



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DA PARAÍBA CAMPUS CAMPINA GRANDE**

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

SISTEMAS EMBARCADOS

DOCENTE: ALEXANDRE SALES VASCONCELOS

**RELATÓRIO DE PROJETO FINAL DA DISCIPLINA DE SISTEMAS
EMBARCADOS**

CAMPUS CAMPINA GRANDE

Setembro, 2024

Introdução

Este projeto tem como objetivo explorar a integração de sensores inerciais e atuadores mecânicos para o controle de movimento, utilizando o sensor MPU6050 e servomotores controlados por uma ESP32. A proposta é coletar dados de orientação (pitch e roll) e utilizá-los para ajustar automaticamente os ângulos dos servos. Esse tipo de sistema é aplicável em diversas áreas, como robótica, controle de veículos e estabilização de plataformas. Ao longo das três etapas deste trabalho, foram desenvolvidas soluções modulares para leitura de sensores e controle preciso dos atuadores, demonstrando o potencial de sistemas embarcados para aplicações que envolvem feedback e automação.

Descrição

Este projeto foi desenvolvido em três etapas, com o objetivo de integrar um sensor de movimento e dois servomotores para controle baseado nos ângulos de inclinação. Na primeira etapa, foi implementado o sensor MPU6050, responsável por fornecer dados de aceleração e giroscópio. Na segunda etapa, foram utilizados dois servomotores, que têm a capacidade de ajustar sua posição angular. A terceira e última etapa integrou ambos os dispositivos, permitindo que, ao capturar os valores de pitch e roll do MPU6050, os servos se moverem para os ângulos correspondentes.

Toda a comunicação foi feita utilizando um ESP32 e as bibliotecas do framework ESP-IDF. O sistema calcula quaternions e ângulos de Euler a partir dos dados do sensor, permitindo um controle preciso dos servomotores de acordo com a orientação detectada pelo MPU6050. Este projeto busca demonstrar a aplicação prática de sensores inerciais no controle de atuadores.

MPU 6050

O MPU 6050 é um sensor inercial que combina um acelerômetro de três eixos e um giroscópio de três eixos em um único módulo, oferecendo medições precisas de movimento e orientação. Ele é amplamente utilizado em projetos de controle de movimento, onde é necessário monitorar a aceleração e a rotação de um objeto em diferentes eixos. O sensor se comunica com microcontroladores por meio da interface I2C, tornando sua integração simples e eficiente em sistemas embarcados.

No projeto, o sensor foi utilizado para capturar os valores de aceleração e rotação nos eixos X, Y e Z, sendo esses dados essenciais para calcular os ângulos de pitch, roll e yaw, que são usados para controlar os servomotores. A seguir, as funções implementadas para o sensor:

Funções Implementadas

1. **esp_err_t imu_init()**: Inicializa a comunicação com o MPU6050 via protocolo I2C, configurando os registradores adequados para iniciar as leituras.
2. **esp_err_t imu_read_acceleration(AccelerationData *accel)**: Lê os dados do acelerômetro e armazena os valores de aceleração nos eixos X, Y e Z.
3. **esp_err_t imu_read_gyroscope(GyroscopeData *gyro)**: Captura os dados do giroscópio, fornecendo as taxas de rotação nos três eixos.
4. **esp_err_t imu_read_data(IMUData *data)**: Realiza a leitura de todos os dados do sensor, incluindo aceleração e giroscópio, armazenando-os em uma estrutura IMUData.
5. **esp_err_t imu_calculate_quaternion(const IMUData *data, Quaternion *quaternion)**: Utiliza os dados de aceleração para calcular os quaternions, que representam a orientação do sensor no espaço tridimensional.
6. **esp_err_t imu_calculate_euler_angles(const Quaternion *quaternion, EulerAngle *euler)**: Converte os quaternions calculados em ângulos de Euler, especificamente os ângulos de pitch, roll e yaw, utilizados para controlar os servos.

