SmartElex Triple Axis Magnetometer - MLX90393

The MLX90393 is a tri-axial magnetic sensor capable of sensing very small fields while behaving as one would want and expect during saturation in larger fields (like a nearby magnet).The MLX90393 can be used as a compass sensor but also works well as a non-contact controller (joystick), flow meter (with magnetic impeller), or a linear actuator position sensor.

In this guide, we'll get going by getting some basic readings from the sensor, then look at how to configure the sensor on different I^2C addresses.

Hardware Overview

Let's look over a few characteristics of the MLX90393 so we know a bit more about how it behaves.

Characteristic	Range
Operating Voltage	2.2V - 3.6V
Operating Temperature	-20°C - 85°C

Resolution	128 Hz - 3.3 kHz
Current Consumption	100 µA (Typ.)
I²C Address	0xC0

Pins

The characteristics of the available pins on the magnetometer are outlined in the table below.

Pin Label	Pin Function	Input/Output	Notes
3.3V	Power Supply	Input	Should be between **2.2V - 3.6V**
GND	Ground	Input	0V/common voltage.
SDA	I²C Data Signal	Bi-directional	Bi-directional data line. Voltage should not exceed power supply (e.g. 3.3V).
SCL/SCLK	I²C Clock Signal	Input	Master-controlled clock signal. Voltage should not exceed power supply (e.g. 3.3V).
CS	Chip Select	Input	Chip Select pin, digital input. Tied high for I²C operation, cut trace jumper and tie low for SPI.
INT	Interrupt	Output	Interrupt pin, active high, digital output. Also configurable as a data ready pin
TRIG	Reset	Input/Output	Trigger pin, active high, digital output. Also configurable as interrupt pin.

Optional Features

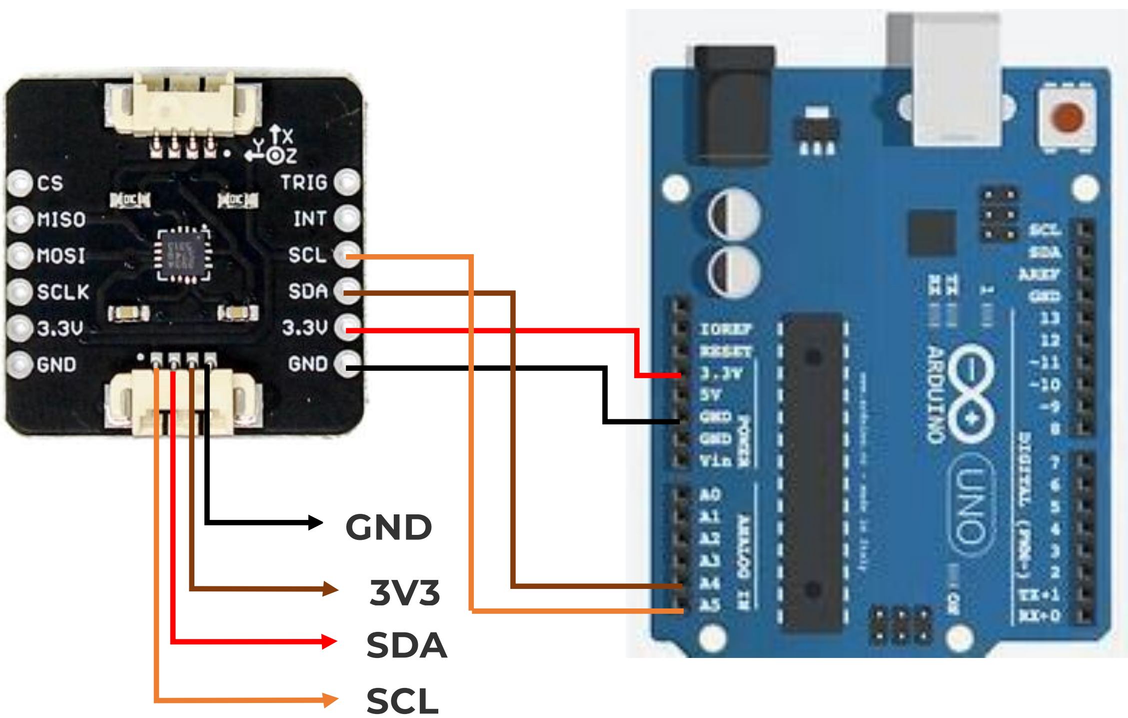
The MLX90393 has onboard I²C pull up resistors; if multiple sensors are connected to the bus with the pull-up resistors enabled, the parallel equivalent resistance will create too strong of a pull-up for the bus to operate correctly. As a general rule of thumb, disable all but one pair of pull-up resistors if multiple devices are connected to the bus. If you need to disconnect the pull up resistors they can be removed by cutting the traces on the corresponding jumpers highlighted below.

