

//Currently in progress as of 06/09/2021

```
#include <I2Cdev.h>
#include <ArduinoBLE.h>
#include <SD.h>
#include <Arduino_LPS22HB.h>
#include <Arduino_LSM9DS1.h>
#include <SensorFusion.h>
#include <Scheduler.h>
float pressure = float(0);
int led1 = LEDR;
int led2 = LEDG;
int led3 = LEDB;
SF fusion;

BLEService customService ("180C");
BLEStringCharacteristic ble_pressure ("2A56", BLERead | BLENotify, 13);

void setup() {
// Setup the 3 pins as OUTPUT
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);

    // join I2C bus (I2Cdev library doesn't do this automatically)
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif
delay(1000);
Serial.begin(9600);
while (!Serial);

if (!BLE.begin()) {
    Serial.println("starting BLE failed!");
    while (1);
}

    BLE.setLocalName ("Alan's Datalogger");
    BLE.setAdvertisedService(customService);
    customService.addCharacteristic(ble_pressure);
    BLE.addService(customService);
//    ble_pressure.writeValue();
    BLE.advertise();
```

```

Serial.println("bluetooth is active");

delay(1000);
if (!IMU.begin()) {
  while (1);
}
delay(1000);
if (!BARO.begin()) {
  Serial.println("Failed to initialize pressure sensor!");
  while (1);
}
delay(1000);
if (!SD.begin(4)) {
  Serial.println("initialization failed!");
//   while (1); //Use this line to stop the code if the SD card isnt working
properly.
}
Serial.println("initialization done.");
delay(1000);
Scheduler.startLoop(blueetooth);

}

void loop() {
  float gx, gy, gz, ax, ay, az, mx, my, mz;
  float pitch, roll, yaw;
  float deltat;
  float pressure = BARO.readPressure();

  IMU.readGyroscope(gx, gy, gz);
  IMU.readAcceleration(ax, ay, az);
  IMU.readMagneticField(mx, my, mz);

//change variables to strings so they can be printed vis serial or onto SD card
easier
  String comma = String (',' );
  String accelXraw = String (ax);
  String accelYraw = String (ay);
  String accelZraw = String (az);
  String GyrXraw = String (gx);
  String GyrYraw = String (gy);
  String GyrZraw = String (gz);

```

```
String PitchS = String (pitch);
String RollS = String (roll);
String YawS = String (yaw);
String PressureS = String (pressure);
String alldata = (PitchS + comma + RollS + comma + YawS + comma + PressureS
+comma+ accelXraw + comma + accelYraw + comma + accelZraw + comma + GyrXraw + comma
+ GyrYraw + comma +GyrZraw );
```

```
digitalWrite (led1, LOW);
digitalWrite (led3, HIGH);
```

```
deltat = fusion.deltatUpdate(); //this have to be done before calling the fusion
update
```

```
//fusion.MahonyUpdate(gx * DEG_TO_RAD, gy * DEG_TO_RAD, gz * DEG_TO_RAD, ax, ay,
az, deltat); //mahony is suggested if there isn't the mag and the mcu is slow
fusion.MadgwickUpdate(gx * DEG_TO_RAD, gy * DEG_TO_RAD, gz * DEG_TO_RAD, ax, ay,
az, mx, my, mz, deltat); //else use the magwick, it is slower but more accurate
```

```
pitch = fusion.getPitch();
roll = fusion.getRoll(); //you could also use getRollRadians() ecc
yaw = fusion.getYaw();
```

```
File dataFile= SD.open("datalog.txt", FILE_WRITE);
```

```
if (dataFile) {
  dataFile.println(alldata);
  dataFile.close();
}
```

```
Serial.print("Pitch:\t"); Serial.println(pitch);
Serial.print("Roll:\t"); Serial.println(roll);
Serial.print("Yaw:\t"); Serial.println(yaw);
Serial.print(PressureS);
Serial.print('\t');
Serial.print(gx);
Serial.print('\t');
Serial.print(gy);
Serial.print('\t');
Serial.print(gz);
```

```

Serial.print('\t');
Serial.print(ax);
Serial.print('\t');
Serial.print(ay);
Serial.print('\t');
Serial.print(az);
Serial.print('\t');
Serial.print(mx);
Serial.print('\t');
Serial.print(my);
Serial.print('\t');
Serial.print(mz);
Serial.println('\t');

digitalWrite(led1, HIGH);
digitalWrite(led1, LOW);

}

void loop2() {
  digitalWrite(led1, HIGH);
}

void bluetooth() {
  float pressure = BARO.readPressure();
  String PressureS = String(pressure);
// wait for a BLE central
  BLEDevice central = BLE.central();
// if a central is connected to the peripheral:
  if (central) {
    Serial.print("Connected to central: ");
    // print the central's BT address:
    Serial.println(central.address());
    // turn on the LED to indicate the connection:
    digitalWrite(LED_BUILTIN, HIGH);

    // check the battery level every 200ms
    // while the central is connected:
    while (central.connected()) {
      ble_pressure.writeValue(PressureS);
    }
  }
}
}

```