



Carro

Relatório de Microcontroladores e Interação com Sensores e Atuadores

Docente: Rui Escadas

Grupo 3: Ana Beatriz Alexandre (108607), Luís Diogo (108668), João Figueiredo (107816), Miriam de Jesus (112878), Pedro Portinha (108605), Tiago Soure (108769)

Objetivos:

Este relatório tem como objetivo fazer o acompanhamento do projeto realizado no âmbito do módulo Microcontroladores e Interação com Sensores e Atuadores, da UC de Competências Transferíveis II. Neste projeto, usando Sensores e Atuadores associados a uma Placa Arduino, fizemos um programa que com o auxílio do Microcontrolador, manipula os dispositivos.

Introdução:

O projeto idealizado consiste na programação de um carro, que se move conforme o comandado. Para isto, recorreu-se a um modelo de carro com 4 rodas e 4 motores DC incorporados, Sensores (botões) que, conforme referido anteriormente, comandam a direção do movimento do carro e Atuadores (LEDs, um Buzzer e motores)

Carro modelo:



Figura 1 - Carro modelo utilizado no projeto

Placa Arduino:

A placa escolhida é uma Placa de Arduino UNO R3 SMD que utiliza uma linguagem de programação baseada nas linguagens C/C++. Esta placa tem várias componentes: microcontrolador ATmega328; 20 pinos I/O digitais; 6 portas analógicas com dois tipos de conectores (macho e fêmea) que lhe confere grande compatibilidade com shields Arduino, módulos e sensores; etc

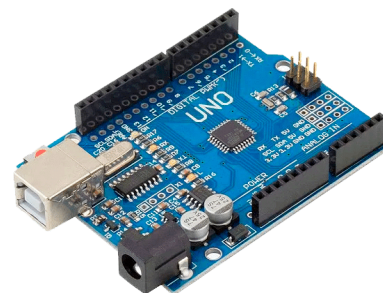


Figura 2 - Placa Arduino UNO R3 SMD

Sensores:

1. **Botões:** Este dispositivo apesar de ser bastante simples, é extremamente utilizado no ramo da Eletrônica pois permite a passagem de corrente elétrica num certo momento. Após o botão ser pressionado duas peças condutoras, que se encontram dentro do botão, entram em contacto criando, então, um circuito elétrico. Neste projeto foram utilizados botões momentâneos de:



Figura 3 - Botões momentâneos

Dois terminais: é composto por duas peças condutoras que entram em contacto quando o botão é pressionado, conectando assim os dois terminais e fechando o circuito elétrico;



Figura 4 - Dois terminais do circuito elétrico

Quatro terminais: é composto por duas peças condutoras que entram em contacto quando o botão é pressionado, fechando assim o circuito elétrico. Os terminais deste botão estão organizados em pares. Quando o botão não está pressionado os terminais laterais estão conectados. No entanto, quando pressionado, os terminais diagonais são conectados, criando continuidade no circuito



Figura 5 - Botão momentâneo

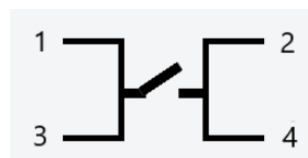


Figura 6 - Quatro terminais do circuito elétrico

Atuadores:

1. **Motor DC:** O motor de corrente contínua converte energia elétrica em energia mecânica. Este conjunto de motor é projetado para integração com plataformas como Arduino, tornando-o facilmente controlável por meio de software. Tem suporte para tensões entre 3V e 6V e é adequado para uma variedade de fontes de energia, tornando-o flexível e adaptável a diferentes configurações de projeto.



Figura 7 - Motor DC

Componentes Eletrônicos

1. **Breadboard:** é uma base de construção usada para construir protótipos semi-permanentes de circuitos eletrônicos.

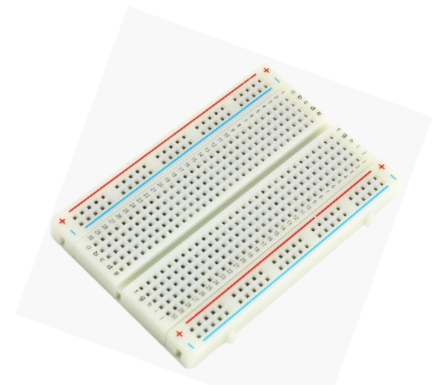


Figura 8 - Breadboard

2. **Transistor NMOs:** Este componente, de modo simplificado, permite o controle da corrente de modo similar a um interruptor a partir do sinal transmitido pelo terminal porta (G). No projeto foram utilizados 2.

