User's Guide

IoT Microcontroller Development Kit

1.0 Introduction

This microcontroller development kit is a low cost, small form factor, programmable, open source microcontroller to aid in Internet of Things applications. It is a reference design using the CC2650MODA for the processor with the same dimensions and pinout as the Adafruit Feather.

1.1 Features

The following is a list of the features on the board:

- Power LED & User Programmable LED
- Jumpers to Disable LEDs
- MCU Reset Button & User Programmable Button
- Pin Headers Matching the Form Factor of the Adafruit Feather
- 15 GPIO Pins
- JTAG Connector for Programming
- Current Measurement Jumper
- Light Sensor (Rev 2.0+)
- Bluetooth Low Energy

1.2 Board Pictures

The following figures show Rev2.0, fully assembled and both sides of an empty PCB.

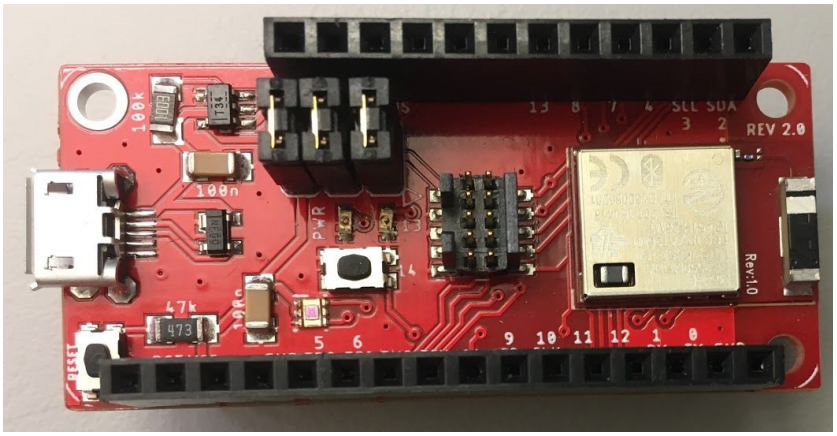

Figure 1 - Fully Assembled Rev2.0

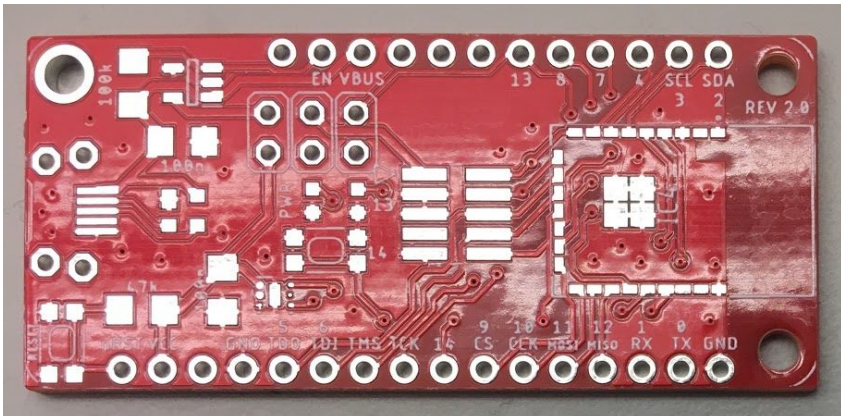

Figure 2 - Top Side of Rev2.0

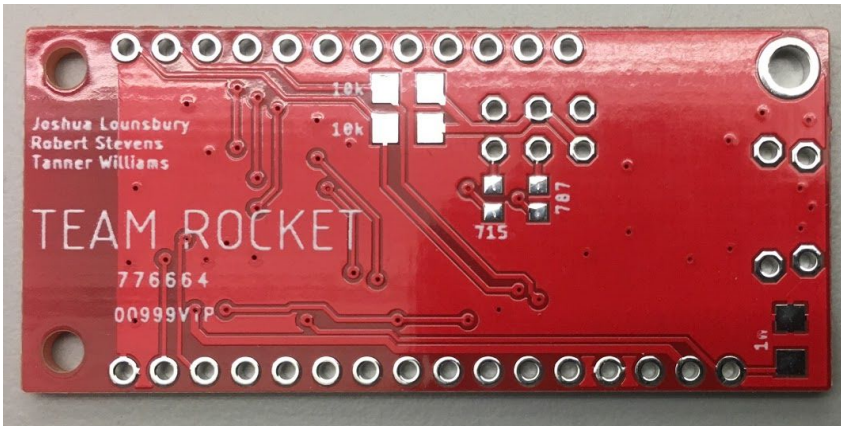

Figure 3 - Bottom Side of Rev2.0

2.0 Hardware

2.1 Bill of Materials

The full Bill of Materials for Rev2.0 can be found in the following table:

Component	Model	Qnty	Price (individual)	Price (1000+)	Package/Size
