Objective

This code example demonstrates how to interface a PSoC® 6 MCU with a BMI160 motion sensor. This example reads steps counted by the sensor to emulate a pedometer. Raw motion data is also read and used to estimate the orientation of the board.

Overview

This example configures and reads data from a BMI160 motion sensor using PSoC 6 MCU. The example uses the BMI160 motion sensor to detect and count steps from activities such as walking or running, emulating the functionality of a pedometer. The motion sensor's accelerometer data is also read and converted to indicate the orientation of the sensor with respect to the ground. The step count and orientation information is displayed on the E-INK display.

This code example assumes that you are familiar with the PSoC 6 MCU device and the PSoC Creator™ IDE. If you are new to PSoC 6 MCU, see the application note AN210781 - Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

Requirements

Tool: PSoC Creator 4.2; Peripheral Driver Library (PDL) 3.0.1

Programming Language: C (Arm® GCC 5.4.1 and Arm MDK 5.22)

Associated Parts: All PSoC 6 MCU parts

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Hardware Setup

Set the switches and jumpers as shown in Table 1.

Table 1. Switch and Jumper Selection

Switch / Jumper	Position	Location
SW5	3.3 V	Front
SW6	PSoC 6 BLE	Back
SW7	V_{DDD} / KitProg2	Back
J8	Installed	Back

Note: This code example does not support supply voltages other than 3.3 V due to limitations on the voltage required for the inertial measurement unit (IMU) and RGB LED.

Figure 1. Hardware Setup

Software Setup

Install the CY8CKIT-62-BLE PSoC 6 BLE Pioneer Kit software, which contains all the required software to evaluate this code example. No additional software setup is required.

Operation

1. Connect the Pioneer Board to your PC using the provided USB cable through the USB connector (J10).
2. Open a terminal software such as Tera Term and select the KitProg2's COM port with a baud rate setting of 115200 bps. Use the default settings for other serial port parameters.
3. Build the project and program it into the PSoC 6 MCU. Choose **Debug** > **Program**. For more information on device programming, see the CY8CKIT-062-BLE kit guide. Flash for both CPUs is programmed in a single program operation.

   Note: Do not replace or delete the *stdio_user.h* file.
4. On successful programming, the program loads and refreshes the E-INK display as shown in Figure 2. Confirm that the terminal application displays the code example title and the initial step counts as shown in Figure 3.

Figure 2. E-INK Display on Program Startup

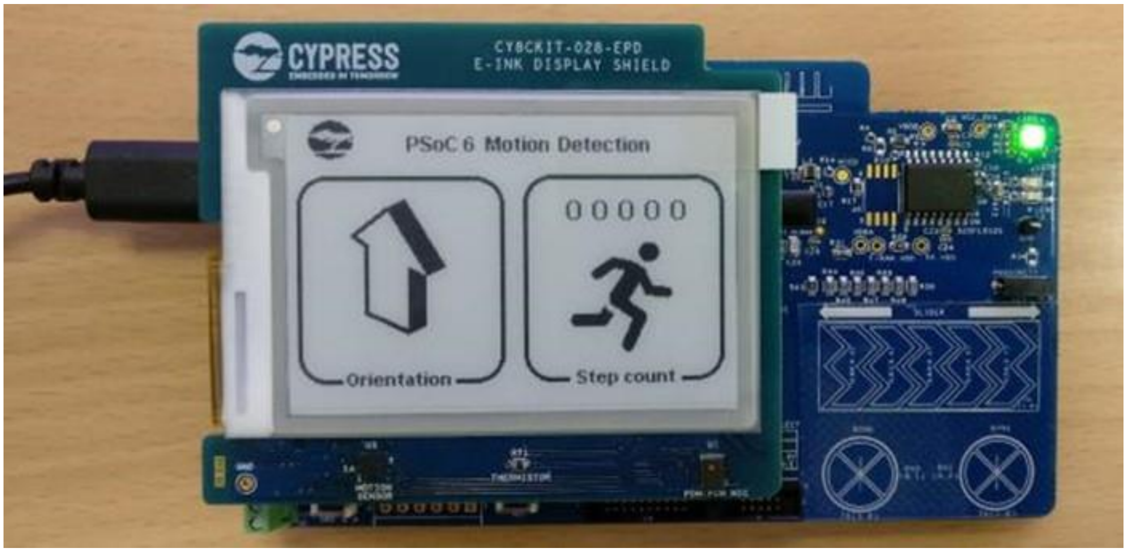
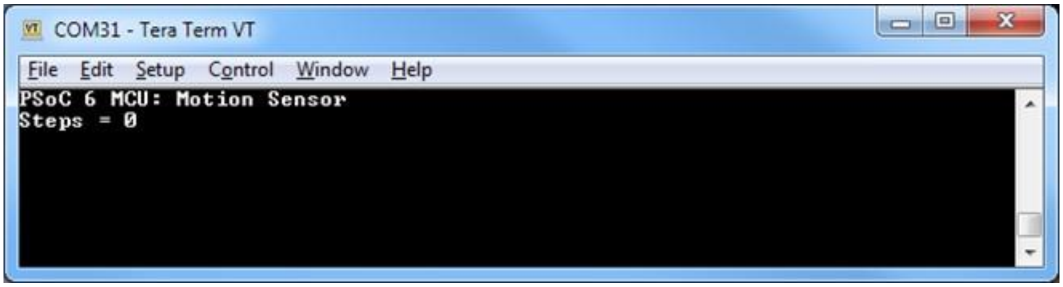


