RELATÓRIO

DE

PRINCIPIO DE CONTROLE

Alunos:

* Andrei Moliterno 1420354/X
* Erico Rocha 1324406/5

**Introdução**

Nosso projeto tem como objetivo fazer um carro conseguir controlar suas rodas com utilização de sensores e um microcontrolador, sabendo quando o mesmo deve parar uma das rodas para fazer uma curva ou quando quiser ir reto deixando as duas rodas em velocidade constante.

Para fazer isso utilizaremos um carro que com o módulo seguidor de linha, utilizaremos 3 sensores para detectamos as linhas , terá um sensor na frente, um que estará mais a esquerda e outro que estará mais a direita, os sensores quando identificarem uma linha, enviaram para o arduino o valor digital um e quando não encontrar nenhuma linha e o valor digital zero.Dessa forma o arduino consegue de acordo com o que o sensor manda, controlar os servo motores, que os mesmos que estarão conectado as rodas do carro, vão controlar se mantém elas girando ou deixam elas paradas.

**Componentes:**

**Módulo Seguidor de Linha Infravermelho 4 Canais:**

O Módulo Seguidor de Linha Infravermelho 4 Canais é baseado em quatro sensores seguidores de linha, o seu diferencial está no fato dele otimizar o projeto, uma vez que o caminho pode ser ao mesmo tempo monitorado em quatro trechos distintos, facilitando dessa forma a estratégia adotada. Cada sensor é independente e possui um trimpot de ajuste de luminosidade de detecção.

O Módulo Seguidor de Linha Infravermelho 4 Canais é compatível com Arduino, Pic, Arm, Raspberry Pi entre outros**.**

****

**- Especificações:**

- Tensão de alimentação: 3.3V - 5V;

- Consumo de Corrente: 1A;

- Temperatura de Operação: -10°C ~ +50°C;

- Distância de detecção: 1mm ~ 60cm (Ajustável);

- Sinal de saída: TTL (Pode ser conectado direto ao microcontrolador);

- Tipo de saída: Nível lógico baixo (0) quando obstáculo detectado, nível lógico alto (1) quando sem obstáculo;

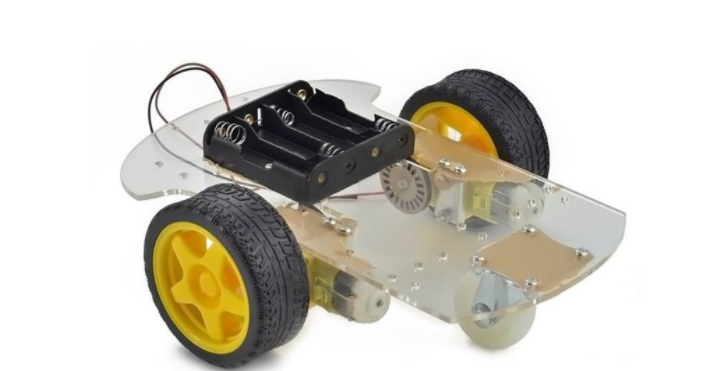
- LED: LED indica quando o sensor foi ativado (Obstáculo detectado);

- Comprimento do jumper: 20cm;

- Dimensões do módulo: 44x40x09mm.

**Kit Chassi 2WD**

O Kit Chassi 2WD foi desenvolvido com a finalidade de prover ao maker interessado em robótica educacional, uma ferramenta de montagem de um carrinho para aplicações em robótica. O Kit Chassi 2WD é composto pela estrutura fabricada em acrílico, motores com caixa de redução(1:48), rodas com pneu, roda boba, suporte para pilhas e peças de montagem do chassi. O Kit Chassi 2WDpode ser utilizado na construção de um robô seguidor de linha, robô de combate, robô que desvia de obstáculos. Para o controle do Kit Chassi 2WD, podem ser utilizadas diversas placas microcontroladas, Arduino, Raspberry Pi, Beaglebone Black, Orange Pi entre outras.



**- Composição:**

01 - Chassi em acrílico;

02 - Motores DC (3~6v);

02 - Rodas de Borracha;

01 - Roda Boba;

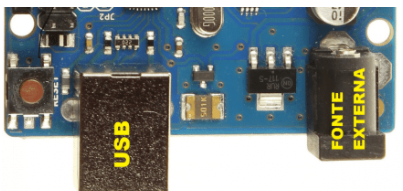
02 - Discos de Encoder;

01 - Suporte para 4 Pilhas;

01 - Jogo de Parafusos.

