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

|  |  |
| --- | --- |
| **활동 개요** | 1.high pass filter, low pass filter의 개념  2, 칼만 필터의 기본 알고리즘  3. esp32 보드의 내장 블루투스 이용해서 bmp mpu6050 데이터를 핸드폰이나 노트북으로 받아보기 |
| **일시** | 2024년 05월 05일 23시 |
| **작성자** | 노은지 |
| **활동 내용** | **- high pass filter, low pass filter의 개념**  1. 가속도 센서 특성상 고주파 영역의 노이즈를 제거하고자 low pass filter를 쓴다.  센서의 공진 주파수에 의해 신뢰 불가능한 값을 출력한다.    하지만 이러한 가속도 센서도 정말 낮은 가속도는 다음과 같은 이유로 측정이 어렵습니다.  압전식 센서는 외력의 변화가 없으면 전기적인 출력이 나오지 않는다. 이때, 방전이 되면 측정을 할 수 없습니다  https://m.post.naver.com/viewer/postView.naver?volumeNo=31701634&memberNo=43432960    허용 오차 범위  <https://blog.naver.com/its-21/221520099011>  또한, 고주파는 저주파와 붙었을 때 노이즈가 심하다.    https://blog.naver.com/PostView.naver?blogId=roboholic84&logNo=220945857964&categoryNo=39&parentCategoryNo=0  가속도는 정적인 상태에서 정확한 기울기를 얻으며 저주파에 해당한다(느린 움직임)  압전 가속도계 = 공진 피크가 있는 저역 통과  압전 세라믹은 특정 진동에서 공진 동작을 보인다.  더 넓은 작동 주파수 범위를 얻으려면 공진 주파수를 높여야 한다.  https://m.blog.naver.com/cndtec/223141346853  2. 자이로 센서는 저주파 영역에서 drift현상이 발생해 high pass filter를 쓴다.  Drift: 자이로 센서의 노이즈 및 오차가 적분할 때 누적되어 진행방향 각도가 다르게 나오는 현상  <https://blog.naver.com/PostView.naver?blogId=jb3ddrone&logNo=220992774151>  <https://mechaworld.tistory.com/11>    [🛠️ ISO 18436-2 진동 기초 용어(주기/주파수/진폭/변위/속도/가속도) : 네이버 블로그 (naver.com)](https://blog.naver.com/PostView.naver?blogId=tmdqls1230&logNo=223352928183) —---변위,속도, 가속도에 따른 주파수 측정 범위  각속도는 동적인 상태에서 안정적인 값을 얻으며 고주파에 해당한다(빠른 움직임)  [라즈이노 iOT :: 【 아두이노모듈#30】 #1.기울기 센서(6축 가속도&자이로)에 대해 알아봅시다.(GY-521/MPU6050 센서 실습 #1) (tistory.com)](https://rasino.tistory.com/entry/%E3%80%90-%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8%EB%AA%A8%EB%93%8830%E3%80%91-1%EA%B8%B0%EC%9A%B8%EA%B8%B0-%EC%84%BC%EC%84%9C6%EC%B6%95-%EA%B0%80%EC%86%8D%EB%8F%84%EC%9E%90%EC%9D%B4%EB%A1%9C%EC%97%90-%EB%8C%80%ED%95%B4-%EC%95%8C%EC%95%84%EB%B4%85%EC%8B%9C%EB%8B%A4GY-521MPU6050-%EC%84%BC%EC%84%9C-%EC%8B%A4%EC%8A%B5-1)  **- 칼만 필터의 기본 알고리즘**    <https://jbground.tistory.com/82>    https://gaussian37.github.io/autodrive-ose-basic-kalman\_filter/    https://blog.naver.com/ycpiglet/222139077774  **-MPU 6050과 BMP280의 데이터를 ESP32 내장 블루투스를 이용하여 값 받기**  **//BMP280**  ESP32 with BME280 Wiring Schematic Diagram I2C  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    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);  }    데이터 출력이 안 되는 것으로, 저번에 센서를 교체 하였을 때 데이터가 나오는 것을 확인하였다. 이는 센서 고장으로 볼 수 있으며 정상 작동하는 센서를 이용하면 값을 받아볼 수 있을 것으로 기대된다.  참고: https://randomnerdtutorials.com/esp32-bme280-arduino-ide-pressure-temperature-humidity/  **//MPU6050**  MPU-6050 Accelerometer Gyroscope Wiring to ESP32 Schematic Diagram Circuit회로도  // 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);  }    아두이노 출력 확인  참고: <https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/>  // mpu6050 블루투스 통신    #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);  } |
| **활동 사진** |  |
| **활동 결과** | Mpu6050과 esp32 블루투스 연결 완료 |
| **계획** | 정상적 bmp280받아 측정 |

2024년 05월 05일

작성자 : 노은지 (인)