MCU	CC2650MODA	1	$11.29	$6.050	MOH (module)
GPIO LED	APHD1608LCGCK	1	$0.34	$0.111	0603
PWR LED	APTD1608LSURCK	1	$0.33	$0.117	0603
Push Button	KMR241GLFS	2	$0.52	$0.287	N/A
Ambient Light Sensor	OPT3001DNPT	1	$2.66	$1.410	DNP0006A
JTAG connector	FTSH-105-01-F-DV-K-P	1	$2.90	$1.682	N/A
JTAG Cable	1675	1	$2.95	$2.360	N/A
Micro-USB connector	2040001	1	$2.16	$1.402	N/A
Micro-USB cable	895-CUB-100-BK	1	$2.95	$2.940	N/A
FeatherWing Pins	2830	1	$1.25	$1.25	N/A
1x2 Jumpers (Current/LED)	649-54101-T05-00	3	$0.10	$0.044	N/A
CONN SHUNT 2POS	SAM8858-ND	3	$0.31	$0.145	N/A
Resistor, 100k	RC1206FR-07100KL	1	$0.10	$0.006	1206
Resistor, 47k	RCG120647K0JNEA	1	$0.10	$0.014	1206
Resistor, 10k	RC1206FR-0710KL	2	$0.10	$0.006	1206
Resistor, 787	CR0603-FX-7870ELF	1	$0.16	$0.009	0603
Resistor, 715	CRCW0603715RFKEB	1	$0.10	$0.006	0603
Capacitor, 100nF	VJ1206Y104JXBTW1BC or VJ1206Y104JXAAR	2	$0.10	$0.050	1206
Capacitor, 1nF	VJ1206Y102JXBAC	1	$0.10	$0.040	1206
TI Switching Voltage Regulators	TPS73133DBVR	1	$1.09	$0.497	SOT-23-5
TI ESD Suppressor	TPD2E001DRLR	1	$0.55	$0.218	SOT-553-5
Total			$31.70	$19.37	

Table 1 - Full BoM for Rev2.0, including prices.

2.2 Pin Map Diagram

The following figure shows the Pin Map Diagram for the custom CC2650MODA board.

