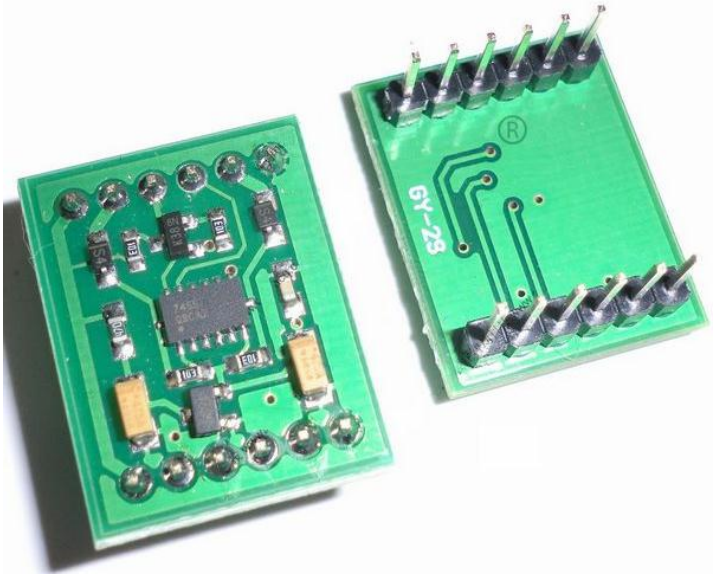
MMA7455 3-axis digital accelerometer module



Instruction

The MMA7455L is a Digital Output (I2C/SPI), low power, low profile capacitive micromachined accelerometer featuring signal conditioning, a low pass filter, temperature compensation, self-test, configurable to detect 0g through interrupt pins (INT1 or INT2), and pulse detect for quick motion detection. 0g offset and sensitivity are factory set and require no external devices. The 0g offset can be customer calibrated using assigned 0g registers and g-Select which allows for command selection for 3 acceleration ranges (2g/4g/8g). The MMA7455L includes a Standby Mode that makes it ideal for handheld battery powered electronics.

This MMA7455 module has low pressure regulator and IO port level converter. The power supply ranges from 3V~5V. And all the digital I/O interfaces are compatible with 3V~5V. Its voltage flexibility allows users to integrate it into most hardware systems.

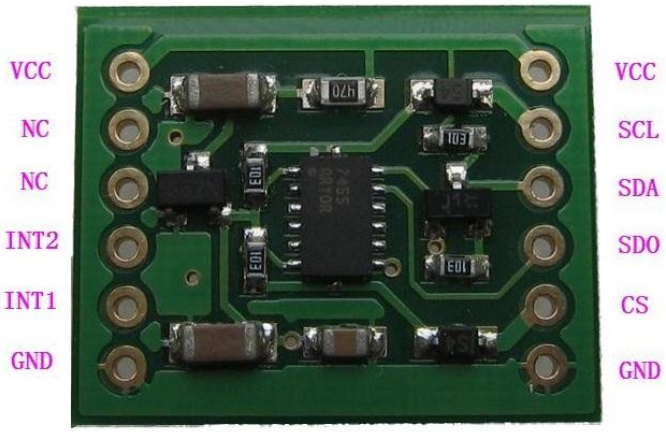
Features

- *Digital Output (I2C/SPI)*
- *Low Current Consumption*
- *Self-Test for Z-Axis*
- *User Assigned Registers for Offset Calibration*
- *Programmable Threshold Interrupt Output*
- *Level Detection for Motion Recognition (Shock, Vibration, Freefall)*
- *Pulse Detection for Single or Double Pulse Recognition*

- *Sensitivity (64 LSB/g @ 2g and @ 8g in 10-Bit Mode)*
- *Selectable Sensitivity (±2g, ±4g, ±8g) for 8-bit Mode*
- *Robust Design, High Shocks Survivability (5,000g)*
- *Environmentally Preferred Product*

Parameter

- *Power supply: 3V ~ 5 V*
- *Current: 30uA*
- *Interface: IIC / SPI*
- *Operating Temperature: -40 ° ~ 85 °*
- *Dimensions (L * W * H): 18.2mm × 23.3mm × 11.6mm*



Interface

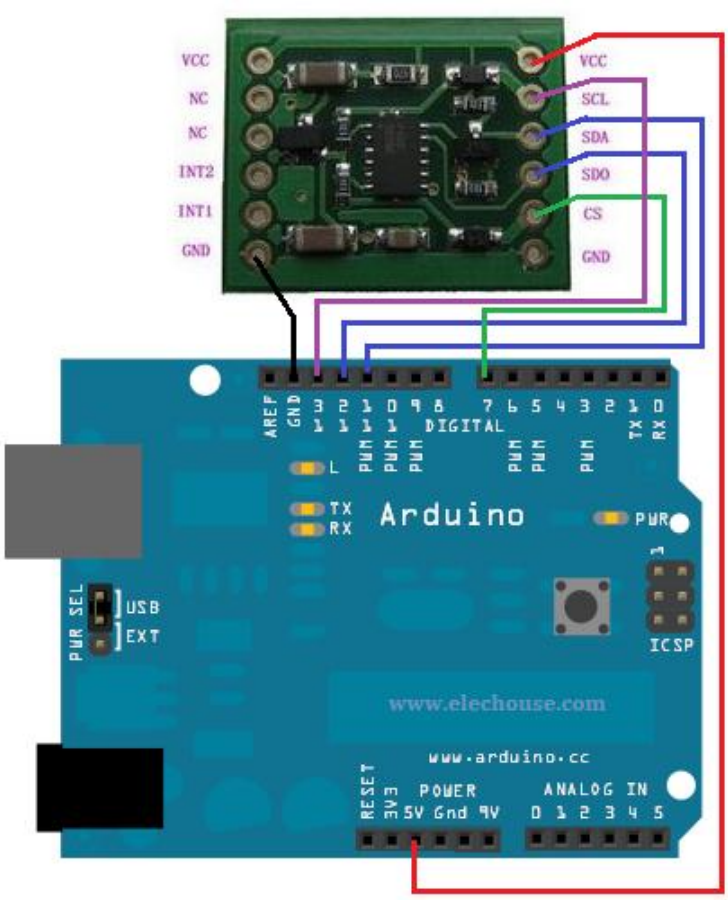
All the interfaces are compatible with 3V~5V.

