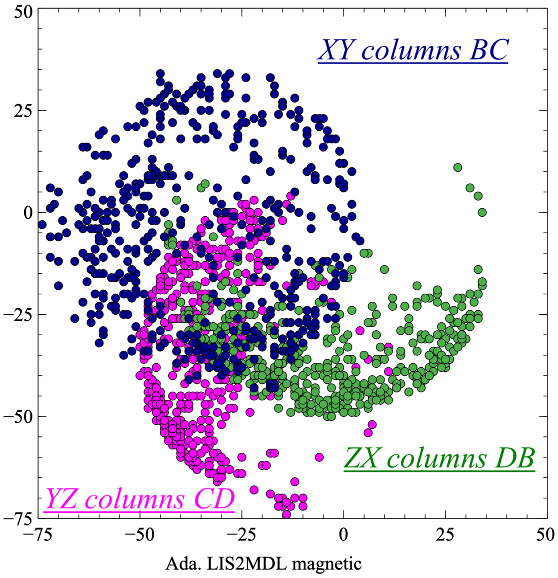
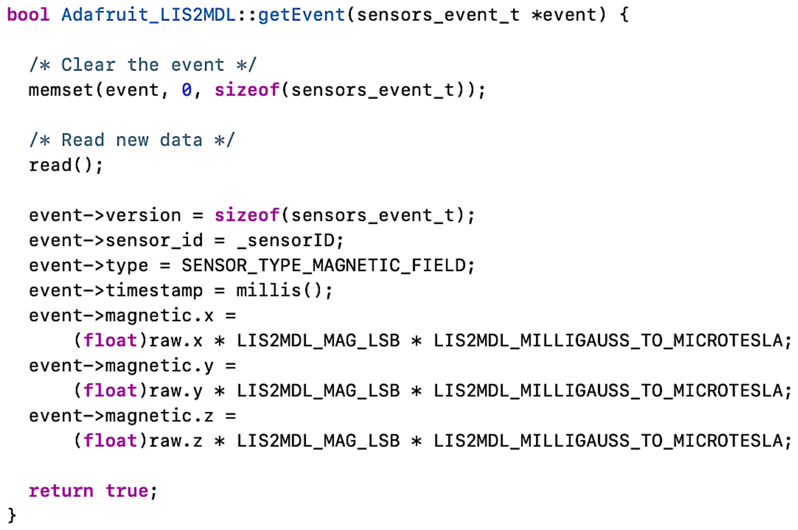
Compass Study

