

Kütüphaneler

Sensor faydalı

- #include <Wire.h> → sensörlerin I2C haberleşmesi için kütüphane
- #include <Adafruit_MPU6050.h> → MPU6050 sensör mühüm kütüphaneleri
- #include <Adafruit_HMC5883L.h> → HMC5883L manyometre mühüm kütüphaneleri
- #include <Adafruit_Sensor.h> → sensör kütüphanelerini ortak yapı katadırma kütüphane
- #include <Adafruit_AHRS.h> → AHRS(Attitude and Heading Reference System) filtrelerini (madwick ve mahony) içen, sensör verilerinden yön hesaplaması yapar.

Nesneler

- Adafruit_MPU6050 mpu; → MPU6050 sensörünü temsil eden nesne
- Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345);
HMC5883L manyometreyi temsil eden nesne. Poemetre 12345 sadece nesne ID'si.
- Adafruit_MadGwick filter; → MadGwick algoritması ile iume, gyro ve manyometre verilerini kullanarak akıllı uzaydaki yönünü hesaplar.
- float samplefreq = 50.0f / 150 Hz → filtrenin güncellenme frekansı

Setup fonksiyonu Program çalıştırıldığında yalnızca bir kez çalışır.

- void setup() { ... } → Geri dönüp değeri "void loop()" olan fonksiyon tanımı
- Serial.begin(115200); → Seri haberleşmesi başlatır. 115200 baud hızında veri göndermeceğini belirtir.
- while(!Serial); → USB seri portu hazır olana kadar beklemek.
- Serial.println("MPU6050 + HMC5883L Başlatılıyor"); → Seri monitöre istenilen mesajı yazdırır.
- if (!mpu.begin()) { ... } → MPU6050 sensörünün başlatıp başlatmadığını kontrol eder.
- while(1); Eğer bukenmeja hata mesajını verir ve sensör döngüye girer.
- mpu.setAccelerometerRange(MPU6050_ANC1E - 8 - G);
İume ölçer ölçüm odaklılarını 8G'ye ayarlar.
- mpu.setGyroRange(MPU6050_ANC1E - 500 - DEG); → Gyro ölçüm odaklı $\pm 500^{\circ}/s$
- mpu.setFilterBandwidth(MPU6050_BAND_21 - Hz);
Alınarak iume/gyro verilenin digital düzük geçiren filtresinin kesim frekansını 21Hz'ye ayarlar. Böylece sensörden gelen veriler gürültüden arındırılır.

- filter.begin(samplefreq); → Madgwick filtresini başlatır ve örnekleme frekansı ayarları.
- Serial.print("Sensörler hazır!"); → istenilen mesajları monitöre yazdırır.
- delay(100); → 100ms gecikme verilir. Sensörlerin stabil olması için.

Döngü fonksiyonu

- void loop() { ... } → her döngüdeki "void (loop)" adlı fonksiyon tanımı.
- SensorEvent t accel, gyro, temp; → sensör verilerini tutmak için Adafruit Kütüphanelerden gelen değişkenler
- mpu.getEvent(&accel, &gyro, &temp); → mpu6050'den ivme, gyro ve sıcaklık değerlerini okur.
- mpu.getEvent(&magEvent); → hmc5883L'den manyometre verisini okur.
- static unsigned long lastUpdate = 0;
- unsigned long now millis(); → millis fonksiyonu ile geçen süre milisaniye biriminde hesaplar. Bu değişken bunun içindeler.
- float dt = (now - lastUpdate) / 1000.0;
- lastUpdate = now; lastUpdate, milisaniye biriminde güncellenir.
- filter.update(gyro, gyro.x, gyro.y, gyro.z, gyro, gyro.z, accel.acceleration.x, accel.acceleration.y, accel.acceleration.z, magEvent, magnetic.x, magEvent, magnetic.y, magEvent, magnetic.z);

Gyro, ivme ve manyometre verilerini Madgwick algoritmasını günceller.

Filtre Euler açılarını hesaplar.

- float roll = filter.getRoll(); }
 - float pitch = filter.getPitch(); }
 - float yaw = filter.getYaw(); }
- cihatın, yatay ve dikey eksenler.
 Roll: Süp - sol spin
 Pitch: Üst - alt spin
 Yaw : Pusula yönü.

- Serial.print("Pitch"); \rightarrow Seri monitörde gösterir.
- Serial.print(pitch, 2); \rightarrow "2" ile ondalık basamak gösterir.
- Serial.print("° | Roll: ");
- Serial.print(" | yaw: ");
- Serial.print(yaw, 2);
- Serial.print(" | sıcaklık: ");
- Serial.print(temp, temperature);
- Serial.println(" °C ");
- delay(20); \rightarrow Döngüyü 20 ms geciktirir, yaklaşıklık 50 Hz veri okuma ve filtre güncellene sefalarır.