GENERAL DESCRIPTION

The MC3610 is an ultra-low power, low noise, integrated digital output 3-axis accelerometer with a feature set optimized for wearables and the Internet of Moving Things (IoMT) devices. With its low power modes and tiny footprint this accelerometer extends sensor battery life and reduces the space required for the sensor by up to 3X. Low noise and low power are inherent in the monolithic fabrication approach.

The EVA3610A is a prebuilt circuit board with the MC3610 accelerometer sensor, with an internal sample rate from 0.4 to 370 samples/second. It measures acceleration with a wide usage range, from +/-2g up to +/-16g. It offers 6-bit to 14-bit high-precision ADC output which is easy mounted onto a microcontroller platform, such as Arduino. The accelerometer communicates via I2C (or SPI) and gives out motion detection or sample acquisition conditions to trigger an interrupt toward a MCU.

Sensor data is easily read by simply connecting DVDD to 3.3V, GND to ground, and SCL/SDA pins to your Arduino I2C clock and data pin respectively. Download the MC3610 library from Github onto the board, run the example sketch, and then sensor data shortly comes out in raw data count and SI unit accelerometer measurements.

An easy-to-use demonstration on the mCube EV3610A eval board using the Arduino platform is described within.

MC3610 FEATURES

Range, Sampling & Power

- ±2,4,8,12 or 16g ranges
- 8, 10 or 12-bit resolution with FIFO
  - 14-bit single samples
- 0.4 - 370 samples/sec
- Ultra-Low Power with FIFO
  - 0.6 µA typical sniff current
  - 1.3 µA typical current @ 25Hz
  - 4.7 µA typical current @ 50Hz
  - 14 µA @ 50Hz, low noise

Simple System Integration

- I2C interface, up to 400 kHz
- SPI Interface, up to 2 MHz
- 2 × 2 × 0.94 mm 12-pin package
- Single-chip 3D silicon MEMS
- Low noise down to 280 µg / √Hz

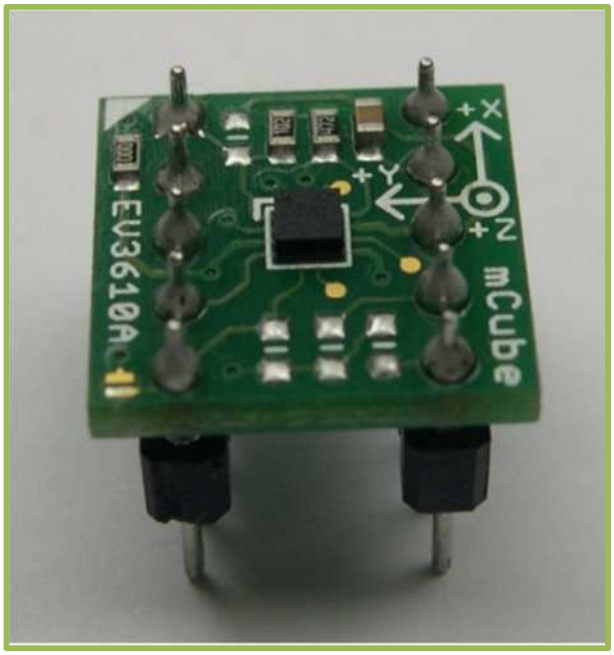
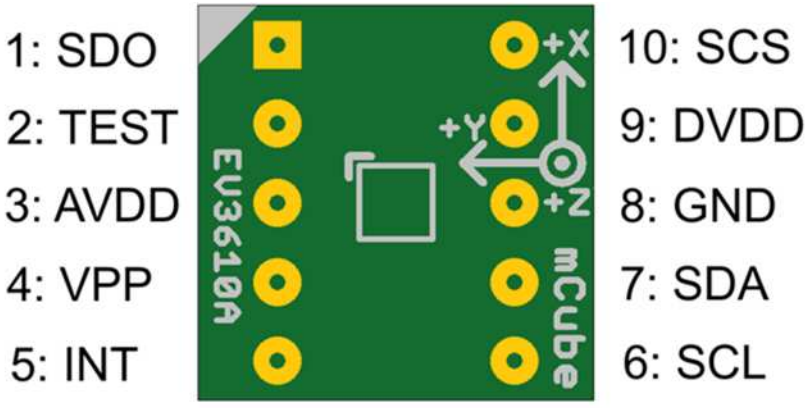


