

Relatório do Projeto Final - Sistemas de Tempo-Real

Johnny Marcos Silva Soares¹, José Robertty de Freitas Costa¹,
André Ribeiro Braga¹

¹Universidade Federal do Ceará (UFC) – Campus Quixadá

johnnymarcos, robertty {@alu.ufc.br}, andrerbraga@yahoo.com.br

1. Arquitetura de Software

Pelo ponto de vista de interfaces e dispositivos, a aplicação é modelada em tarefas que são escalonadas pelo *TI RTOS*. As tarefas são responsáveis pela leitura e escrita nos dispositivos que possuem comunicação I2C, além de analisar as interrupções de *Hardware* que podem ser oriundas dos sensores LM393, que indicam velocidade de giro dos motores ou do botão que indica a seleção da trajetória que será percorrida. O PCA 9685 envia sinais PWM para os módulos L298N, enquanto uma das tarefas envia as direções de giro dos motores.

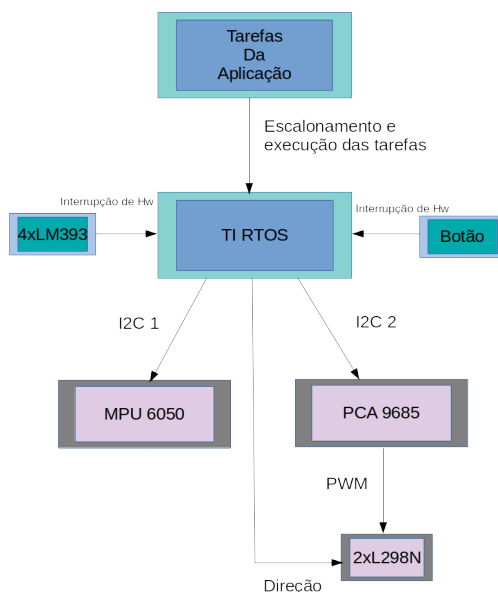


Figura 1. Funcionamento do sistema

O código fonte da aplicação é dividido em seis arquivos para a linguagem *c*. Cada código possui uma funcionalidade específica listada abaixo:

- **funcMath:** Possui as funções de arredondamento, valor absoluto e o mínimo que são utilizadas pela aplicação.
- **i2c:** Possui as funções de leitura e escrita de um endereço da memória de um escravo com o protocolo i2c.

- **mpu**: Possui as definições dos endereços do MPU 6050, a conversão de escala de aceleração e da velocidade de giro e possui a função de configuração do módulo.
- **pca**: Possui as definições dos endereços de memória utilizados, como por exemplo os endereços das saídas *PWMs* utilizadas e a função de configuração do módulo PCA 9685.
- **configuracaoInit**: Possui o mapeamento das *GPIOs* para os *encoders*, pinos de direção dos motores dos dois módulos L298N, do botão de controle, dos LEDs da placa que indicam qual trajetória será descrita e das constante de controle. Nesses arquivos são feitas as chamadas para as configurações iniciais dos módulo MPU 6050 e PCA 9685 e calibração, além de, configurar a trajetória selecionada, verificar o espaço e o ângulo das curvas e possuir as funções que serão chamadas pelas interrupções de *Clock* e de *Hardware*.
- **main**: Função principal da execução da aplicação, sendo responsável pela criação dos *Clocks*, tarefas e semáforos. Além de habilitar as interrupções de *Hardware* e configurar os pinos de interrupção.

2. Descrição das tasks

Em nosso sistema foi configurado um total de três tasks. A primeira task, denominada *taskConfigInit*, será responsável por calibrar o sensor MPU 6050, configurar o sensor PCA 9685, selecionar a trajetória que será percorrida pelo robô móvel e configurar o sentido de rotação do motor. Depois de configurar o robô e sua trajetória a task será suspensa e só será reativada quando o robô móvel completar a trajetória. A segunda task do sistema, denominada *taskEspaco* será responsável por verificar se o robô móvel alcançou a meta, ou seja, o fim da trajetória. Em caso positivo, a task deve parar o carro e reativar a task *taskConfigInit* para que seja possível selecionar a nova trajetória. A terceira e última task, denominada *taskPID*, será responsável por garantir a mesma velocidade de rotações de todas as quatro rodas do robô móvel. Para garantir a mesma velocidade de rotação utilizamos o controlador Proporcional-Integral-Derivativo (PID) para atuar em cima do valor de *Pulse Width Modulation* (PWM) das rodas.

Na tabela abaixo apresentamos as principais características de cada *task* do sistema. Observe que o fato da *taskConfigInit* não possuir período e nem frequência deve-se ao fato de se tratar de uma *task* aperiódica.

| <i>task</i> | Prioridade | Período | Frequência |
|-----------------------|------------|---------|------------|
| <i>taskConfigInit</i> | 14 | - | - |
| <i>taskEspaco</i> | 5 | 0.01 s | 100 Hz |
| <i>taskPID</i> | 12 | 0.1 s | 10 Hz |

Tabela 1. Propriedades de cada task do sistema

3. Estados do sistema

Os estados do sistema poderão ser vistos pelas *tasks* do sistema. Cada *task* no sistema tem sua função bem definida. Abaixo apresentamos um diagrama geral do sistema.