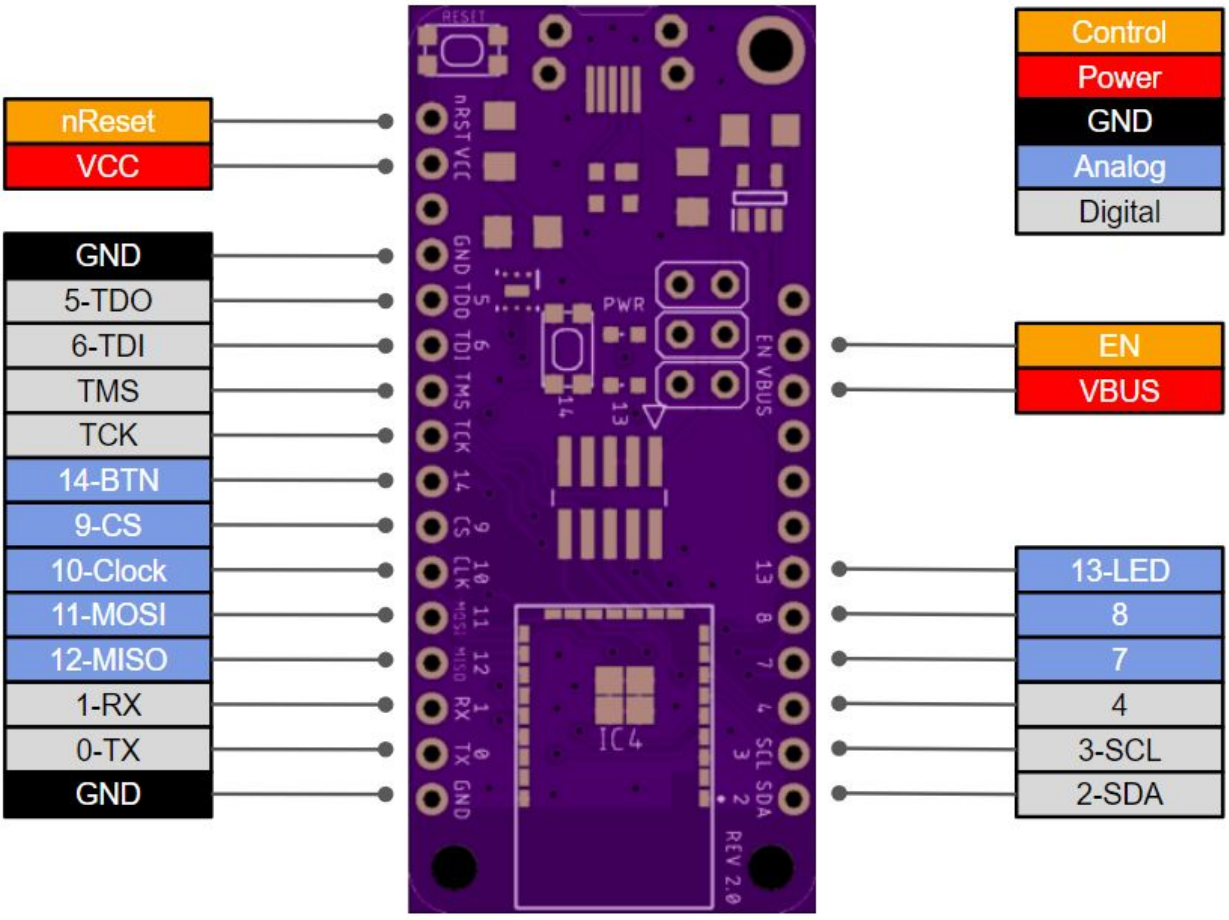

Figure 4 - Pin Map Diagram of custom CC2650MODA board Rev2.0

2.3 Block Diagram

The following figure illustrates the block diagram for Rev2.0:

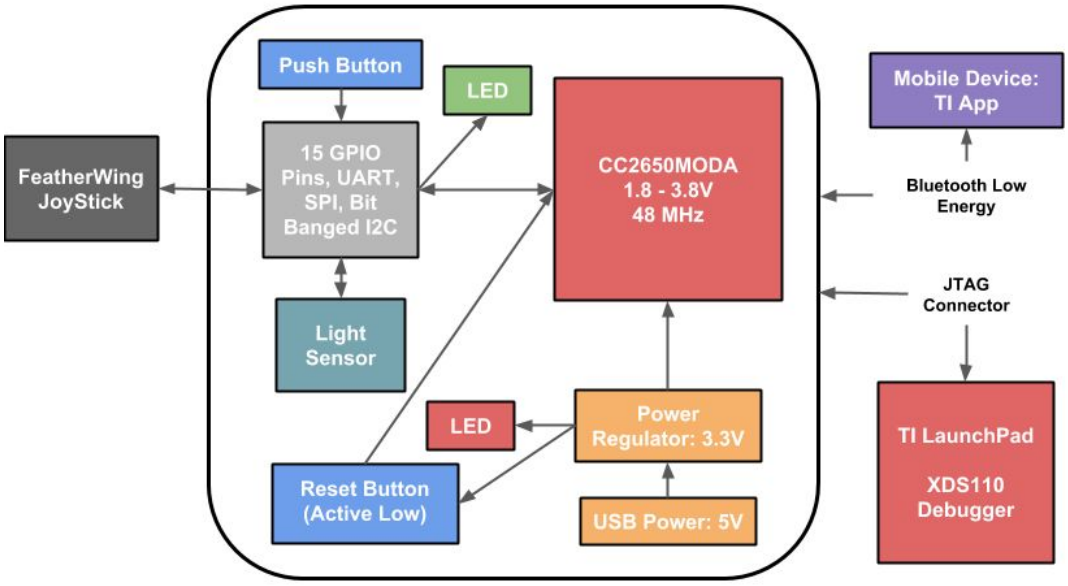

Figure 5 - Rev2.0 Block Diagram

The block diagram for just the CC2650MODA can be found in its datasheet, which is linked to in the **5.0 Resources** section.

2.4 Board Revisions

In the development of this reference design, three major board revisions were built: Rev1.0, Rev1.1, and Rev2.0. After, realizing the power regulator on all three of these revisions was not operating correctly, an unnecessary GND signal was removed from one of the power regulator pins, producing Rev1.2 and Rev2.1. These two revisions were never physically assembled, but the pinouts for Rev1.1 through Rev2.1 are all identical.

The following table lists the major board revisions and what was changed between each one:

Rev #	Notes
Rev1.0	First board with LEDs and Buttons working
Rev1.1	Moved LEDs and Buttons to better locations
Rev1.2	Fixed power regulator signals from Rev1.1
Rev2.0	Added OPT3001 light sensor from Rev1.1
Rev2.1	Fixed power regulator signals from Rev2.0

Table 2 - Differences between board revisions

Note: any software that works with Rev1.1 should work with any revision after it.

2.5 CC2650MODA MCU

The CC2650MODA contains a powerful ARM® Cortex® - M3 processor with up to a 48-MHz clock speed. It has 128KB of In-System Programmable Flash and works with 2-Pin cJTAG and JTAG Debugging. The Ultra-Low-Power Sensor Controller makes it compatible with TI's Sensor Controller Studio software. The processor also contains a 2.4-GHz RF transceiver compatible with bluetooth low energy (BLE).

This MCU was selected for the development kit because of the integrated antenna and compatibility with bluetooth low energy. Some other factors that went into this decision were the 15 GPIO pins and its compatible with the I2C protocol. All information on the CC2650MODA can be found on the datasheet which is linked to under the **5.0 Resources** section.

2.6 Power

The board can be powered several ways. Connecting a USB cable will supply 5V to the TPS73133DBVR Power Regulator, which is also accessible on the VBUS header pin. The Power Regulator can accept up to 5.5V, and will output 3.3V to the VCC header pin.

To completely bypass the power regulator and power the board with an external power source and regulator, remove the power jumper, optionally connect the EN pin to GND, and supply 1.8V to 3.8V to the VCC header pin. Connecting the EN pin on the power regulator to GND will completely disable it and enable reverse current protection.

Power can also be supplied via the JTAG connector, which connects to the VCC power signal. It is also worth noting that **external power, from JTAG or another source, should NOT be supplied to the board if it is also connected to USB power.**

2.7 Compatibility with Adafruit Products

The board's pin headers are designed to match the form factor of the Adafruit feather. Depending on how many GPIO pins are required, certain Adafruit featherwing products are stackable and usable with this board. Before selecting a FeatherWing, check how many GPIO pins would be required to operate that device and check to make sure the protocol used for communication is supported by the CC2650MODA MCU. Additionally, make sure that the FeatherWing will be properly powered when stacked on the board. The TPS73133DBVR on board power regulator is capable of supplying up to 150 mA to the CC2650MODA and its peripherals.

