

Kütüphaneler

Sensör kütüphaneleri

- #include <Wire.h> → Sensörlerin I2C haberleşmesi için kütüphane
- #include <Adafruit_MPU6050.h> → MPU6050 sensör modülü kütüphanesi
- #include <Adafruit_HMC5883L.h> → HMC5883L manyetometre modülü kütüphanesi
- #include <Adafruit_Sensor.h> → Sensör kütüphanelerini ortak yapı kütüphanesi
- #include <Adafruit_AHRS.h> → AHRS (Attitude and Heading Reference System) filtrelerini (Madwick ve Mahony) içerir; sensör verilerinden yön hesaplaması yapar.

Nesneler

- Adafruit_MPU6050 mpu; → MPU6050 sensörünü temsil eden nesne
- Adafruit_HMC5883_Unified mag ⇒ Adafruit_HMC5883_Unified(12345);
HMC5883L manyetometreyi temsil eden nesne. Parametre 12345 sadece nesne ID'si.
- Adafruit_Madwick filter; → Madwick algoritması ile ivme, gyro ve manyetometre verilerini kullanarak cihazın uzaydaki yönünü hesaplar.

Değişkenler

- float sampleFreq = 50.0f / 150 Hz → Filtrenin güncellenme frekansı

Setup fonksiyonu Program çalıştırıldığında yalnızca bir kez çalışır.

- void setup() { ... } → Geri dönüş değeri "void (boş)" olan fonksiyon tanımı
- Serial.begin(115200); → Seri haberleşmeyi başlatır. 115200 baud hızında veri göndereceğini belirtir.
- while(!Serial); → USB seri portu hazır olana kadar bekleme.
- Serial.println("MPU6050 + HMC5883L başlatılıyor"); → Seri monitöre gönderilen mesajı yazdırır.
- if (!mpu.begin()) { ... } → MPU6050 sensörünün başlatıp başlanmadığını kontrol eder.
- while(1); Eğer bulunmazsa hata mesajını verir ve sensör döngüsüne girer.
- mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
İvmeölçer ölçek ölçeğini 8G'ye ayarlar.
- mpu.setGyroRange(MPU6050_RANGE_500_DEG); → Gyro ölçek ölçeğini ±500°/s
- mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Alınacak ivme/gyro verilerini dijital düşük geçiren filtresinin kesim frekansını 21Hz'ye ayarlar. Böylece sensörden gelen veriler gürültüden arındırılır.

- `filter.begin(samplefreq);` → Madgwick filtresini başlatır ve örnekleme frekansı ayarlar.
- `Serial.println("Sensörler hazır!");` → istenilen mesaj seri monitöre yazdırılır.
- `delay(100);` → 100ms gecikme verilir. Sensörlerin stabil kalması için.

Döngü fonksiyonu

- `void loop() { ... }` → Heri döngü değeri "void loop()" den fonksiyon tanımı.
- `SensorEvent t accel, gyro, temp;` → Sensör verilerini tutmak için Adafruit kütüphanesinden gelen değişkenler
- `mpu.getEvent(&accel, &gyro, &temp);` → MPU6050'den ivme, gyro ve sıcaklık değerlerini okur.
- `mpu.getEvent(&magnEvent);` → HMC5883L'den manyometre verisini alır.
- `static unsigned long lastUpdate = 0;`
- `unsigned long now = millis();` → millis fonksiyonu ile geçen süre milisaneye cihinden hesaplanır. Bu değişken burun cihinden
- `float dt = (now - lastUpdate) / 1000.0;`
- `lastUpdate = now;`
lastUpdate, milisaneye cihinden güncellenir.
- `filter.update(gyro.gyro.x, gyro.gyro.y, gyro.gyro.z, accel.acceleration.x, accel.acceleration.y, accel.acceleration.z, magnEvent.magnetic.x, magnEvent.magnetic.y, magnEvent.magnetic.z);`

Gyro, ivme ve manyometre verilerini Madgwick algoritmasıyla gönderir.

Filtre Euler açılarını hesaplar.

- `float roll = filter.getRoll();`
 - `float pitch = filter.getPitch();`
 - `float yaw = filter.getYaw();`
- } Cihazın, yatay ve dikey eksenler, Roll: Sağ-sol eğim
Pitch: Öne-arka eğim
Yaw: Puzula yönü.

- `Serial.print("Pitch:");` \longrightarrow Seri monitöre yazar.
- `Serial.print(pitch, 2);` \longrightarrow ",2" ile ondalık basamak gösterilir.
- `Serial.print("° Roll: ");`
- `Serial.print(roll, 2);`
- `Serial.print("° yaw: ");`
- `Serial.print(yaw, 2);`
- `Serial.print("° Sıcaklık: ");`
- `Serial.print(temp, temperature);`
- `Serial.println(" °C");`
- `delay(20);` \rightarrow Döngüde 20 ms geciktirir, yaklaşık 50 Hz veri okuma ve filtre güncellemeye sağlar.