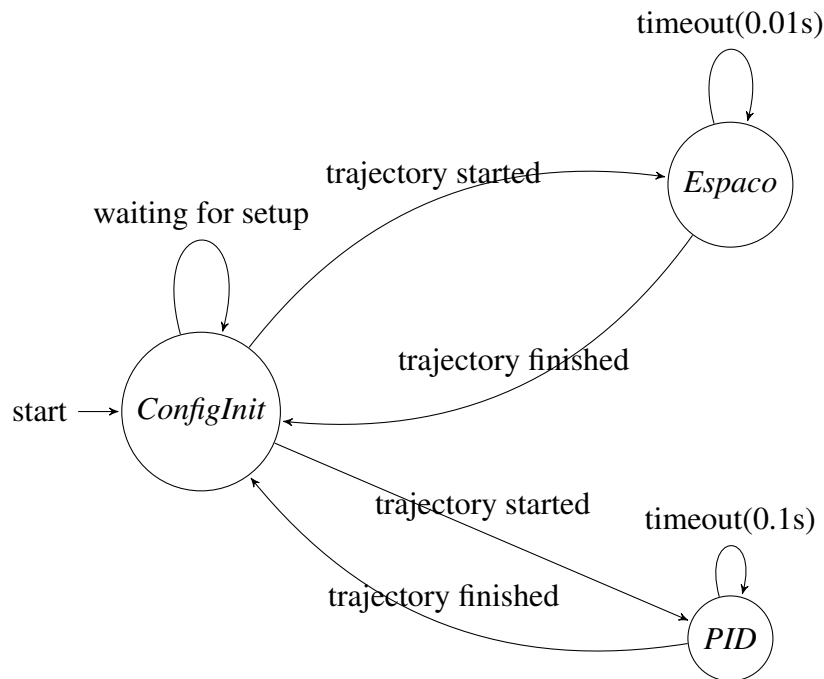


Figura 2. Diagrama de estados do sistema

Como podemos ver na figura acima o sistema inicia com a *taskConfigInit* que será responsável por configuração dos sensores PCA 9685 e MPU 6050. Após isso, a *task* fica em um estágio de espera (*waiting for setup*) esperando o pressionamento de um botão. Ao pressionar o botão a primeira trajetória é selecionada e a *taskConfigInit* configura o sentido dos motores e inicia. A partir deste momento a *taskConfigInit* se bloqueia e coloca em execução as duas outras *tasks*, desta forma, iniciando a trajetória selecionada (*trajectory started*). A *taskPID* irá controlar a velocidade de rotações por segundo (rps) de cada uma das rodas utilizando controle Proporcional-Integral-Derivativo (PID). Este controle de rotações é feito a cada 100 ms (*timeout(0.1s)*) e a velocidade estabelecida é de 3 rps em todas as trajetórias. A *taskEspaco* irá verificar se o objetivo da trajetória (ou um dos objetivos) foi completado. Esta verificação é feita a cada 10 ms (*timeout(0.01s)*). Ao terminar uma trajetória (*trajectory finished*) as *tasks Espaco* e *PID* ficam bloqueadas e a *task taskConfigInit* é desbloqueada. A cada nova ativação da *task taskConfigInit* a trajetória é incrementa, ou seja, na primeira ativação o sistema executará a trajetória 1, na segunda executará a trajetória 2 e assim por diante.

Apresentamos até então uma visão geral dos estados dos sistemas. Agora vamos descrever os estados do sistema para realizar cada uma das trajetórias.

3.1. Trajetória 1

A trajetória 1 que o carro deve ser capaz de executar é uma linha reta com comprimento igual a 1 metro. Abaixo apresentamos o diagrama de estados e as tabelas descrevendo os estados e descrevendo as transições.



Figura 3. Diagrama de estados da trajetória 1

| Estado | Descrição |
|--------|-----------------------------------------------------------|
| $q0$ | Configura a direção das rodas e inicia os PWMs das rodas. |
| $q1$ | Para o carro. |

Tabela 2. Descrição dos estados

| Transição | Descrição |
|-----------|-----------------------------------------|
| a | A distância de um metro foi percorrida. |

Tabela 3. Descrição das transições

3.2. Trajetória 2

A trajetória 2 que o carro deve ser capaz de executar é uma linha reta com comprimento igual a 2 metros. Abaixo apresentamos o diagrama de estados e as tabelas descrevendo os estados e descrevendo as transições.