3.0 Device Setup

3.1 Required Hardware and Software

Hardware

- 1 Custom Board
- 1 CC2650 Launchpad with XDS110 Debugger or another debugger
- 10-pin JTAG debug cable or other wires to connect the appropriate JTAG and power pins
- Micro USB cable

Software

- Code Composer Studio
- Sensor Controller Studio
- TI-RTOS for CC2650 (tirtos_cc13xx_cc26xx-2_21_00_06 or newer)
- Bluetooth low energy software stack (BLE-STACK-2-2-1 or newer)

3.2 Hardware Setup

In **Figure 6** (below), the custom board is shown correctly connected to the CC2650 LaunchPad. Use the following instructions to set up the hardware:

1. Connect the 10-pin JTAG cable to the JTAG connector on the CC2650MODA custom board and to the XDS110 output pins on the CC2650 LaunchPad
2. Ensure that the necessary jumpers are removed to isolate the XDS110 from the onboard CC2650 of the LaunchPad
3. Verify that the custom board has power as it will not be provided by all TI LaunchPads
4. Plug the USB cable into the LaunchPad and computer.

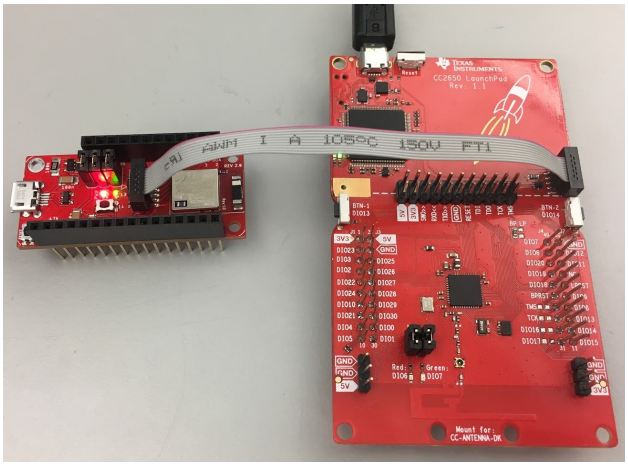

Figure 6 - Custom Board connected to CC2650 LaunchPad via JTAG ribbon cable

4.0 Software

This section details how to run standalone applications on the custom board that incorporates the CC2650MODA module. To check the functionality of a bluetooth low energy connection between a smartphone and the Custom Board a user can run the BLE simple_peripheral application (C:\ti\simplelink\ble_sdk_2_02_01_18\examples \cc2650lp\simple_peripheral\ccs) in CCS just by editing the necessary board files discussed in **Section 4.3**. Due to the fact that all device firmware resides and executes internally on the CC2650 wireless MCU, other TI BLE sample applications can run on the Custom Board if the project is configured to account for the correct mapping of external peripheral I/O connections.

4.1 Downloading and Running TI BLE Sample Applications

Refer to the CC26X0 SimpleLink™ Bluetooth® low energy Software Stack 2.2.X for information on building and downloading the sample applications using Code Composer Studio. A link to this document can be found in section **5.0 Resources**.

4.2 Running Provided Example Applications on Custom Board

To program the Custom Board with the provided software examples in **Section 4.4**, download and unzip the desired .zip file from GitHub, and then import its contents into Code Composer Studio. The default location for project folders for Code Composer Studio is: *C:\Users\<User Name>\workspace_v7*

Next, follow the instructions in section **3.2 Hardware Setup** to correctly connect the Custom Board and XDS110 debugger to your computer. Once the device is connected, build the project, press the *Debug* button, and the press the *Play* button. If the program prints anything to the console, it should now be visible and the program is running. The board can now be disconnected from the CC2650 launchpad and it will still run the program after it is reset and has power.