TABLE OF CONTENTS

1 GENERAL OPERATION

1.1 PINOUTS

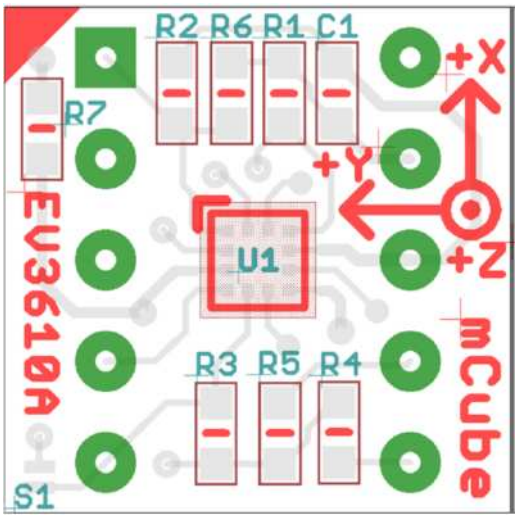


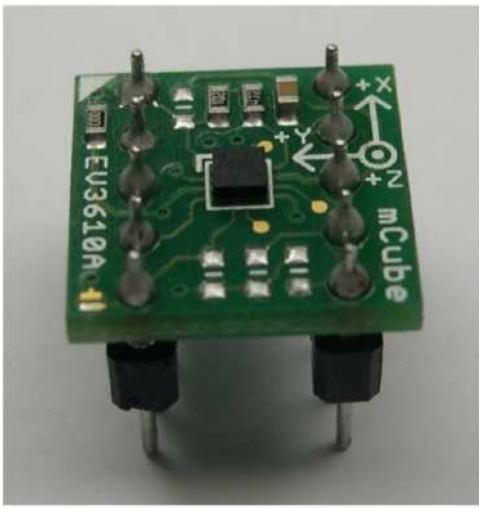
1.2 POWER PINS

- **DVDD** – 3.3V Power Supply Input
- **GND** – Ground Pin for Power and Logic

1.3 I2C PINS

- Connect the **SCL** (I2C clock pin) to your microcontroller's I2C clock line.
- Connect the **SDA** (I2C data pin) to your microcontroller's I2C data line.





R4, R5: If using I2C and I2C pull-up resistors are needed for your application then install~4.7KΩ resistors into R4 (SCL clock pin) and R5 (SDA data pin) which are not installed by factory default. In addition, besides soldering resisters on R4/R5, you can add axial lead 4.7K ohm resistors to the SDA and SCL pin respectively. It will work the same either way.

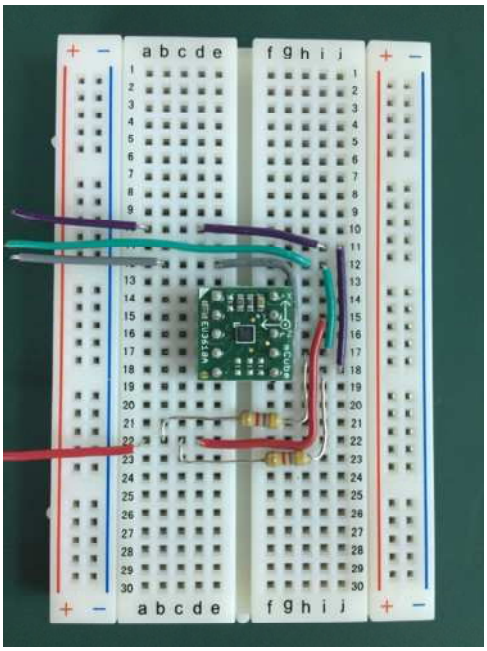
NOTE: DO NOT install more than one setup pull-up resistors per I2C bus.

