Arduino Magnetometer

by amamitof7

What are we building?

Humans can't detect magnetic fields, but we use devices that rely on magnets all the time. Motors, compasses, rotation sensors, and wind turbines, for example, all require magnets for operation. This tutorial describes how to build an Arduino based magnetometer that senses magnetic field using three Hall effect sensors. The magnetic field vector at a location is displayed on a small screen using isometric projection.

What is an Arduino?

An Arduino is a small open-source user-friendly microcontroller. It has digital input and output pins. It also has analog input pins, which are useful for reading input from sensors. Different Arduino models are available. This tutorial describes how to use either the Arduino Uno or the Arduino MKR1010. However other models can be used too.

Before you begin this tutorial, download the Arduino development environment as well as any libraries needed for your particular model. The development environment is available at https://www.arduino.cc/en/main/software, and installation instructions are available at https://www.arduino.cc/en/main/software.

What is a magnetic field?

Permanent magnets exert forces on other permanent magnets. Current carrying wires exert forces on other current carrying wires. Permanent magnets and current carrying wires exert forces on each other too. This force per unit test current is a magnetic field.

If we measure the volume of an object, we get a single scalar number. However, magnetism is described by a vector field, a more complicated quantity. First, it varies with position throughout all space. For example, the magnetic field one centimeter from a permanent magnet is likely to be larger than the magnetic field ten centimeters away.

Next, the magnetic field at each point in space is represented by a vector. The magnitude of the vector represents the strength of the magnetic field. The direction is perpendicular to both the direction of the force and the direction of the test current.

We can picture the magnetic field at a single location as an arrow. We can picture the magnetic field throughout space by an array of arrow at different locations, possibly of different sizes and pointing in different directions. A nice visualization is available at https://www.falstad.com/vector3dm/. The magnetometer we are building displays the magnetic field at the location of the sensors as an arrow on the display.

What is a Hall effect sensor, and how does it work?

A Hall effect sensor is a small, inexpensive device that measures the strength of the magnetic field along a particular direction. It is made from a piece of semiconductor doped with excess charges. The output of some Hall effect sensors is an analog voltage. Other Hall effect sensors have an integrated comparator and produce a digital output. Other Hall effect sensors are integrated into larger instruments which measure flow rate, rotation speed, or other quantities.

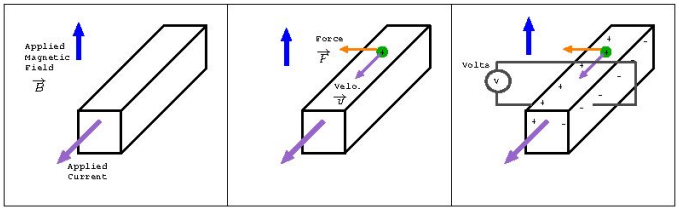
The physics behind the Hall effect is summarized by the Lorentz force equation. This equation describes the force on a moving charge due to an external electric and magnetic field.

$$\vec{F} = Q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

$\vec{F}$ = Force in newtons
Q = Charge in coulombs
$\vec{E}$ = Electric field intensity in volts per meter
$\vec{v}$ = velocity in meters per second
$\vec{B}$ = Magnetic flux density in webers per meter squared

The figure below illustrates the Hall effect. Suppose we want to measure the strength of the magnetic field in the direction of the blue arrow. As shown in the left part of the figure, we apply a current through a piece of semiconductor perpendicular to the direction of the

