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

|  |  |
| --- | --- |
| **활동 개요** | Teensy + bmp390 + bmi270 코드 병합 및 raw데이터 테스트 |
| **일시** | 2024년 05월 23일 19시 |
| **작성자** | 문기준 |
| **활동 내용** | 1. 코드 병합 기존의 예제 코드를 병합함 bmp390 센서 필터 정보를 추가함 각 기능을 함수로 쪼개어 작성함  |  | | --- | | //print with BMI270, BMP390  #include <Wire.h>  #include <Adafruit\_Sensor.h>  #include "Adafruit\_BMP3XX.h"  #include "SparkFun\_BMI270\_Arduino\_Library.h"  //BMP390와 esp32 연결 핀 지정 (INT)  int BMP\_INT = 2;  #define SEALEVELPRESSURE\_HPA (1013.25)  Adafruit\_BMP3XX bmp;  //BMI270  BMI270 imu;  #define IMU\_SCL 16  #define IMU\_SDA 17  int interruptPin = 3;  volatile boolean dataReady = false;  uint8\_t i2cAddress = BMI2\_I2C\_PRIM\_ADDR; // 0x68  void handleInterrupt() {  dataReady = true;  }  bool BMP\_INI(TwoWire \*wire, int INT){  wire->begin();  wire->setClock(1000000); // 1MHz 최대hz  if (!bmp.begin\_I2C(0x77)){  Serial.println("연결 실패: BMP390");  return false;  }  bmp.setTemperatureOversampling(BMP3\_OVERSAMPLING\_8X);  /\*BMP3\_OVERSAMPLING\_2X,  BMP3\_OVERSAMPLING\_4X,  BMP3\_OVERSAMPLING\_8X,  BMP3\_OVERSAMPLING\_16X,  BMP3\_OVERSAMPLING\_32X \*/  bmp.setPressureOversampling(BMP3\_OVERSAMPLING\_4X);  /\*BMP3\_OVERSAMPLING\_2X,  BMP3\_OVERSAMPLING\_4X,  BMP3\_OVERSAMPLING\_8X,  BMP3\_OVERSAMPLING\_16X,  BMP3\_OVERSAMPLING\_32X\*/  bmp.setIIRFilterCoeff(BMP3\_IIR\_FILTER\_COEFF\_3);  /\*BMP3\_IIR\_FILTER\_DISABLE (no filtering),  BMP3\_IIR\_FILTER\_COEFF\_1,  BMP3\_IIR\_FILTER\_COEFF\_3,  BMP3\_IIR\_FILTER\_COEFF\_7,  BMP3\_IIR\_FILTER\_COEFF\_15,  BMP3\_IIR\_FILTER\_COEFF\_31,  BMP3\_IIR\_FILTER\_COEFF\_63,  BMP3\_IIR\_FILTER\_COEFF\_127\*/  bmp.setOutputDataRate(BMP3\_ODR\_50\_HZ);  /\*BMP3\_ODR\_200\_HZ,  BMP3\_ODR\_100\_HZ,  BMP3\_ODR\_50\_HZ,  BMP3\_ODR\_25\_HZ,  BMP3\_ODR\_12\_5\_HZ,  BMP3\_ODR\_6\_25\_HZ,  BMP3\_ODR\_3\_1\_HZ,  BMP3\_ODR\_1\_5\_HZ,  BMP3\_ODR\_0\_78\_HZ,  BMP3\_ODR\_0\_39\_HZ,  BMP3\_ODR\_0\_2\_HZ,  BMP3\_ODR\_0\_1\_HZ,  BMP3\_ODR\_0\_05\_HZ,  BMP3\_ODR\_0\_02\_HZ,  BMP3\_ODR\_0\_01\_HZ,  BMP3\_ODR\_0\_006\_HZ,  BMP3\_ODR\_0\_003\_HZ, or BMP3\_ODR\_0\_001\_HZ\*/  // 인터럽트 핀 설정  pinMode(INT, INPUT\_PULLUP);  attachInterrupt(digitalPinToInterrupt(BMP\_INT), handleInterrupt, CHANGE);  Serial.println("BMP390 센서 초기화 완료");  return true;  }  bool IMU\_INI(TwoWire \*wire, int scl, int sda)//bmi270 초기화 및 설정  {  wire->begin();  wire->setSCL(scl);  wire->setSDA(sda);  wire->setClock(1000000); // 1MHz 최대hz  if (imu.beginI2C(i2cAddress, \*wire) != BMI2\_OK)  {  Serial.println("연결 실패: BMI270");  return false;  }  // 가속도 전원모드  /\*  /// BMI2\_POWER\_OPT\_MODE 0.78Hz to 400Hz  /// BMI2\_PERF\_OPT\_MODE 12.5Hz to 1600Hz  \*/  