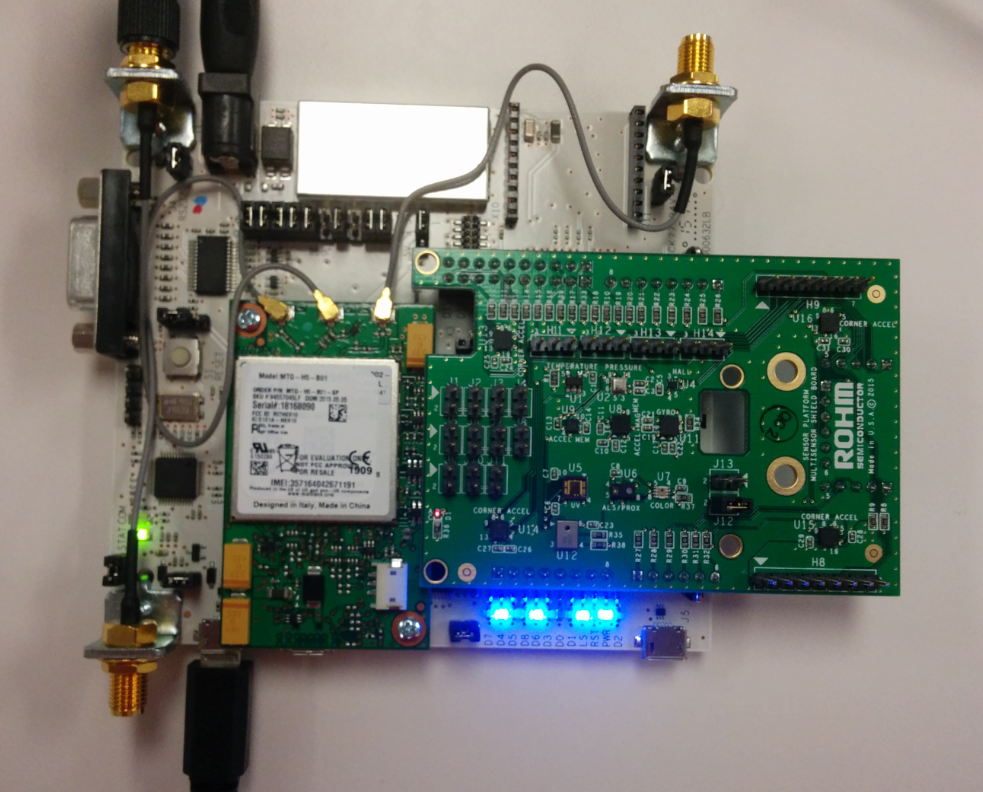
Connecting the Multi-Sensor Shield to the MultiTech Dragonfly



Above: ROHM Sensor Platform Shield directly connected to the MultiTech DragonFly EVK

17 November 2015, Revision A

Table of Contents

[Copyright and License 3](#_Toc430956979)

[Revision History 3](#_Toc430956980)

[Introduction 4](#_Toc430956981)

[Getting Started 4](#_Toc430956982)

[Hardware Connection for ROHM BDE0600G Temp Sensor to the Dragonfly Platform 5](#_Toc430956983)

[Software Explanation for ROHM BDE0600G Temp Sensor to the Dragonfly 5](#_Toc430956984)

[Hardware Connection for LAPIS ML8511 UV Sensor to the Dragonfly Platform 7](#_Toc430956985)

[Software Explanation for LAPIS ML8511 UV Sensor to the Dragonfly Platform 7](#_Toc430956984)

[Hardware Connection for ROHM BU52014 Hall Sensor to the Dragonfly Platform 8](#_Toc430956986)

[Software Explanation for ROHM BU52014 Hall Sensor to the Dragonfly Platform 8](#_Toc430956987)

[Hardware Connection for Kionix KMX62 Accel+Mag Sensor to the Dragonfly Platform 9](#_Toc430956988)

[Software Explanation for Kionix KMX62 Accel+Mag Sensor to the Dragonfly Platform 9](#_Toc430956989)

[Hardware Connection for ROHM BM1383GLV Barometric Pressure Sensor to the Dragonfly Platform 11](#_Toc430956990)

[Software Explanation for ROHM BM1383GLV Barometric Pressure Sensor to the Dragonfly Platform 11](#_Toc430956991)

[Hardware Connection for ROHM RPR-0521 3-in-1 Ambient Light Sensor, Proximity Sensor, and IR LED Combo Package for the Dragonfly Platform 13](#_Toc430956992)

