# KE2059 KEYES MMA8452Q Module Triaxial Digital Acceleration Tilt Sensor

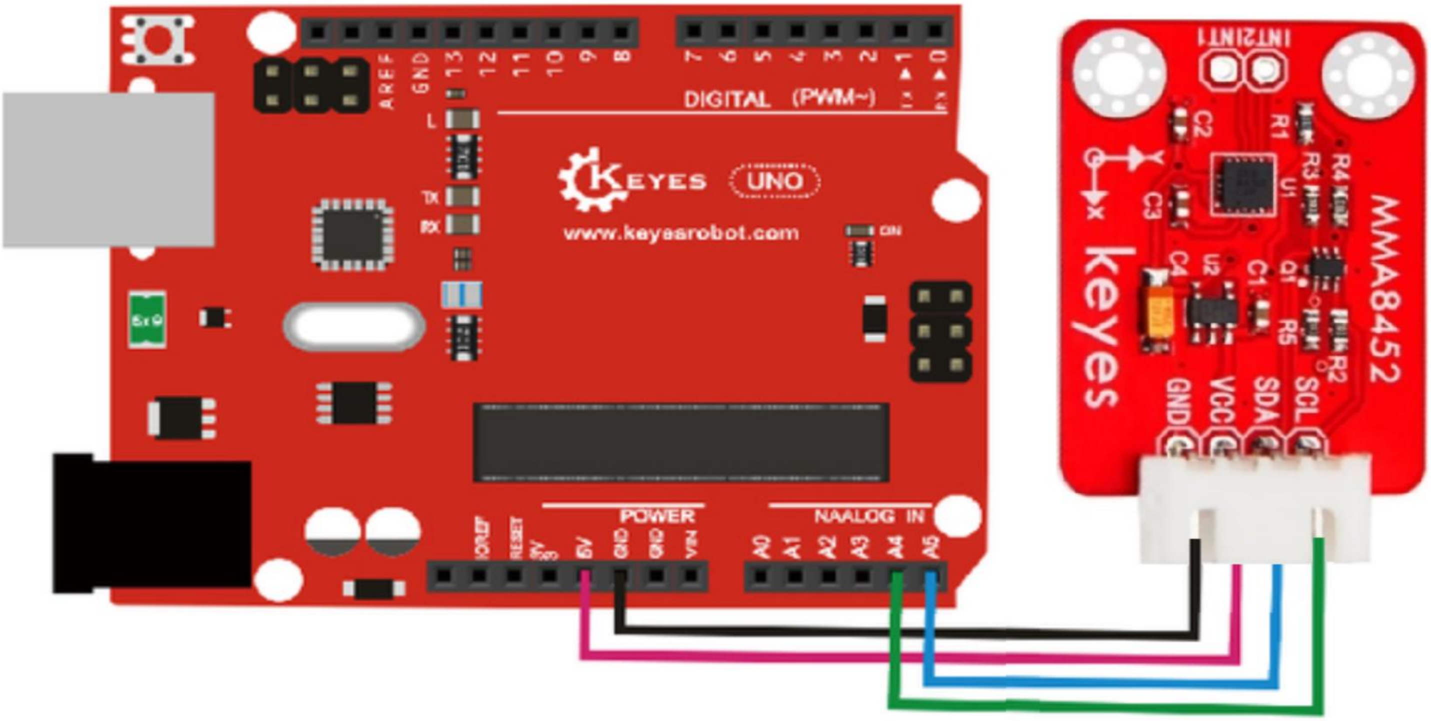
**Parameters:**

Working Voltage: 3.3 ~ 5VDC

Colour: Red

Size: 39x22x9mm.

