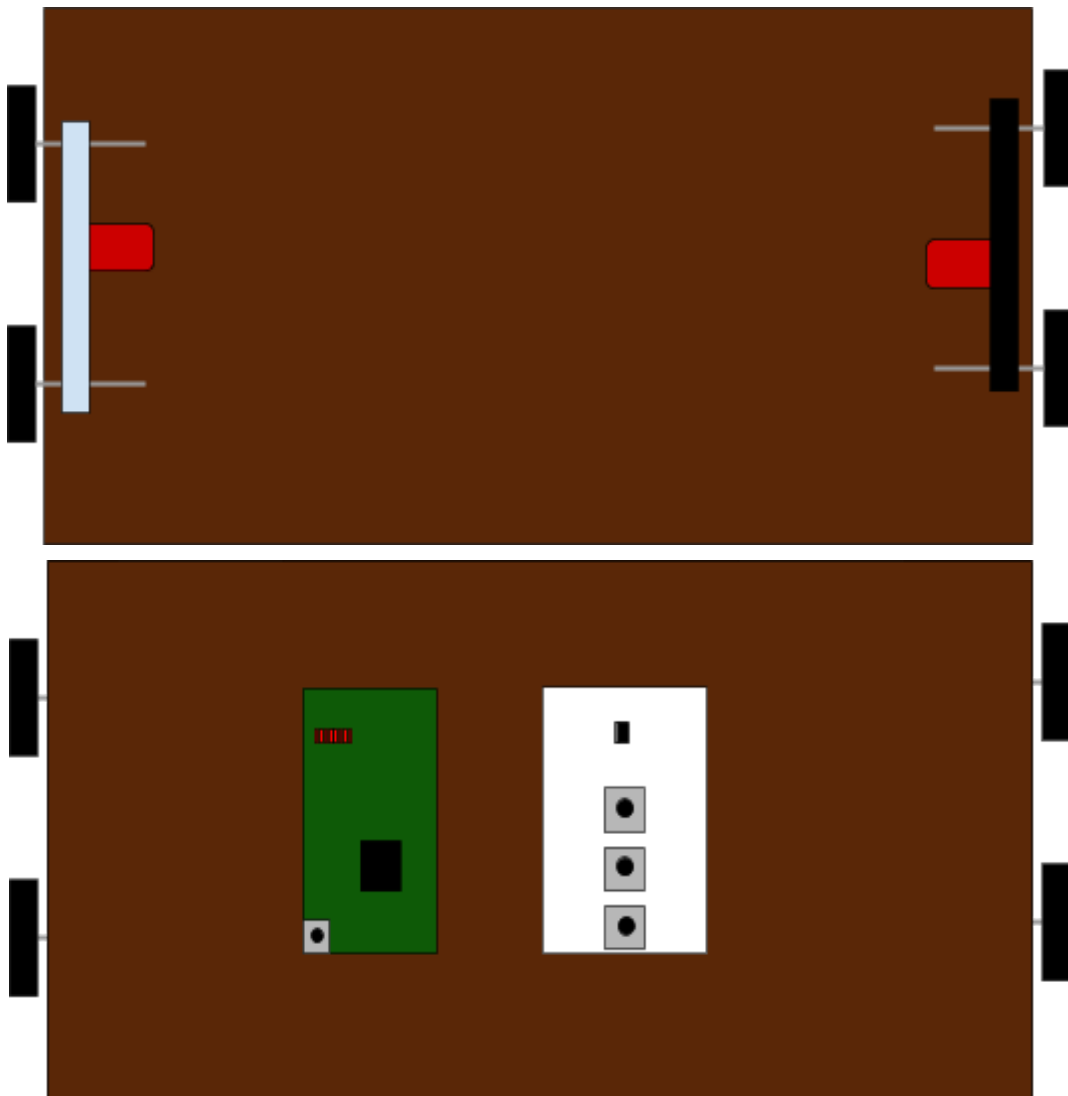