[Software Explanation for ROHM RPR-0521 3-in-1 Ambient Light Sensor, Proximity Sensor, and IR LED Combo Package to the Dragonfly Platform 13](#_Toc430956993)

[Hardware Connection for ROHM BH1745NUC Color Sensor for the Dragonfly Platform 15](#_Toc430956994)

[Software Explanation for ROHM BH1745NUC Color Sensor to the Dragonfly Platform 15](#_Toc430956995)

[Hardware Connection for Kionix KX122 Accelerometer Sensor for the Dragonfly Platform 17](#_Toc430956996)

[Software Explanation for Kionix KX122 Accelerometer Sensor to the Dragonfly Platform 17](#_Toc430956997)

[Hardware Connection for Kionix KXG03 Gyroscopic Sensor for the Dragonfly Platform 19](#_Toc430956998)

[Software Explanation for Kionix KXG03 Gyroscopic Sensor to the Dragonfly Platform 19](#_Toc430956999)

# Copyright and License

[Need to add MultiTech References Here]

Please note that all references to ROHM’s Multi-Sensor Shield are also shared under open source guidelines under the GNU General Public License, Version 3. Details can be found at the following link: <https://github.com/ROHMUSDC/ROHM_SensorPlatform_Multi-Sensor-Shield>.

# Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Description** | **Revision ID** |
| 25 September 2015 | First Draft | A |
| 17 November 2015 | Added Additional Sensor Information | A |
|  |  |  |

# Introduction

The following document was written to provide a brief connection guide and starting point for using ROHM’s Multi-sensor shield with the MultiTech Dragonfly Development Kit. This guide assumes that the user has basic functional knowledge of both the ROHM Multi-Sensor Shield and the Dragonfly Development Platform itself. If this is not correct, please see the following links for other guides and information on these products.

ROHM’s Multi-Sensor Shield GitHub Repository Page: <https://github.com/ROHMUSDC/ROHM_SensorPlatform_Multi-Sensor-Shield>

MultiTech DragonFly MBED Platform Page: <http://www.multitech.com/brands/multiconnect-dragonfly>

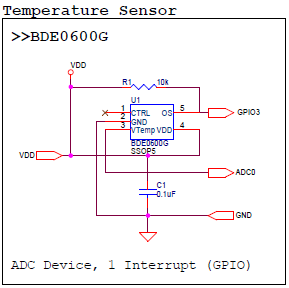
Please note that the MultiTech DragonFly Development Kit is a microcontroller platform; thus, this document will explain and show examples of how to connect to and convert values for the ROHM Sensor Kit Sensors. This includes explanations on the following sensors:

* ROHM BDE0600G – Analog Temperature Sensor
* LAPIS ML8511 – Analog UV Sensor
* ROHM BU52014HFV – Hall Switch Sensor
* KIONIX KMX62 – Digital Accelerometer and Magnetometer
* ROHM BM1383GLV – Digital Barometric Pressure Sensor
* ROHM RPR-0521 – Digital Ambient Light Sensor and Proximity Sensor
* ROHM BH1745 – Digital Color Sensor
* KIONIX KX122 – Digital Accelerometer
* KIONIX KXG03 – Digital Gyroscopic Sensor

# Getting Started

1. Initial Setup
   1. ROHM Multi-Sensor Shield Board
      1. Connect ROHM’s Multi-Sensor Shield board directly to the Dragonfly dev kit board using the standard Shield Headers on the MTUDK2-ST
   2. The following are required for using the MultiTech DragonFly application board
      1. Dragonfly Module Board and MTUDK2-ST Shield Board
      2. PC with a Serial COM Port Terminal Application (TeraTerm, Terminal, HyperTerm, etc.)
      3. MBED account for online compiler access
      4. USB Cable to program
      5. Import the example code from the following repository on the MBED site into your online MBED complier
         1. <https://developer.mbed.org/teams/ROHMUSDC/code/MultiTech_DragonFly_ROHMSensorShield_Sam/>

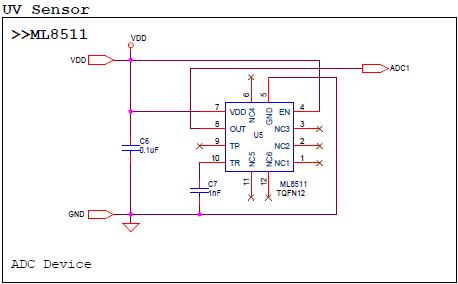
## Hardware Connection for ROHM BDE0600G Temp Sensor to the Dragonfly Platform

* ROHM BDE0600G – Analog Temperature Sensor
  + 
* As seen in the schematic above, this device connects 4 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + ADC0 – ADC Output for Temperature Readings, Connected to pin A4 on MTUDK2 Board
  + GPIO0 – This pin is used to trigger the thermostat output function. This pin is not used in the example application.