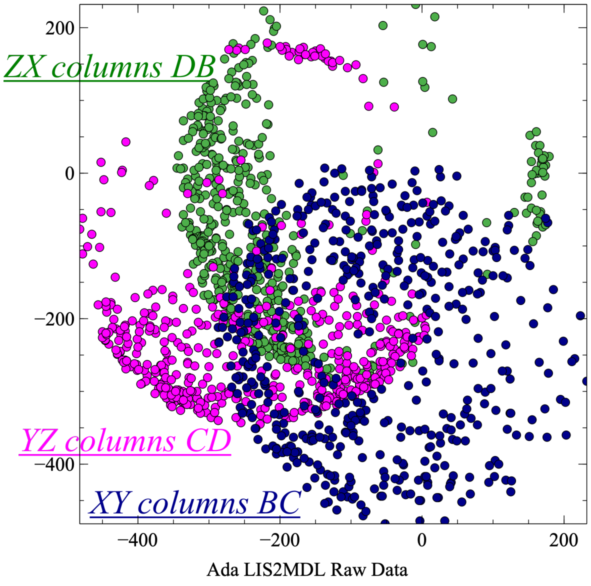
LIS2MDL



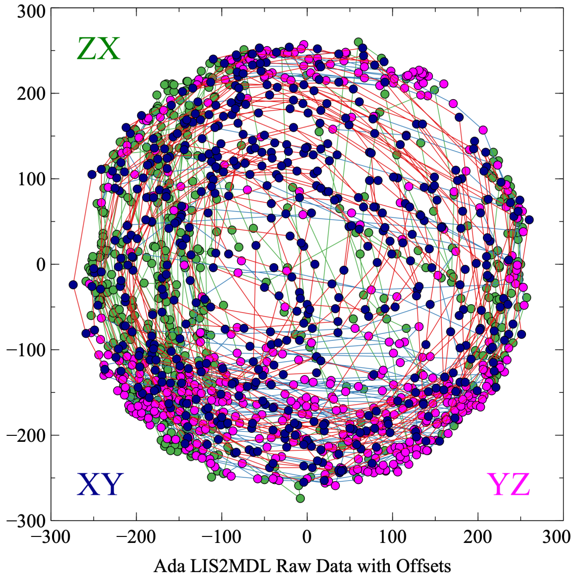


```
bool Adafruit_LIS2MDL::getEvent(sensors_event_t *event) {

  /* Clear the event */
  memset(event, 0, sizeof(sensors_event_t));

  /* Read new data */
  read();

  event->version = sizeof(sensors_event_t);
  event->sensor_id = _sensorID;
  event->type = SENSOR_TYPE_MAGNETIC_FIELD;
  event->timestamp = millis();
  event->magnetic.x =
      (float)raw.x * LIS2MDL_MAG_LSB * LIS2MDL_MILLIGAUSS_TO_MICROTESLA;
  event->magnetic.y =
      (float)raw.y * LIS2MDL_MAG_LSB * LIS2MDL_MILLIGAUSS_TO_MICROTESLA;
  event->magnetic.z =
      (float)raw.z * LIS2MDL_MAG_LSB * LIS2MDL_MILLIGAUSS_TO_MICROTESLA;

  return true;
}
```

The Adafruit LIS2MDL code *getEvent->magnetic.x, .y and .z* was used for the chart. The data comma separated values (csv) file has four columns, (A) tick counter, (B) magnetic.x, (C) magnetic.y, (D) magnetic.z. In the chart there are three data sets: (1) the xy-plane using column series B and C, (2) the yz-plane using column series C and D, and finally (3) the zx-plane using column series D and B. The property range x-axis and y-axis of the chart has equal spacings. Each data set seems to equate to an area of a circle. Each data set seems to be offset from the origin (0, 0). The raw values come from an buffer array loaded by the LIS2MDL output registers.



The next chart uses the *raw.x, .y and .z* data instead of the magnetic calculation to produce the similar four columns. Notice the positions of the colored sets when comparing the two charts. (TODO repeat experiment for repeatability.) If these colored sets are confirmed to be different, then it begs the question as to *why are they different*… Another property between the two charts, the latter is in millitesla and the former is in microtesla. Their *ranges are different*. Both the LIS2MDL magnetic and raw data sets failed the Magnetometer Accuracy Test which is explained at the Communication repository at MageMCU at Github.



```
// Raw Offsets (Use Calibration)
// -17, -221, -83

// Calculate axis with Offsets
// For offsets, use opposite sign (-/+)
XAxis = (float)(mag.raw.x + 17);
YAxis = (float)(mag.raw.y + 221);
// ZAxis = (float)(mag.raw.z + 83);

// Calculate the angle of the arctangent(y,x) (Degrees)
heading = atan2(YAxis, XAxis) * RAD_TO_DEG;
```

The chart here uses an offset for each axis obtained by using the calibration code. Notice how all three sets are centered about the origin (0, 0). The compass code using the offsets as shown in the snippet code the passed Magnetometer Accuracy Test.

Why must the xy-plane be centered about the origin in order to pass the Magnetometer Accuracy Test? It might have to do with the function arctangent() shown in the code snippet next to the third chart. Why were the initial values misaligned and why was it necessary to use offsets? Is there a physics (science) explanation for this behavior?

There is a lot of information about how to calibrate magnetometers on the internet and the study is only scratching the surface… **The compass study is work in progress…**

TOOLS USED FOR THE STUDY:

(1) Arduino Uno creates the comma separated values (CSV) by copy and paste the data into a text application with the extension (.csv).

(2) Veusz is a scientific plotting package. It is designed to be easy to use, easily extensible, but powerful. The program features a graphical user interface (GUI), which works under Unix/Linux, Windows or Mac OS. (FREE OPEN SOURCE) Source: **Github**: https://github.com/veusz/veusz **Download**: https://veusz.github.io/download/

Note: The order (or labeling) the 2D-planes was the right hand rule: XY, YZ, ZX.

Carpenter Software
Article Compass Study 20230510.
Edited 20230511…
by Jesse Carpenter