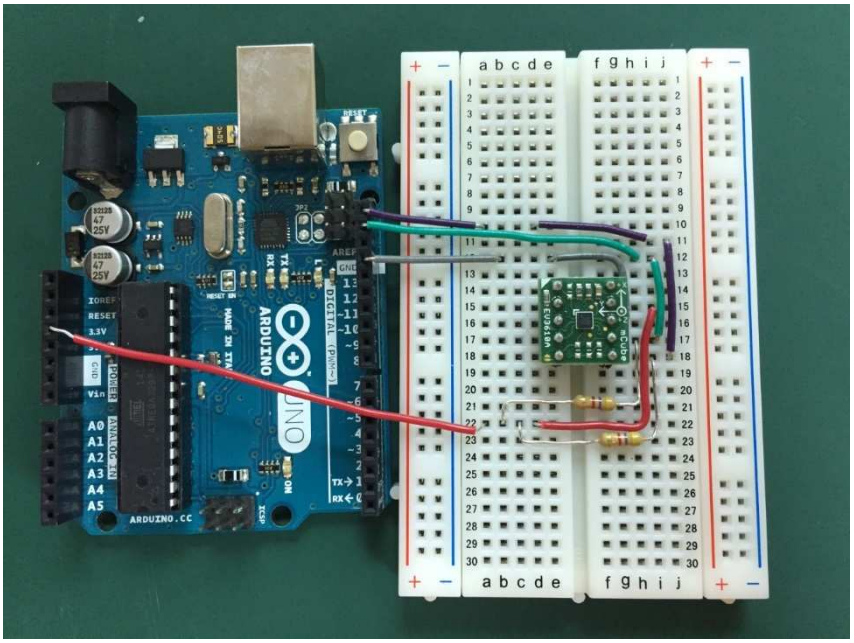
1.4 INTERRUPT PINS

INT - HW interrupt signal pin. It will be driven by the chip when data is ready to read, or a motion event is detected by the accelerometer. (Not currently supported in the library for the interrupt pin, so please check the datasheet for the I2C commands toward related registers).

2 ASSEMBLY AND TEST

You can easily wire this breakout to any microcontroller; we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C with repeated-start support, then port the code. Please refer to the illustration below to connect the related pins.





- Connect **DVDD** to the power supply, **3.3V**.
  (Providing higher voltage, like 5V may damage the sensor.)
- Connect **GND** to common power/data ground.
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino.
- Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino.