## Software Explanation for ROHM BDE0600G Temp Sensor to the Dragonfly

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***AnalogTemp***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Begin Loop()
    1. Read back the Analog Sensor Output from Pin PC\_4 (Header Reference A2)
    2. Convert value to V, then to Temperature reading
       - Known Values
         * Temperature Sensitivity = -10.68mV/degC
         * Temperature Sensitivity = -0.01068V/degC
         * Temp Known Point = 1.753V @ 30 degC
       - Calculation
         * ADC\_Voltage = (sensorValue / 65535) \* 3.3V
         * ADC\_Voltage = sensorValue \* (3.3/65535)
         * ADC\_Voltage = sensorValue \* 0.000050354
         * Temperature (in degC) = (ADC\_Voltage - 1.753)/(-0.01068) + 30
    3. Format Serial Output and Return Information

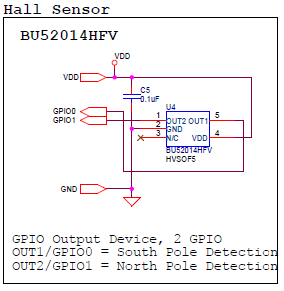
## Hardware Connection for LAPIS ML8511 UV Sensor to the Dragonfly Platform

* LAPIS ML8511 – Analog UV Sensor
  + 
* As seen in the schematic above, this device connects 3 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + A4 – ADC Output for UV Sensor Output

**Software Explanation for LAPIS ML8511 UV Sensor to the Dragonfly Platform**

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***AnalogUV***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Begin Loop()
    1. Read back the Analog Sensor Output from Pin PC\_1 (Reference Shield Pin A4)
    2. Convert value to V, then to UV Intensity
       - Known Values
         * UV Sensitivity = 0.129 V/(mW/cm2)
         * UV Known Point = 2.2V @ 10mW/cm2
       - Calculation
         * ADC\_Voltage = (sensorValue / 65535) \* 3.3V
         * ADC\_Voltage = sensorValue \* (3.3V/65535)
         * ADC\_Voltage = sensorValue \* 0.000050354
         * UV Intensity (in mW/cm2)= (ADC\_Voltage - 2.2)/(0.129) + 10
    3. Format Serial Output and Return Information

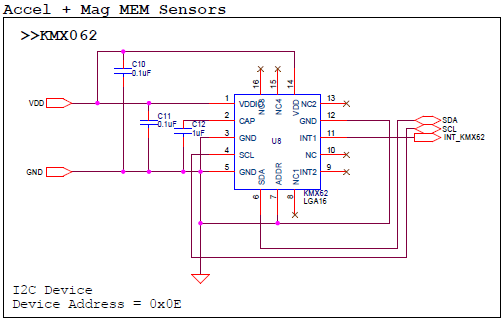
## Hardware Connection for ROHM BU52014 Hall Sensor to the Dragonfly Platform

* ROHM BU52011HFV – Hall Switch Sensor
  + 
* As seen in the schematic above, this device connects 4 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + GPIO0, GPIO1 – Indicates Hall Switch Output
    - GPIO0: Indicates South Pole Detection, Connected to pin D10
    - GPIO1: Indicates North Pole Detection, Connected to pin D11

## Software Explanation for ROHM BU52014 Hall Sensor to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***HallSensor***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Begin setup()
    1. Setup Pins PC\_8 (Shield D10) and PB\_5 (Shield D11) as Input pins
  + Begin loop()
    1. Perform digital reads on pins PC\_8 and PB\_5
       - Output will be either 1 or 0 and will indicate presence of north or south magnetic fields
    2. Format Serial Output and Return Information

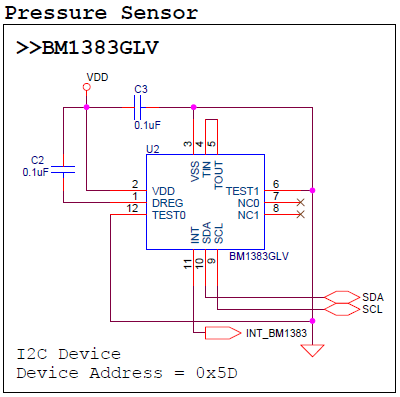
## Hardware Connection for Kionix KMX62 Accel+Mag Sensor to the Dragonfly Platform

* Kionix KMX62 – Digital Accelerometer and Magnetometer
  + 
* As seen in the schematic above, this device connects 6 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + SDA/SCL – I2C Connection, connected to D14 and D15
  + INT\_KMX62 - This pin is used monitor the interrupt pins on the KMX61. This pin is not used in the example application.