**PINOUT Instruction:**



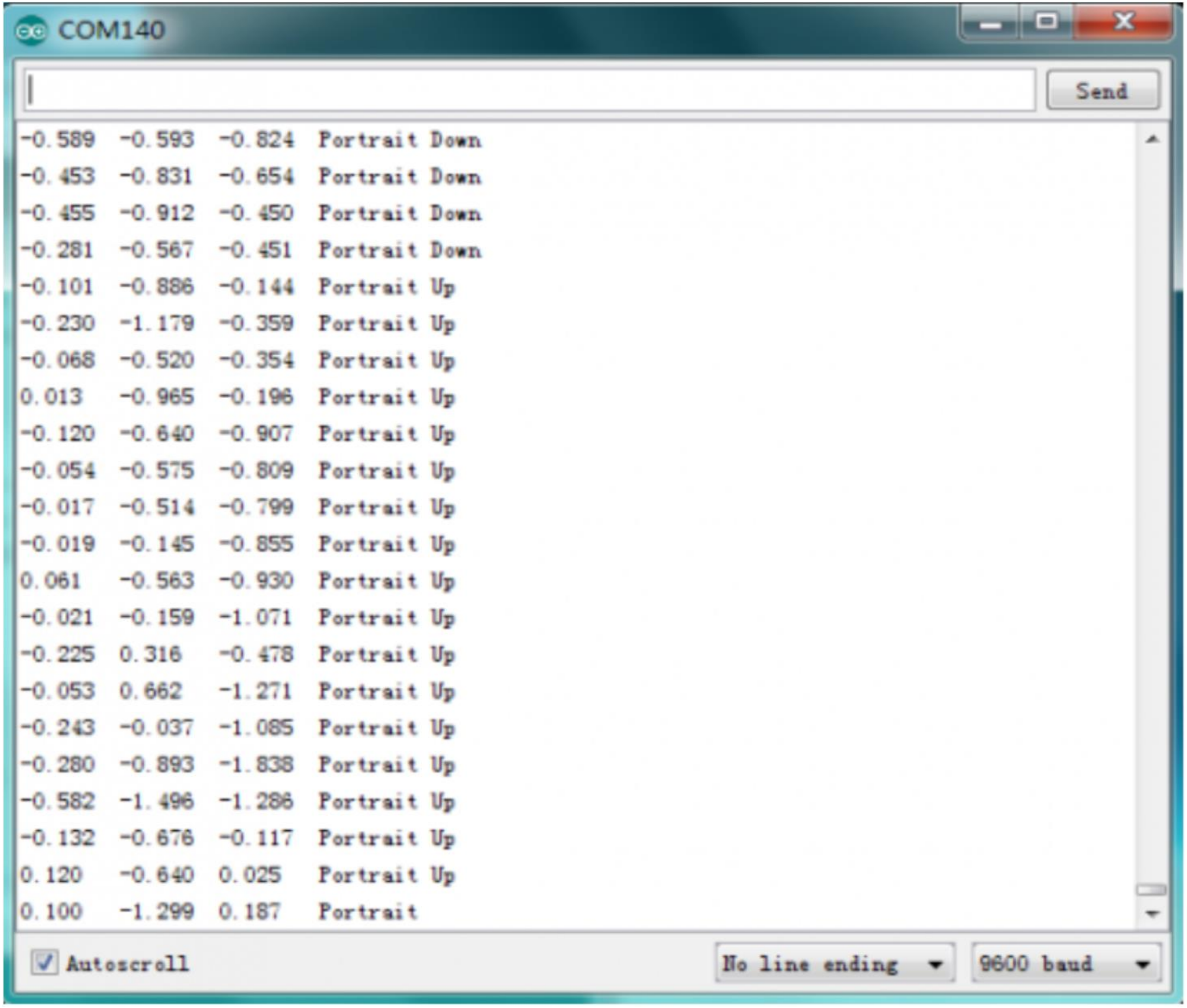
Note: before compiling the code, do remember to place the library into libraries directory of Arduino IDE. Otherwise, compiling will fail.