**Arduino UNO:**

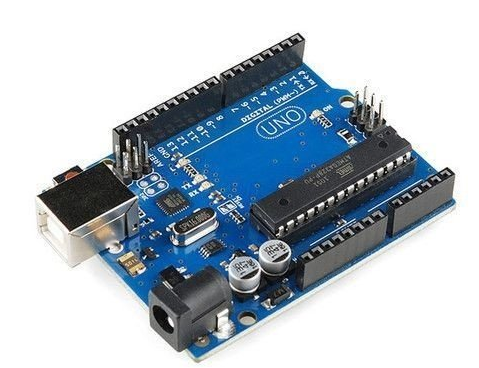
A placa pode ser alimentada pela conexão USB ou por uma fonte de alimentação externa, conforme exibido na figura abaixo:

****

Arduino é uma plataforma de computação open-source baseado em uma simples placa com entradas e saídas tanto digitais como analógicas. Possui um próprio ambiente de desenvolvimento que implementa a Linguagem C. O Arduino pode ser usado para desenvolver objetos interativos autônomos ou pode ser conectado a um software em seu computador (ex. Flash, Processing, MaxMSP). O Ambiente de desevolvimento(IDE) open-souce pode ser obtido gratuitamente (atualmente disponível para Mac OS X, Windows, e Linux).

Este é o Arduino Uno R3. É uma placa com microcontrolador Atmega328 ([datasheet](http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf)). Possui 14 entradas/saídas digitais (das quais 6 podem ser usadas como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, conexão USB, uma entrada para fonte, soquetes para ICSP, e um botão de reset. A placa contém todo o necessário para usar o microcontrolador. Simplesmente conecte-a a um computador com o [cabo USB - AB](https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=92) (não acompanha - deve ser comprado separadamente) ou ligue a placa com uma [fonte AC-DC](https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=110)(ou bateria). O Uno seleciona automáticamente a fonte de alimentação (USB ou fonte externa). Esta placa já vem pronta e testada com o microcontrolador ATMega328 pré-carregado com "bootloader"

A placa Uno se diferencia das outras por não utilizar o chip da FTDI USB-to-Serial. Ao invés deste chip, um Atmega8U2 já programado faz a função de converter os dados da USB para Serial.

****

**- Especificações:**

- Microcontrolador: ATMEGA328P-PU

- Tensão de Operação: 5V

- Tensão de Entrada: 7~12V

- Portas Digitais: 14(6 tem função PWM)

- Corrente Pinos de I/O: 40mA

- Corrente Pinos 3.3V: 50ma

- Memória Flash: 32Kb

- SRAM: 2Kb

- EEPROM: 1Kb

- Velocidade Clock: 16Mhz

# **Servo TowerPro MG995 Metálico**

O Servo Motor TowerPro MG995 é um servo motor de alto torque de 13kg/cm a 4.8V e 15kg/cm a 6V, sua rotação pode chegar a 180 graus, por essas características, ele é muito utilizado em projetos de robótica.

Esse modelo pode ser utilizado com as principais plataformas microcontroladas do mercado, seja ela, pic, arduino, arm entre outros. As conexões do servo motor MG995 segue o padrão dos principais fabricantes do mercado, são elas: Hitec, GWR, Futaba, Cirrus entre outros



**- Especificações:**

- Modelo: TowerPro MG995;

- Tensão de operação: 4,8-7,2V;

- Tipo de Engrenagem: Metálica;

- Modulação: Analógica;

- Velocidade de operação: 0,17seg/60graus (4,8V sem carga);

- Velocidade de operação: 0,13seg/60graus (6,0V sem carga);

- Torque: 13 kg.cm (4,8V) e 15 kg.cm (6,0V);

- Faixa de Rotação: 180°;

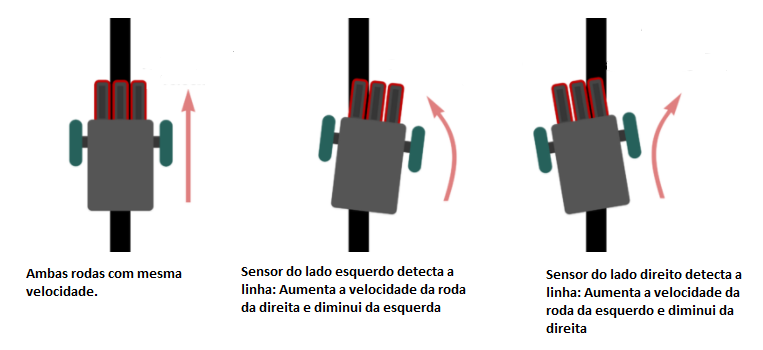
- Tamanho cabo: 300mm;

- Dimensões: 40 x 19 x 43mm

# **Protoboard**

É constituída por uma base plástica, contendo inúmeros orifícios destinados à inserção de terminais de componentes eletrônicos. Internamente existem ligações determinadas que interconectam os orifícios, permitindo a montagem de circuitos eletrônicos sem a utilização de solda.

