDR (offset)	REG ACC TYPE	REGISTER NAME	ABBR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	POR
20	R/W	Raw GyroY Data	RGYYD	RGYY D15	RGYY D14	RGYY D13	RGYY D12	RGYY D11	RGYY D10	RGYY D9	RGYY D8	RGYY D7	RGYY D6	RGYY D5	RGYY D4	RGYY D3	RGYY D2	RGYY D1	RGYY D0	0x00
21	R/W	Raw GyroZ Data	RGYZD	RGYZ D15	RGYZ D14	RGYZ D13	RGYZ D12	RGYZ D11	RGYZ D10	RGYZ D9	RGYZ D8	RGYZ D7	RGYZ D6	RGYZ D5	RGYZ D4	RGYZ D3	RGYZ D2	RGYZ D1	RGYZ D0	0x00
22	R/W	Cmp Data	CMD	CMD15	CMD14	CMD13	CMD12	CMD11	CMD10	CMD9	CMD8	CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0	0x00
23	R/W	Mag FluxX Data	MGFXD	MGFX D15	MGFX D14	MGFX D13	MGFX D12	MGFX D11	MGFX D10	MGFX D9	MGFX D8	MGFX D7	MGFX D6	MGFX D5	MGFX D4	MGFX D3	MGFX D2	MGFX D1	MGFX D0	0x00
24	R/W	Mag FluxY Data	MGFYD	MGFY D15	MGFY D14	MGFY D13	MGFY D12	MGFY D11	MGFY D10	MGFY D9	MGFY D8	MGFY D7	MGFY D6	MGFY D5	MGFY D4	MGFY D3	MGFY D2	MGFY D1	MGFY D0	0x00
25	R/W	Mag FluxZ Data	MGFZD	MGFZ D15	MGFZ D14	MGFZ D13	MGFZ D12	MGFZ D11	MGFZ D10	MGFZ D9	MGFZ D8	MGFZ D7	MGFZ D6	MGFZ D5	MGFZ D4	MGFZ D3	MGFZ D2	MGFZ D1	MGFZ D0	0x00
26	R/W	Raw MagX Data	RMGXD	RMGX D15	RMGX D14	RMGX D13	RMGX D12	RMGX D11	RMGX D10	RMGX D9	RMGX D8	RMGX D7	RMGX D6	RMGX D5	RMGX D4	RMGX D3	RMGX D2	RMGX D1	RMGX D0	0x00
27	R/W	Raw MagY Data	RMGYD	RMGY D15	RMGY D14	RMGY D13	RMGY D12	RMGY D11	RMGY D10	RMGY D9	RMGY D8	RMGY D7	RMGY D6	RMGY D5	RMGY D4	RMGY D3	RMGY D2	RMGY D1	RMGY D0	0x00
28	R/W	Raw MagZ Data	RMGZD	RMGZ D15	RMGZ D14	RMGZ D13	RMGZ D12	RMGZ D11	RMGZ D10	RMGZ D9	RMGZ D8	RMGZ D7	RMGZ D6	RMGZ D5	RMGZ D4	RMGZ D3	RMGZ D2	RMGZ D1	RMGZ D0	0x00
29	R/W	OriX Data	ORXD	ORXD 15	ORXD 14	ORXD 13	ORXD 12	ORXD 11	ORXD 10	ORXD 9	ORXD 8	ORXD 7	ORXD 6	ORXD 5	ORXD 4	ORXD 3	ORXD 2	ORXD 1	ORXD 0	0x00
2A	R/W	OriY Data	ORYD	ORYD 15	ORYD 14	ORYD 13	ORYD 12	ORYD 11	ORYD 10	ORYD 9	ORYD 8	ORYD 7	ORYD 6	ORYD 5	ORYD 4	ORYD 3	ORYD 2	ORYD 1	ORYD 0	0x00
2B	R/W	OriZ Data	ORZD	ORZD 15	ORZD 14	ORZD 13	ORZD 12	ORZD 11	ORZD 10	ORZD 9	ORZD 8	ORZD 7	ORZD 6	ORZD 5	ORZD 4	ORZD 3	ORZD 2	ORZD 1	ORZD 0	0x00
2C	R/W	OriW Data	ORWD	ORW D15	ORW D14	ORW D13	ORW D12	ORW D11	ORW D10	ORW D9	ORW D8	ORW D7	ORW D6	ORW D5	ORW D4	ORW D3	ORW D2	ORW D1	ORW D0	0x00
2D	R/W	InclX Data	INXD	INXD15	INXD14	INXD13	INXD12	INXD11	INXD10	INXD9	INXD8	INXD7	INXD6	INXD5	INXD4	INXD3	INXD2	INXD1	INXD0	0x00
2E	R/W	InclY Data	INYD	INYD15	INYD14	INYD13	INYD12	INYD11	INYD10	INYD9	INYD8	INYD7	INYD6	INYD5	INYD4	INYD3	INYD2	INYD1	INYD0	0x00
2F	R/W	InclZ Data	INZD	INZD15	INZD14	INZD13	INZD12	INZD11	INZD10	INZD9	INZD8	INZD7	INZD6	INZD5	INZD4	INZD3	INZD2	INZD1	INZD0	0x00
30	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00

