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

|  |  |
| --- | --- |
| **활동 개요** | esp32 보드의 내장 블루투스 이용해서 bmp mpu6050 데이터를 핸드폰이나 노트북으로 받아보기 |
| **일시** | 2024년 04월 11일 |
| **작성자** | 김민주 |
| **활동 내용** | - bmp280  bmp280은 칩 기반의 대기압, 고도, 온도, 습도를 측정해주는 센서이다.   |  | | --- | | 대기압, 고도, 온도, 습도를 나타내는 소스코드 #I2C 통신 | | #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(115200);      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(1013.25)); /\* Adjusted to local forecast! \*/      Serial.println(" m");      Serial.println();      delay(2000);  } |   - mpu6050  mpu6050 은 6축 자이로 가속도 센서이다.   |  | | --- | | 자이로 가속도 값을 나타내는 소스코드 (필터까지 이용해서 각을 추정) #I2C 통신 | | #include<Wire.h>  const int MPU\_ADDR = 0x68;  int16\_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;  double angleAcX, angleAcY, angleAcZ;  double angleGyX, angleGyY, angleGyZ;  double angleFiX, angleFiY, angleFiZ;  const double RADIAN\_TO\_DEGREE = 180 / 3.14159;  const double DEGREE\_PER\_SECOND = 32767 / 250;  const double ALPHA = 1 / (1 + 0.04);  unsigned long now = 0;  unsigned long past = 0;  double dt = 0;  double baseAcX, baseAcY, baseAcZ;  double baseGyX, baseGyY, baseGyZ;  void setup() {    initSensor();    Serial.begin(115200);    calibrateSensor();    past = millis();  }  void loop() {    getData();    getDT();    angleAcY = atan(-AcX / sqrt(pow(AcY, 2) + pow(AcZ, 2)));    angleAcY \*= RADIAN\_TO\_DEGREE;    angleGyY = ((GyY - baseGyY) / DEGREE\_PER\_SECOND) \* dt;    double angleTmp = angleFiY + angleGyY;    angleFiY = ALPHA \* angleTmp + (1.0 - ALPHA) \* angleAcY;      Serial.println(angleFiY);  }  void initSensor() {    Wire.begin();    Wire.beginTransmission(MPU\_ADDR);    Wire.write(0x6B);    Wire.write(0);    Wire.endTransmission(true);  }  void getData() {    Wire.beginTransmission(MPU\_ADDR);    Wire.write(0x3B);    Wire.endTransmission(false);    Wire.requestFrom(MPU\_ADDR, 14, true);    AcX = Wire.read() << 8 | Wire.read();    AcY = Wire.read() << 8 | Wire.read();    AcZ = Wire.read() << 8 | Wire.read();    Tmp = Wire.read() << 8 | Wire.read();    GyX = Wire.read() << 8 | Wire.read();    GyY = Wire.read() << 8 | Wire.read();    GyZ = Wire.read() << 8 | Wire.read();  }  void getDT() {    now = millis();    dt = (now - past) / 1000.0;    past = now;  }  void calibrateSensor() {    double sumAcX = 0, sumAcY = 0, sumAcZ = 0;    double sumGyX = 0, sumGyY = 0, sumGyZ = 0;    getData();    for (int i = 0; i < 10 ; i++) {      getData();      sumAcX += AcX; sumAcY += AcY; sumAcZ += AcZ;      sumGyX += GyX; sumGyY += GyY; sumGyZ += GyZ;      delay(100);    }    baseAcX = sumAcX / 10;    baseAcY = sumAcY / 10;    baseAcZ = sumAcZ / 10;    baseGyX = sumGyX / 10;    baseGyY = sumGyY / 10;    baseGyZ = sumGyZ / 10;  } |   - esp32  마이크로컨트롤러이다. 와이파이 및 블루투스 무선 기능과 듀얼 코어 프로세서를 포함하고 있다.   |  | | --- | | 다른 기기와 블루투스 통신을 주고 받는 소스코드 # mpu6050: I2C 통신 , bmp280: SPI 통신 | | #include <BluetoothSerial.h>  BluetoothSerial SerialBT;    void setup() {    // put your setup code here, to run once:    Serial.begin(115200);    SerialBT.begin("EPS32test");    Serial.println("The device started, now you can pair it with bluetooth!");  }  void loop() {    // put your main code here, to run repeatedly:    if ( Serial.available()){      SerialBT.write(Serial.read());    }    if ( SerialBT.available()){      Serial.write(SerialBT.read());    }    delay(20);  } |   - esp32 보드의 내장 블루투스 이용해서 bmp280 mpu6050 데이터를 핸드폰으로 받기   |  | | --- | | 소스코드 | | #include <Wire.h>  #include <SPI.h>  #include <Adafruit\_BMP280.h>  #include <BluetoothSerial.h>  #define BMP\_SCK  (18)  #define BMP\_MISO (19)  #define BMP\_MOSI (23)  #define BMP\_CS   (5)  //Adafruit\_BMP280 bmp; // I2C  //Adafruit\_BMP280 bmp(BMP\_CS); // hardware