The I²C address of the Magnetometer can be changed using the A0 and A1 jumpers on the back of the board. Simply cut the traces connecting each pad to ground (0) and solder the other side to connect it to 3.3V (1).

To operate the Magnetometer in SPI mode, cut the chip select trace jumper (labeled below) and ensure that the CS pin is then connected to ground.

Wiring

Connecting the MLX90393 to Arduino:



MLX90393	Arduino
SCL	SCL(A5)
SDA	SDA(A4)
3V3	3.3v
GND	GND

- If you are running a 3.3v Arduino (Uno, etc.), connect **Arduino 3.3v** to **board VIN**
- Connect **Arduino GND** to **board GND**
- Connect **Arduino SCL** to **board SCL**
- Connect **Arduino SDA** to **board SDA**

Example Code

Basic Readings

Before we get into programming the Magnetometer, we'll need to download and install the MLX90393 magnetometer library.By Adafruit.

```
#include "Adafruit_MLX90393.h"

Adafruit_MLX90393 sensor = Adafruit_MLX90393();

#define MLX90393_CS 10

void setup(void)

{

  Serial.begin(115200);

  /* Wait for serial on USB platforms. */

  while (!Serial) {

     delay(10);

  }

  Serial.println("Starting Adafruit MLX90393 Demo");

  if (! sensor.begin_I2C()) {          // hardware I2C mode, can pass in address & alt Wire

  //if (! sensor.begin_SPI(MLX90393_CS)) {  // hardware SPI mode

    Serial.println("No sensor found ... check your wiring?");

    while (1) { delay(10); }

  }

  Serial.println("Found a MLX90393 sensor");


  sensor.setGain(MLX90393_GAIN_1X);

  // You can check the gain too

  Serial.print("Gain set to: ");

  switch (sensor.getGain()) {

    case MLX90393_GAIN_1X: Serial.println("1 x"); break;
```

```
    case MLX90393_GAIN_1_33X: Serial.println("1.33 x"); break;
    case MLX90393_GAIN_1_67X: Serial.println("1.67 x"); break;
    case MLX90393_GAIN_2X: Serial.println("2 x"); break;
    case MLX90393_GAIN_2_5X: Serial.println("2.5 x"); break;
    case MLX90393_GAIN_3X: Serial.println("3 x"); break;
    case MLX90393_GAIN_4X: Serial.println("4 x"); break;
    case MLX90393_GAIN_5X: Serial.println("5 x"); break;
  }
  // Set resolution, per axis. Aim for sensitivity of ~0.3 for all axes.
  sensor.setResolution(MLX90393_X, MLX90393_RES_17);
  sensor.setResolution(MLX90393_Y, MLX90393_RES_17);
  sensor.setResolution(MLX90393_Z, MLX90393_RES_16);
  // Set oversampling
  sensor.setOversampling(MLX90393_OSR_3);
  // Set digital filtering
  sensor.setFilter(MLX90393_FILTER_5);
}
void loop(void) {
  float x, y, z;
  // get X Y and Z data at once
  if (sensor.readData(&x, &y, &z)) {
     Serial.print("X: "); Serial.print(x, 4); Serial.println(" uT");
     Serial.print("Y: "); Serial.print(y, 4); Serial.println(" uT");
     Serial.print("Z: "); Serial.print(z, 4); Serial.println(" uT");
  } else {
     Serial.println("Unable to read XYZ data from the sensor.");
  }
  delay(500);
  /* Or....get a new sensor event, normalized to uTesla */
  sensors_event_t event;
  sensor.getEvent(&event);
  /* Display the results (magnetic field is measured in uTesla) */
  Serial.print("X: "); Serial.print(event.magnetic.x);
```

```
 Serial.print(" \tY: "); Serial.print(event.magnetic.y);
 Serial.print(" \tZ: "); Serial.print(event.magnetic.z);
 Serial.println(" uTesla ");
 delay(500);
}
```

Opening up your serial monitor to a baud rate of 9600 should display output similar to that shown below. The numbers are in units of µT.