To import **Example 4.4.5 I2C Light Sensor**, unzip the contents downloaded from GitHub to the default location for Sensor Controller Studio projects, which is: *C:\Users\<User Name>\Documents\Texas Instruments\Sensor Controller Studio\projects*

In Sensor Controller Studio, under the *Task Testing* tab, click the *Connect* button, then the *Run Initialization Code* button, and finally the *Run Task* button. The program should now be running and the data from the light sensor should be displayed on the *Graph* tab. If the Custom Board is unplugged from the computer after being programmed with SCS, it will **NOT** retain the program instructions to read data from the light sensor. For more information refer to the Sensor Controller Studio Guide in section **5.0 Resources**.

4.3 Required Board-File Changes

To change the GPIO pin mapping from existing board files, there are three main files that need to be changed: board.h, CC2650DK_5XD.h, CC2650DK_5XD.c. The last two files may have different names depending on the project they were originally imported from, but they should be similar to <processor name>.h and .c.

Examples of these files can be found in the attached software examples.

In board.h, there should be some `#define` statements for the button, LED, UART, etc. The important lines for changing the GPIO mapping are:

```
/* These #defines allow us to reuse TI-RTOS across other device
families */
#define     Board_LED0              Board_DK_LED1

#define     Board_BUTTON0           Board_KEY_SELECT

#define     Board_UART0             Board_UART
```

In CC2650DK_5XD.h, the following lines need to be changed accordingly to map the GPIO lines correctly:

```
/* Leds */
#define Board_LED_ON              1
#define Board_LED_OFF             0
#define Board_DK_LED1             IOID_13

/* Button Board */
#define Board_KEY_SELECT          IOID_14

/* UART Board */
#define Board_UART_RX             IOID_1
#define Board_UART_TX             IOID_0

/* SPI Board */
#define Board_SPI0_MISO           IOID_12
#define Board_SPI0_MOSI           IOID_11
#define Board_SPI0_CLK            IOID_10
#define Board_SPI0_CSN            IOID_9
```

And in CC2650DK_5XD.c, the following lines would be used to initialize the GPIO lines:

```
const PIN_Config BoardGpioInitTable[] = {
  Board_LED0 | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL|
     PIN_DRVSTR_MAX,
  Board_KEY_SELECT | PIN_INPUT_EN | PIN_PULLUP | PIN_HYSTERESIS,
  Board_UART_TX | PIN_GPIO_OUTPUT_EN | PIN_GPIO_HIGH |
     PIN_PUSHPULL,
  PIN_TERMINATE
};
```

Note that in the Resource Explorer in Code Composer Studio, under:

Software\TI-RTOS for CC2650\Development Tools\CC2650 Development Kit (5XD)

any of the existing software examples can be ported for use on this board if the GPIO pins are correctly remapped.

4.4 Software Examples

The following examples can be found on the team GitHub page.

4.4.1 Blink LED

This example simply blinks the green LED and prints when the LED is turned on or off to the CCS console.

Note: The LED blinks at a faster rate when running through the debugger compared to when the board is power reset or the reset button is reset.

4.4.2 Button LED

This example toggles the green LED when the programmable button is pressed and prints Hello World to the CCS console.

4.4.3 GPIO Test

This example is an extension of the Blink LED example. The program will cycle through each pin (IOID_X), toggling it on then off and moving to IOID_X+1. The name of the pin being toggled is also printed to the CCS console.

4.4.4 UART Echo

This example was ported from the existing UART Echo example in the Resource Explorer. The green LED will be turned on after the task initialization is completed. The Rx and Tx GPIO pins will need to be connected to the RXD and TXD pins on the XDS110 Debugger for the echoed characters to be seen on the computer.

Use device manager in Windows to determine which COM port the board is connected to. In a serial monitoring program, such as PuTTY, select the appropriate COM port to run with a baud rate of 9600, 8 data bits, 1 stop bit, no parity or flow control. **Figure 7** shows an example of the PuTTY configuration.