SPI  Adafruit\_BMP280 bmp(BMP\_CS, BMP\_MOSI, BMP\_MISO,  BMP\_SCK);  BluetoothSerial SerialBT;  const int MPU\_ADDR = 0x68;  int16\_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;  double angleAcX, angleAcY, angleAcZ;  double angleGyX, angleGyY, angleGyZ;  double angleFiX, angleFiY, angleFiZ;  const double RADIAN\_TO\_DEGREE = 180 / 3.14159;  const double DEGREE\_PER\_SECOND = 32767 / 250;  const double ALPHA = 1 / (1 + 0.04);  unsigned long now = 0;  unsigned long past = 0;  double dt = 0;  double baseAcX, baseAcY, baseAcZ;  double baseGyX, baseGyY, baseGyZ;  void setup() {    initSensor();    Serial.begin(115200);    calibrateSensor();    past = millis();      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. \*/       SerialBT.begin("EPS32test");   Serial.println("The device started, now you can pair it with bluetooth!");  }  void loop() {    getData();    getDT();      angleAcY = atan(-AcX / sqrt(pow(AcY, 2) + pow(AcZ, 2)));    angleAcY \*= RADIAN\_TO\_DEGREE;    angleGyY = ((GyY - baseGyY) / DEGREE\_PER\_SECOND) \* dt;    double angleTmp = angleFiY + angleGyY;    angleFiY = ALPHA \* angleTmp + (1.0 - ALPHA) \* angleAcY;      SerialBT.println("=================");    SerialBT.print(F("Temperature = "));    SerialBT.print(bmp.readTemperature());    SerialBT.println(" \*C");    SerialBT.print(F("Pressure = "));    SerialBT.print(bmp.readPressure());    SerialBT.println(" Pa");    SerialBT.print(F("Approx altitude = "));    SerialBT.print(bmp.readAltitude(1013.25)); /\* Adjusted to local forecast! \*/    SerialBT.println(" m");      SerialBT.print(F("Angle = "));    SerialBT.println(angleFiY);    SerialBT.println("=================");    delay(200);    if ( Serial.available()){      SerialBT.write(Serial.read());      delay(200);    }    if ( SerialBT.available()){      Serial.write(SerialBT.read());      delay(200);    }    delay(2000);  }  void initSensor() {    Wire.begin();    Wire.beginTransmission(MPU\_ADDR);    Wire.write(0x6B);    Wire.write(0);    Wire.endTransmission(true);  }  void getData() {    Wire.beginTransmission(MPU\_ADDR);    Wire.write(0x3B);    Wire.endTransmission(false);    Wire.requestFrom(MPU\_ADDR, 14, true);    AcX = Wire.read() << 8 | Wire.read();    AcY = Wire.read() << 8 | Wire.read();    AcZ = Wire.read() << 8 | Wire.read();    Tmp = Wire.read() << 8 | Wire.read();    GyX = Wire.read() << 8 | Wire.read();    GyY = Wire.read() << 8 | Wire.read();    GyZ = Wire.read() << 8 | Wire.read();  }  void getDT() {    now = millis();    dt = (now - past) / 1000.0;    past = now;  }  void calibrateSensor() {    double sumAcX = 0, sumAcY = 0, sumAcZ = 0;    double sumGyX = 0, sumGyY = 0, sumGyZ = 0;    getData();    for (int i = 0; i < 10 ; i++) {      getData();      sumAcX += AcX; sumAcY += AcY; sumAcZ += AcZ;      sumGyX += GyX; sumGyY += GyY; sumGyZ += GyZ;      delay(100);    }    baseAcX = sumAcX / 10;    baseAcY = sumAcY / 10;    baseAcZ = sumAcZ / 10;    baseGyX = sumGyX / 10;    baseGyY = sumGyY / 10;    baseGyZ = sumGyZ / 10;  } | |
| **활동 사진** | - esp32 보드의 내장 블루투스 이용해서 bmp280 mpu6050 데이터를 핸드폰으로 받기  텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진  자동 생성된 설명 텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진  자동 생성된 설명  전자제품, 전자 공학, 텍스트, 케이블이(가) 표시된 사진  자동 생성된 설명 |
| **활동 결과** | Eps32를 통해 주변기기와 통신할 수 있다. bmp280과 mpu6050을 통해 각도, 온도, 습도, 대기압, 고도 등을 파악할 수 있다. 이러한 데이터를 다른 기기에게 전송할 수 있다. |
| **참고 문헌** | <https://yepb.tistory.com/14>  <https://blog.naver.com/darknisia/221168423562>  <https://m.blog.naver.com/eduino/222066669634>  <https://gamjathebest.tistory.com/7>  <https://arsviator.blogspot.com/2018/11/esp32.html>  <https://blog.naver.com/darknisia/222861214925> |

2024년 05월 05일

작성자 : 김민주 (인)