VREG ADDR (offset)	REG ACC TYPE	REGISTER NAME	ABBR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	POR
31	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
32	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
33	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
34	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
35	R/W	Reserved	-	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	0x00
36	R	Product ID	PID	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0	0x00
37	R	Vendor ID	VID	VID15	VID14	VID13	VID12	VID11	VID10	VID9	VID8	VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	0x00
38	R	Device ID	DID	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	0x00
39	R	Exponent1	exp1	expO	expO	expO	expO	expC3	expC2	expC1	expC0	expG3	expG2	expG1	expG0	expA3	expA2	expA1	expA0	0x00
3A	R	Exponent2	exp2	expRA3	expRA2	expRA1	expRA0	RES	RES	RES	RES	RES	RES	RES	RES	expIn3	expIn2	expIn1	expIn0	0x00
3B	R	Exponent3	exp3	Res	Res	Res	Res	Res	Res	Res	Res	expRG3	expRG2	expRG1	expRG0	expRM3	expRM2	expRM1	expRM0	0x00
3C	R	ExponentCS1	expCSs1	expCSsO	expCSO	expCSO	expCSO	expCSC3	expCSC2	expCSC1	expCSC0	expCSG3	expCSG2	expCSG1	expCSG0	expCSA3	expCSA2	expCSA1	expCSA0	0x00
3D	R	ExponentCS2	expCS2	expCSRA3	expCSRA2	expCSRA1	expCSRA0	RES	RES	RES	RES	RES	RES	RES	RES	expCSIn3	expCSIn2	expCSIn1	expCSIn0	0x00
3E	R	ExponentCS3	expCS3	Res	Res	Res	Res	Res	Res	Res	Res	expCSRG3	expCSRG2	expCSRG1	expCSRG0	expCSRM3	expCSRM2	expCSRM1	expCSRM0	0x00
3F	R/W	Status 1	stat1	Raw Acc status 1	Raw Acc status 0	RES	RES	RES	RES	Inc status 1	Inc status 0	Ori status 1	Ori status 0	Cmp status 1	Cmp status 0	GyroStatus1	GyroStatus0	AccStatus1	AccStatus0	0x00
40	R/W	Status 2	stat2	Ori CS/ data2	Ori CS/ data1	Ori CS/ data0	Cmp CS/ data2	Cmp CS/ data1	Cmp CS/ data0	Gyro CS/ data2	Gyro CS/ data1	Gyro CS/ data0	Acc CS/ data2	Acc CS/ data1	Acc CS/ data0	Raw gyro Status 1	Raw gyro Status 0	Raw Mag Status 1	Raw Mag Status 0	0x00

VREG ADDR (offset)	REG ACC TYPE	REGISTER NAME	ABBR	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	POR
41	R/W	Status 3	stat3	RES	Raw Mag CS/ data2	Raw Mag CS/ data1	Raw Mag CS/ data0	Raw Acc CS/ data2	Raw Acc CS/ data1	Raw Acc CS/ data0	RES	RES	RES	RES	RES	RES	Incl CS/ data2	Incl CS/ data1	Incl CS/ data0	0x00
42	R/W	Status 4	stat4	RES	RES	RES	RES	RES	RES	MM sleep / wake status 2	MM sleep / wake status 1	MM sleep / wake status 0	MM Reset status 1	MM Reset status 0	MM Start status 1	MMStart status 0	Raw Mag CS/ data2	Raw Mag CS/ data1	Raw Mag CS/ data0	0x00

APPENDIX C: APPLICATION NOTE REVISION HISTORY

TABLE C-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS00001873A (12-22-14)	Document Release	

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 9781632768872

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14