Servos Motores (GS90)

Os Servomotores são atuadores eletromecânicos que permitem o controle preciso de posição angular, geralmente entre 0 e 180 graus. Eles são amplamente utilizados em sistemas de controle, como robótica e automação, devido à sua capacidade de mover e manter uma posição com precisão. No projeto, foram utilizados dois servomotores para ajustar suas posições com base nos valores de pitch e roll captados pelo MPU6050. O controle do ângulo dos servos foi realizado através de sinais PWM (Pulse Width Modulation) gerados pela ESP32.

A seguir, as funções implementadas para o controle dos servomotores:

Funções Implementadas

1. **`esp_err_t servo_init(ServoConfig *config)`**: Inicializa o servomotor com base nas configurações fornecidas, verificando se o servo responde corretamente aos comandos.
2. **`esp_err_t servo_set_angle(ServoConfig *config, int angle)`**: Define o ângulo do servomotor, movendo-o para a posição angular especificada de acordo com o valor calculado.
3. **`esp_err_t servo_get_angle(const ServoConfig *config, ServoAngle *angle)`**: Obtém o ângulo atual do servomotor, verificando a posição do eixo.
4. **`esp_err_t hw_servo_init(uint8_t gpio_num)`**: Configura o pino GPIO da ESP32 para gerar sinais PWM, usados no controle do servomotor.

Integração do MPU 6050 com os Servomotores

Na aviação, os ângulos de roll e pitch são fundamentais para o controle e estabilidade da aeronave. O roll é a inclinação lateral da aeronave, controlada pelos ailerons, e afeta o movimento em torno do eixo longitudinal. O pitch é o movimento de subida e descida do nariz da aeronave, controlado pelos elevadores, e afeta o movimento em torno do eixo lateral.

No contexto deste projeto, os valores de roll e pitch, medidos pelo sensor, são usados para ajustar a posição dos servomotores, simulando o comportamento de uma aeronave. O roll controla um dos servos, enquanto o pitch controla o outro, permitindo que o sistema responda de maneira semelhante ao controle de atitude de um avião.

Como o `main.c` faz a correlação entre MPU e servos

No arquivo `main.c`, após a inicialização do MPU 6050 e dos servomotores, o sistema entra em um loop onde realiza as seguintes etapas:

1. **Leitura dos dados do MPU6050:** As funções `imu_read_data()` e `imu_calculate_quaternion()` são utilizadas para capturar os dados de aceleração e rotação, que são convertidos em quaternions.
2. **Cálculo dos ângulos de Euler:** A função `imu_calculate_euler_angles()` converte os quaternions em ângulos de Euler (pitch, roll e yaw). Esses ângulos são fundamentais para controlar os servos.

3. **Controle dos servos:** Os ângulos de roll e pitch são utilizados diretamente para mover os servos, com o roll controlando o servo 1 e o pitch controlando o servo 2. Os servos ajustam sua posição de acordo com os valores calculados, conforme a lógica:

- servo_set_angle(&servo_1, roll)
- servo_set_angle(&servo_2, pitch)

4. **Atualização contínua:** A cada ciclo, novos dados são lidos e os servos são ajustados para refletir a nova orientação medida pelo sensor.

Esse comportamento simula como os controles de voo de um avião utilizam os valores de roll e pitch para ajustes precisos, permitindo que o sistema de controle reaja às mudanças de inclinação.

Conclusão

O desenvolvimento deste projeto permitiu a criação de um sistema funcional que integra a leitura de dados inerciais com o controle de servomotores. A implementação do MPU 6050 para capturar os valores de pitch e roll, juntamente com a resposta direta dos servomotores, demonstrou a possibilidade de conexão direta entre os dois dispositivos. O sistema desenvolvido pode ser adaptado para diversas aplicações, como controle de drones, braços robóticos ou sistemas de estabilização. Este trabalho também destacou a importância da modularização no desenvolvimento de sistemas embarcados, facilitando a expansão e manutenção do código, usando da plataforma ESP - IDF seguindo as boas práticas da Espressif.