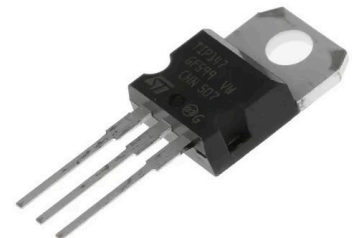


Figura 9 - Transistor

3. **Integrado L293D:** Este chip funciona como um circuito ponte H dupla tendo, assim, a capacidade de controlar dois motores bidirecionais individualmente a partir do uso de quatro canais de controle de saída. Além destes quatro canais de controle ainda possui outros dois canais de controle de velocidade dos motores através de sinais PWM. O L293D também possui diodos de proteção interna para proteger todo o circuito de picos de corrente gerados pelo motor.

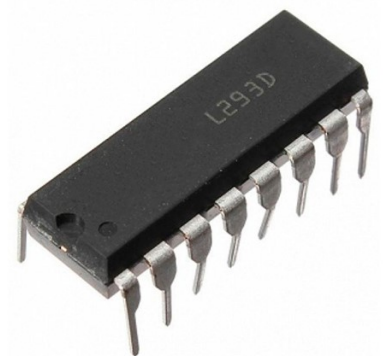


Figura 10 - Integrado L293D

4. **Potenciômetro:** funciona como uma resistência de valores variáveis. Consiste numa resistência com um contacto móvel fisicamente controlado que percorre a faixa de resistência controlando consequentemente a resistência elétrica imposta. O potenciômetro possui três terminais. O terminal central está conectado ao contacto móvel, enquanto os outros dois terminais estão conectados às extremidades da faixa resistiva.



Figura 11 e 12 - Potenciômetro e o seu funcionamento

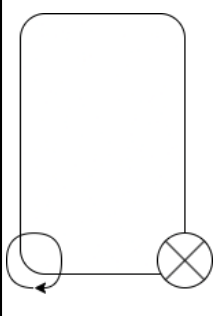
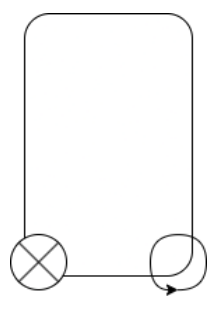
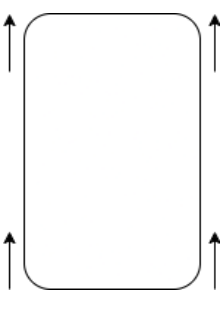
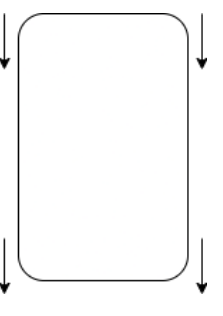
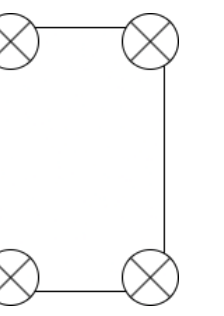
Execução do projeto:

Antes de se começar a desenvolver o software desenhou-se um fluxograma de forma a esquematizar o projeto.