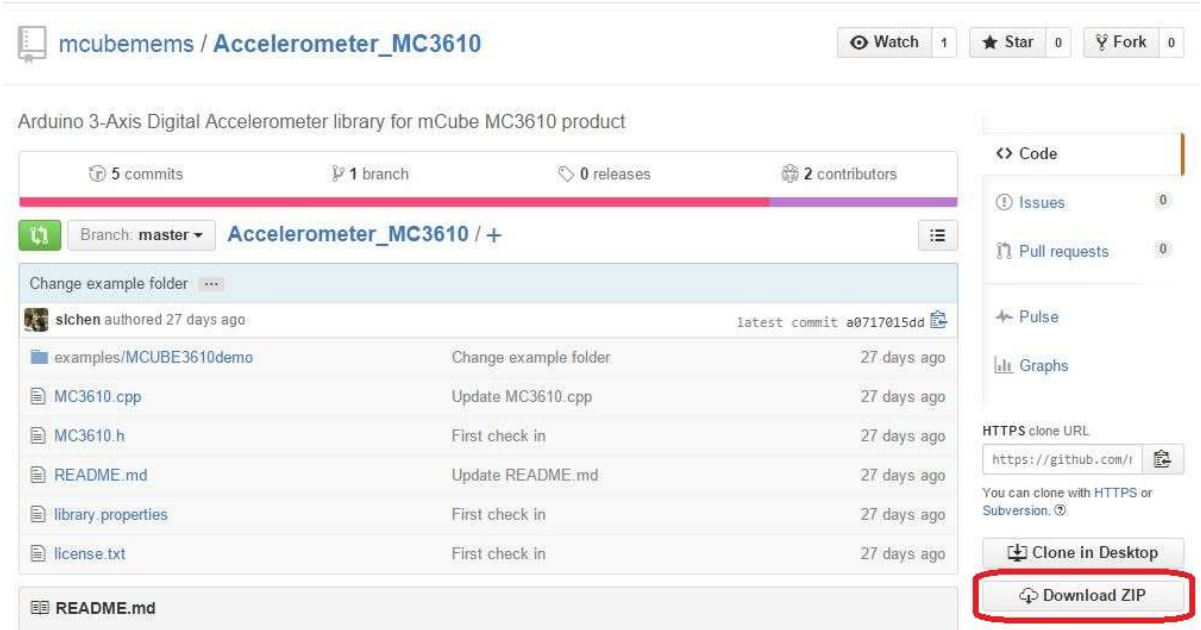
The MC3610 has a default I2C address of 0x6C and can be changed to 0x4C by tying the DOUT pin to GND

3 DEMO

3.1 DOWNLOAD THE DRIVER FROM GITHUB

To begin reading sensor data, you will need to download the MC3610_Library from the github repository. You can do that by visiting the github repository and manually downloading or simply click this button the attached URL to download the zip file.

https://github.com/mcubemems/Accelerometer_MC3610



Rename the uncompressed folder **Accelerometer_MC3610** and check that the Accelerometer_MC3610 folder contains **MC3610.cpp** and **MC3610.h**

Place the Accelerometer_MC3610 library folder to your **Arduino_sketch_folder/libraries/** folder.

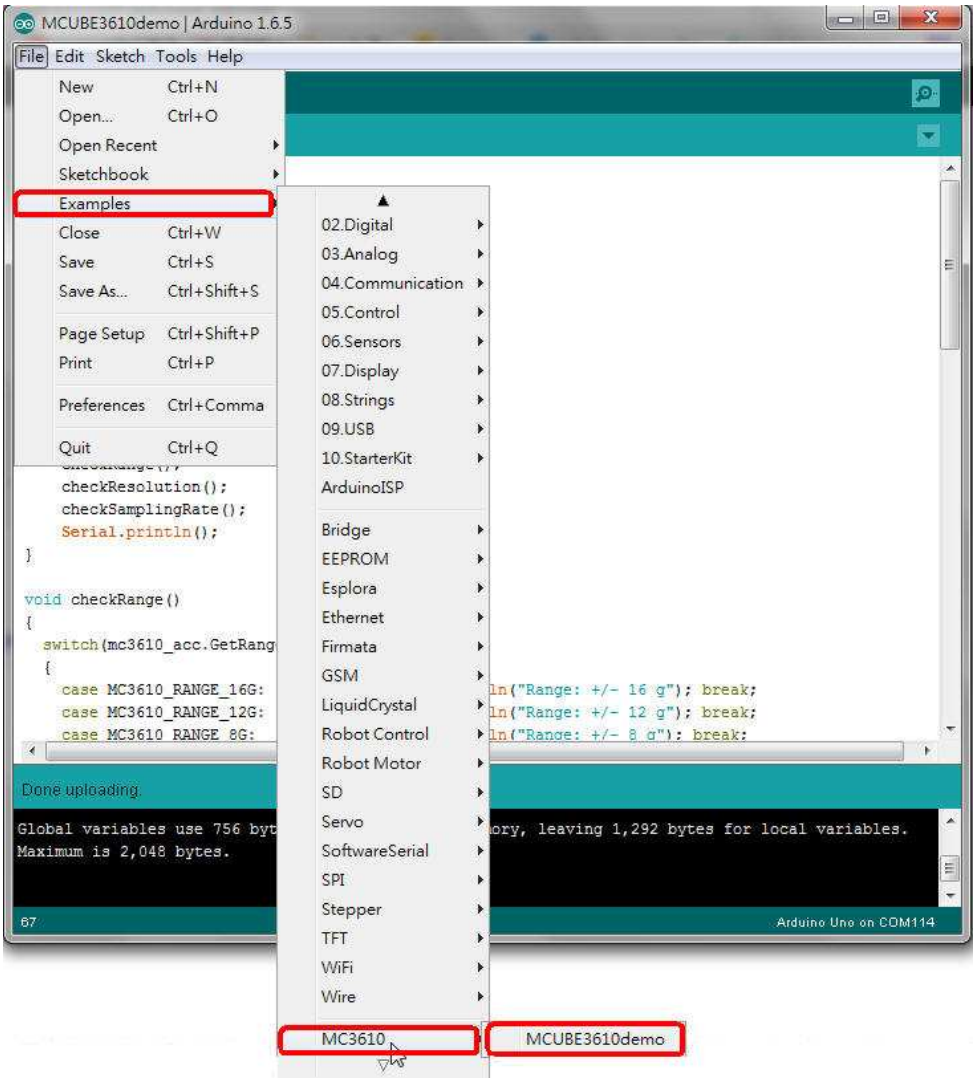
You may need to create the library subfolder if it is your first library. Then just restart the IDE.

