

**UFS – UNIVERSIDADE FEDERAL DE SERGIPE
CCET – CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DCOMP – DEPARTAMENTO DE COMPUTAÇÃO**

COMP0417 - FUNDAMENTOS DE SISTEMAS EMBARCADOS

PROF. MARCO TULIO CHELLA

GABRIEL LUCAS SANTANA - 201500305653

JOÃO VICTOR GUIMARÃES FLORENCIO - 201700100141

WILLIAN MOTA OLIVEIRA - 201800017520

YOHAN ALEXANDER DANTAS DE FRANCA - 201800017208

**São Cristóvão – SE
QUINTA-FEIRA, 21 DE MARÇO DE 2019**

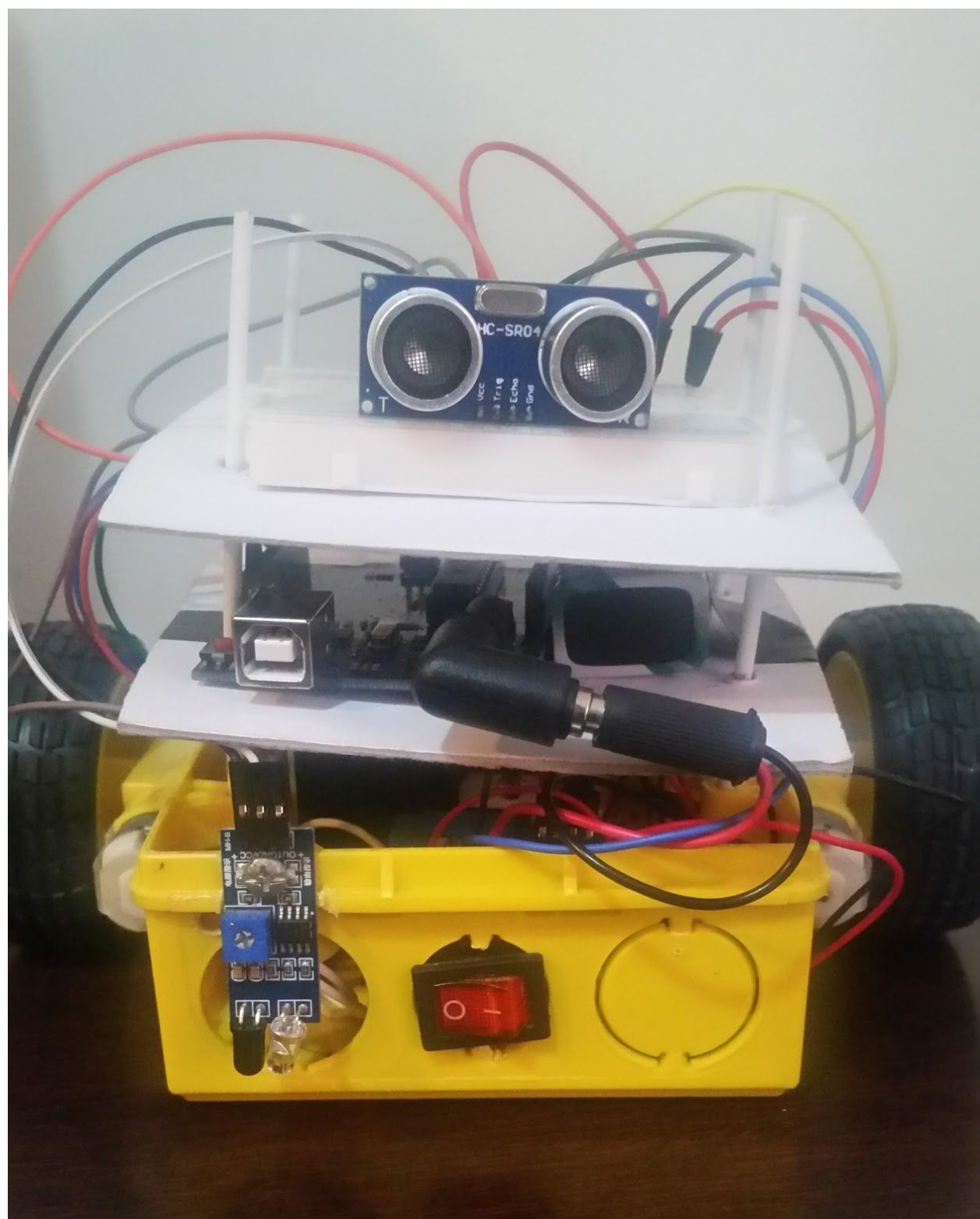
1 - Componentes Utilizados

- **1 Caixa de luz**
- **2 Motores DC 3-6V com Caixa de Redução e Roda**
- **2 Sensores de obstáculo infravermelho IR**
- **1 Arduino Uno**
- **1 Sensor de distância ultrassônico HC-SR04**
- **1 Driver Motor Ponte H L298N**
- **1 Bateria 9V**
- **8 pilhas tipo AA de 1,5 V**
- **2 soquetes para 4 pilhas cada**
- **Jumper Macho - Fêmea**
- **1 Protoboard**
- **2 Plataformas de papelão**
- **4 Tubos de caneta**

