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

|  |  |
| --- | --- |
| **활동 개요** | esp32 보드의 내장 블루투스 이용해서 bmp mpu6050 데이터를 핸드폰이나 노트북으로 받아보기 |
| **일시** | 2024년 05월 11일 |
| **작성자** | 양주호 |
| **활동 내용**  **활동 사진** | 1. mpu6050    코드  #include "I2Cdev.h"  #include "MPU6050.h"  MPU6050 accelgyro;  int16\_t ax, ay, az;  int16\_t gx, gy, gz;  void setup()  {  Wire.begin();  Serial.begin(115200);  accelgyro.initialize();  }  void loop()  {  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);  Serial.print(ax); Serial.print(","); // x 방향 기울기 성분 ( 중력 가속도 성분)  Serial.print(ay); Serial.print(","); // y 방향 기울기 성분 ( 중력 가속도 성분)  Serial.print(az); Serial.print(",");  Serial.print(gx); Serial.print(",");  Serial.print(gy); Serial.print(",");  Serial.print(gz);  Serial.println("");  }  결과    2. bmp    코드  #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());  Serial.println(" Pa");  Serial.print(F("Approx altitude = "));  Serial.print(bmp.readAltitude(1011.9)); /\* Adjusted to local forecast! \*/  Serial.println(" m");  Serial.println();  delay(200);  }  결과    3. 블루투스 연결 후 컴퓨터로 데이터 받기  #include <BluetoothSerial.h>  #include <Adafruit\_MPU6050.h>  #include <Adafruit\_Sensor.h>  #include <Wire.h>  #if !defined(CONFIG\_BT\_ENABLED) || !defined(CONFIG\_BLUEDROID\_ENABLED)  #error Bluetooth is not enabled! Please run `make menuconfig` to and enable it  #endif    BluetoothSerial SerialBT;  Adafruit\_MPU6050 mpu;  void setup(void) {  Serial.begin(115200);  SerialBT.begin("ESP32\_test");  Serial.println("connect");  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) {  if(SerialBT.available()){  if(SerialBT.read() == 'S'){  while(1){  sensors\_event\_t a, g, temp;  mpu.getEvent(&a, &g, &temp);  if(SerialBT.read() == 'P'){  SerialBT.println("stop");  break;  }  SerialBT.print("X: ");  SerialBT.print(g.gyro.x + 0.02);  SerialBT.print(", Y: ");  SerialBT.print(g.gyro.y - 0.02);  SerialBT.print(", Z: ");  SerialBT.print(g.gyro.z + 0.01);  SerialBT.println();  delay(500);  }  }  }  } |
| **활동 결과** | 각 센서들로 데이터를 받는 데 성공했지만 컴퓨터와 블루투스 연결이 안되어 블루투스 통신으로 데이터를 받는 데 실패함 |
| **계획** | 블루투스 연결 계속 해보겠음 |

2024년 5월 11일

작성자 : 양주호 (인)