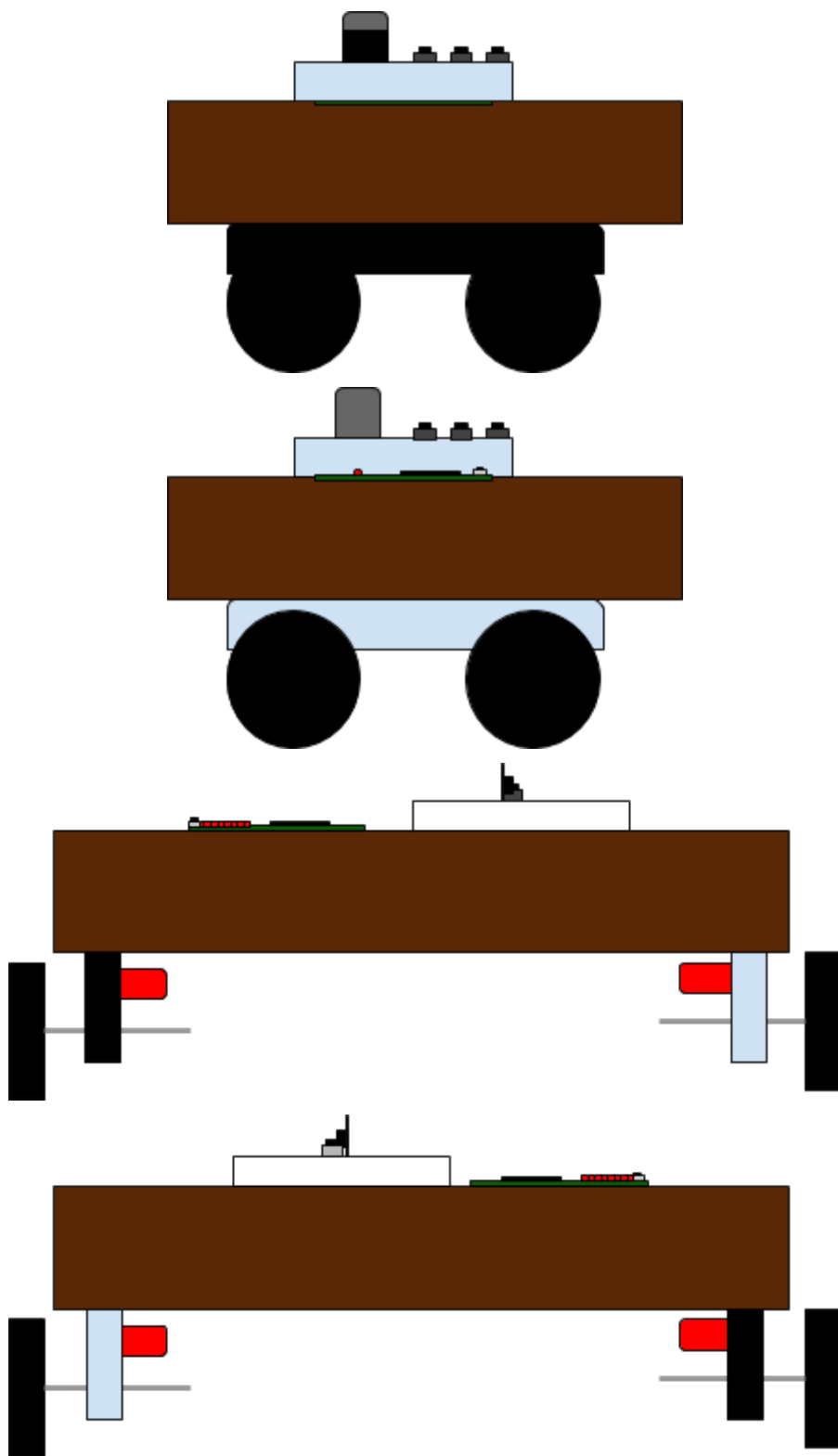