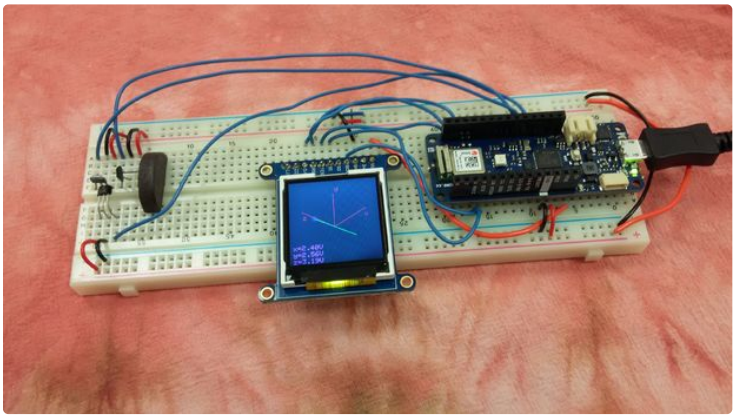
field to be measured. Current is flow of charges, so a charge in the semiconductor moves with some velocity. This charge will feel a force due to the external field, as shown in the middle part of the figure. Charges will move due to the force and accumulate on the edges of the semiconductor. Charges build up until the force due to the accumulated charges balances the force due to the external magnetic field. We can measure the voltage across the semiconductor, as shown in the right part of the figure. The voltage measured is proportional to the strength of the magnetic field, and it is in the direction perpendicular to the current and the direction of the magnetic field.



What is isometric projection?

At each point in space, the magnetic field is described by a three dimensional vector. However, our display screen is two dimensional. We can project the three dimensional vector into a two dimensional plane so that we can draw it on the screen. There are multiple ways to accomplish this such as isometric projection, orthographic projection, or oblique projection.

In isometric projection, the *x*, *y*, and *z* axes are 120 degrees apart, and they appear equally foreshortened. Additional information about isometric projection, as well as the formulas needed, can be found on Wikipedia's page on the topic.



Step 1: Gather Supplies

- **Arduino** and **Cable**

The Arduino is the brains of the magnetometer. These instructions describe how to use either an Arduino Uno or an Arduino MKR1010. In either case, a cable is needed to connect it to the computer.

Option 1: Arduino Uno and USB AB Cable

https://www.digikey.com/product-detail/en/arduino/A000066/1050-1024-ND/2784006

https://www.digikey.com/product-detail/en/stewart-connector/SC-2ABE003F/380-1424-ND/8544570

Option 2: Arduino MKR1010 and microUSB cable

https://www.digikey.com/product-detail/en/arduino/ABX00023/1050-1162-ND/9486713

https://www.digikey.com/product-detail/en/stewart-connector/SC-2AMK003F/380-1431-ND/8544577

- **TFT Display**

TFT stands for Thin Film Transistor. This 1.44" display contains 128 by 128 pixels. It is small, bright, and colorful. It comes attached to a breakout board. However, the header pins come separate, so you have to solder them on. (Solder and a soldering iron are needed.)

https://www.digikey.com/product-detail/en/adafruit-industries-llc/2088/1528-1345-ND/5356830

- **Analog Hall Effect Sensors**

Three Hall effect sensors are required. The link below is for Allegro part number A1324LUA-T. For this sensor, pin 1 is the supply voltage, pin 2 is ground, and pin 3 is the output. Other Hall sensors should work too, but make sure they are analog, not digital. If you use a different sensor, check the pinout and adjust the wiring if needed. (I actually used a different sensor from the same company for testing purposes. However, the one I used is obsolete, and this sensor is its replacement.)

https://www.digikey.com/product-detail/en/allegro-microsystems-llc/A1324LUA-T/620-1432-ND/2728144

- **Small Breadboard** and **Wire**

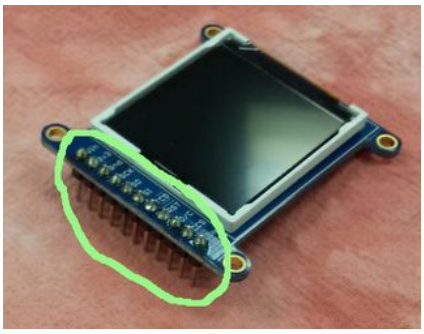
https://www.digikey.com/product-detail/en/adafruit-industries-llc/239/1528-2143-ND/7244929

- **Permanent Magnets for Testing**

Refrigerator magnets will work fine.