PIN NUMBER	NAME	DESCRIPTION
1	VCC	3V~5V DC
2	NC	NC
3	NC	NC
4	INT1	Interrupt 1
5	INT2	Interrupt 2
6	GND	GND
7	GND	GND
8	CS	SPI / I2C CS
9	SDO	SPI Serial Data Output (MOSI)
10	SDA/SDIO	I2C Serial Data (SDA), SPI Serial Data Input (SDI/ MISO), 3-wire interface Serial Data Output (SDO)
11	SCL	SPI / IIC clock line
12	VCC	3V~5V DC

Example

Here we show an example of this module working with Arduino.

Hardware connection



Code

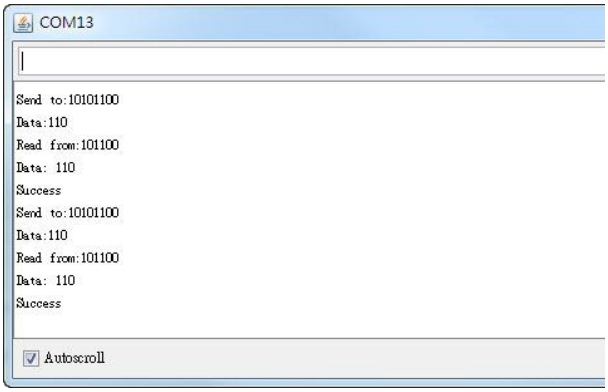
Update the following code to Arduino

```
/*
 Circuit:
 CS: pin 7
 MOSI: pin 11
 MISO: pin 12
 SCK: pin 13
*/
// the sensor communicates using SPI, so include the library:
#include <SPI.h>

//Sensor's memory register addresses:
//Data ready status is output to INT1/DRDY PIN
//SPI is 4 wire mode
//Self test is not enabled
//2G is selected for measurement range
//Measurement Mode
```

```
const int Mode = 0x05;

const int Mode_address = 0x16;

const int X_address = 0x00;
const int Y_address = 0x02;
const int Z_address = 0x04;

const byte READ = 0b00000000;     //read command
const byte WRITE = 0b10000000;   // write command

// pins used for the connection with the sensor
// the other you need are controlled by the SPI library):
const int CSPin = 7;
unsigned int read_buffer;

void setup() {
  Serial.begin(9600);

  // start the SPI library:
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  SPI.setClockDivider(SPI_CLOCK_DIV128);

  // initalize the  data ready and chip select pins:
  pinMode(CSPin, OUTPUT);
  digitalWrite(CSPin, HIGH);   //pull high


}

void loop() {
    //Configure
  writeRegister(Mode_address,Mode);

  // give the sensor time to set up:
  delay(100);
  read_buffer = readRegister(Mode_address);
  if (read_buffer==Mode)
  {
    Serial.println("Success");
  }
  else
    {
    Serial.println("Fail");
  }
  delay(1000);

}

//Read from or write to register from the SCP1000:
unsigned int readRegister(byte thisRegister) {
  unsigned int result = 0;   // result to return
  //left shift 1 bit
  thisRegister = thisRegister << 1;
  byte dataToRead = thisRegister | READ;
  // take the chip select low to select the device:
  digitalWrite(CSPin, LOW);
  // send the address you want to read:
  Serial.print("Read from:");
  Serial.println(dataToRead,BIN);
  SPI.transfer(dataToRead);
  // send a value of 0 to read the first byte returned:
  result = SPI.transfer(0x00);
  Serial.print("Data: ");
  Serial.println(result,BIN);
```

```
  // take the chip select high to de-select:
  digitalWrite(CSPin, HIGH);
  // return the result:
  return(result);
}


//Sends a write command to MMA7450L
void writeRegister(byte thisRegister, byte thisValue)
{
  // take the chip select low to select the device:
  digitalWrite(CSPin, LOW);
    // shift one byte:
  thisRegister = thisRegister<<1;
  // combine the register address and the command into one byte:
  byte dataToSend = thisRegister | WRITE;
     Serial.print("Send to:");
  Serial.println(dataToSend,BIN);
  Serial.print("Data:");
  Serial.println(thisValue,BIN);
  SPI.transfer(dataToSend); //Send register location
  SPI.transfer(thisValue);  //Send value to record into register
  // take the chip select high to de-select:
  digitalWrite(CSPin, HIGH);
}
```

Result



Other information

For Arduino users, you can refer to this topic for more information.

Contact

Web: www.elechouse.com Email: elechouse@elechouse.com

Disclaimer and Revisions

The information in this document may change without notice.
Revision History

Rev.	Date	Author	Description
A	Oct. 22nd, 2011	Wilson Shen	Initial version