An excellent tutorial on Arduino library installation is located at:

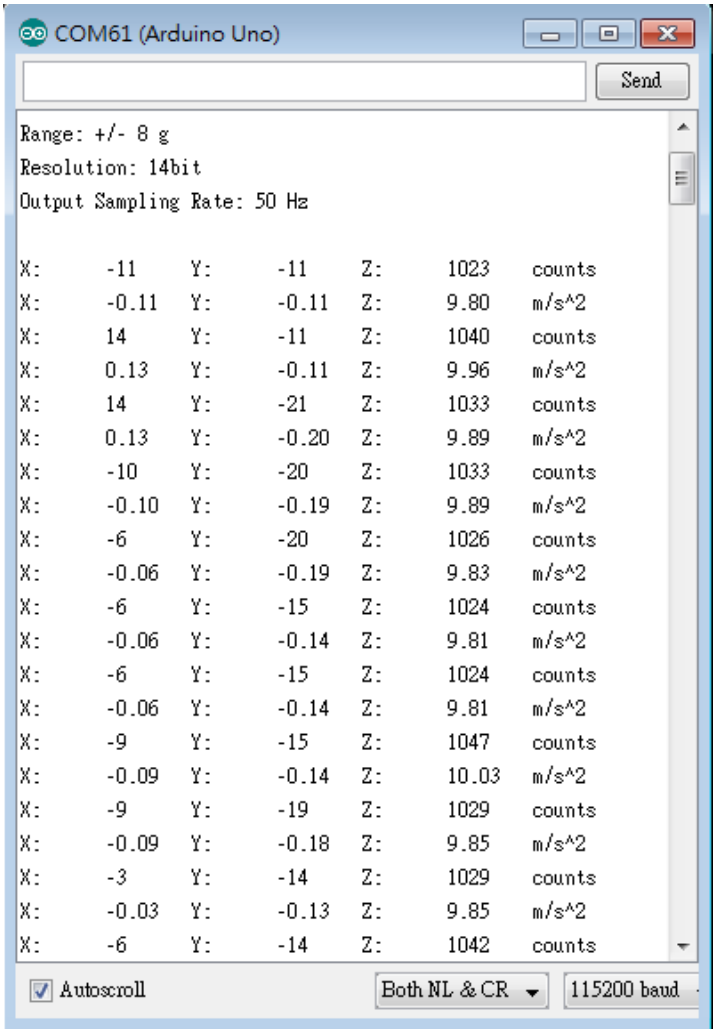
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use

3.2 LOAD THE DEMO

Open up File->Examples->MC3610-> MC3610demo and upload to your Arduino wired up to the sensor



Now open up the serial terminal window at 115200 baud rate speed to begin the test.