## Software Explanation for Kionix KMX62 Accel+Mag Sensor to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***KMX62***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Initialization Section
    1. Configure the Accel/Mag Sensor once by performing the following writes
       - 1. CNTL2 Register (0x3A), write 0x5F (4g Sensor Resolution, Max RES, Enable Temp/Mag/Accel)
         * Note: Please see the KMX62 Datasheet for additional information on these registers
  + Main while loop Section
    1. Read back the Accelerometer output by reading 6 Bytes starting from address 0x0A. [0][1]...[5]
    2. Format Each of the X, Y, and Z axis acceleration
       - *Note:* to save some time with conversions, please note that these outputs are meant to be unsigned 14bit values. Thus, to make it easier to convert we recommend the following…
         * Xout = (float)(([1]<<8) | ([0]))/4
         * Yout = (float)(([3]<<8) | ([2]))/4
         * Zout = (float)(([5]<<8) | ([4]]))/4
    3. Convert Returned Value to G
       - Axis\_ValueInG = MEMS\_Accel\_axis / 2048
    4. Read back the Magnetometer output by reading 6 Bytes starting from address 0x10. [0][1]...[5]
    5. Format Each of the X, Y, and Z axis mag data
       - *Note:* to save some time with conversions, please note that these outputs are meant to be unsigned 14bit values. Thus, to make it easier to convert we recommend the following…
         * Xout = (float)(([1]<<8) | ([0]))/4
         * Yout = (float)(([3]<<8) | ([2]))/4
         * Zout = (float)(([5]<<8) | ([4]]))/4
    6. Convert Returned Value to uT
       - Axis\_ValueInuT = MEMS\_Mag\_axis \* 0.146
    7. Format Serial Output and Return Information

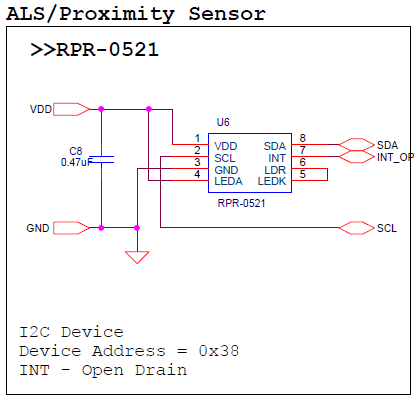
## Hardware Connection for ROHM BM1383GLV Barometric Pressure Sensor to the Dragonfly Platform

* ROHM BM1383GLV Barometric Pressure Sensor
  + 
* As seen in the schematic above, this device connects 6 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + SDA/SCL – I2C Connection, connected to D14 and D15
  + INT\_BM1383 - This pin is used monitor the interrupt pins on the BM1383. This pin is not used in the example application.

## Software Explanation for ROHM BM1383GLV Barometric Pressure Sensor to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***Pressure***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Initialization Section
    1. Configure the Pressure Sensor once by performing the following writes:
       - 1. PWR\_DOWN Register (0x12), write (0x01)
       - 2. SLEEP Register (0x13), write (0x01)
       - 3. Mode Control Register (0x14), write (0xC4)
       - Note: Please see the BM1383 datasheet for additional information on these registers
  + Main while loop section
    1. Read back the pressure and temperature output by reading 5 Bytes starting from address 0x1A. [0][1]...[4]
    2. Format raw temperature and pressure
       - Raw ADC Temp = ([0]<<8) | ([1])
       - Raw ADC Pressure (Integer Portion) = ([2]<<3) | ([3]>>5)
       - Raw ADC Pressure (Decimal Portion) = (([3] & 0x1f) << 6 | (([4] >> 2)));
    3. Convert Returned Raw Value to Real Values
       - Temp in degC = Raw ADC Temp / 32
       - Pressure in hPa =
         * (Raw ADC Pressure(int)) + (Raw ADC Pressure(dec)\*2-11)
    4. Format Serial Output and Return Information

## Hardware Connection for ROHM RPR-0521 3-in-1 Ambient Light Sensor, Proximity Sensor, and IR LED Combo Package for the Dragonfly Platform

* ROHM RPR-0521 ALS/PROX Sensor
  + 
* As seen in the schematic above, this device connects 6 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + SDA/SCL – I2C Connection, connected to D14 and D15
  + INT\_Optical - This pin is used monitor the interrupt pins on the RPR-0521. This pin is not used in the example application.

