

# Firmware MPU-6050 – Transmissor ESP-NOW

---

## Visão Geral

`esp32s3_codes/mpu-6050/mpu-6050.ino` é a versão simplificada do transmissor de orientação, baseada no módulo MPU6050 (MPU6050\_light). Ele envia roll/pitch/yaw estimados para o ESP32-S3 principal via ESP-NOW e aceita comandos de recalibração remota, permitindo que o controlador redefina o zero sem acessar fisicamente o sensor.

## Fluxo Principal

### 1. Setup

- Inicializa I2C (`Wire.begin(21, 22)`) e a biblioteca `MPU6050_light`.
- Executa `mpu.calcOffsets()` com o sensor em repouso (o firmware instrui o usuário via Serial).
- Configura ESP-NOW (`WiFi.mode(WIFI_STA)`), adiciona peer `receiverMac[]`, registra `onDataRecv`).

### 2. Loop

- Chama `mpu.update()` continuamente.
- A cada `sendIntervalMs` (50 ms ≈ 20 Hz), preenche `OrientationData` com `getAngleX/Y/Z()` e envia via `esp_now_send`.

### 3. Recalibração

- `onDataRecv` recebe `OrientationData` vindo do ESP32-S3 principal.
- Se `recalibra == true`, o firmware executa novamente `mpu.calcOffsets()` e escreve logs no Serial.

## Estrutura de Dados (ESP-NOW)

```
typedef struct {
    float roll;
    float pitch;
    float yaw;
    bool recalibra;
} OrientationData;
```

- `ori` – pacote enviado (roll/pitch/yaw atuais, `recalibra=false`).
- `recebido` – comando recebido (apenas `recalibra` é utilizado).

## Configurações Importantes

Constante	Descrição
<code>receiverMac[]</code>	MAC do ESP32-S3 receptor (ajuste para o seu hardware).

Constante	Descrição
<code>sendIntervalMs</code>	Intervalo entre envios (50 ms).
Pinos I2C	<code>Wire.begin(21, 22)</code> – troque conforme o módulo.

## Integração com o Sistema

- Os dados enviados por este firmware chegam ao controlador principal (firmware PID) e são repassados ao backend via telemetria serial/WebSocket. Assim, páginas como [accelerometer.html](#) e [actuators.html](#) exibem os ângulos em tempo real.
- O backend/UX pode solicitar uma nova calibração (botão “Recalibrar” na interface). Esse comando percorre: UI → backend → firmware PID → ESP-NOW → este transmissor.

## Passos de Teste

1. Compile/flash `mpu-6050.ino` com a biblioteca `MPU6050_light` instalada.
2. No Serial Monitor (115200), confirme:
  - Status `MPU6050 status: 0`.
  - Mensagem “Calibração concluída!”.
  - MAC do transmissor e log “ESP-NOW pronto. Enviando ângulos...”.
3. Mova o sensor e observe os logs `Send -> R:....` para garantir que os pacotes estão sendo transmitidos.

## Arquivos Relacionados

- [PID-CONTROL-FILTER-SPIKE-BNO.md](#) – descreve como o ESP32-S3 controlador consome os dados do MPU6050 (quando não há BNO085).
- [ACCELEROMETER-README.md](#) – interface que exibe roll/pitch/yaw recebidos deste firmware.