Figure 3. Terminal Application Displaying Startup Message



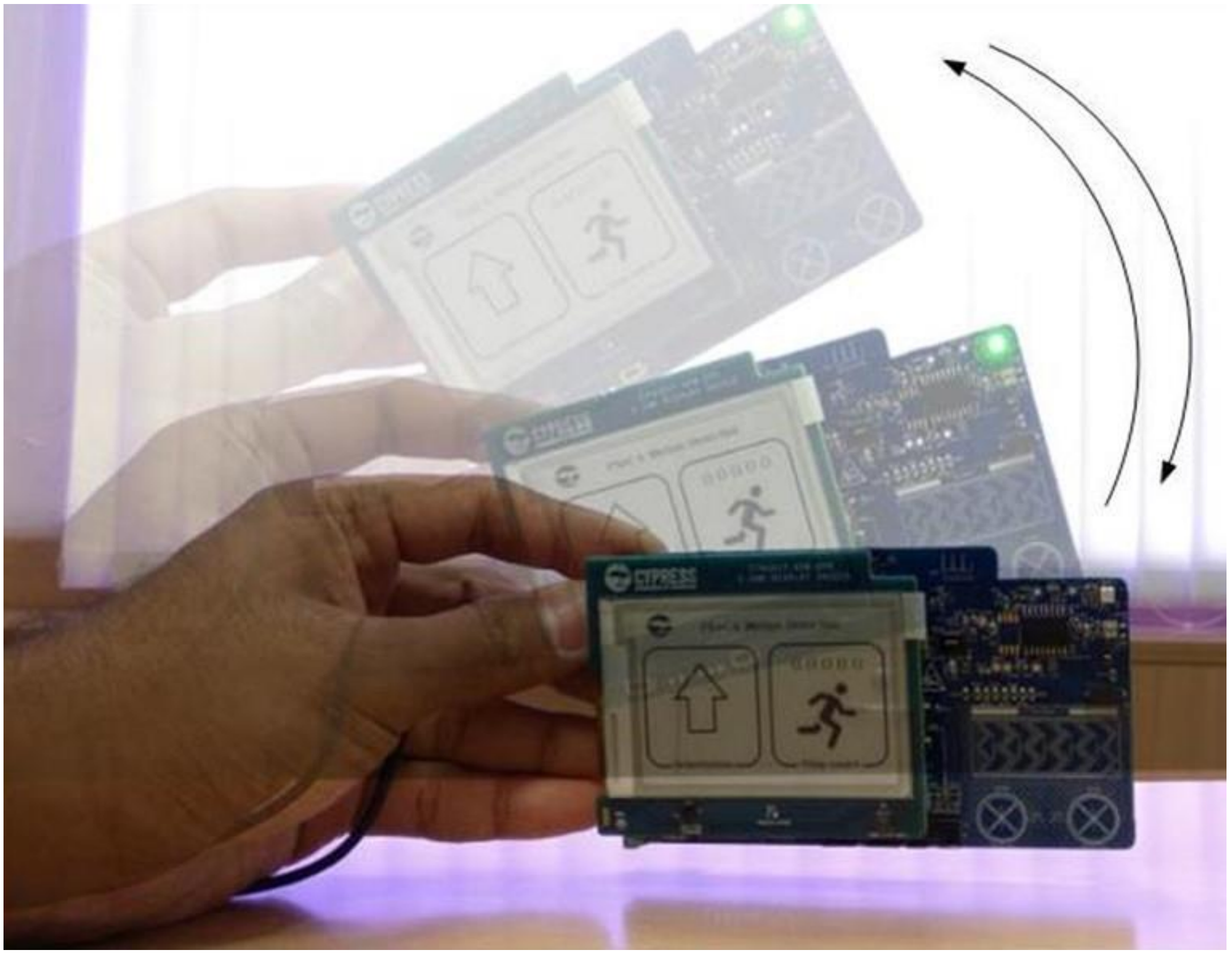
Note: If the terminal displays an error message, check the connection of the motion sensor/E-INK display shield with the Pioneer Baseboard.