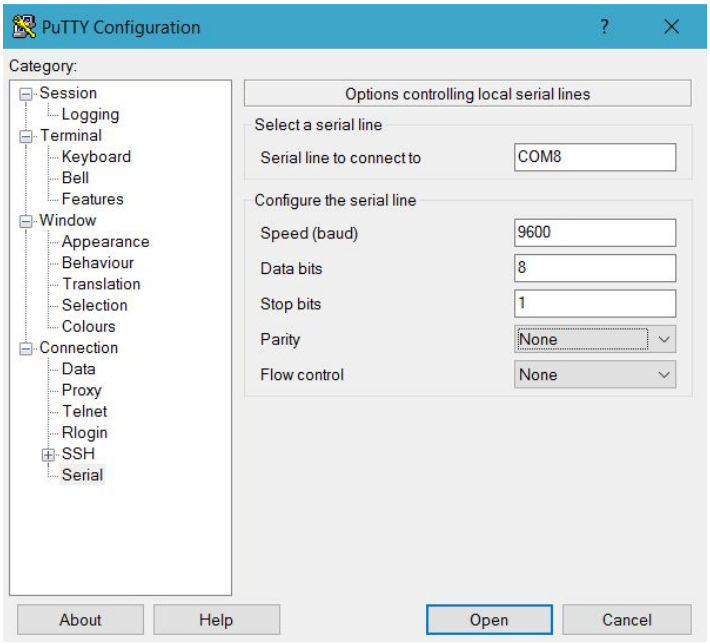

Figure 7 - PuTTY configuration for UART Echo example

4.4.5 I2C Light Sensor (Sensor Controller Studio)

This example must be run in Sensor Controller Studio. The digital value read from the light sensor is displayed on the graph on the *Task Testing* tab. From the existing I2C light sensor example for the SensorTag, at line 28 the following lines were added to the Event Handler Code to toggle the LED at certain light levels:

```
gpioClearOutput(1);

if (value < 200){
     gpioSetOutput(1);
}
```

Below in **Figure 8**, the correct IO mapping for the project is displayed.

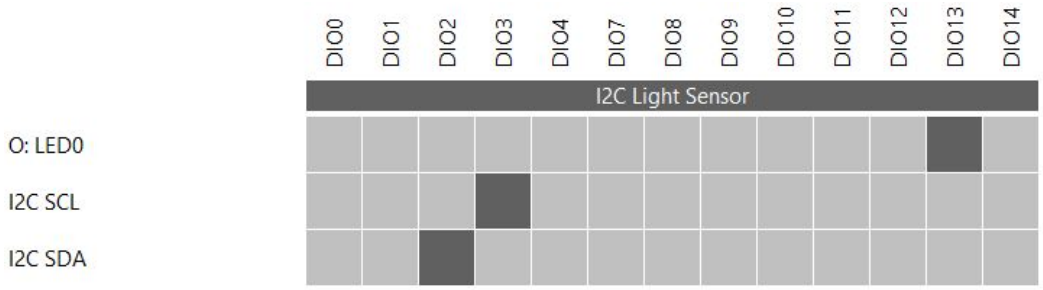

Figure 8 - Sensor Controller Studio I2C Light Sensor Example IO Mapping

5.0 Resources

5.1 Hardware

Eagle Files and Software Examples (GitHub):
https://github.com/TannerAWilliams/Senior-Design-Project

MCU CC2650MODA Datasheet:
http://www.ti.com/lit/ds/symlink/cc2650moda.pdf

Power Regulator TPS73133DBVR Datasheet:
http://www.ti.com/lit/ds/symlink/tps731.pdf

5.2 Software

CC26X0 SimpleLink Bluetooth low energy Software Stack 2.2.X Developer's Guide:
http://www.ti.com/lit/ug/swru393e/swru393e.pdf

Sensor Controller Studio Guide:
http://www.ti.com/lit/ml/swru439e/swru439e.pdf

TI-RTOS 2.20 User's Guide:
http://www.ti.com/lit/ug/spruhd4m/spruhd4m.pdf

Using TI Bluetooth low energy Module (CC2650MODA) as Single-Chip Wireless MCU:
http://www.ti.com/lit/an/swra534a/swra534a.pdf