

Entrega Parcial - RobôCIn

• Você está tendo dificuldades para a realização desse projeto? sim, quais?

Resumo: configuração do ambiente de desenvolvimento (PlatformIO, compilador), acesso e leitura de arquivos CSV em C++, entendimento das funções de odometria (integração, conversão de unidades).

A dificuldade inicial que me deparei foi a de nunca ter usado a extensão PlatformIO. Eu já tinha uma certa familiaridade com o uso da Pymark, pois já fiz alguns projetos de IoT usando ESP32-WROOM-32D e micropython. Acredito que isso também me fez ter algumas dificuldades em usar C++ e algumas bibliotecas que informarei em seguida. Antes de começar a fazer propriamente dito os códigos para o projeto, fiz teste básicos para poder me familiarizar com a extensão PlatformIO - apenas escutava sobre o seu poderoso uso. Então decidi fazer um projetinho de acender o led interno de um ESP32-WROOM-32D, que antes tinha em micropython e tive que fazer em C++ agora. O projeto serviu para poder esbarrar em diversos bugs e problemas. Sei que isso pode aparecer durante o caminho, mas, antecipar alguns já adianta demais, tendo em vista que é mais fácil encontrar documentação sobre ESP32 do que STM32F767ZI. (rindo de nervoso). Tive que instalar e voltar a usar o linux para poder ser mais otimizador e ter mais liberdade.

A intenção era poder realizar testes de comunicação, é uma questão que tenho muita dificuldade, mesmo quando participei de alguns programas de extensão tive problemas para poder entender o conceito e praticá-lo. Depois de conversar no discord percebi que poderia abstrair bastante coisa na primeira parte.

• Quais abordagens foram estudadas para tratar os problemas mencionados no projeto?

Resumo: Fiquei bastante horas tentando fazer com o que eu fosse elaborar rodasse em um ambiente simulado. Tentei usar um framework chamado *native* para poder fazer isso, mas tive muita dor de cabeça e não tive êxito. Daí parti para o que era mais básico e foi o que vi no discord. Apliquei coisas da disciplina de Algoritmos e fiz um programa para poder ler o arquivo CSV de logs recebido, compilar e mostrar algo no "serial monitor". O programa lê os dados de um CSV, integra a velocidade angular para estimar o ângulo, e compara o resultado com os valores medidos, permitindo avaliar a precisão da integração.

- Leitura de Arquivos CSV e Simulação de Dados: Estudo de métodos para ler dados de sensores a partir de arquivos CSV, usando ``ifstream`` e ``stringstream`` em C++. Simulação de leituras em tempo real (ou pseudo-real) para replicar o comportamento de um sensor.
- Integração de Velocidade Angular Conceitos básicos de odometria inercial para integrar velocidade angular (``local_w``) ao longo do tempo e obter o ângulo. Filtros simples (média móvel, remoção de offset).

• Quais são os materiais utilizados como referência para o desenvolvimento do projeto?

- **Documentação do Sensor MPU-6050:** Datasheet e Register Map da InvenSense (TDK) para entender a configuração, registradores, e possíveis escalas de leitura.
- **Documentação da Equipe RobôCIn:** Repositórios no GitHub, repositório *ssl-firmware* e materiais informados no enunciado (para ver exemplos de códigos já usados).
- **Tutoriais e Documentação de C++:** Sites de referência (cppreference.com, cplusplus.com) para comandos de leitura e manipulação de arquivos, uso de `std::vector`, `std::string`, etc.
- **Tutoriais PlatformIO:** Documentação oficial do PlatformIO para entender como lidar com bibliotecas, compilações e projetos no mesmo ambiente.
- **Materiais sobre Odometria Inercial e Movimentos Circulares:** Estudei alguns materiais da Disciplina IC do projeto do carrinho. Conceitos básicos de integração de giroscópio, possíveis filtros complementares, e exemplos de odometria inercial. Também em projetos open source no GitHub. Por fim, a ajuda do GPT para revisar códigos.