*※보고서 제목 양식 예 : 이름\_20240403*

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

|  |  |
| --- | --- |
| **활동 개요** | esp32,bmp280,mpu6050,을 이용해서 블루투스로 데이터 받기 |
| **일시** | 2024년 5월 5일 14시 |
| **작성자** | 문시경 |
| **활동 내용** | 이번 활동은 ESP32모듈과 BMP280, MPU6050을 이용하여 환경 데이터와 운동 데이터를 측정하고, 이를 블루투스를 통해 무선으로 송수신하는 시스템을 구축하는 것을 목표로 한 것이다.  사용한 하드웨어들의 기능과 특징은 다음과 같다.  ESP32 : Bluetooth 및 Wi-Fi 기능을 내장한 저전력 시스템 온 칩(SoC)으로, 다양한 입출력 기능을 지원한다. 각종 센서와의 손쉬운 통신을 위해 다수의 GPIO 핀을 제공하며, 블루투스와 같은 무선 통신 모듈이 내장되어 있어 모바일 기기나 다른 네트워크 장비와의 연결이 가능하다.  BMP280 : 온도와 기압을 측정할 수 있는 디지털 환경 센서이다. I2C 또는 SPI 통신을 통해 마이크로컨트롤러에 연결된다. 기압 데이터를 활용하여 상대적 고도를 추정할 수 있기 때문에 드론, 스마트워치 및 기상 관측 장비에서 사용된다.  MPU6050 : 가속도계와 자이로스코프 기능을 통합한 센서로, 3축 가속도와 3축 각속도를 측정할 수 있다. 운동 추적, 자세 제어 등 다양한 애플리케이션에서 활용되며 I2C 인터페이스를 통해 데이터를 전송한다.  이러한 기기들이 사용되는 과정은 다음과 같다.    ESP32는 I2C 통신을 사용하여 BMP280과 MPU6050으로부터 데이터를 주기적으로 읽어온다. BMP280에서는 온도와 기압 데이터를, MPU6050에서는 X, Y, Z 축에 대한 가속도와 각속도 데이터를 수집한다. 이후 수집된 데이터는 ESP32에서 초기 처리를 거쳐 사용 가능한 형태로 변환된다. 이 활동의 궁극적 목적이 위성에 탑재하는 것임을 고려한다면, 기압 데이터는 고도로, 가속도 데이터는 움직임의 정도로 해석될 수 있다. 처리된 데이터는 내장된 블루투스 모듈을 통해 주변의 블루투스 지원 장치에게 전송된다. 이 데이터는 스마트폰 앱, PC 소프트웨어 또는 다른 ESP32 기기에서 수신될 수 있다.  이제 기기들을 연결하여 직접 데이터를 받아보기로 하자. 아래는 데이터를 받기 위한 코드이다.  #include<Wire.h>  #include<Adafruit\_Sensor.h>  #include<Adafruit\_BMP280.h>  #include<MPU6050\_light.h>  #include<BluetoothSerial.h>  BluetoothSerial SerialBT;  Adafruit\_BMP280 bmp;  MPU6050 mpu(Wire);  voidsetup(){  Serial.begin(115200);  SerialBT.begin("ESP32\_Bluetooth");// Bluetooth 디바이스 명  Wire.begin();  if(!bmp.begin()){  Serial.println("Could not find a valid BMP280 sensor, check wiring!");  while(1);  }  mpu.begin();  mpu.calcOffsets();// 자이로와 가속도계 보정  }  voidloop(){  // BMP280 데이터 읽기  floattemp = bmp.readTemperature();  floatpressure = bmp.readPressure();  // MPU6050 데이터 읽기  mpu.update();  floataccX = mpu.getAccX();  floataccY = mpu.getAccY();  floataccZ = mpu.getAccZ();  floatgyroX = mpu.getGyroX();  floatgyroY = mpu.getGyroY();  floatgyroZ = mpu.getGyroZ();  // Bluetooth를 통해 데이터 전송  SerialBT.print("Temperature: ");  SerialBT.print(temp);  SerialBT.print(" C, Pressure: ");  SerialBT.print(pressure);  SerialBT.print(" Pa, ");  SerialBT.print("AccX: ");  SerialBT.print(accX);  SerialBT.print(" m/s2, AccY: ");  SerialBT.print(accY);  SerialBT.print(" m/s2, AccZ: ");  SerialBT.print(accZ);  SerialBT.print(" m/s2, GyroX: ");  SerialBT.print(gyroX);  SerialBT.print(" deg/s, GyroY: ");  SerialBT.print(gyroY);  SerialBT.print(" deg/s, GyroZ: ");  SerialBT.println(gyroZ);  SerialBT.println(" deg/s");  delay(1000);  }  아래 활동사진 칸에 기기를 연결하고 코드를 실행한 사진이 있다.  기기를 연결하고 코드를 실행하였지만 BMP280 센서의 연결에 문제가 생겨 결과물을 만들지는 못했다. 다른 예제 코드를 사용하였을 때 MPU6050 센서에는 문제가 없는 것으로 확인되었고, 코드에 문제점이 확인되지 않으므로 연결 과정에서의 문제 혹은 센서의 문제로 파악된다. 따라서 센서의 문제가 없다고 가정하였을 때 코드가 정상적으로 작동될 것이라 예상되며, 블루투스 통신 또한 원활히 이루어질 것으로 보인다. |
| **활동 사진** |  |
| **활동 결과** | 세 가지의 기기를 이용하여 데이터를 받아오는 과정과 각 기기들의 기능, 역할 등을 알아본 후 필요한 내용이 포함된 코드를 사용하여 실습을 진행하였으나 실행하는 것은 실패하였다. 그러나 하드웨어 적인 문제가 없다면 정상적인 결과가 나올 것으로 보인다. |
| **계획** | 문제를 해결한 후 코드가 정상적으로 작동하는지 확인한다. 이후 미흡한 부분이 발견되면 해당 내용에 수정을 가하고, PC와의 직접적인 연결 없이 아두이노 보드와 모듈, 센서들만을 이용하여 원격으로 데이터를 받아 실제로 위성같은 곳에 사용이 가능할 지를 테스트한다. |

2024년 5 월 5 일

작성자 : 문시경 (인)