Sumário

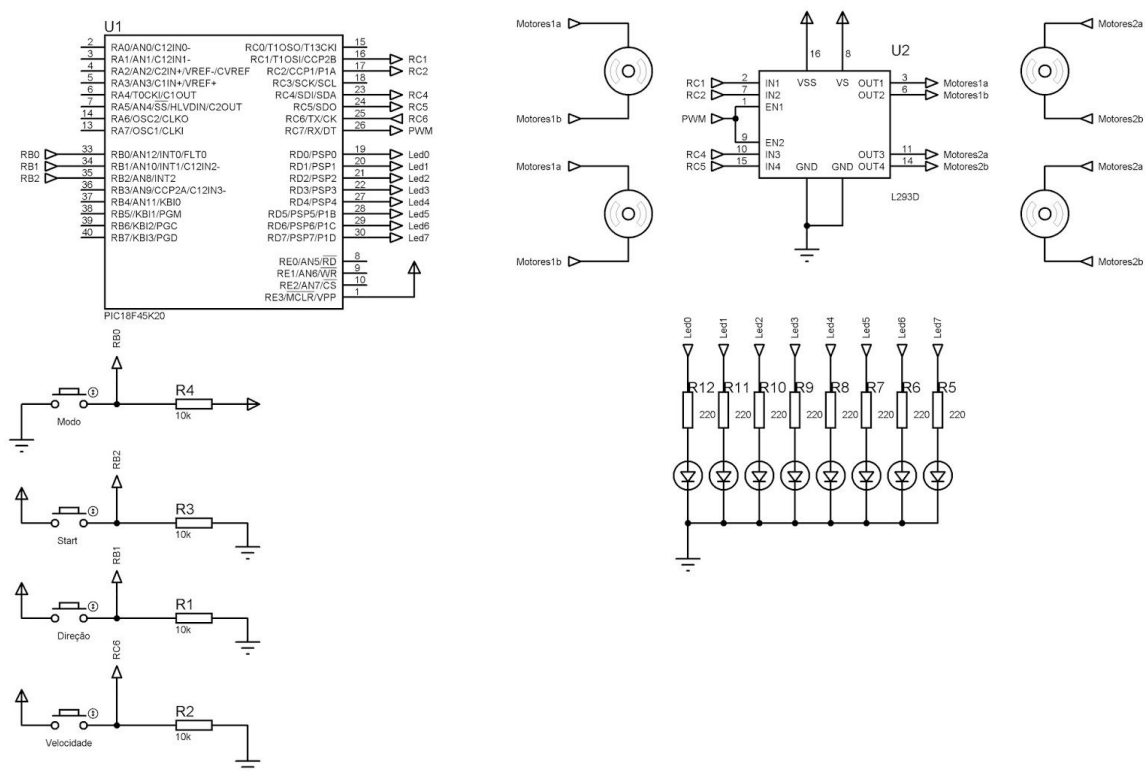
- 1. Protótipo de montagem
- 2. Protótipo dos componentes eletrônicos
- 3. Descrição das funcionalidades
 - 3.1 Estados
 - 3.2 Display dos estados
 - 3.3 Potência dos motores
 - 3.4 Andar
 - Em frente!
 - Circulando
 - Round n' round
- 4. Códigos fonte