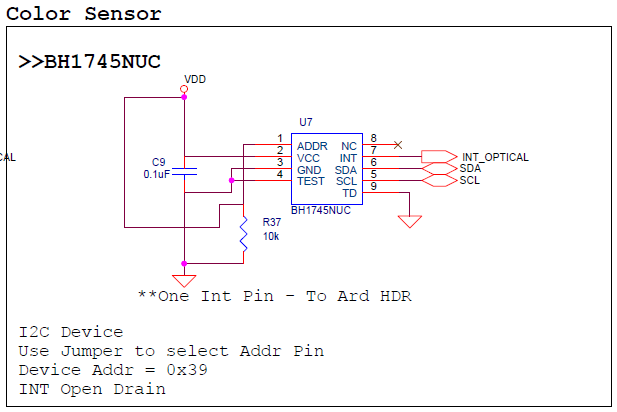
## Software Explanation for ROHM RPR-0521 3-in-1 Ambient Light Sensor, Proximity Sensor, and IR LED Combo Package to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***RPR0521***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Initialization Section
    1. Configure the Ambient Light Sensor and Proximity Sensor once by performing the following writes:
       - 1. Mode Control Register (0x41), write (0xC6)
       - 2. ALS\_PS\_Control Register (0x42), write (0x03)
       - 3. Persist Register (0x43), write (0x20)
       - Note: Please see the RPR-0521 datasheet for additional information on these registers
  + Main while loop section
    1. Read back the ALS output by reading 6 Bytes starting from address 0x44. [0][1]...[5]
    2. Format Raw ambient Light and proximity
       - RAW PROX ADC = ([1]<<8) | ([0])
       - RAW ALS DATA0 = ([3]<<8) | ([2])
       - RAW ALS DATA1 = ([5]<<8) | ([4])
    3. Convert Raw Returned Value to Real Values
       - ALS can be returned by using the following code block to return ALS value in lx (RPR0521\_ALS\_OUT)



* + 1. Format Serial Output and Return Information

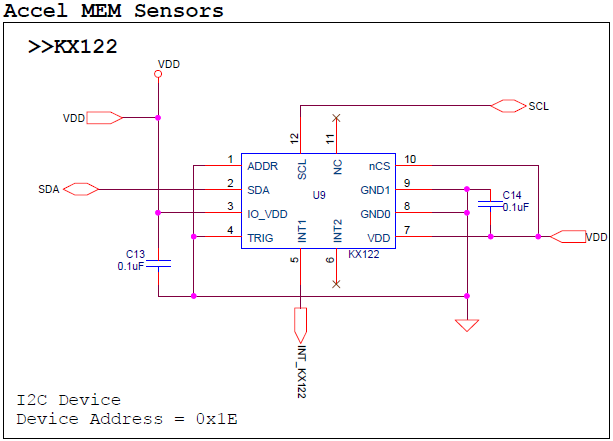
## Hardware Connection for ROHM BH1745NUC Color Sensor for the Dragonfly Platform

* ROHM BH1745NUC Color Sensor
  + 
* As seen in the schematic above, this device connects 6 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + SDA/SCL – I2C Connection, connected to D14 and D15
  + INT\_Optical - This pin is used monitor the interrupt pins on the RPR-0521. This pin is not used in the example application.

## Software Explanation for ROHM BH1745NUC Color Sensor to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***COLOR***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Initialization Section
    1. Configure the Color Sensor once by performing the following writes:
       - 1. Persistence Register (0x61), write (0x03)
       - 2. Mode Control 1 Register (0x41), write (0x00)
       - 3. Mode Control 2 Register (0x42), write (0x92)
       - 4. Mode Control 3 Register (0x43), write (0x02)
       - Note: Please see the color sensor datasheet for additional information on these registers
  + Main while loop section
    1. Read back the color output by reading 6 Bytes starting from address 0x50. [0][1]...[5]
    2. Format Raw color
       - RED ADC = ([1]<<8) | ([0])
       - GREEN ADC = ([3]<<8) | ([2])
       - BLUE ADC = ([5]<<8) | ([4])
    3. Convert Raw Returned Value to Real Values
       - Coming Soon!
    4. Format Serial Output and Return Information