You will see the output from the serial terminal showing the current range scale and resolution of the sensor in the first three lines followed by two lines of output sensor data at some output data rate which depict “raw count" data for line 1: X: -16 Y: -8 Z: 1030 with 8G range, 14bit ADC resolution.

Line 2 indicates the SI units for measuring acceleration as X:-0.15 m/s^2 Y:-0.08 m/s^2 Z: 9.86 m/s^2.

4 LIBRARY REFERENCE

4.1 CREATE MCUBE_MC3610 OBJECT

You can create the MCUBE_MC3610 object with:

```
MC3610 mc3610_acc = MC3610();
```

4.2 INITIALIZE AND CONFIGURE SENSOR

Initialize and configure the sensor with:

```
Mc3610_acc.start();
```

4.3 SET RANGE

Set the accelerometer max range to ±2g, ±4g, ±8g or ±16g with:

```
mc3610_acc.SetRangeCtrl(MC3610_RANGE_2G);
mc3610_acc.SetRangeCtrl(MC3610_RANGE_4G);
mc3610_acc.SetRangeCtrl(MC3610_RANGE_8G);
mc3610_acc.SetRangeCtrl(MC3610_RANGE_16G);
```

4.4 READ RANGE

Read the current range with:

```
mc3610_acc.GetRangeCtrl();
```

This returns: 0 for ±2g, | 1 for ±4g, | 2 for ±8g | 3 for ±16g.

4.5 READ RESOLUTION

Read the current resolution with:

```
mc3610_acc.GetResolutionCtrl();
```

This returns: 0 for 6-bit | 1 for 7-bit | 2 for 8-bit | 3 for 10-bit | 4 for 12-bit | 5 for 14-bit

4.6 READ RAW COUNT DATA

Read the raw count data and SI unit measurement with:

```
mc3610_acc.readRawAccel();
```