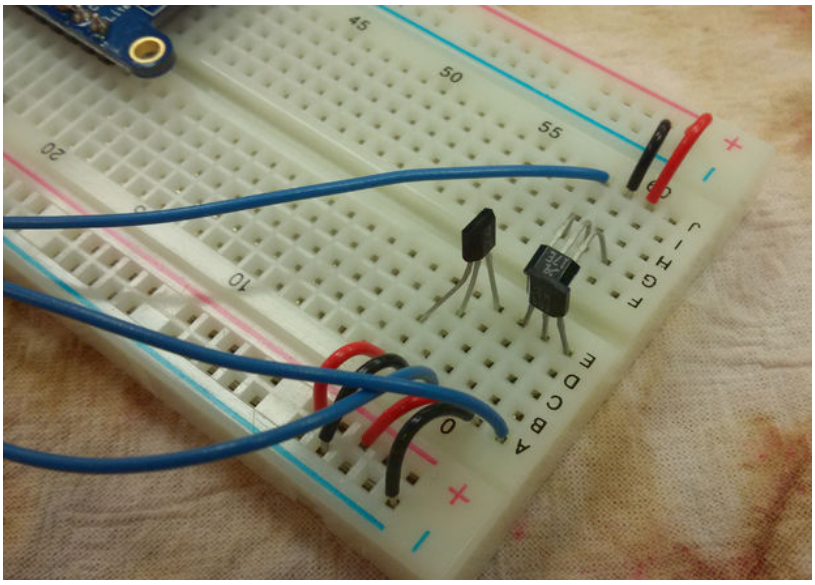
Step 2: Wiring

Solder the headers on the display.



Place the sensors at one end of the breadboard, and place the display and Arduino on the opposite end. Current in the wires in the Arduino and display generate magnetic fields, which we do not want the sensors to read. Additionally, we may want to put the sensors near permanent magnets, which could adversely impact the current in the wires of the display and sensor. For these reasons, we want the sensors far from the display and Arduino. Also for these reasons, this magnetometer should be kept away from very strong magnetic fields.

Place the sensors perpendicular to each other but as close to each other as possible. Gently bend the sensors to get them perpendicular. Each pin of each sensor must be in a separate row of the breadboard so it can be separately connected.