5. To test the step counter, hold the kit in your hand and rock it back and forth to simulate walking motion as shown in Figure 4. Do not shake the board vigorously because this motion pattern might be rejected by the motion sensor's internal filter.

   Upon detecting a valid motion pattern, the sensor provides a step detector interrupt signal. Notice that the green LED toggles with each step detected.

   When starting to count steps from an idle state, the sensor starts incrementing the step counter internally, but waits for a few consecutive steps (typically 4) to be registered before reporting the count. Henceforth, consecutive steps are reported as they are detected. When steps are no longer detected and motion sensor is idle, this process is repeated and step counting is resumed.

Figure 4. Simulating Step Motion



6. The accelerometer sensor data is used to estimate the board's orientation. The E-INK display shows an "arrow" graphic indicating one of six orientation states as shown in Table 2. Changing the orientation of the board between portrait and landscape orientations updates the arrow graphic to keep it pointing towards the ceiling. When the board is placed so that the display is parallel to the ground (face up/face down), a 3-D arrow pointing away from or into the board is displayed.

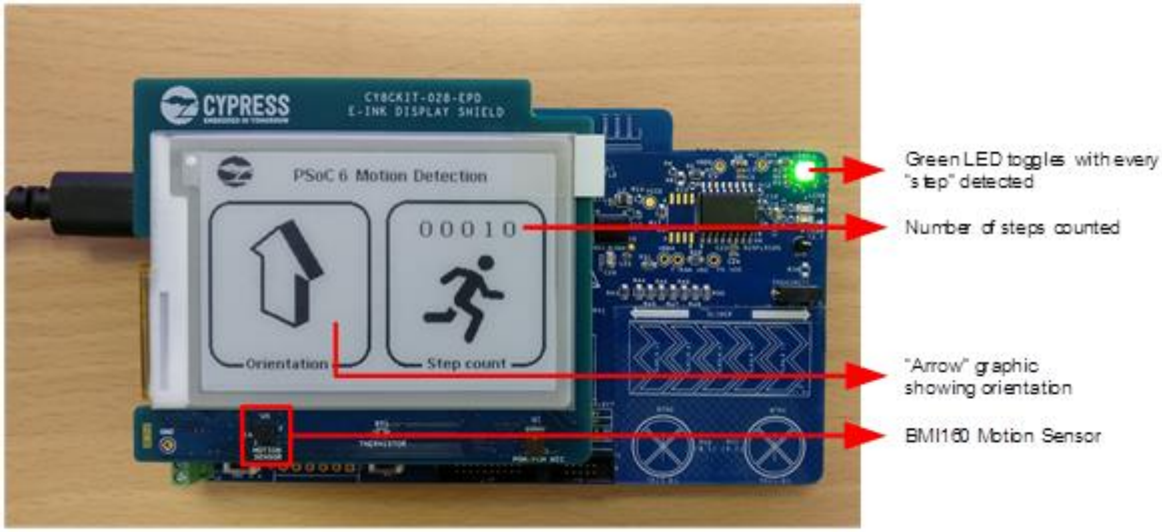
Table 2. Orientation States

 ORIENTATION_TOP_EDGE = 0 Top edge of the board points towards the ceiling	 ORIENTATION_BOTTOM_EDGE = 1 Bottom edge of the board points towards the ceiling
 ORIENTATION_LEFT_EDGE = 2 Left edge of the board points towards the ceiling	 ORIENTATION_RIGHT_EDGE = 3 Right edge of the board points towards the ceiling
 ORIENTATION_DISP_UP = 4 Display faces up (towards the sky/ceiling)	 ORIENTATION_DISP_DOWN = 5 Display faces down (towards the ground)

Design and Implementation

The E-INK Display Shield (CY8CKIT-028-EPD) contains a BMI160 motion sensor (U5) which is a low-power inertial measurement unit (IMU) providing 3-axis acceleration and 3-axis gyroscopic measurements. PSoC 6 MCU interfaces to this sensor and reads motion data, which is converted into two outputs: Orientation and Step count, and displayed on the E-INK display.