**Explicação Técnica do funcionamento:**

Os Servo motores irão fazer girar as rodas(a roda direita e esquerda) dentro do loop fazendo o carro andar em linha reta pois as duas rodas vão estar na mesma velocidade constante. Quando um sensor detecta a linha, o arduino manda um comando para o motor do lado que o sensor detectou a linha fazendo o mesmo parar, assim somente o do outro lado vai girar fazendo ele desviar da linha e continua dessa forma. Conforme a imagem mostra.****

OBS: Durante a montagem do carro, observando que as rodas não estavam com velocidade constante ou seja, uma delas estava girando mais rápido do que a outra, mesmo mandando o valor igual em módulo. A lógica abordada é bem simples, quando o sensor da direita mandar mandar o sinal identificando uma linha, irá parar a roda da direita e aumentar a velocidade da roda da esquerda, e visse versa, porém utilizamos uma variável de ajuste para tentar igualar a velocidade da roda, por isso na implementação não colocamos os mesmo nas rodas.

**Codigo de impementação**

#include <Servo.h>

//Define os pinos para o trigger e echo

#define pino\_trigger1 1

#define pino\_trigger2

#define pino\_trigger3 3

//Inicializa as variaveis

int led = 13;

int para\_roda = 90;

int dir = 0;

float meio = 1;

float esq = 0;

int Vel = 20;

Servo mot\_esquerdo,mot\_direito;

//Configura os pinos

void setup()

{

mot\_esquerdo.attach(8);

mot\_direito.attach(9);

//mot\_esquerdo.writeMicroseconds (1500);

//mot\_direito.writeMicroseconds (1500);

}

void loop()

{

int dir = digitalRead(3);

int meio = digitalRead(2);

int esq = digitalRead(1);

int para\_Roda = 0;

int Vel = 30;

int Ajust1= 10;

int Ajust2 = -25;

//CondiÃ§oes e regras

if( dir == 1){

mot\_direito.write(90);

delay(10);

mot\_esquerdo.write(90+ Vel+ Ajust1);

}

else if(esq == 1){

mot\_esquerdo.write(90);

delay(10);

mot\_direito.write(90 - Vel - Ajust2);

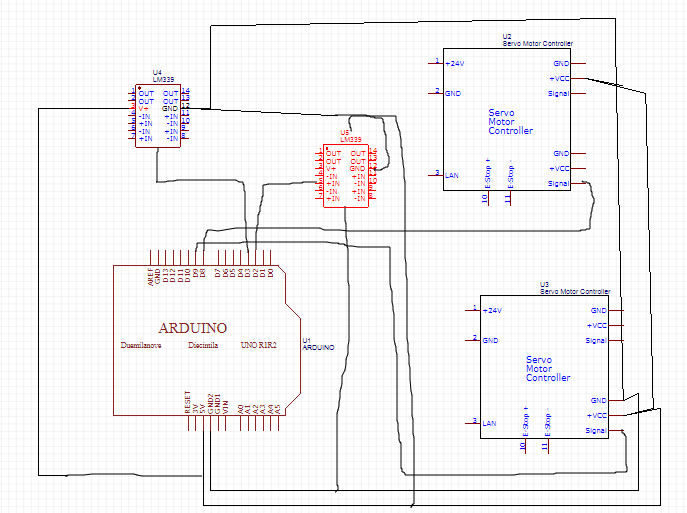
}

mot\_esquerdo.write(90 + Vel+ Ajust1);

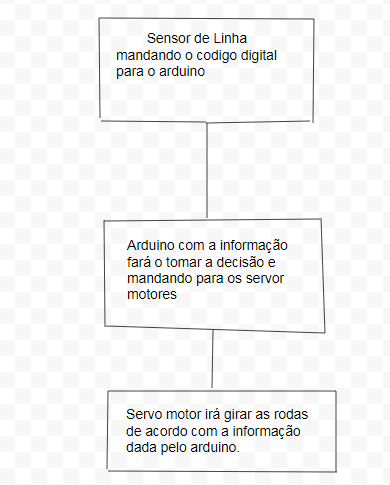
mot\_direito.write(90 - Vel- Ajust2);

}

**Esquemático**

****

**Diagrama de bloco**

****

**Conclusão**

Foi encontrado algumas dificuldades no trabalho, o fato do sensor não está muito perto da linha, causando um sinal não muito preciso. Um aumento na quantidade de sensores poderia dar mais precisão, como as rodas são soldadas manualmente, causou o problema de uma roda não girar com mesma velocidade que a outra.

bibliografia :

<https://www.filipeflop.com/blog/projeto-robo-seguidor-de-linha-arduino/>

<https://www.autocorerobotica.com.br>

<https://www.robocore.net>