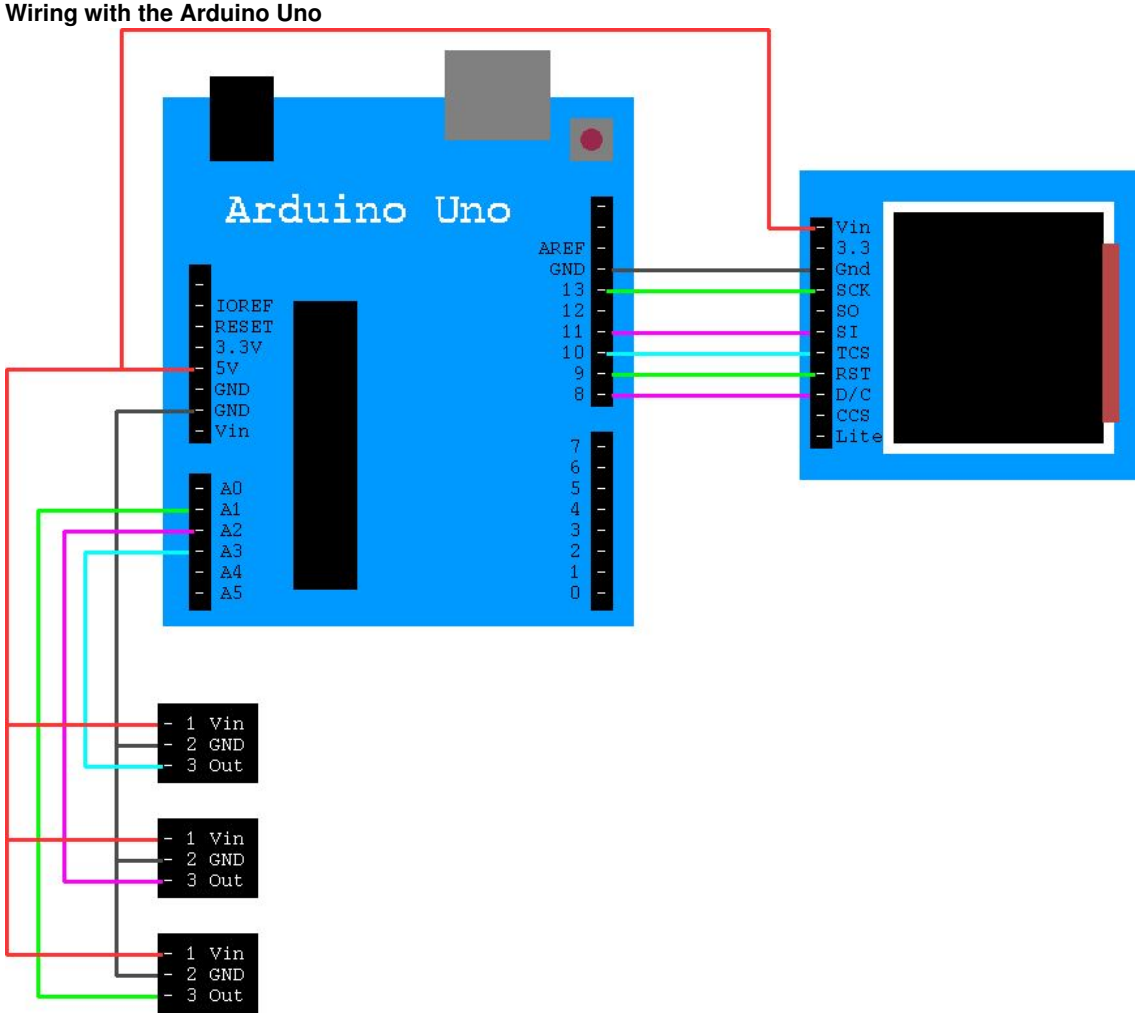


The wiring is slightly different between the MKR1010 and the Uno for two reasons. First, the Arduino and display communicate by SPI. Different Arduino models have different dedicated pins for certain SPI lines. Second, analog inputs of the Uno can accept up to 5 V while analog inputs of the MKR1010 can accept only up to 3.3 V. The

recommended supply voltage for the Hall effect sensors is 5 V. The sensor outputs are connected to Arduino analog inputs, and these can be as large as the supply voltages. For the Uno, use the recommended 5 V supply for the sensors. For the MKR1010, use 3.3 V so that the analog input of the Arduino never sees a voltage larger than it can handle.

Follow the diagrams and instructions below for the Arduino you are using.