1. Protótipo de montagem



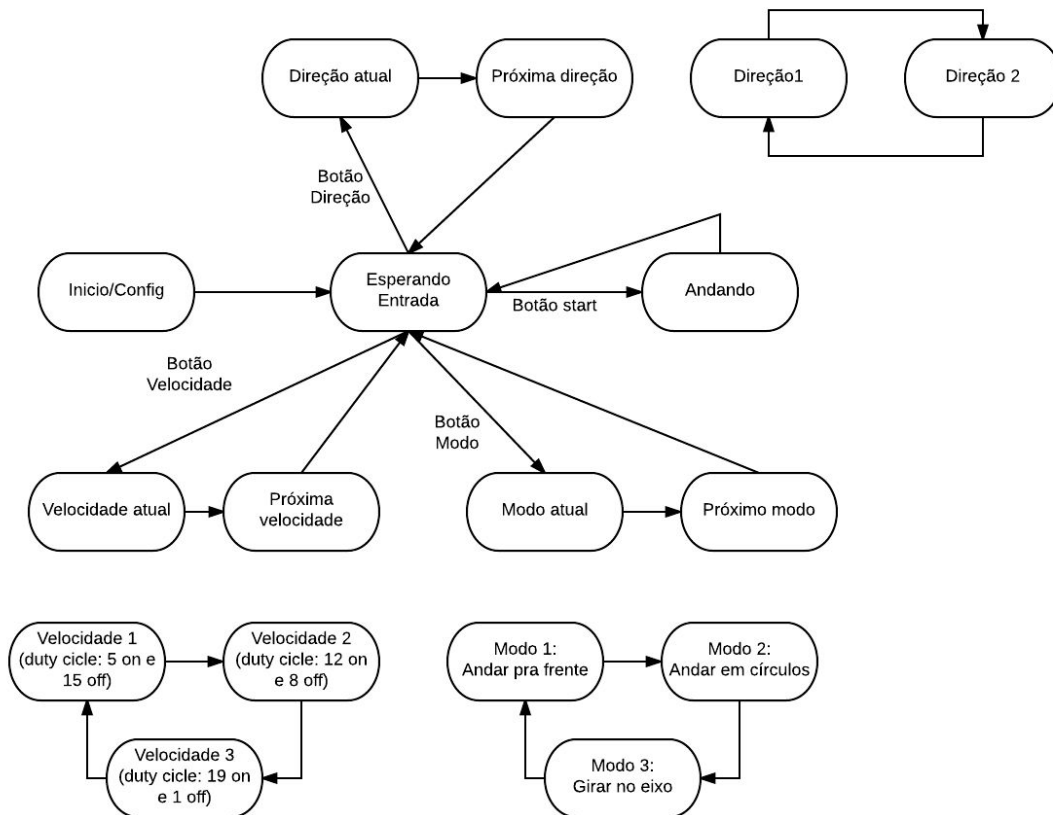


2. Protótipo dos componentes eletrônicos



3. Descrição das funcionalidades

3.1 Estados



Os diversos estados atuais são salvos em variáveis globais que são utilizadas em diversos trechos do código.

3.2 Display dos estados

Foi implementada uma função para atualizar todos os leds de acordo com os estados atuais do robo.

O display do estado atual foi feito através das leds nativas da placa do pic18f45k20, utilizando as seguinte codificações:

1. **Velocidade:** possui os valores 01, 10 e 11 (valores binários das 3 velocidades)
2. **Direção:** possui os valores 1 e 0 (bit para alternar a direção do movimento no estado de modo)
3. **Movimento:** possui os valores 01, 10 e 11 (valores binários dos 3 tipos de movimento)

Led 0	Led 1	Led 2	Led 4	Led 6	Led 7
bit 1 velocidade	bit 2 velocidade	indicador de potência do PWM	bit direção	bit 1 modo	bit 2 modo

3.3 Potência dos motores

A regulagem da potência foi realizada através de um PWM, tendo 3 potências diferentes. Considerando um duty cycle qualquer de valor X:

1. a velocidade 1 dos motores tem 25% e 75% do total em **on** e **off** respectivamente;
2. a velocidade 2 dos motores tem 60% e 40% do total em **on** e **off** respectivamente;
3. a velocidade 3 dos motores tem 90% e 10% do total em **on** e **off** respectivamente;

3.4 Andar

Para auxiliar na locomoção do Walle (nome do robô de rodas), foram implementadas funções auxiliares.

A função **SetPWM()** que recebe um bool status e o estado de velocidade atual. Caso o status seja **False** o PWM será desligado, caso o status seja **True** o PWM será ligado de acordo com a **Potência dos motores** para o estado de velocidade recebido.

A função **Direction()** que recebe uma direção e liga os motores adequados para realizar o movimento para a direção recebida.

E a função **AtivaModo()** que recebe os estados de modo atual, direção e velocidade atuais do Walle. De acordo com essas informações, chamados as funções **Direction()** e **SetPWM()** com o intuito de realizar o movimento do modo atual.

Em frente!

Ao receber o modo **Andar para frente** na função **AtivaModo()** verificamos qual o estado de direção atual do Walle. Caso seja **Direção 1** é chamada a função **Direction()** com o parâmetro **frente**, caso contrário o parâmetro será **tras**.

Depois ligamos o PWM utilizando a função **SetPWM()**, esperamos um X qualquer de segundos e desligamos o PWM.

Circulando

Ao receber o modo **Andar em círculo** na função **AtivaModo()** verificamos qual o estado de direção atual do Walle. Caso seja **Direção 1** a curva realizada pelo walle será para direita, caso contrário a curva será para **esquerda**.

O Walle andará para frente por um X qualquer de segundos, depois desligamos o PWM e chamamos a função **Direction()** com a direção da curva decidida ligando o PWM logo em seguida por uma quantidade Y de segundos. Esse processo é realizado 4 vezes para completar o 'círculo'.

Round n' round

Ao receber o modo **Girar no eixo** na função **AtivaModo()** verificamos qual o estado de direção atual do Walle. Caso seja **Direção 1** o giro realizado pelo walle será para direita, caso contrário o giro será para **esquerda**. Após isso chamamos a function **Direction()** passando o sentido do giro decidido, ligamos o PWM e esperamos um X qualquer de segundos antes de desligar o PWM.

4. Códigos fonte

O projeto está disponível no <https://github.com/rodrigondec/walle>.
O arquivo principal é o [main.c](#).