imu.setAccelPowerMode(BMI2\_PERF\_OPT\_MODE);  // 가속도 주기  /\*  / BMI2\_ACC\_ODR\_0\_78HZ  / BMI2\_ACC\_ODR\_1\_56HZ  / BMI2\_ACC\_ODR\_3\_12HZ  / BMI2\_ACC\_ODR\_6\_25HZ  / BMI2\_ACC\_ODR\_12\_5HZ  / BMI2\_ACC\_ODR\_25HZ  / BMI2\_ACC\_ODR\_50HZ  / BMI2\_ACC\_ODR\_100HZ (default)  / BMI2\_ACC\_ODR\_200HZ  / BMI2\_ACC\_ODR\_400HZ  / BMI2\_ACC\_ODR\_800HZ  / BMI2\_ACC\_ODR\_1600HZ  \*/  imu.setAccelODR(BMI2\_ACC\_ODR\_1600HZ);  // 가속도 필터 설정  /\*  / BMI2\_ACC\_OSR4\_AVG1// Oversampling Ratio  / BMI2\_ACC\_OSR2\_AVG2  / BMI2\_ACC\_NORMAL\_AVG4 (default)  / BMI2\_ACC\_CIC\_AVG8//Cascaded Integrator-Comb  / BMI2\_ACC\_RES\_AVG16//Resolution Enhancement  / BMI2\_ACC\_RES\_AVG32  / BMI2\_ACC\_RES\_AVG64  / BMI2\_ACC\_RES\_AVG128  \*/  imu.setAccelFilterBandwidth(BMI2\_ACC\_NORMAL\_AVG4);  // 자이로 전원모드  /\*  / BMI2\_POWER\_OPT\_MODE  / BMI2\_PERF\_OPT\_MODE (default)  / BMI2\_POWER\_OPT\_MODE  / BMI2\_PERF\_OPT\_MODE (default)  \*/  imu.setGyroPowerMode(BMI2\_PERF\_OPT\_MODE, BMI2\_PERF\_OPT\_MODE);  // 자이로 주기 설정  /\*  / BMI2\_GYR\_ODR\_25HZ  / BMI2\_GYR\_ODR\_50HZ  / BMI2\_GYR\_ODR\_100HZ  / BMI2\_GYR\_ODR\_200HZ (default)  / BMI2\_GYR\_ODR\_400HZ  / BMI2\_GYR\_ODR\_800HZ  / BMI2\_GYR\_ODR\_1600HZ  / BMI2\_GYR\_ODR\_3200HZ  \*/  imu.setGyroODR(BMI2\_GYR\_ODR\_3200HZ);  // 자이로 필터 설정  /\*  / BMI2\_GYR\_OSR4\_MODE // Oversampling Ratio 4  / BMI2\_GYR\_OSR2\_MODE // Oversampling Ratio 2 3200/2 Hz  / BMI2\_GYR\_NORMAL\_MODE (default) //오버샘플링 없이 센서 데이터를 그대로 출력합니다.  / BMI2\_GYR\_CIC\_MODE // Cascaded Integrator-Comb  \*/  imu.setGyroFilterBandwidth(BMI2\_GYR\_OSR2\_MODE);  // 인터럽트 핀 설정  /\*  / BMI2\_FFULL\_INT  / BMI2\_FWM\_INT  / BMI2\_DRDY\_INT  / BMI2\_ERR\_INT  / BMI2\_SIG\_MOTION\_INT  / BMI2\_WRIST\_GESTURE\_INT  / BMI2\_ANY\_MOTION\_INT  / BMI2\_NO\_MOTION\_INT  / BMI2\_STEP\_COUNTER\_INT  / BMI2\_STEP\_DETECTOR\_INT  / BMI2\_STEP\_ACTIVITY\_INT  / BMI2\_WRIST\_WEAR\_WAKE\_UP\_INT  \*/  imu.mapInterruptToPin(BMI2\_DRDY\_INT, BMI2\_INT1); // DRDY: Data Ready  bmi2\_int\_pin\_config intPinConfig;  intPinConfig.pin\_type = BMI2\_INT1;  intPinConfig.int\_latch = BMI2\_INT\_NON\_LATCH;  intPinConfig.pin\_cfg[0].lvl = BMI2\_INT\_ACTIVE\_HIGH;  intPinConfig.pin\_cfg[0].od = BMI2\_INT\_PUSH\_PULL;  intPinConfig.pin\_cfg[0].output\_en = BMI2\_INT\_OUTPUT\_ENABLE;  intPinConfig.pin\_cfg[0].input\_en = BMI2\_INT\_INPUT\_DISABLE;  imu.setInterruptPinConfig(intPinConfig);  attachInterrupt(digitalPinToInterrupt(interruptPin), handleInterrupt, RISING);  return true;  }  void BMP390\_(){  if (dataReady){  dataReady = false;  Serial.print("Temperature = ");  Serial.print(bmp.temperature);  Serial.println(" \*C");  Serial.print("Pressure = ");  Serial.print(bmp.pressure / 100.0);  Serial.println(" hPa");  Serial.print("Approx. Altitude = ");  Serial.print(bmp.readAltitude(SEALEVELPRESSURE\_HPA));  Serial.println(" m");  Serial.println();  delay(2000);  }  }  void IMU\_()  {  if (dataReady)  {  // Reset flag for next interrupt  dataReady = false;  Serial.print("Interrupt occurred!");  Serial.print("\t");  // Get the interrupt status to know which condition triggered  uint16\_t interruptStatus = 0;  imu.getInterruptStatus(&interruptStatus); // 인터럽트 정보 저장 (자이로가 준비 되었는지, 가속도가 준비 되었는지 확인용)  if (interruptStatus & BMI2\_GYR\_DRDY\_INT\_MASK) // 비트 연산자 &를 이용하여 자이로 데이터 준비 인터럽트인지 확인  {  Serial.print("Gyro data ready!");  Serial.print("\t");  imu.getSensorData();  Serial.print("X: ");  Serial.print(imu.data.gyroX, 3);  Serial.print("\t");  Serial.print("Y: ");  Serial.print(imu.data.gyroY, 3);  Serial.print("\t");  Serial.print("Z: ");  Serial.print(imu.data.gyroZ, 3);  Serial.print("\t");  }  if (interruptStatus & BMI2\_ACC\_DRDY\_INT\_MASK) // 비트 연산자 &를 이용하여 가속도 데이터 준비 인터럽트인지 확인  {  Serial.print("Accelerometer data ready!");  Serial.print("\t");  imu.getSensorData();  Serial.print("X: ");  Serial.print(imu.data.accelX, 3);  Serial.print("\t");  Serial.print("Y: ");  Serial.print(imu.data.accelY, 3);  Serial.print("\t");  Serial.print("Z: ");  Serial.print(imu.data.accelZ, 3);  }  if (!(interruptStatus & (BMI2\_GYR\_DRDY\_INT\_MASK | BMI2\_ACC\_DRDY\_INT\_MASK))) // 예외처리  {  Serial.print("Unknown interrupt condition!");  }  Serial.println();  delay(2000);  }  }  void setup() {  Serial.begin(115200);  if (!BMP\_INI(&Wire, BMP\_INT)) {  Serial.println("BMP390 초기화 실패");  }  if (!IMU\_INI(&Wire1, IMU\_SCL, IMU\_SDA)) {  Serial.println("BMI270 초기화 실패");  }  }  void loop() {  BMP390\_();  IMU\_();  } |  1. 코드 별 플로우차트 작성   도표, 기술 도면, 평면도, 개략도이(가) 표시된 사진  자동 생성된 설명  <Figure 1 : 2023 통합 코트 플로우차트>  도표, 기술 도면, 평면도, 스케치이(가) 표시된 사진  자동 생성된 설명  <Figure 2 : 사출 플로우차트>  텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진  자동 생성된 설명  <Figure 3 : 센싱 플로우차트>  도표, 텍스트, 평면도, 기술 도면이(가) 표시된 사진  자동 생성된 설명  <Figure 4 : 통신 플로우차트>   1. 함수 네이밍 고안  |  | | --- | | Setup  :  set\_imu //bmi270 셋업  set\_bmp //bmp390 셋업  set\_lora //lora 모듈 셋업  set\_eject //사출 모듈 셋업  set\_sdcard //sd카드 모듈 셋업  Update  :  read\_imu //bmi270 값 받아오기  read\_bmp //bmp390 값 받아오기  read\_eject //사출 조건 받아오기  write\_sdcard //sd카드 쓰기  Loop  :  create\_struct //lora 구조체 작성  check\_eject //사출 조건 확인  print\_data //테스트 데이터 출력  (update\_lora / get\_lora) //lora 통신 | |
| **활동 사진** |  |
| **활동 결과** | 1. Teensy 보드에 사용가능한 bmp390 및 bmi270 코드 작성 완료 2. 플로우차트 작성 완료 3. 통합코드 함수 네이밍 |
| **참고 문헌** |  |

2024년 5월 25일

폰트, 상징, 화이트, 서예이(가) 표시된 사진

자동 생성된 설명

작성자 : 문기준 (인)