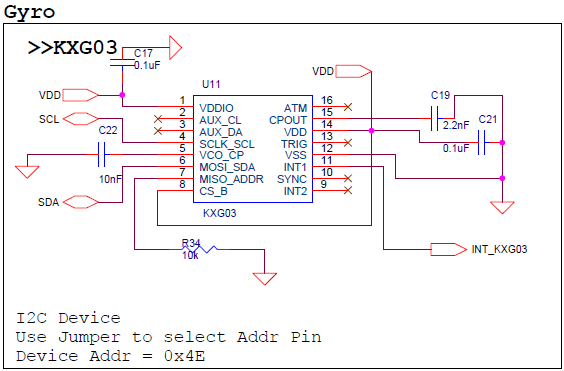
## Hardware Connection for Kionix KX122 Accelerometer Sensor for the Dragonfly Platform

* Kionix KX122 Accelerometer Sensor
  + 
* As seen in the schematic above, this device connects 6 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + SDA/SCL – I2C Connection, connected to D14 and D15
  + INT\_KX122 - This pin is used monitor the interrupt pins on the KX122. This pin is not used in the example application.

## Software Explanation for Kionix KX122 Accelerometer Sensor to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***KX122***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Initialization Section
    1. Configure the Accelerometer Sensor once by performing the following writes:
       - 1. CNTL1 Register (0x18), write (0x41)
       - 2. ODCNTL Register (0x1B), write (0x02)
       - 3. CNTL3 Register (0x1A), write (0xD8)
       - 4. TILT\_TIMER Register (0x22), write (0x01)
       - 5. CNTL1 Register (0x18), write (0xC1) (Enable bit to ON)
       - Note: Please see the KX122 datasheet for additional information on these registers
  + Main while loop section
    1. Read back the Accelerometer output by reading 6 Bytes starting from address 0x06. [0][1]...[5]
    2. Format Raw acceleration
       - ACCEL X Axis Raw = ([1]<<8) | ([0])
       - ACCEL Y Axis Raw = ([3]<<8) | ([2])
       - ACCEL Z Axis Raw = ([5]<<8) | ([4])
    3. Convert Raw Returned Value to g
       - ACCEL X in g = ACCEL X Axis Raw / 16384
       - ACCEL X in g = ACCEL Y Axis Raw / 16384
       - ACCEL X in g = ACCEL Z Axis Raw / 16384
    4. Format Serial Output and Return Information

## Hardware Connection for Kionix KXG03 Gyroscopic Sensor for the Dragonfly Platform

* Kionix KXG03 Gyroscopic Sensor
  + 
* As seen in the schematic above, this device connects 6 pins.
  + VDD – Connect to 3.3V output pin on power connector
  + GND – Connect to GND pin on the power connector
  + SDA/SCL – I2C Connection, connected to D14 and D15
  + INT\_KXG03 - This pin is used monitor the interrupt pins on the KXG03. This pin is not used in the example application.

## Software Explanation for Kionix KXG03 Gyroscopic Sensor to the Dragonfly Platform

* Code Segments pertaining to this sensor can be found by defining and seeing code within the “***KXG03***” #ifDef statements
* Pseudo-Code Explanation
  + Define Relevant Variables
  + Initialization Section
    1. STDBY Register (0x43), write 0x00.

Note: Please see the KXG03 datasheet for additional information on

these registers.

* + Begin loop()
    1. Read back the Gyroscopic output by reading 6 Bytes starting from address 0x02. [0][1]...[5]
    2. Format Each of the X,Y, and Z axis Gyro data
       - Gyro\_X\_RawOUT = ([1]<<8) | ([0])
       - Gyro\_Y\_RawOUT = ([3]<<8) | ([2])
       - Gyro\_Z\_RawOUT = ([5]<<8) | ([4])
    3. Cancel for offset value
       - Measure RawOUT data 10 times in order to proofread offset value.
       - Calculate average offset value. This data is defined as proofread data.
       - Subtract proofread data from each RawOUT data.
    4. Convert Returned Value to deg/sec
       - Gyro\_X = (float)KXG03\_Gyro\_X\_RawOUT2 \* 0.007813 + 0.000004
       - Gyro\_Y = (float)KXG03\_Gyro\_Y\_RawOUT2 \* 0.007813 + 0.000004
       - Gyro\_Z = (float)KXG03\_Gyro\_Z\_RawOUT2 \* 0.007813 + 0.000004
    5. Format Serial Output and Return Information