Figure 5. E-INK Display Showing Motion Sensor Outputs



Note: E-INK displays consume no power to retain the display. For more details on E-INK display, see the code example CE218133 - PSoC 6 MCU E-INK Display with CapSense.

The BMI160 motion sensor is interfaced to PSoC 6 MCU using an I^2C interface and one interrupt pin. BMI160 has a hardware-selectable I^2C slave address, depending on the logic driven on the SDO pin. On the E-INK Display Shield, the SDO pin is pulled to GND, which selects the slave address 0b1101000 (0x68).

BMI160 provides two output pins (INT1 and INT2) to which various interrupt events can be assigned. In this example, the “Step Detector” event is assigned to INT1. On the E-INK Display Shield, INT1 pin is connected to pin 2 of J3. On the Pioneer Baseboard, this connects to P13[1] of PSoC 6 MCU. See the BMI160 datasheet for more details on interrupt outputs.

The BMI160 step detector interrupt output has configurable active level and pulse width. In this example, the INT1 output is configured to provide a rising-edge signal with a pulse width of 2.5 ms. On PSoC 6 MCU, P13[1] is configured as an input pin and is internally pulled down. The interrupt is used to detect when new steps are detected; the step counts are fetched by reading a register using I^2C.

In PSoC Creator, an SCB-based I2C Component is used to implement the I2C Master interface to BMI160. The I2C Data Rate is set to 400 kbps. Configuration of the motion sensor and reading accelerometer and step count information are performed over this interface.

Step counts are polled whenever the step detector interrupt is detected. A green LED (StepDetected_LED pin / P1[1]) is toggled with every step detected.

Raw accelerometer data is continuously read and processed to compute the orientation. The orientation is represented using a set of graphic icons (arrows) on the E-INK display.