5 DOWNLOADS

5.1 MC3610 ACCELEROMETER DATASHEET

http://www.mcubemems.com/product/mc3610-3-axis-accelerometer/

5.2 EV3610A EVAL BOARD QUICK START GUIDE

http://www.mcubemems.com/product/mc3610-3-axis-accelerometer/

5.3 EV3610A EVAL BOARD DATASHEET

http://www.mcubemems.com/product/mc3610-3-axis-accelerometer/

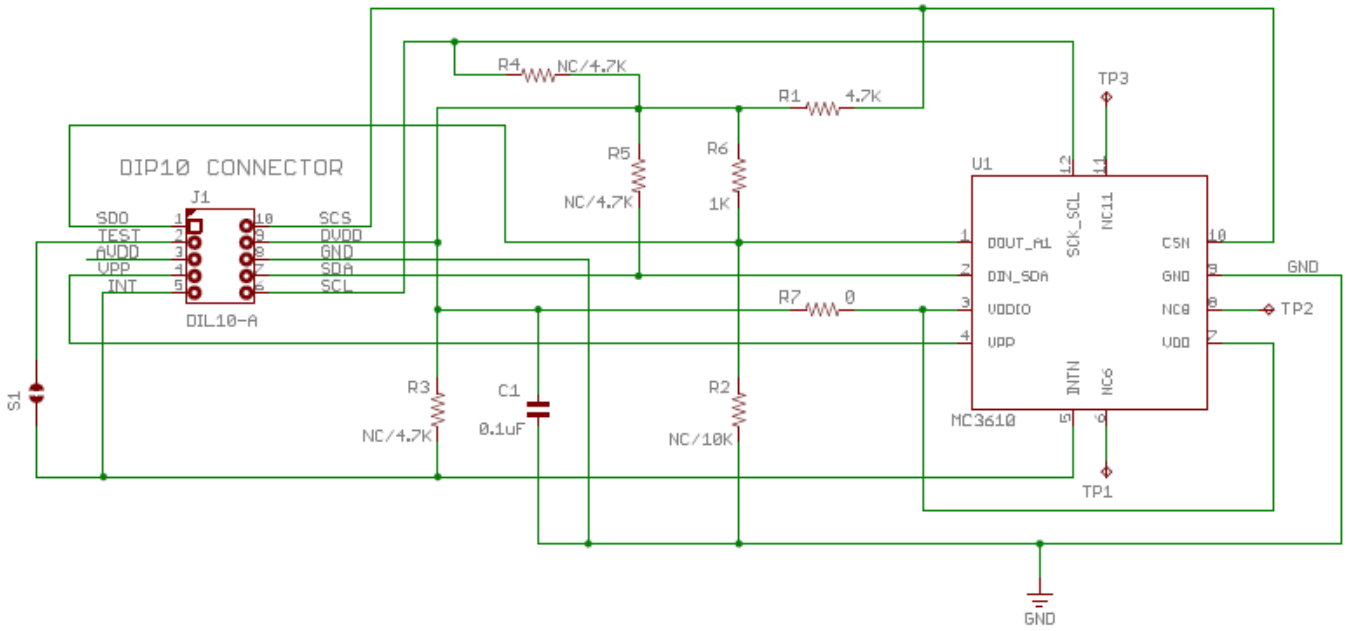
5.4 MC3610 DRIVER AT GITHUB

https://github.com/mcubemems/Accelerometer_MC3610

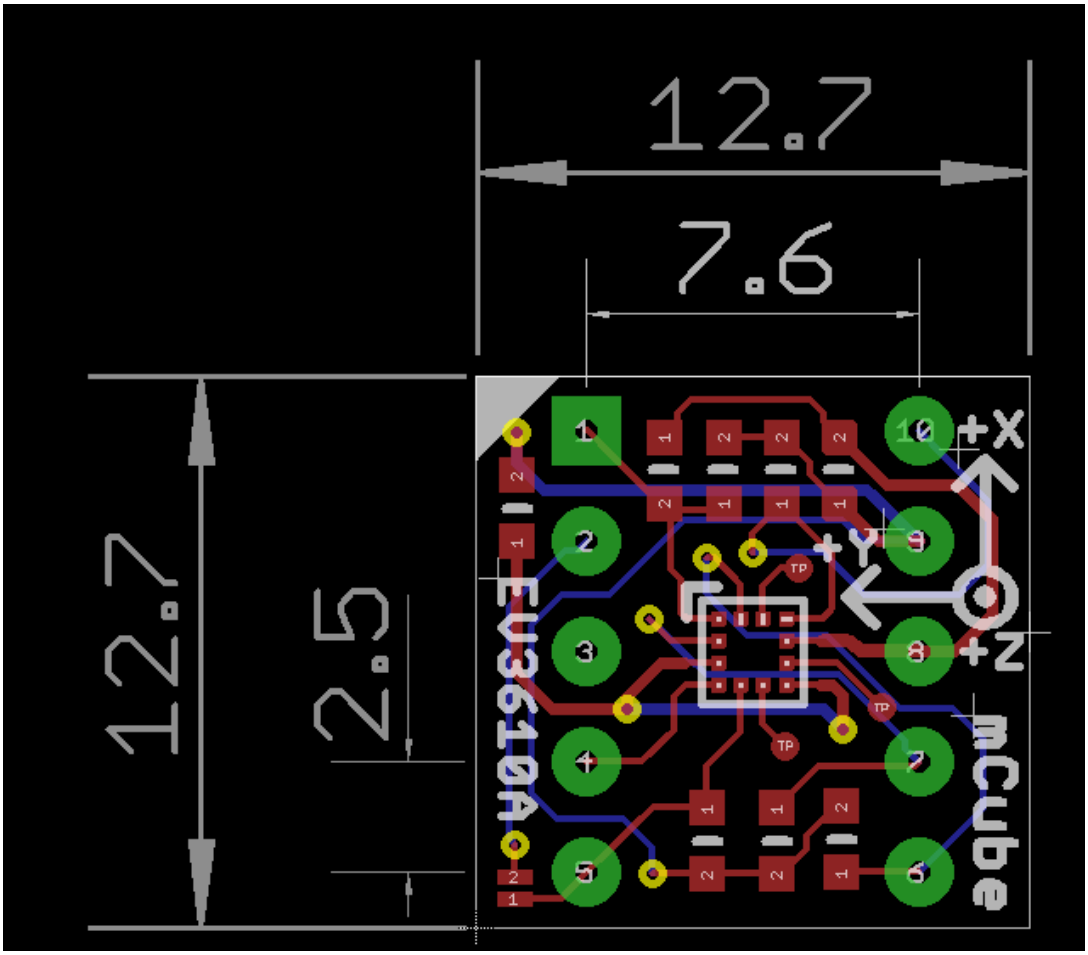
5.5 ALL OTHER MCUBE DOCUMENTATION

http://www.mcubemems.com/resources-support/resources/

6 SCHEMATICS



7 FABRICATION PRINT



8 REVISION HISTORY

Date	Revision	Description
2015-08	APS-045-0017v1.0	First release.
2015-08	APS-045-0017v1.1	Minor edits and schematic upgrade

9 LEGAL

1. M-CUBE reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and to this document at any time and discontinue any product without notice. The information contained in this document has been carefully checked and is believed to be accurate. However, M-CUBE shall assume no responsibilities for inaccuracies and make no commitment to update or to keep current the information contained in this document.

2. M-CUBE products are designed only for commercial and normal industrial applications and are not suitable for other purposes, such as: medical life support equipment; nuclear facilities; critical care equipment; military / aerospace; automotive; security or any other applications, the failure of which could lead to death, personal injury or environmental or property damage. Use of the products in unsuitable applications are at the customer's own risk and expense.

3. M-CUBE shall assume no liability for incidental, consequential or special damages or injury that may result from misapplication or improper use of operation of the product.

4. No license, express or implied, by estoppel or otherwise, to any intellectual property rights of M-CUBE or any third party is granted under this document.

5. M-CUBE makes no warranty or representation of non-infringement of intellectual property rights of any third party with respect to the products. M-CUBE specifically excludes any liability to the customers or any third party regarding infringement of any intellectual property rights, including the patent, copyright, trademark or trade secret rights of any third party, relating to any combination, machine, or process in which the M-CUBE products are used.

6. Examples of use described herein are provided solely to guide use of M-CUBE products and merely indicate targeted characteristics, performance and applications of products. M-CUBE shall assume no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein

7. Information described in this document including parameters, application circuits and its constants and calculation formulas, programs and control procedures are provided for the purpose of explaining typical operation and usage. "Typical" parameters that may be provided in M-CUBE data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters including "Typicals," must be validated for each customer application by customer's technical experts. In no event shall the information described be regarded as a guarantee of conditions or characteristics of the products. Therefore, the customer should evaluate the design sufficiently as whole system under the consideration of various external or environmental conditions and determine their application at the customer's own risk. M-CUBE shall assume no responsibility or liability for claims, damages, costs and expenses caused by the customer or any third party, owing to the use of the above information.

is a trademark of M-CUBE, Inc.

M-CUBE and the M-CUBE logo are trademarks of M-CUBE, Inc.,

All other product or service names are the property of their respective owners.