Wiring with the Arduino Uno

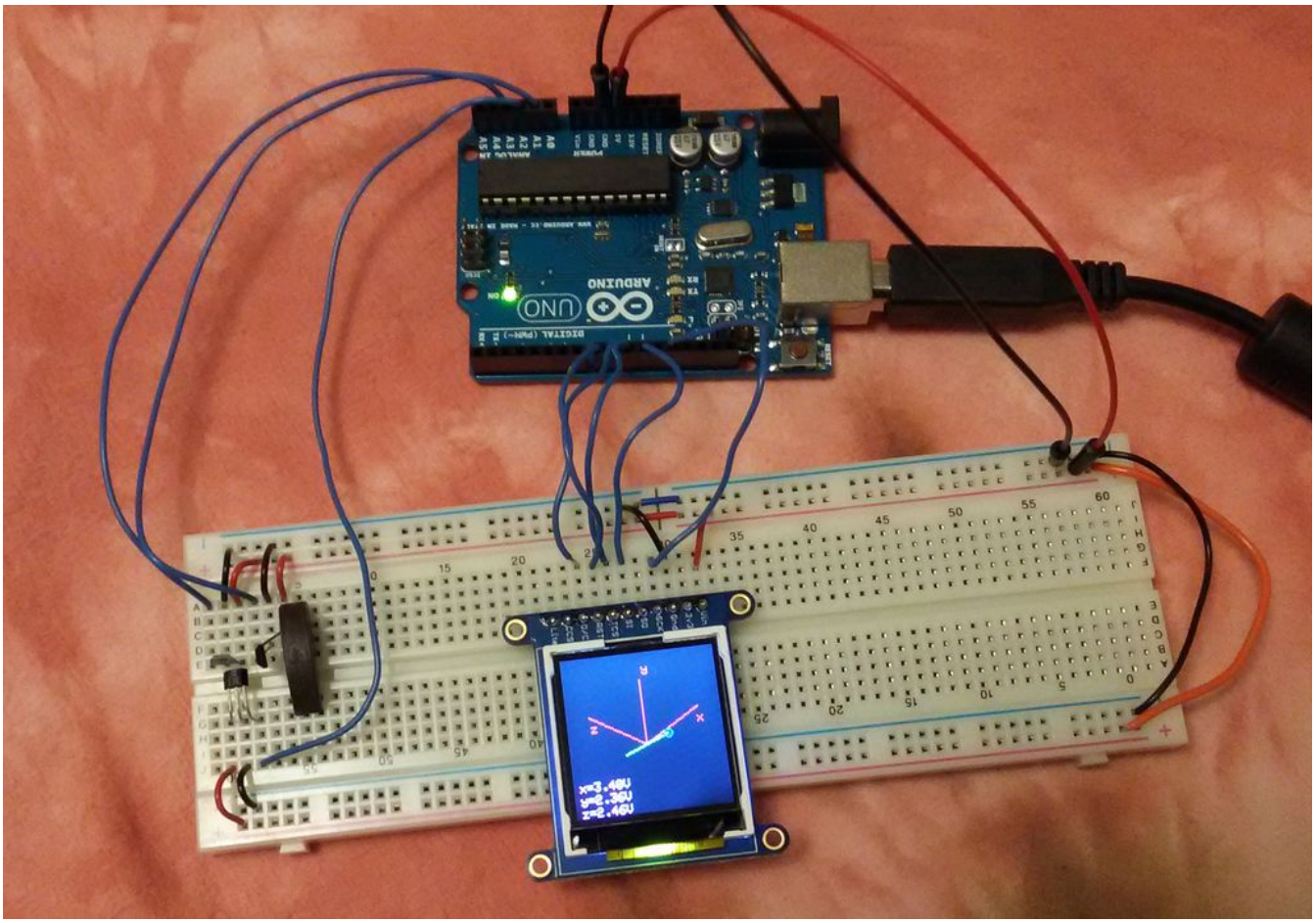


The display has 11 pins. Connect them to the Arduino Uno as follows. (NC means not connected.)

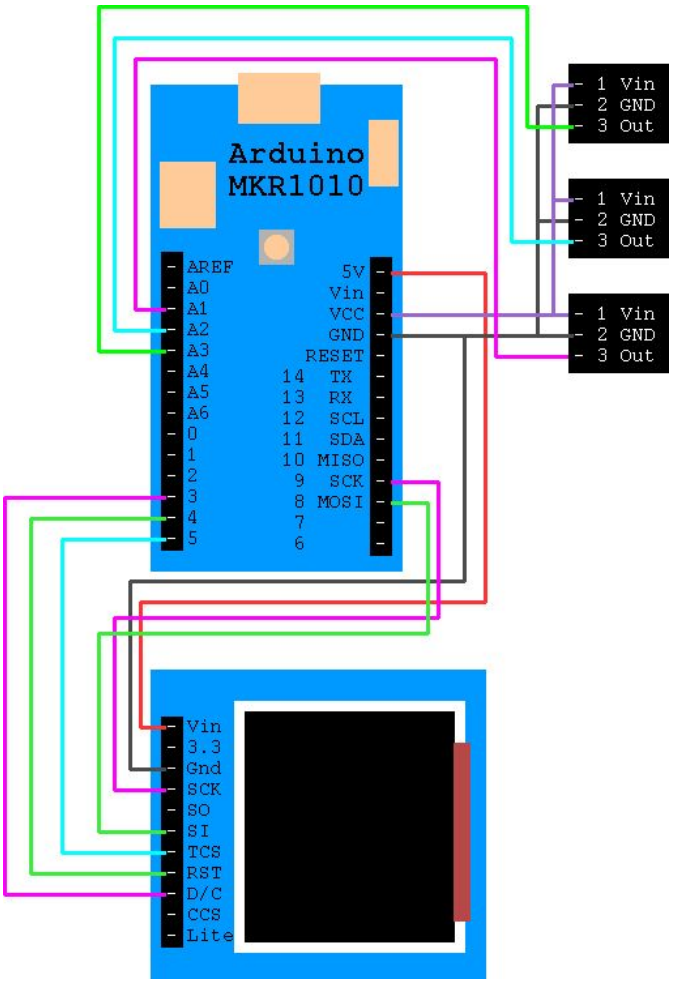
- Vin →5V
- 3.3 →NC
- Gnd →GND
- SCK →13
- SO → NC
- SI →11
- TCS →10
- RST →9
- D/C →8
- CCS → NC
- Lite → NC