2 - Construção Física

Na montagem física do robô, foi utilizada uma caixa de luz como base para toda a estrutura. Onde logo em seguida foram acoplados os motores, juntamente com as rodas na lateral da base. Os sensores de infravermelho foram colocados, um na parte frontal do robô. e outro na parte traseira do mesmo. A primeira plataforma de papelão foi utilizada tanto para isolar a fiação, quanto para servir de base para o arduino e a bateria de 9v que serve de alimentação para o mesmo.

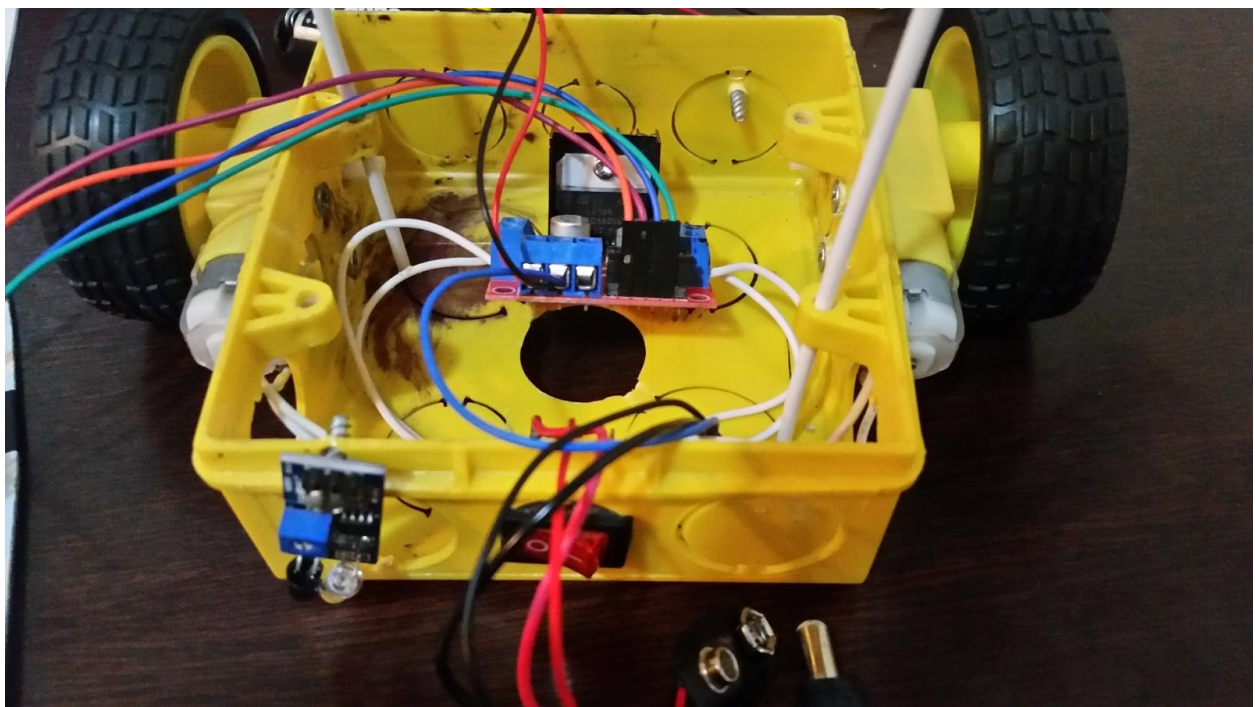
Os tubos foram utilizados como ligação e apoio às plataformas de papelão. Após a base do arduino, a segunda plataforma de papelão serviu como base para o protoboard. O sensor ultrassônico foi ligado diretamente a protoboard, juntamente com os jumpers dos sensores infravermelho, e as saídas de 5v e 0v do arduino. Os suportes com as pilhas que servem para alimentação dos motores, foram colocados na parte inferior para balancear o peso da estrutura.



