

## **PROJETO UNIDADE 2**

### **COMPUTADOR DE BORDO COM ATMEGA328p**

DISCENTES:

FERNANDO LUCAS SOUSA SILVA  
TEÓFILO VITOR DE CARVALHO CLEMENTE

DOCENTE:

IGNACIO SANCHEZ GENDRIZ

DOCENTE ASSISTENTE:

MATEUS ARNAUD SANTOS DE SOUSA GOLDBARG

## **1. INTRODUÇÃO**

O presente relatório visa apresentar e explicar como foi construído o circuito no Tinker Cad que tinha por objetivo projetar um computador de bordo utilizando-se do registrador ATmega328p através do arduino e de componentes descritos na atividade, a seguir será apresentada a solução que permitiu obter a simulação com sucesso e também de reforçar o aprendizado adquirido até então em sala de aula.

## **2. METODOLOGIA**

Para o desenvolvimento deste projeto utilizamos de pesquisas no material didático disponibilizado pelo professor para entender bem o funcionamento do ATmega328p e também baseado na experiência adquirida com as atividades realizadas até então, além disso o software usado para desenvolvimento foi o TinkerCad, nele foi montado todo circuito, além da realização da codificação e simulação. Para melhor compreensão este relatório foi dividido em partes que apresentam as explicações.

## **3. RESULTADOS**

Aqui serão apresentados os componentes utilizados, o circuito montado a partir dos componentes fornecidos e as explicações de como cada parte funciona.

### **3.1 COMPONENTES**

Para o desenvolvimento do projeto foi necessário utilizar componentes eletrônicos para a construção do circuito a ser embarcado, que são: sete resistores de 10k $\Omega$ ; dois potenciômetros; um ATmega328p (conectado à placa arduino); quatro push button; quatro LEDs; quatro resistores de 220 $\Omega$ ; um servo motor.

### **3.2 CIRCUITO**

O circuito foi montado utilizando os componentes citados, como mostrado na figura abaixo, o circuito completo:

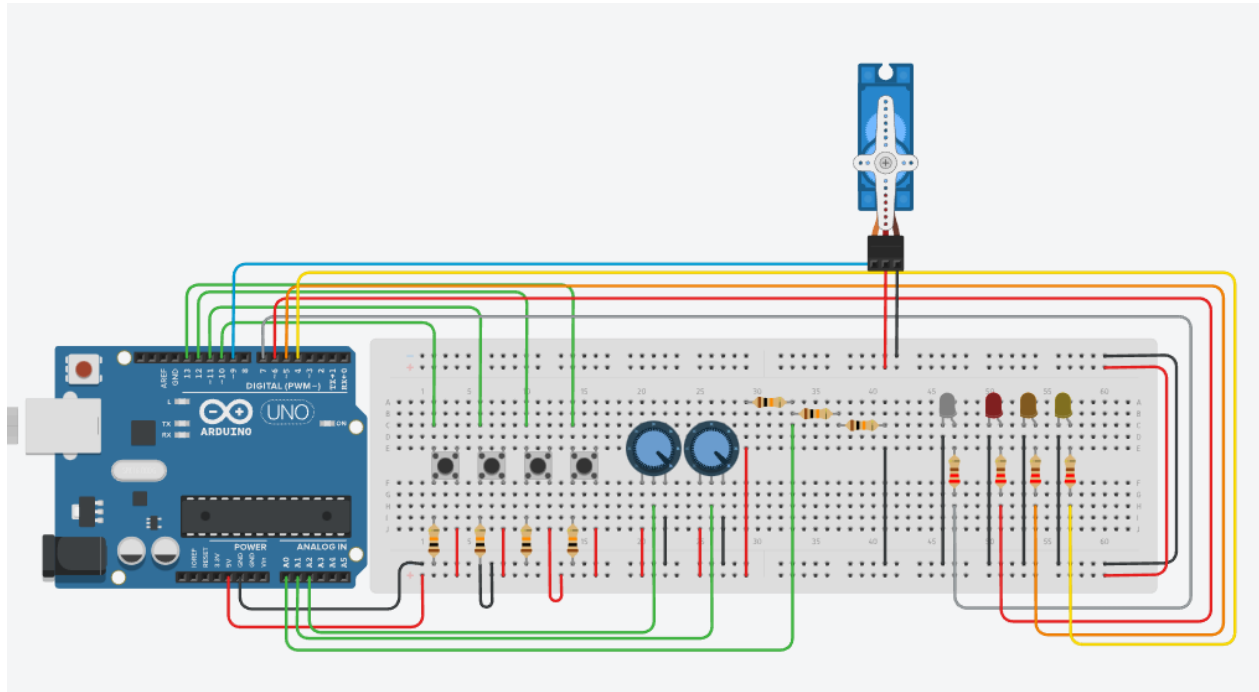


Figura 1 - Circuito completo desenvolvido no TinkerCad.

Como vemos na imagem os push buttons são como as manivelas do carro, onde ao manter eles pressionados as setas direita e esquerda (leds laranja e amarelo) ficam piscando e ao soltá-los elas param, no caso do botão dos faróis ao pressionar o botão ele ligará o farol (led branco) e ele permanece ligado e ao pressionar novamente ele desliga. Por fim, o botão que aciona o led vermelho e junto com ele o limpador, função desempenhada pelo servo motor.

Enquanto que os potenciômetros atuaram como os controladores do nível de gasolina e velocidade, desse modo ao movê-los (girá-los) teremos uma variação seja na quantidade de combustível ou na velocidade. Já o sistema de ignição é composto por três resistores em série onde o jumper (fio) é conectado em uma das três posições, configurando assim um divisor de tensão, assim obteremos valores diferentes de ignição para cada estado.

### 3.3 FUNCIONAMENTO

O funcionamento do circuito é baseado em uma conexão (Ignição do carro) que permite a seleção de 3 estados, sendo eles: desligado (o sistema não funciona), ligar faróis (o sistema permite que seja acionado os faróis, setas e leitura do nível de combustível) e ligar motor (o sistema em completo funcionamento). Assim, cada estado possui características específicas que devem ser definidas em código na linguagem de programação C (AVR).

Com isso, desenvolvemos o código para as funcionalidades e ele pode ser visualizado no

TinkerCad<sup>1</sup>, além das figuras abaixo que mostram brevemente o que foi desenvolvido:

```
bool luz = false;
bool e = true;
bool d = true;
void setup()
{
  DDRB = 0b00000000;
  PORTB = 0b00000000;
  Serial.begin(9600);
  ADMUX |= (1<<REFS0);
  ADCSRA |= (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);
}
```

Figura 2 - Primeira parte do código função setup() e variáveis.

```
void loop()
{
  ADMUX &= 0b11110000;
  ADMUX |= 0b00000000;
  ADCSRA |= (1<<ADSC);
  while(ADCSRA & (1<<ADSC));
  int A0 = ADC;

  if(A0 == 341){
    if(PINB & (1<<4) && luz == false){ //Pino para os faróis (branco)
      PORTD |= (1<<7);
      luz = true;
      delay(50);
    }
    else if(PINB & (1<<4) && luz == true){ //Pino para os faróis (branco)
      PORTD &= ~(1<<7);
      luz = false;
      delay(50);
    }
  }

  if(PINB & 1<<3){ //Pino da seta-esquerda (laranja)
    if(e){
      PORTD |= (1<<5);
    }
    else {
      PORTD &= ~(1<<5);
    }
    e = !e;
  }
  else {
    PORTD &= ~(1<<5);
  }

  if(PINB & 1<<2){ //Pino da seta-direita (amarelo)
    if(d){
      PORTD |= (1<<4);
    }
    else {
      PORTD &= ~(1<<4);
    }
    d = !d;
  }
  else {
    PORTD &= ~(1<<4);
  }
}
```

Figura 3 - Segunda parte do código: função loop() e segundo estado da ignição.