Connect Vin of the sensors to 5V of the Arduino. Connect ground of the sensor to ground of the Arduino. Connect

the output of the sensors to analog inputs A1, A2, and A3 of the Arduino.



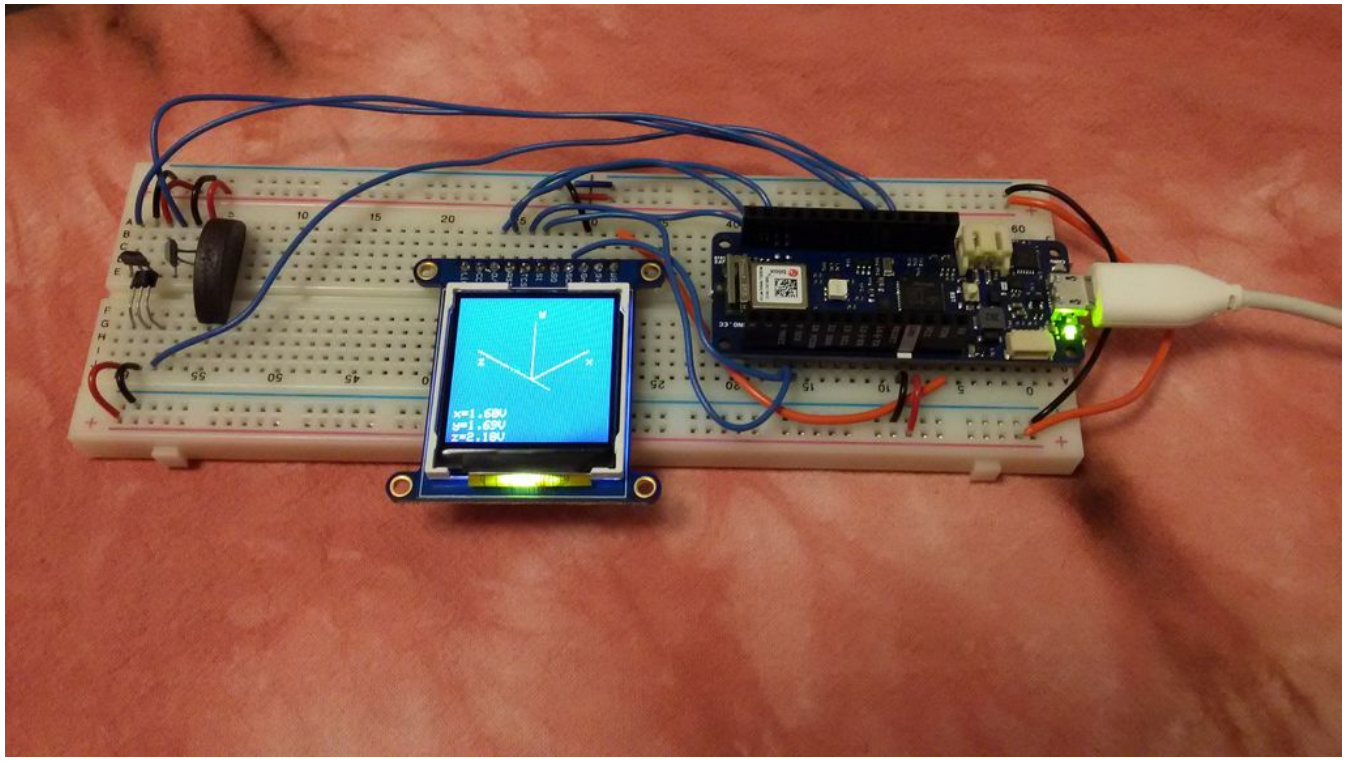
Wiring with the Arduino MKR1010

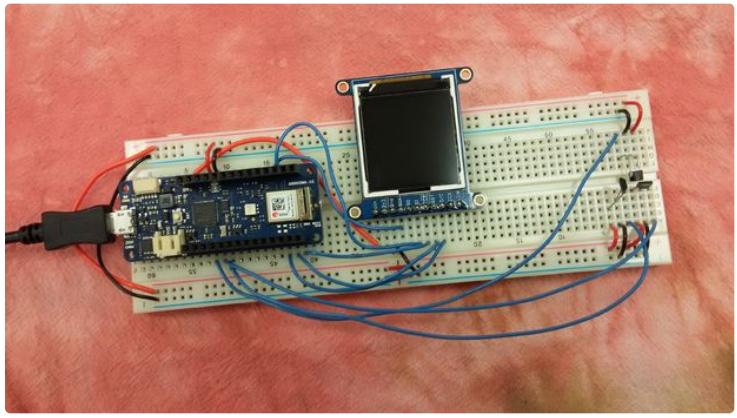


The display has 11 pins. Connect them to the Arduino as follows. (NC means not connected.)

- Vin →5V
- 3.3 →NC
- Gnd →GND
- SCK →SCK 9
- SO → NC
- SI →MOSI 8
- TCS →5
- RST →4
- D/C →3
- CCS → NC
- Lite → NC

Connect Vin of the sensors to Vcc of the Arduino. This pin is at 3.3V, not 5V. Connect ground of the sensor to ground of the Arduino. Connect the output of the sensors to analog inputs A1, A2, and A3 of the Arduino.


x=1.68V
y=1.69V
z=2.18V



Step 3: Test Out the Display

Let's get the TFT display working. Fortunately, Adafruit has some user friendly libraries and an excellent tutorial to go along with them. These instructions closely follow the tutorial, https://learn.adafruit.com/adafruit-1-44-color-tft-with-micro-sd-socket/overview.

Open the Arduino development environment. Go to Tools → Manage Libraries. Install the Adafruit_GFX, Adafruit_ZeroDMA, and Adafruit_ST7735 libraries. Restart the Android development environment.

The graphicstest example is included with the libraries. Open it. File → Examples → Adafruit ST7735 and ST7789 Library → graphicstest. To select the 1.44" display comment out line 95 and uncomment line 98.

Original version:

```
94 // Use this initializer if using a 1.8" TFT screen:
95 tft.initR(INITR_BLACKTAB);     //Init ST7735S chip, black tab
96
97 //OR use this initializer (uncomment) if using a 1.44" TFT:
98 //tft.initR(INITR_144GREENTAB); // Init ST7735R chip, green tab<br>
```

Correct version for 1.44" display:

```
94 // Use this initializer if using a 1.8" TFT screen:
95 //tft.initR(INIT_BLACKTAB);     //Init ST7735S chip, black tab
96
97 //OR use this initializer (uncomment) if using a 1.44" TFT:
98 tft.initR(INITR_144GREENTAB); //Init SST35R chip, green tab<br>
```

The display communicates using SPI, and different model Arduinos use different dedicated pins for some communication lines. The graphicstest example is set up to work with the Uno pins. If you are using the MKR1010, add the following lines between lines 80 and 81.

Corrections for the MKR1010:

```
80
   #define TFT_CS     5
   #define TFT_RST    4
   #define TFT_DC     3
   #define TFT_MOSI   8
   #define TFT_SCLK   9
   Adafruit_ST7735 tft=Adafruit_ST7735(TFT_CS, TFT_DC, TFT_MOSI, TFT_SCLK, TFT_RST);
81  float p=3.1415926;
```

Save the modified graphicstest example. Plug the Arduino into the computer if you have not yet done so. Go to Tools → Board and Tools → Port to verify that the computer can find the Arduino. Go to Sketch → Upload. If the example works, the display will show lines, rectangles, text, and the complete demo. The Adafruit tutorial provides more detail if troubleshooting is needed.