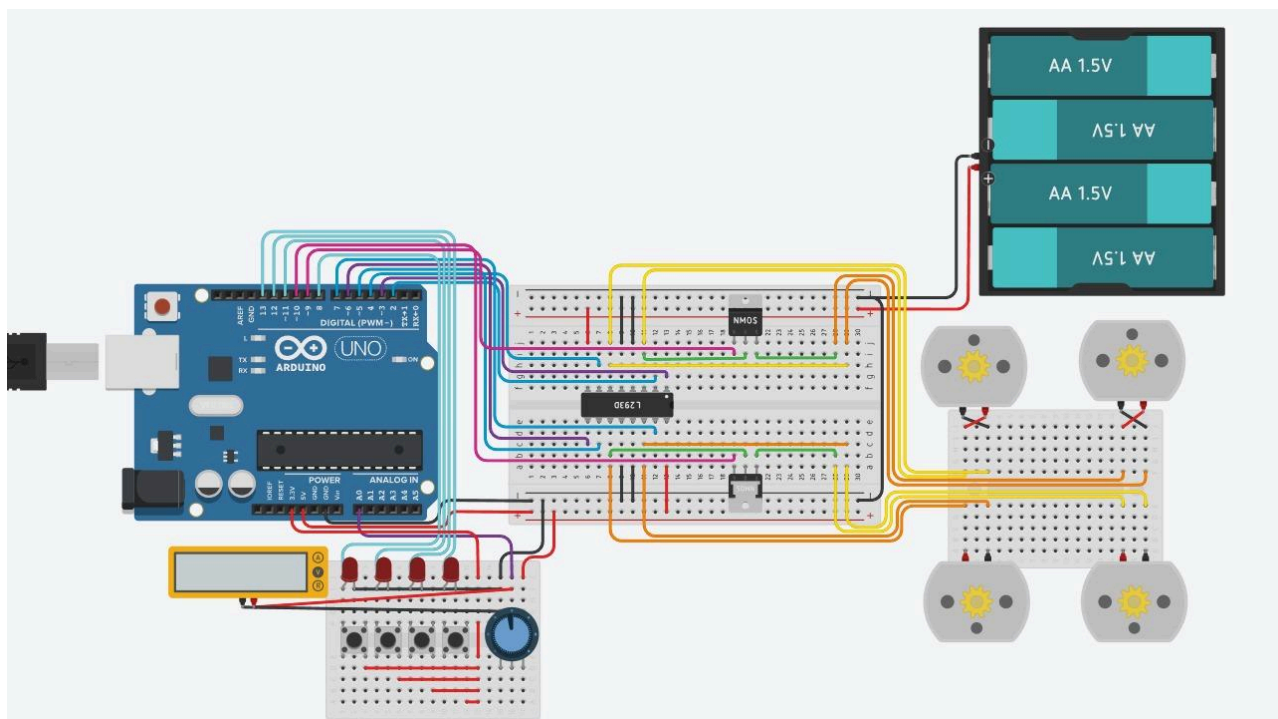


Figura 13 - Fluxograma do projeto

- Iniciou-se à avaliação das possibilidades permitidas pelo hardware disponível.
- Procedeu-se ao estudo de controle de movimento do carro, definindo quais as formas de realizar os movimentos pretendidos e possíveis tendo em conta as nossas limitações.

				
Virar para a direita: - motor direito a travar - motor esquerdo a rodar para a frente - motores dianteiros neutros	Virar para a esquerda: - motor direito a rodar para a frente - motor esquerdo a travar - motores dianteiros neutros	Movimentar para frente: - todos os motores a rodar para a frente	Movimentar para trás: - todos os motores a andar para trás	Travar: - todos os motores a travar

- Para melhor gestão de tempo e recursos humanos foi executada a montagem do circuito do projeto no emulador Tinkercad.



- Seguidamente, foi realizada uma tabela de verdade, posteriormente utilizada para o cálculo de funções booleanas para diminuir a complexidade do programa.

Sinais provenientes dos botões				Sinais de controle do L293D e dos NMOS						
reverse	left	right	brake	in1_1	in1_2	in2_1	in2_2	tra1	tra2	movimento
x	x	x	1	1	1	1	1	1	1	travar
x	0	0	0	0	0	0	0	0	0	neutro
0	0	1	0	1	0	1	1	0	0	avanço
0	1	0	0	1	1	1	0	x	1	esquerda
0	1	1	0	1	0	1	0	0	0	direita
1	0	1	0	1	0	1	1	0	0	esquerda
1	1	0	0	1	1	1	0	0	0	direita
1	1	1	0	0	1	0	1	1	x	ré

Legenda: O x nos sinais de controle assinalam a impossibilidade de controle, onde o valor de saída do transistor irá ser igual ao de entrada independentemente do sinal transmitido à porta (G).

- As funções booleanas resultantes do cálculo a partir da tabela de verdade:
 $in1_1 = brake \text{ or } (not \text{ reverse and left}) \text{ or } (right \text{ xor left})$
 $in1_2 = brake \text{ or } (reverse \text{ and left}) \text{ or } (left \text{ and not right})$
 $in2_1 = brake \text{ or } (not \text{ reverse and left}) \text{ or } (right \text{ xor left})$
 $in2_2 = brake \text{ or } (inv \text{ and right}) \text{ or } (not \text{ left and right})$
 $tra1 = brake \text{ or } (left \text{ and right})$
 $tra2 = brake \text{ or } (left \text{ and right})$
- Como passo final da prototipagem, foi escrito o programa a ser executado pelo Arduino.
- Por fim, avançou-se para a montagem do circuito, upload do programa para o Arduino e testagem do carro.

Software:

A classe botão foi criada para moldar o comportamento do botão momentâneo de inversão de sentido de modo a guardar o estado, permitindo assim, ao ser premido o botão, o estado do sentido de marcha seria invertido e apenas revertido após ser novamente premido.

```
class Btn {  
  
    private:  
        int pin;  
        bool state;  
        bool clicked;  
        bool not_clicked;  
  
    public:  
  
    Btn(int pinNum){  
        pin = pinNum;  
        state = false;  
        clicked = false;  
        pinMode(pinNum, INPUT);  
    };  
  
    void checkPress(){  
        bool buttonState = digitalRead(pin);  
        if (buttonState == HIGH) {  
            if (!clicked) {  
                state = !state;  
            }  
            clicked = true;  
        } else {  
            clicked = false;  
        }  
    };  
  
    bool getState(){  
        return state;  
    };  
};
```

