Application Note 607002

Title: **Getting started with the EM7180SFP**

Product Family: **Motion and optical sensing**

Part Number: EM7180SFP

Keywords: Start

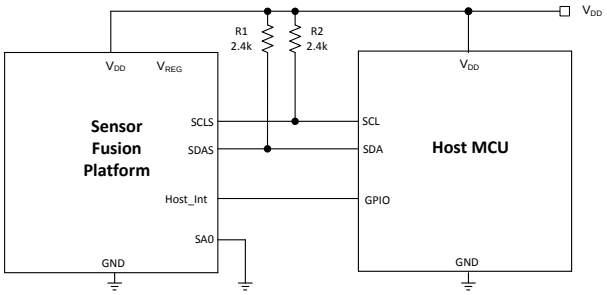
## 1- Introduction

You have received the first samples of EM7180SFP castellated module and you want to make the first trials, please read this application note.

## 2- Scope

This application note aims at giving a quick overview of how to start using the EM7180SFP sensor fusion platform and how to extract the quaternion data calculated by the sensor fusion platform.

## 3- Connections




Connection example between the host and one SFP

The host communicates with I2C communication using the SDAS (Serial Data Line) and SCL (Serial Clock Line). The Host_int pin provides the desired interruption to the host controller.

## 4- Initializing the SFP

At start-up (ie Power-up, watchdog reset or I2C reset), the configuration is uploaded from the E2PROM. The module E2PROM is already programmed so that the module is already functional. The host must wait that the Host_Int signal goes high which indicates that the boot process is completed.
Then, the SentralStatus register can be checked to verify that the software upload was done correctly (SentralStatus=01011).

- Wait for Host_int=1

*While(host_int==0) wait*

- Verify that the upload is correct

*Read SentralStatus?=0B*

After this first step, the host sets the platform parameters to values allowing proper behaviour. Output Data Rates (ODR) are set as follows:
ODRmag=$MagRate_d$
ODRacc=$AccelRate_d$*10
ODRgyro=$GyroRate_d$*10 and
ODRquat= $GyroRate_d$*10 / $QRateDivisor_d$

For best performance, the magnetometer should be set to 30Hz. The magnetometer output data rate can be reduced down to 8Hz.
We usually recommend data rates of 100Hz for the accelerometer and 200Hz for the gyrometer. The proper settings can be sent as follows:
$MagRate_d$=30Hz
$AccelRate_d$*10=100Hz
$GyroRate_d$*10=200Hz and
$GyroRate_d$*10 / $QRateDivisor_d$=200Hz

- Set sensor ODR

*Write MagRate 1E*
*Write AccelRate 0A*
*Write GyroRate 14*

- Set Quaternion ODR

*Write QRateDivisor 01*

Finally, the interrupt on quaternion availability needs to be set (in addition to error and CPU-reset cases) and the platform can enter in normal mode thanks to the following commands:

- Configure Trigger

*Write EnableEvents 07*

- Enter in normal Operation Mode

*Write HostControl 01*

Where "Write" is the operation consisting in writing the second argument value into the first argument address.

## 5- Extracting the quaternions

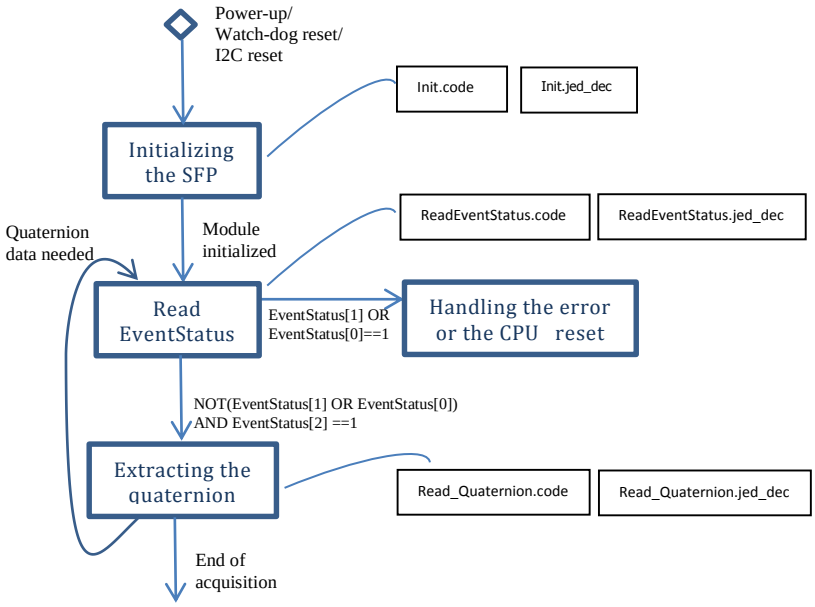
Once the initialization phase has correctly occurred, it is possible to extract the quaternions calculated by the fusion algorithm embedded in the sensor fusion platform.
When the host_int goes high, the host must read the EventStatus register. The reading of EventStatus is needed to make sure that the next quaternion data is a new one (EventStatus is cleared). When EventStatus[1] and EventStatus[0] are both low, the quaternion data must be ready.

- Trigger when hostInt=1 and read the data

While(host_Int==0) wait
*Read EventStatus*
If EventStatus[1] OR EventStatus[0]==1,
then the error or the CPU reset must be handled (see datasheet)
else
if EventStatus[2]==1
*read QX*
*read QY*
*read QZ*
*read QW*
*read QTIME*
else "an error occurred"

## 6- Simplified quaternion extraction flow

The block diagram below summarizes the steps from power-up/reset to quaternion extraction. Embedded in the block diagram, you can find the corresponding communication signals stored in form of standard jedec and symbolic code file, both files including common comments (starting with ;). Please refer to these signals as an example in order to get started fast and to obtain a starting point for your system including the sensors fusion module.




*Note:.code and .jed_dec files are inserted as attachments in this PDF file*