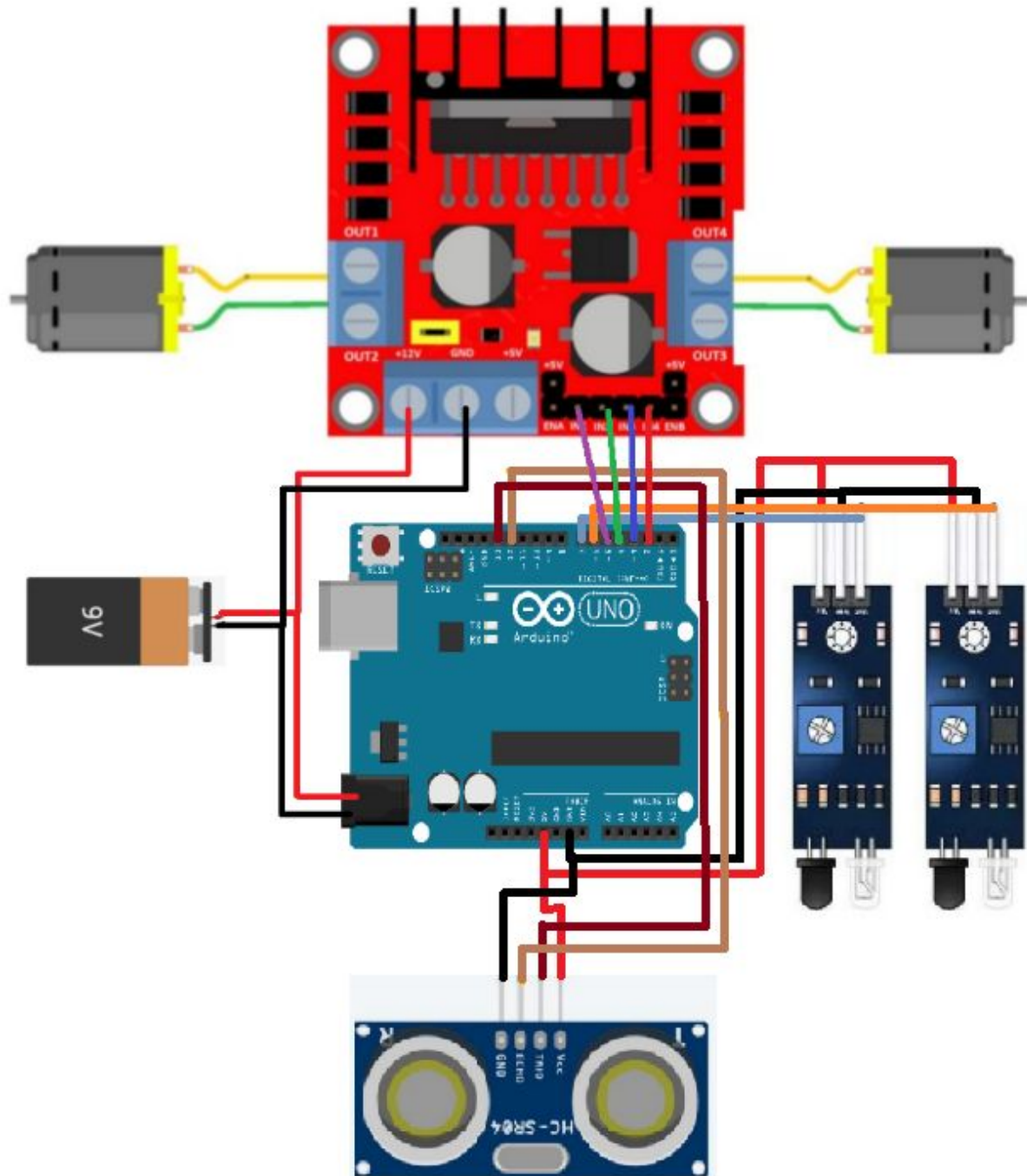
3 - Conexões Eletrônicas

Para as conexões eletrônicas foram utilizados jumpers que mapearam os sensores de ultrassom e infravermelho na protoboard, e logo depois no arduino, para a alimentação pela saída de 5v do arduino, e aterramento no 0v.

Os motores tem o giro controlado por meio do driver de ponte, e este que teve as conexões mapeadas por meio de jumpers nas saídas digitais do arduino. Além disso, a fonte de alimentação era diferente do arduino, foi feita uma conexão terra em comum para o acionamento correto dos motores. No total, foram somados 12v para a alimentação dos motores, por meio da ligação em série dos dois soquetes de pilhas.



4 - Diagrama das Conexões



5 - Estratégia

A estratégia de combate consiste basicamente em ler os sinais dos sensores infravermelho e ultrassom para perceber em qual local da arena o robô está no momento, e a qual distância se encontra o robô oponente. Recebida ambas informações, o código instrui os seguintes comportamentos para os motores nestes casos:

- **Se a distância do oponente for menor que 20 cm e os sensores não detectarem a linha, ir pra frente.**
- **Se a distância do oponente for maior que 40 cm, girar em torno de uma posição fixa até detecção do oponente.**
- **Se o sensor traseiro detectar a linha, ir pra frente e parar.**
- **Se o sensor frontal detectar a linha, dar ré e girar 90°.**

6 - Código

O código foi modularizado por meio de funções para que pudéssemos acionar os motores de acordo com o retorno dos sensores. Além do código principal que contém a estratégia de combate do robô.