Figure 6. PSoC Creator Schematic Showing Motion Sensor Interface and Debug Outputs

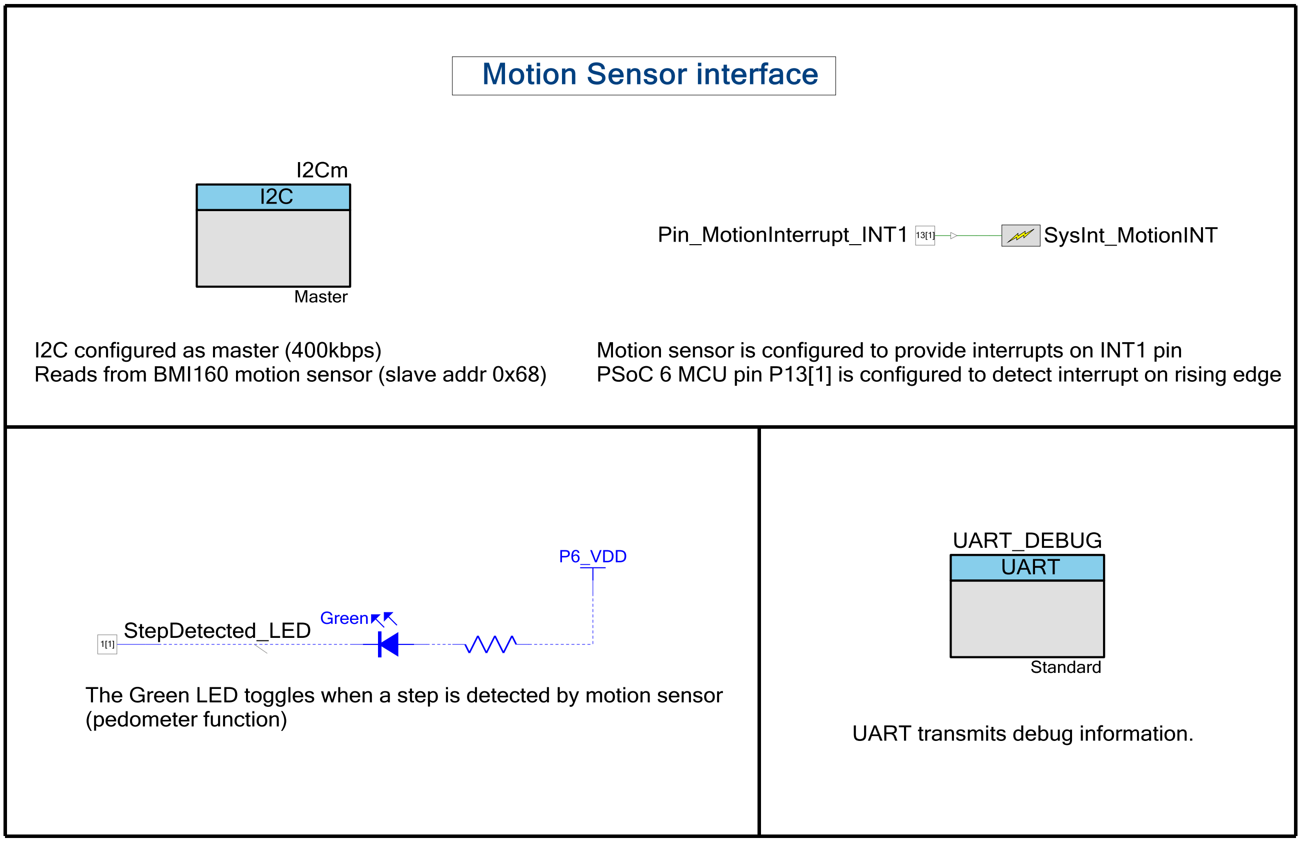
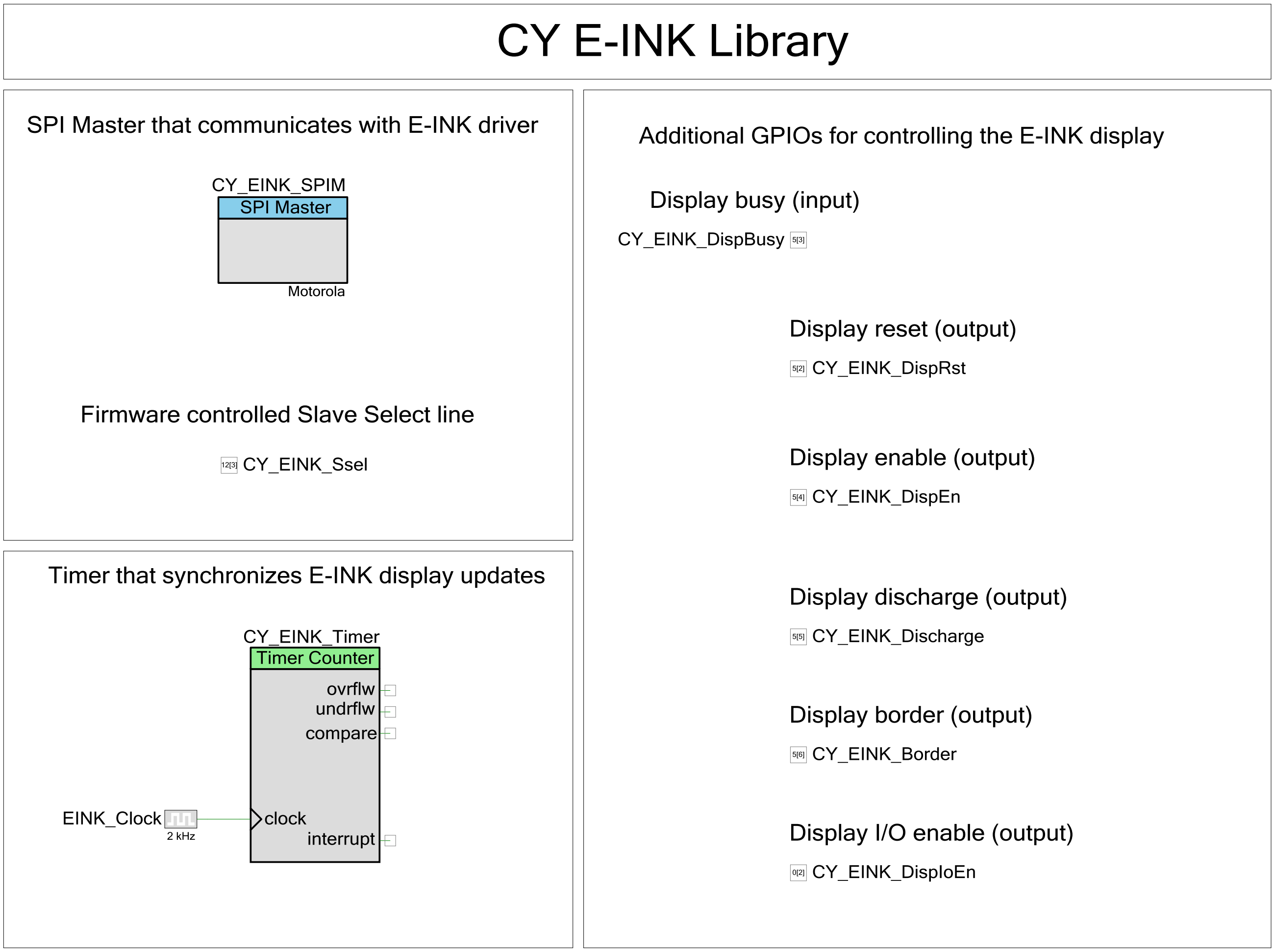