Step 4: The Magnetometer Code

Download the attached code, and open it in the Arduino development environment.

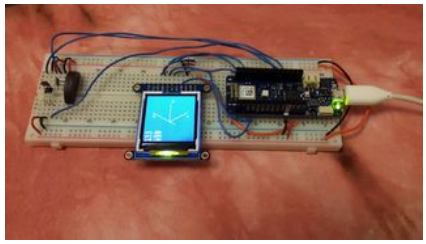
This program uses six functions:

- **Setup()** initializes the display.
- **Loop()** contains the main loop of the program. It blacks out the screen, draws the axes, reads the inputs, and draws the arrow representing the magnetic field vector. It has a refresh rate of one seconds which can be changed by altering line 127.
- **DrawAxes3d()** draws and labels the *x*, *y*, and *z* axes.
- **DrawArrow3d()** takes in an *x*, *y*, and *z* input ranging from 0 to 1023. From these values, it calculates the end points of the arrow in space. Next, it uses the isometricxx() and isometricyy() functions to calculate the end points on the screen. Finally, it draws the arrow and prints the voltages on the bottom of the screen.
- **Isometricxx()** finds the *x* coordinate of the isometric projection. It takes in *x*, *y*, and *z* coordinates of a point and returns the corresponding *x* pixel location on the screen.
- **Isometricyy()** finds the *y* coordinate of the isometric projection. It takes in *x*, *y*, and *z* coordinates of a point and returns the corresponding *y* pixel location on the screen.

Before running the code, we need to specify which pins to use for SPI communication with the display, and we need to specify the source voltage for the sensors. If you are using the MKR1010, comment out lines 92-96 as well as line 110. Then, uncomment lines 85-89 as well as line 108. If you are using the Uno, comment out lines 85-89 as well as line 108. Then, uncomment lines 92-96 as well as line 110.

Upload the code, Sketch → Upload.

You should see the x, y, and z axes in red. A green arrow with a blue circle for the tip represents the magnetic field vector at the sensors. Voltage readings are displayed on the lower left. As you bring a magnet closer to the sensors, the voltage readings should change, and the size of the arrow should grow.



https://www.instructabl... Download

Step 5: Future Work

The next step would be to calibrate the device.The sensor data sheet provides information on how to convert raw sensor voltage values to magnetic field strength. Calibration could be verified by comparing to a more accurate magnetometer.

Permanent magnets interact with current carrying wires. Wires near the display and in the Arduino generate magnetic fields which could affect sensor readings. Additionally, if this device is used to measure near a strong permanent magnet, the magnetic field from the device under test will interact with, introduce noise into, and possibly damage the Arduino and display. Shielding could make this magnetometer more robust. The Arduino can withstand larger magnetic fields if it is shielded in a metal box, and less noise will be introduced if shielded cables connect the sensors instead of bare wires.

Magnetic field is a function of position, so it is different at every point in space. This device uses three sensors, one to measure the *x*, the *y*, and the *z* component of the magnetic field at a point. The sensors are near to each other but not at a single point, and this limits the magnetometer resolution. It would be cool to save magnetic field readings at different points then display them as an array of arrows at the corresponding locations. However, that is a project for another day.

References

Information on Adafruit Arduino Graphics libraries

- https://learn.adafruit.com/adafruit-1-44-color-tft-with-micro-sd-socket/overview

Magnetic field visualization

- https://www.falstad.com/vector3dm/

Information on Hall effect and Hall effect sensors

- https://sensing.honeywell.com/index.php?ci_id=47847
- https://www.allegromicro.com/~/media/Files/Datasheets/A1324-5-6-Datasheet.ashx

Information on isometric projection

- https://en.wikipedia.org/wiki/3D_projection
- https://en.wikipedia.org/wiki/Isometric_projection