

/*

Advanced_I2C.ino

Brian R Taylor

brian.taylor@bolderflight.com

Copyright (c) 2017 Bolder Flight Systems

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*/

#include "MPU9250v2.h"

#include <SD.h>

```

// an MPU9250 object with the MPU-9250 sensor on I2C bus
0 with address 0x68
MPU9250 IMU(Wire,0x68);
int status;
const int chipSelect = 4;

void setup() {
  // serial to display data
  Serial.begin(115200);
  SD.begin(chipSelect);

  // while(!Serial) {}

  // start communication with IMU
  status = IMU.begin();

  // setting the accelerometer full scale range to +/-8G
  IMU.setAccelRange(MPU9250::ACCEL_RANGE_8G);
  // setting the gyroscope full scale range to +/-500
deg/s
  IMU.setGyroRange(MPU9250::GYRO_RANGE_500DPS);
  // setting DLPF bandwidth to 20 Hz
  IMU.setDlpfBandwidth(MPU9250::DLPF_BANDWIDTH_460HZ);
  IMU.setSrd(0);
}

void loop() {

  IMU.readSensor();
  String dataString = "";
  dataString += String(IMU.getAccelX_mss(),2);
  dataString += ",";
  dataString += String(IMU.getAccelY_mss(),2);

```

```
dataString += ",";
dataString += String(IMU.getAccelZ_mss(),2);
```

```
File dataFile = SD.open("datalog.txt", FILE_WRITE);
dataFile.println(dataString);
dataFile.close();
Serial.println(dataString);
```

```
/*
// read the sensor
IMU.readSensor();
```

```
// display the data
```

```
Serial.print(IMU.getAccelX_mss(),2);
Serial.print("\t");
Serial.print(IMU.getAccelY_mss(),2);
Serial.print("\t");
Serial.print(IMU.getAccelZ_mss(),2);
Serial.println("");
*/
```

```
delay(0);
```

```
}
```