**Sample Code:**

```
#include <Wire.h> // Must include Wire library for I2C
#include <SparkFun_MMA8452Q.h> // Includes the SFE_MMA8452Q library
// Begin using the library by creating an instance of the MMA8452Q
//  class. We'll call it "accel". That's what we'll reference from
//  here on out.
MMA8452Q accel;
// The setup function simply starts serial and initializes the
//  accelerometer.
void setup()
{
  Serial.begin(9600);
  Serial.println("MMA8452Q Test Code!");

  // Choose your adventure! There are a few options when it comes
  // to initializing the MMA8452Q:
  //  1. Default init. This will set the accelerometer up
  //     with a full-scale range of +/-2g, and an output data rate
  //     of 800 Hz (fastest).
  accel.init();
  //  2. Initialize with FULL-SCALE setting. You can set the scale
  //     using either SCALE_2G, SCALE_4G, or SCALE_8G as the value.
  //     That'll set the scale to +/-2g, 4g, or 8g respectively.
  //accel.init(SCALE_4G); // Uncomment this out if you'd like
  //  3. Initialize with FULL-SCALE and DATA RATE setting. If you
  //     want control over how fast your accelerometer produces
  //     data use one of the following options in the second param:
  //     ODR_800, ODR_400, ODR_200, ODR_100, ODR_50, ODR_12,
  //     ODR_6, or ODR_1.
  //     Sets to 800, 400, 200, 100, 50, 12.5, 6.25, or 1.56 Hz.
  //accel.init(SCALE_8G, ODR_6);
}

// The loop function will simply check for new data from the
//  accelerometer and print it out if it's available.
void loop()
{
  // Use the accel.available() function to wait for new data
  //  from the accelerometer.
  if (accel.available())
  {
    // First, use accel.read() to read the new variables:
    accel.read();

    // accel.read() will update two sets of variables.
    // * int's x, y, and z will store the signed 12-bit values
    //   read out of the accelerometer.
    // * floats cx, cy, and cz will store the calculated
    //   acceleration from those 12-bit values. These variables
    //   are in units of g's.
    // Check the two function declarations below for an example
    // of how to use these variables.
    printCalculatedAccels();
    //printAccels(); // Uncomment to print digital readings
    // The library also supports the portrait/landscape detection
    //  of the MMA8452Q. Check out this function declaration for
    //  an example of how to use that.
    printOrientation();

    Serial.println(); // Print new line every time.
  }
}

// The function demonstrates how to use the accel.x, accel.y and
//  accel.z variables.
// Before using these variables you must call the accel.read()
//  function!
void printAccels()
{
  Serial.print(accel.x, 3);
  Serial.print("\t");
  Serial.print(accel.y, 3);
  Serial.print("\t");
  Serial.print(accel.z, 3);
  Serial.print("\t");
}

// This function demonstrates how to use the accel.cx, accel.cy,
//  and accel.cz variables.
// Before using these variables you must call the accel.read()
//  function!
void printCalculatedAccels()
{
  Serial.print(accel.cx, 3);
  Serial.print("\t");
  Serial.print(accel.cy, 3);
  Serial.print("\t");
  Serial.print(accel.cz, 3);
  Serial.print("\t");
}
// This function demonstrates how to use the accel.readPL()
// function, which reads the portrait/landscape status of the
// sensor.
void printOrientation()
{
  // accel.readPL() will return a byte containing information
  // about the orientation of the sensor. It will be either
  // PORTRAIT_U, PORTRAIT_D, LANDSCAPE_R, LANDSCAPE_L, or
  // LOCKOUT.
  byte pl = accel.readPL();
  switch (pl)
  {
  case PORTRAIT_U:
    Serial.print("Portrait Up");
    break;
  case PORTRAIT_D:
    Serial.print("Portrait Down");
    break;
  case LANDSCAPE_R:
   Serial.print("Landscape Right");
   break;
  case LANDSCAPE_L:
   Serial.print("Landscape Left");
   break;
  case LOCKOUT:
   Serial.print("Flat");
   break;
  }
 }
```

**Result:**

Wiring as the above diagram and burning the code, after powered-on,then open the serial monitor to display the triaxial acceleration of sensor and its status, as the graph shown below.



```
-0.589   -0.593   -0.824   Portrait Down
-0.453   -0.831   -0.654   Portrait Down
-0.455   -0.912   -0.450   Portrait Down
-0.281   -0.567   -0.451   Portrait Down
-0.101   -0.886   -0.144   Portrait Up
-0.230   -1.179   -0.359   Portrait Up
-0.068   -0.520   -0.354   Portrait Up
0.013    -0.965   -0.196   Portrait Up
-0.120   -0.640   -0.907   Portrait Up
-0.054   -0.575   -0.809   Portrait Up
-0.017   -0.514   -0.799   Portrait Up
-0.019   -0.145   -0.855   Portrait Up
0.061    -0.563   -0.930   Portrait Up
-0.021   -0.159   -1.071   Portrait Up
-0.225   0.316    -0.478   Portrait Up
-0.053   0.662    -1.271   Portrait Up
-0.243   -0.037   -1.085   Portrait Up
-0.280   -0.893   -1.838   Portrait Up
-0.582   -1.496   -1.286   Portrait Up
-0.132   -0.676   -0.117   Portrait Up
0.120    -0.640   0.025    Portrait Up
0.100    -1.299   0.187    Portrait
```

No line ending
9600 baud

**Datasheet:** https://www.mantech.co.za/Datasheets/Products/MMA8452Q.pdf

**Reference:**

https://wiki.keyestudio.com/Ks0270_keyestudio_MMA8452Q_Module_Triaxial_Digital_Acceleration_Tilt_Sensor

**Download Code and Libraries:**

https://drive.google.com/open?id=1U-be4fsFEqMgt89J-bK2tVNJdrTP2ZvY