|  |
| --- |
| **2024 ALTIS SW 개인 활동 보고서** |

|  |  |
| --- | --- |
| **활동개요** | esp32 보드의 내장 블루투스 이용해서 bmp mpu6050 데이터를 핸드폰이나 노트북으로 받아보기 |
| **일시** | 2024년 05월 05일 |
| **작성자** | 박상하 |
| **활동 내용** | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    This is a library for the BMP280 humidity, temperature & pressure sensor    Designed specifically to work with the Adafruit BMP280 Breakout    ----> http://www.adafruit.com/products/2651    These sensors use I2C or SPI to communicate, 2 or 4 pins are required    to interface.    Adafruit invests time and resources providing this open source code,    please support Adafruit andopen-source hardware by purchasing products    from Adafruit!    Written by Limor Fried & Kevin Townsend for Adafruit Industries.    BSD license, all text above must be included in any redistribution   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <Wire.h>  #include <SPI.h>  #include <Adafruit\_BMP280.h>  #define BMP\_SCK  (13)  #define BMP\_MISO (12)  #define BMP\_MOSI (11)  #define BMP\_CS   (10)  Adafruit\_BMP280 bmp; // I2C  //Adafruit\_BMP280 bmp(BMP\_CS); // hardware SPI  //Adafruit\_BMP280 bmp(BMP\_CS, BMP\_MOSI, BMP\_MISO,  BMP\_SCK);  void setup() {    Serial.begin(9600);    while ( !Serial ) delay(100);   // wait for native usb    Serial.println(F("BMP280 test"));    unsigned status;  //  status = bmp.begin(BMP280\_ADDRESS\_ALT, BMP280\_CHIPID);    status = bmp.begin(0x76);    if (!status) {      Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "                        "try a different address!"));      Serial.print("SensorID was: 0x"); Serial.println(bmp.sensorID(),16);      Serial.print("        ID of 0xFF probably means a bad address, a BMP 180 or BMP 085\n");      Serial.print("   ID of 0x56-0x58 represents a BMP 280,\n");      Serial.print("        ID of 0x60 represents a BME 280.\n");      Serial.print("        ID of 0x61 represents a BME 680.\n");      while (1) delay(10);    }    /\* Default settings from datasheet. \*/   bmp.setSampling(Adafruit\_BMP280::MODE\_NORMAL,     /\* Operating Mode. \*/                   Adafruit\_BMP280::SAMPLING\_X2,    /\* Temp. oversampling \*/                   Adafruit\_BMP280::SAMPLING\_X16,   /\* Pressure oversampling \*/                   Adafruit\_BMP280::FILTER\_X16,     /\* Filtering. \*/                   Adafruit\_BMP280::STANDBY\_MS\_500); /\* Standby time. \*/  }  void loop() {      Serial.print(F("Temperature = "));      Serial.print(bmp.readTemperature());      Serial.println(" \*C");      Serial.print(F("Pressure = "));      Serial.print(bmp.readPressure()/100.0F);      Serial.println(" Pa");      Serial.print(F("Approx altitude = "));      Serial.print(bmp.readAltitude(1013.25)); /\* Adjusted to local forecast! \*/      Serial.println(" m");      Serial.println();      delay(2000);  }    (캡쳐를 잘못해서 일부가 짤렸습니다.)    // Basic demo for accelerometer readings from Adafruit MPU6050  #include <Adafruit\_MPU6050.h>  #include <Adafruit\_Sensor.h>  #include <Wire.h>  Adafruit\_MPU6050 mpu;  void setup(void) {    Serial.begin(115200);    while (!Serial)      delay(10); // will pause Zero, Leonardo, etc until serial console opens    Serial.println("Adafruit MPU6050 test!");    // Try to initialize!    if (!mpu.begin()) {      Serial.println("Failed to find MPU6050 chip");      while (1) {        delay(10);      }    }    Serial.println("MPU6050 Found!");    mpu.setAccelerometerRange(MPU6050\_RANGE\_8\_G);    Serial.print("Accelerometer range set to: ");    switch (mpu.getAccelerometerRange()) {    case MPU6050\_RANGE\_2\_G:      Serial.println("+-2G");      break;    case MPU6050\_RANGE\_4\_G:      Serial.println("+-4G");      break;    case MPU6050\_RANGE\_8\_G:      Serial.println("+-8G");      break;    case MPU6050\_RANGE\_16\_G:      Serial.println("+-16G");      break;    }    mpu.setGyroRange(MPU6050\_RANGE\_500\_DEG);    Serial.print("Gyro range set to: ");    switch (mpu.getGyroRange()) {    case MPU6050\_RANGE\_250\_DEG:      Serial.println("+- 250 deg/s");      break;    case MPU6050\_RANGE\_500\_DEG:      Serial.println("+- 500 deg/s");      break;    case MPU6050\_RANGE\_1000\_DEG:      Serial.println("+- 1000 deg/s");      break;    case MPU6050\_RANGE\_2000\_DEG:      Serial.println("+- 2000 deg/s");      break;    }    mpu.setFilterBandwidth(MPU6050\_BAND\_21\_HZ);    Serial.print("Filter bandwidth set to: ");    switch (mpu.getFilterBandwidth()) {    case MPU6050\_BAND\_260\_HZ:      Serial.println("260 Hz");      break;    case MPU6050\_BAND\_184\_HZ:      Serial.println("184 Hz");      break;    case MPU6050\_BAND\_94\_HZ:      Serial.println("94 Hz");      break;    case