Figure 7. PSoC Creator Schematic: E-INK Display



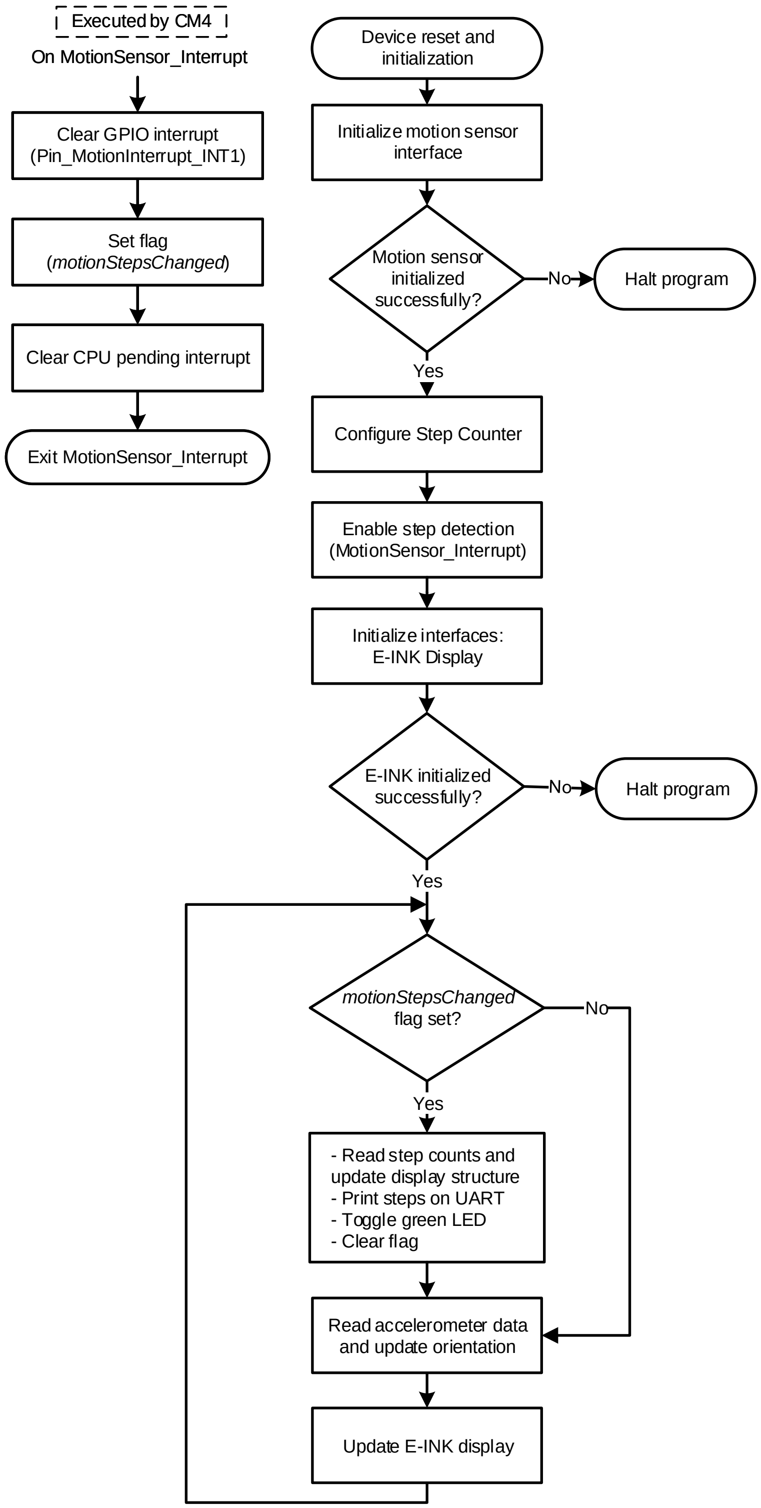
The project consists of the following files:

- ***main_cm4.c*** contains the main function for Arm Cortex®-M4, which is the entry point and execution of the firmware application. The main function calls the initializing functions, continuously processes motion sensor events, reads the data, and updates the E-INK display.
- ***main_cm0+.c*** enables CM4.
- ***bmi160.c/.h*** contains the sensor driver for BMI160 motion sensor. For the latest version of this driver please visit the GitHub repository.
- ***motion_sense.c/.h*** contains the macros and function definitions related to motion sensor application outputs. This includes the functions used to initialize and configure the motion sensor, set up interrupts, and compute motion outputs like orientation from raw accelerometer data.
- ***i2cm_support.c/.h*** contains the I²C master read and write functions.
- ***display.c/.h*** contains the functions that initialize the E-INK display and show the instructions to use this code example at startup[1].
- ***stdio_user.c/.h*** contains functions for the debug and UART functionality.