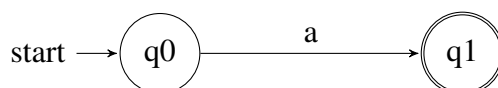


Figura 4. Diagrama de estados da trajetória 2

| Estado | Descrição |
|--------|-----------------------------------------------------------|
| $q0$ | Configura a direção das rodas e inicia os PWMs das rodas. |
| $q1$ | Para o carro. |

Tabela 4. Descrição dos estados

| Transição | Descrição |
|-----------|--------------------------------------------|
| a | A distância de dois metros foi percorrida. |

Tabela 5. Descrição das transições

3.3. Trajetória 3

A trajetória 3 que o carro deve ser capaz de executar é de um quadrado de lado igual a 1 metro. Abaixo apresentamos o diagrama de estados e as tabelas descrevendo os estados e descrevendo as transições.

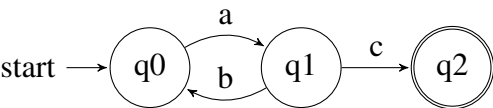


Figura 5. Diagrama de estados da trajetória 3

| Estado | Descrição |
|-----------|-----------------------------------------------------------------------|
| <i>q0</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q1</i> | Configura a direção das rodas para trajetória em curva para esquerda. |
| <i>q2</i> | Para o carro. |

Tabela 6. Descrição dos estados

| Transição | Descrição |
|-----------|--------------------------------------------------------------|
| <i>a</i> | A distância de um metro foi percorrida. |
| <i>b</i> | Obteve uma angulação de 90 graus e não completou o quadrado. |
| <i>c</i> | Completo o quadrado. |

Tabela 7. Descrição das transições

3.4. Trajetória 4

A trajetória 4 que o carro deve ser capaz de executar é descrita na figura geométrica abaixo.

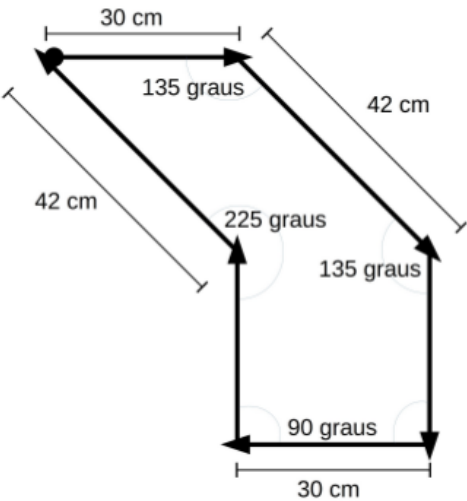


Figura 6. Trajetória 4

Abaixo apresentamos o diagrama de estados e as tabelas descrevendo os estados e descrevendo as transições.

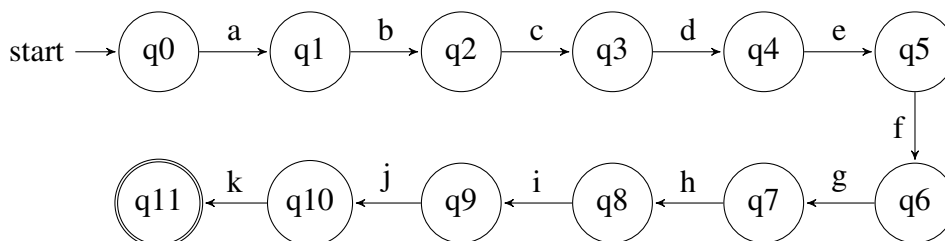


Figura 7. Diagrama de estados da trajetória 3

| Estado | Descrição |
|------------|-----------------------------------------------------------------------|
| <i>q0</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q1</i> | Configura a direção das rodas para trajetória em curva para direita. |
| <i>q2</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q3</i> | Configura a direção das rodas para trajetória em curva para direita. |
| <i>q4</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q5</i> | Configura a direção das rodas para trajetória em curva para direita. |
| <i>q6</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q7</i> | Configura a direção das rodas para trajetória em curva para direita. |
| <i>q8</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q9</i> | Configura a direção das rodas para trajetória em curva para esquerda. |
| <i>q10</i> | Configura a direção das rodas para trajetória em linha reta. |
| <i>q11</i> | Para o carro. |

Tabela 8. Descrição dos estados

| Transição | Descrição |
|-----------|--------------------------------------|
| <i>a</i> | A distância de 30 cm foi percorrida. |
| <i>b</i> | Obteve uma angulação de 45 graus. |
| <i>c</i> | A distância de 42 cm foi percorrida. |
| <i>d</i> | Obteve uma angulação de 45 graus. |
| <i>e</i> | A distância de 30 cm foi percorrida. |
| <i>f</i> | Obteve uma angulação de 90 graus. |
| <i>g</i> | A distância de 30 cm foi percorrida. |
| <i>h</i> | Obteve uma angulação de 90 graus. |
| <i>i</i> | A distância de 30 cm foi percorrida. |
| <i>j</i> | Obteve uma angulação de 45 graus. |
| <i>k</i> | A distância de 42 cm foi percorrida. |

Tabela 9. Descrição das transições

Referências