<sup>1</sup> [Circuit design ProjetoU2 | Tinkercad](#)

```
ADMUX &= 0b11110000;  
ADMUX |= 0b00000010;  
ADCSRA |= (1<<ADSC);  
  
while(ADCSRA & (1<<ADSC));  
long convert = (ADCH << 8) | ADCL;  
long fuel = (1023 - convert)/10.2;  
Serial.print("Combustivel: ");  
Serial.print(fuel);  
Serial.println(" %");  
}
```

Figura 4 - Segunda parte do código: adição do sensor de combustível no segundo estado da ignição.

```
ADMUX &= 0b11110000;  
ADMUX |= 0b00000001;  
ADCSRA |= (1<<ADSC);  
  
while(ADCSRA & (1<<ADSC));  
convert = ADCL | (ADCH << 8);  
long speed = (1023 - convert)/3.3;  
Serial.print(" | Velocidade: ");  
Serial.print(speed);  
Serial.println(" Km/h");  
}  
if(PINB & 1<<5){ //Pino do servo motor  
}  
delay(500); //Espera 500 milisegundos  
}
```

Figura 5 - Terceira parte do código: adição do sensor de velocidade no terceiro estado da ignição.

O funcionamento do código é, simplifiadamente, transformar o que está no circuito em linguagem de máquina. Por conseguinte, foram utilizadas portas digitais e analógicas, sendo mais complexo o desenvolvimento das partes em que a PWM e as analógicas estão inseridas.

Dessa forma, é importante analisar a forma que foi desenvolvido a obtenção de dados das entradas analógicas - potenciômetros (sensores de velocidade e combustível) e a ignição - visto que não é possível utilizar as funções das bibliotecas do arduino que facilitam. Assim, utilizando apenas C (AVR), foi necessário abranger a pesquisa com registradores, sendo usado o ADMUX e outros registradores que são responsáveis pelas análises das portas analógicas do arduino (atmega328p).

Ademais, o limpador de vidros (parabrisa), precisaria abstrair os dados de uma porta PWM em que quando acionado o botão, o servo motor ativa e movimenta de 0° a 180°, como ocorre nos carros.

## 5. CONCLUSÕES

A partir do desenvolvimento do projeto foi possível agregar conhecimento ao adquirido em sala e praticar o conhecimento na resolução de um problema real, que seria a construção de um computador de bordo, então coma a realização do processo vimos que o registrador ATmega328p, como também outros registradores possibilitam construir os mais variados sistemas a depender da necessidade. Um ponto importante é que devido a bugs do TinkerCad foi necessário o uso da void setup e void loop para que o circuito funcionasse normalmente, exceto isso o código foi todo escrito em C (AVR) como pedido no roteiro.

## 6. REFERÊNCIAS

[1] Material didático do Professor Ignacio.

[2] Material didático do Professor Mateus.

[3] Software. **TinkerCad**. Disponível em: <https://www.tinkercad.com>.

[4] Datasheet. **Datasheet ATmega328p**. Disponível em: [https://www.alldatasheet.com/view.jsp?Searchword=Datasheet%20atmega328p&gclid=CjwKCAjwh4ObBhAzEiwAHzZYU6DisblauxMR66BX179RMSVbHrt4XMDwlecLEIQ8HfbKm-LMwi7z9xoCpioQAvD\\_BwE](https://www.alldatasheet.com/view.jsp?Searchword=Datasheet%20atmega328p&gclid=CjwKCAjwh4ObBhAzEiwAHzZYU6DisblauxMR66BX179RMSVbHrt4XMDwlecLEIQ8HfbKm-LMwi7z9xoCpioQAvD_BwE).