[1]**Note:** For a detailed list of files included in the E-INK Library, see the code example CE218133 PSoC 6 MCU E-INK Display with CapSense.

Figure 8 shows the firmware flow for this project.

Figure 8. Firmware Flowchart



Components and Settings

Table 3 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 3. PSoC Creator Components

<table>
<tr><th>Component</th><th>Instance Name</th><th>Purpose</th><th>Non-default Settings</th></tr>
<tr><td>I2C (SCB)</td><td>I2Cm</td><td>I²C master for communicating with the motion sensor</td><td>Mode: Master
Data Rate (kbps): 400</td></tr>
<tr><td>Interrupt</td><td>SysInt_MotionINT</td><td>Component to receive signal from Pin_MotionInterrupt_INT1</td><td>Interrupt Type: Rising-Edge Triggered</td></tr>
<tr><td>UART (SCB)</td><td>UART_DEBUG</td><td>Serial communication block for debug output on terminal</td><td></td></tr>
<tr><td>SPI (SCB)</td><td>CY_EINK_SPIM</td><td>SPI master for communication with E-INK display controller</td><td>Mode: Master
Data Rate (kbps): 16000
Oversample: 6
RX Data Width: 8
TX Data Width:8
Number of SS: 0</td></tr>
<tr><td>Timer Counter (TCPWM)</td><td>CY_EINK_Timer</td><td>Provides timing required by E-INK display controller</td><td>Clock Prescaler: Divide by 2
Run Mode: One Shot
Period: 10000
Compare 0: 5000
Compare or Capture: Compare</td></tr>
<tr><td rowspan="2">Digital Output Pin</td><td>CY_EINK_Ssel
CY_EINK_DispRst
CY_EINK_DispEn
CY_EINK_Discharge
CY_EINK_Border
CY_EINK_DispIoEn</td><td>Pins required by E-INK display controller</td><td>HW connection: Unchecked</td></tr>
<tr><td>StepDetected_LED</td><td>Visual output of step detection</td><td>Initial drive state: High (1)</td></tr>
<tr><td rowspan="2">Digital Input Pin</td><td>CY_EINK_DispBusy</td><td>Pin indicating E-INK display controller busy status</td><td></td></tr>
<tr><td>Pin_MotionInterrupt_INT1</td><td>Pin connected to the motion sensor interrupt signal</td><td>[General tab]
Drive mode: Resistive Pull Down
[Input tab]
Interrupt: Rising Edge</td></tr>
<tr><td>Clock</td><td>CY_EINK_Clock</td><td>Clock source for E-INK Timer</td><td>Frequency: 2 kHz</td></tr>
</table>

For information on the hardware resources used by a Component, see the Component datasheet.

Reusing This Example

This example is designed for the PSoC 6 BLE Pioneer Kit. To port the design to a different PSoC 6 MCU device and/or kit, change the target device using the Device Selector and update the pin assignments in the Design Wide Resources Pins settings as needed. For single-core PSoC 6 MCU devices, port the code from *main_cm4.c* to *main.c*.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU: Dual-Core CPU system Design	Describes the dual-core CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-core design
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
I2C	Supports I2C slave, master, and master-slave operation configurations using SCB
SPI	Provides an industry-standard, 4-wire master SPI interface using SCB hardware
UART	Provides asynchronous communication interface using SCB hardware
Timer Counter (TCPWM)	Supports configuration of the TCPWM hardware for Timer/Counter functionality
Clock	Supports local clock generation
Pins	Supports connection of hardware resources to physical pins
Interrupt	Provides Interrupt component settings
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
BMI160 Motion Sensor datasheet	PSoC 6 MCU: PSoC 62 Datasheet
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE220675 – PSoC 6 MCU: Motion Sensor

Document Number: 002-20675

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6005589	ARVI	02/15/2018	New code example
*A	6075093	AJYA	03/07/2018	Updated for PSoC Creator 4.2

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709