Ultra.h

```
#include<avr/io.h>
#include<util/delay.h>

float dist(void){

    unsigned long duracao = 0;
    float distancia;
    DDRB = 0b11101111;

    while (1){

        // Trigger
        PORTB = 0b00000000;
        delayMicroseconds(2);
        PORTB = 0b00100000;
        delayMicroseconds(10);
        PORTB = 0b00000000;

        // detectar eco sem uso de biblioteca
        duracao = 0;
        while (!(PINB & 16)>> 4);
        while ((PINB & 16)>> 4){duracao++;}

        //calibracao com o sensor
        if (duracao > 0){
            distancia = ((duracao * 0.00945) - 2);
            return distancia;
        }
    }
}
```


Infra.h

```
#include<avr/io.h>
#include<util/delay.h>

int infra(void){

    unsigned char pino7;
    unsigned char pino6;
    DDRD = DDRD|0b00111100;
    Serial.begin(9600);

    while(1){

        pino7 = (PIND & 128) >> 7;
        pino6 = (PIND & 64) >> 6;

        if (pino7 == 1){PORTD = 0b00101000; return 7;}
        if (pino6 == 1){PORTD = 0b00010100; return 6;}
        if (pino6 == 0 && pino7 == 0){PORTD = 0b00111100; return 0;}

    }
}
```

Motor.h

```
#include<avr/io.h>
#include<util/delay.h>

int direita(void){
    DDRD = 0b00111100;
    PORTD = 0b00101000; //direita 1010
    _delay_ms(5);
}

int esquerda(void){
    DDRD = 0b00111100;
    PORTD = 0b00010100; //esquerda 0101
    _delay_ms(5);
}
```

```
int frente(void){
    DDRD = 0b00111100;
    PORTD = 0b00100100; //frente 1001
    _delay_ms(5);
}
```

```
int tras(void){
    DDRD = 0b00111100;
    PORTD = 0b00011000; //tras 0110
    _delay_ms(5);
}
```

```
int parar(void){
    DDRD = 0b00111100;
    PORTD = 0b00000000; //parar
    _delay_ms(5);
}
```

Estrategia.c

```
#include<avr/io.h>
#include<util/delay.h>
#include"motor.h"
#include"infra.h"
#include"ultra.h"
```

```
int main(void){

    while(1){

        if(dist() < 40 && infra() == 0){
            frente();
            frente();
            tras();
        }

        if(infra() == 7){
            tras();
            esquerda();
            parar();
        }
    }
}
```

```
if(infra() == 6){  
    frente();  
    parar();  
}
```

```
else if(dist() > 20){  
    parar();  
    esquerda();  
    parar();
```

```
if(infra() == 7){  
    tras();  
    esquerda();  
    parar();  
}
```

```
if(infra() == 6){  
    frente();  
    parar();  
}
```

```
if(dist() > 20){  
  
    if(infra() == 7){  
        tras();  
        esquerda();  
        parar();  
    }  
  
    if(infra() == 6){  
        frente();  
        parar();  
    }  
  
    direita();  
    direita();  
    parar();
```

```
if(dist() > 20){  
  
    if(infra() == 7){  
        tras();  
        esquerda();  
        parar();  
    }  
  
    if(infra() == 6){  
        frente();  
        parar();  
    }  
  
    esquerda();  
    parar();  
  
    }  
    }  
    }  
}
```

7 - Metodologia de Testes

Durante a fase inicial de construção foram feitos diversos testes para entender a melhor disposição dos componentes na estrutura.

Com tudo bem fixado, partiu-se para as conexões eletrônicas, estas foram projetadas de forma a mapear todos os sinais no arduino com ajuda da protoboard. Alguns problemas foram detectados durante essa fase do processo, porém com ajuda de um multímetro e medição das saídas de níveis lógicos, conseguimos resolvê-los.

Além de um curto circuito causado por conta da solda do motor dando contato em uma placa de metal. Retiramos o excesso de solda, e isolamos bem todos o terminais para evitar este tipo de problema.

Com todos os passos anteriores já bem definidos, partimos para os testes na arena. Após algumas tentativas com falha, o robô respondeu bem às condições estabelecidas para a estratégia no código, permanecendo dentro do perímetro da arena, procurando o oponente para o combate.