O método setup foi utilizado para iniciar todos os pinos a serem utilizados e o serial monitor.

```
int Btn::num_btn = 0;

void setup()
{
    Serial.begin(9600);
    pinMode(CTRL1, OUTPUT);
    pinMode(IN1_1, OUTPUT);
    pinMode(IN1_2, OUTPUT);
    pinMode(CTRL2, OUTPUT);
    pinMode(IN2_1, OUTPUT);
    pinMode(IN2_2, OUTPUT);
    pinMode(TRA1, OUTPUT);
    pinMode(TRA2, OUTPUT);

    pinMode(POT, INPUT);
    pinMode(LEFT, INPUT);
    pinMode(RIGHT, INPUT);
    pinMode(BRAKE, INPUT);

    analogWrite(CTRL1,0);
    digitalWrite(IN1_1,0);
    digitalWrite(IN1_2,0);
    analogWrite(CTRL2,0);
    digitalWrite(IN2_1,0);
    digitalWrite(IN2_2,0);
}
```


O método `motor_ctrl` foi criado para receber os sinais provenientes dos botões e do potenciômetro e a partir destes calcular e enviar os sinais para o chip L293D, para os transistores e apresentar informações úteis para a monitorização do programa pelo serial monitor.

[illegible]

Por fim, o método `loop()` foi empregue com as funções de atualização da instância do botão e a leitura dos valores fornecidos pelos restantes botões e pelo potenciômetro.

```
void loop()
{
    // readings
    btn_inv.checkPress();
    bool inv_state = btn_inv.getState();
    Serial.println( btn_inv.getState());
    bool left_state = digitalRead(LEFT);
    bool right_state = digitalRead(RIGHT);
    bool brake_state = digitalRead(BRAKE);
    int pot_state = analogRead(POT);

    motor_ctrl(pot_state,inv_state,left_state,right_state,brake_state)
    ;
}
```

Hardware:

Com 2 breadboards disponíveis, fez-se uso de uma para a montagem do circuito dos controlos de modo a simular um comando, e da outra para a montagem do circuito responsável pelo controle dos motores.

Na breadboard de comando, foram montados os 4 botões e o potenciômetro. Em todos os 5 componentes foi conectada uma fonte de 5V proveniente do Arduino, e uma conexão terra ao potenciômetro. Posteriormente as saídas dos botões intitulados de esquerda, direita, inversão e travagem foram conectados aos pinos 8, 11, 12 e 13 do Arduino respetivamente com o propósito de uma leitura digital, enquanto a saída do potenciômetro foi conectado ao pino A0 do Arduino para fins de leitura analógica.

Analogicamente foram instalados o chip L293D e os transistores na breadboard de controle dos motores e conectados ao Arduino. As entradas do chip input 1, input 2,

input 3 e input 4 foram conectadas aos pinos 2, 4, 5 e 7 do Arduino respetivamente, com a intenção de uma escrita digital por parte do Arduino. Enquanto as entradas enable 1 & 2 e enable 3 & 4 foram conectadas aos pinos 3 e 6 do Arduino respetivamente, para o uso da sua escrita analógica realizada pelo Arduino. Ainda no L293D foram conectadas às ligações terra e power de acordo com os requisitos. Quanto às saídas do chip, o output 1 foi conectado a dois dos pólos dos motores esquerdos, enquanto o output 2 foi conectado ao outro dos pólos de motor esquerdo traseiro e ainda à entrada drenagem (D) do transistor responsável pelo motor esquerdo dianteiro. Os outputs 3 e 4 foram conectados de maneira simétrica aos outputs 1 e 2, mas em relação aos motores da direita. Em relação aos transistores, a gate (G) do transistor da esquerda foi conectado ao pino 9 do Arduino e a a gate (G) do transistor da direita foi conectado ao pino 10 do Arduino. Já a saída do transistor da esquerda foi conectado de maneira simétrica ao da direita, sendo que neste a source (S) foi conectada ao polo que restava conectar do motor dianteiro direito.

Finalizando, foi feita a ligação da fonte de energia ao circuito na localização previamente planejada.

Conclusão:

Em conclusão, o objetivo foi atingido, apesar de este apenas ter sido bem sucedido no Tinkercad. Neste emulador, os movimentos que se pretendiam obter com o carro, desde virar para a direita, para a esquerda, movimentar para a frente, para trás e travar, eram todos bem executados. Desta forma, concluiu-se que este problema poderia estar relacionado com o hardware e não com o software, podendo este estar associado ao facto da solda dos cabos se terem rompido.