

Relatório de TI

O meu código pode ser dividido em 4 funcionalidades principais: a geração de um número aleatório, o contador, a condição de reset e as condições de vitória/derrota.

Irei começar por falar da atribuição do número aleatório (Figura 1), esta função começa por atribuir um número a sorte de 1 a 32 a variável num, depois esse numero ira ser lido em binário e se acabar em 1 a divisão por resto ira ser 1(impar) enquanto que se for 0 ira ser 0 (par) depois o processo é repetido usando a funcionalidade do deslocamento para a direita(>>).

```
void gera_num(){  
    num=random(1,32);  
    digitalWrite(6, num % 2);  
    digitalWrite(7, (num >> 1) % 2);  
    digitalWrite(8, (num >> 2) % 2);  
    digitalWrite(9, (num >> 3) % 2);  
    digitalWrite(10, (num >> 4) % 2);  
}
```

Figura 1-Numero Aleatório

O contador faz uso do debounce, pois quando há uma alteração de HIGH para LOW ou vice-versa a chapa do botão pode tocar na outra parte metálica “sem querer” havendo assim uma alteração que não é pretendida. Logo, para contrariar as alterações não pretendidas durante X tempo as alterações do botão iram ser ignoradas (Figura 2).

```
int reading = digitalRead(buttonPin);  
if (reading != lastbuttonState){  
    lastDebounceTime= millis();  
}  
if ((millis()-lastDebounceTime)> debounceDelay){  
    if (reading!= buttonState){  
        buttonState=reading;  
        if (buttonState != HIGH){
```

Figura 2-Contador

O reset(Figura 3) faz uso de algo parecido ao debounce em que se o botão for pressionado durante X tempo ira ser executado um conjunto de comandos, esses comandos incluem uma limpeza das variaveis usadas no jogo e no final um chamamento da função loop() .

```
if ((millis()-lastDebounceTime)>resetttime){  
    if (buttonState == LOW){  
        n=0;  
        Serial.println("O programa vai ser resetado");  
        delay(1000);  
        loop();  
    }
```

Figura 3-Reset

Quanto às condições de vitória e derrota(Figura 4), depois de acabar o tempo o programa ira ver se o numero do contador é igual ao numero que foi gerado, se for ocorre um varrimento que passa por acender e apagar as luzes LED sequencialmente com um delay de 200ms, se for diferente é imprimida uma mensagem a dizer que errou e qual era o numero que devia ter sido colocado. Depois de verificada as condições de vitória/derrota o programa limpa as variaveis tal como no reset e o programa volta a correr do inicio.

```
if(num==(n)){
  Serial.println("Ganhou, congratz!!!!");
  n=0;
  //Acende as leds de forma sequencial
  for (int i=6; i<=10; i++){
    digitalWrite(i, 1);
    delay(200);
    digitalWrite(i, 0);
    delay(200);
  }
}
else{
  Serial.print("Falhou o numero correto era:");
  Serial.print(num);
  Serial.println();
  n=0;
}
```

Figura 4-Vitoria/derrota

Segue em anexo o link para o tinkercad e uma foto da montagem(Figura 5)([Link](#)).

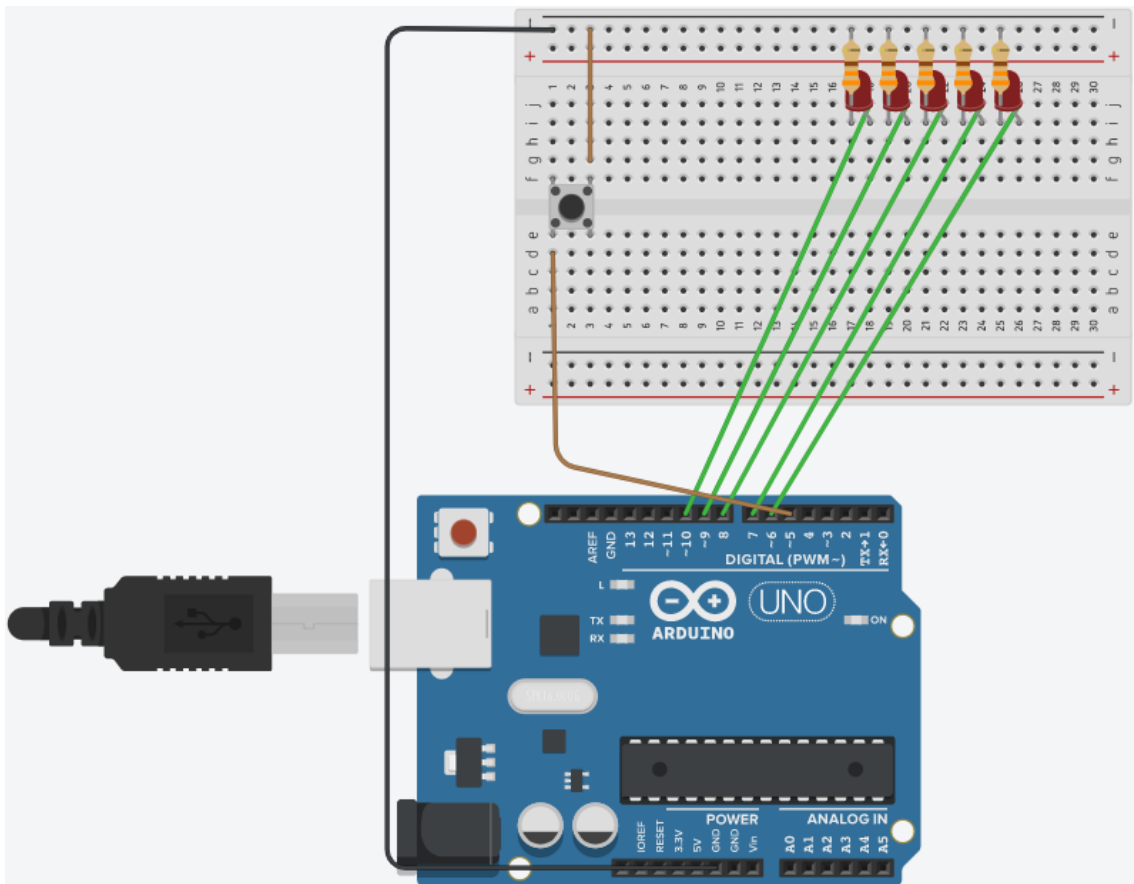


Figura 5-Montagem