MPU6050\_BAND\_44\_HZ:      Serial.println("44 Hz");      break;    case MPU6050\_BAND\_21\_HZ:      Serial.println("21 Hz");      break;    case MPU6050\_BAND\_10\_HZ:      Serial.println("10 Hz");      break;    case MPU6050\_BAND\_5\_HZ:      Serial.println("5 Hz");      break;    }    Serial.println("");    delay(100);  }  void loop() {    /\* Get new sensor events with the readings \*/    sensors\_event\_t a, g, temp;    mpu.getEvent(&a, &g, &temp);    /\* Print out the values \*/    Serial.print("Acceleration X: ");    Serial.print(a.acceleration.x);    Serial.print(", Y: ");    Serial.print(a.acceleration.y);    Serial.print(", Z: ");    Serial.print(a.acceleration.z);    Serial.println(" m/s^2");    Serial.print("Rotation X: ");    Serial.print(g.gyro.x);    Serial.print(", Y: ");    Serial.print(g.gyro.y);    Serial.print(", Z: ");    Serial.print(g.gyro.z);    Serial.println(" rad/s");    Serial.print("Temperature: ");    Serial.print(temp.temperature);    Serial.println(" degC");    Serial.println("");    delay(500);  }    //블루투스 통신으로 출력하기  #include <BluetoothSerial.h>  #include <Adafruit\_MPU6050.h>  #include <Adafruit\_Sensor.h>  #include <Wire.h>  #include "BluetoothSerial.h"  const char \*pin = "1234"; // Change this to more secure PIN.  String device\_name = "ESP32-BT-Slave";  #if !defined(CONFIG\_BT\_ENABLED) || !defined(CONFIG\_BLUEDROID\_ENABLED)  #error Bluetooth is not enabled! Please run `make menuconfig` to and enable it  #endif  #if !defined(CONFIG\_BT\_SPP\_ENABLED)  #error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.  #endif  BluetoothSerial SerialBT;  Adafruit\_MPU6050 mpu;  void setup(void) {    Serial.begin(115200);    SerialBT.begin("ESP32\_test");    Serial.println("connect");    #ifdef USE\_PIN      SerialBT.setPin(pin);      Serial.println("Using PIN");    #endif    while(!Serial){      delay(10);    }    Serial.println("MPU Gyro range");    if(!mpu.begin()){      Serial.println("We can't find MPU");      while(1){        delay(10);      }    }    Serial.println("We found MPU");      mpu.setAccelerometerRange(MPU6050\_RANGE\_8\_G);    Serial.print("Accelerometer range set to: ");    switch (mpu.getAccelerometerRange()) {    case MPU6050\_RANGE\_2\_G:      Serial.println("+-2G");      break;    case MPU6050\_RANGE\_4\_G:      Serial.println("+-4G");      break;    case MPU6050\_RANGE\_8\_G:      Serial.println("+-8G");      break;    case MPU6050\_RANGE\_16\_G:      Serial.println("+-16G");      break;    }      mpu.setGyroRange(MPU6050\_RANGE\_500\_DEG);    Serial.print("Gyro range set to: ");    switch (mpu.getGyroRange()) {    case MPU6050\_RANGE\_250\_DEG:      Serial.println("+- 250 deg/s");      break;    case MPU6050\_RANGE\_500\_DEG:      Serial.println("+- 500 deg/s");      break;    case MPU6050\_RANGE\_1000\_DEG:      Serial.println("+- 1000 deg/s");      break;    case MPU6050\_RANGE\_2000\_DEG:      Serial.println("+- 2000 deg/s");      break;    }      mpu.setFilterBandwidth(MPU6050\_BAND\_21\_HZ);    Serial.print("Filter bandwidth set to: ");      switch (mpu.getFilterBandwidth()) {    case MPU6050\_BAND\_260\_HZ:    Serial.println("260 Hz");      break;    case MPU6050\_BAND\_184\_HZ:      Serial.println("184 Hz");      break;    case MPU6050\_BAND\_94\_HZ:      Serial.println("94 Hz");      break;    case MPU6050\_BAND\_44\_HZ:      Serial.println("44 Hz");      break;    case MPU6050\_BAND\_21\_HZ:      Serial.println("21 Hz");      break;    case MPU6050\_BAND\_10\_HZ:      Serial.println("10 Hz");      break;    case MPU6050\_BAND\_5\_HZ:      Serial.println("5 Hz");      break;    }    Serial.println("");    delay(100);  }  void loop(void) {    /\* Get new sensor events with the readings \*/    sensors\_event\_t a, g, temp;    mpu.getEvent(&a, &g, &temp);    /\* Print out the values \*/    SerialBT.print("Acceleration X: ");    SerialBT.print(a.acceleration.x);    SerialBT.print(", Y: ");    SerialBT.print(a.acceleration.y);    SerialBT.print(", Z: ");    SerialBT.print(a.acceleration.z);    SerialBT.println(" m/s^2");    SerialBT.print("Rotation X: ");    SerialBT.print(g.gyro.x);    SerialBT.print(", Y: ");    SerialBT.print(g.gyro.y);    SerialBT.print(", Z: ");    SerialBT.print(g.gyro.z);    SerialBT.println(" rad/s");    SerialBT.print("Temperature: ");    SerialBT.print(temp.temperature);    SerialBT.println(" degC");    SerialBT.println("");    delay(500);  }  이후 블루투스로 받아보려고 했으나, 앱이 튕기는 현상이 발생해서 마지막 중요한 부분을 받을 수 없었습니다. |
| **활동사진** | 위 코드 결과 사진들로 대체하겠습니다. |
| **활동결과** | Bmp Mpu6050과 esp32 블루투스 연결은 완료하였으나, 앱이 무슨 이유인지 모르겠으나 계속 튕겨서 마지막 확인을 제대로 하지 못했습니다. |
| **계획** | 폰으로 튕김현상 없이 제대로 받기!! |

2024년 05월 05일

작성자 : 박상하 (인)