

## Relatório do Trabalho Prático Individual

Este relatório foi desenvolvido no âmbito da realização de um trabalho prático sobre um Jogo (BIT MATH) na unidade curricular de Tecnologia da Informática na licenciatura em engenharia informática.

### Material utilizado:

1x Arduino UNO  
1x Placa de ensaio  
1x LED RGB 4x LEDs Verdes 4x LEDs Vermelhos 1x Botão  
11x Resistências de 220 ohm  
Cabos

Nas primeiras 23 linhas de código começo por definir as variáveis que vou usar ao longo do programa. Dentro da função setup recorro à função pinMode para configurar os pinos dos leds, do botão e do RGB.

```
void setup()
{
  Serial.begin(9600);
  pinMode(botao, INPUT_PULLUP);
  for (int pin=6; pin<=13; pin++)
  {
    pinMode(pin, OUTPUT);
  }
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  Serial.println("Jogo BIT MATH - Boa sorte!"); //mostra a mensagem
  Serial.println("Clica no botao para dar inicio ao jogo ! ");
}
```

Entrando dentro do loop começo por definir o estado da flag como true, a seguir uso a condição while fazendo com que o programa não saia deste while enquanto eu não premir o botão (usei o **debounce** para assumir que um clique é apenas um clique), pois ao premir o botão, a flag irá ser definida como falsa e assim sai deste while e continua o programa.

```
void loop()
{
  flag=true;
  while(flag) //o programa fica aqui "preso" enquanto
  {
    int leitura = digitalRead(botao);
    if(leitura != ultimoestadobotao)
    {
      lastdebouncetime = millis();
    }
    if ((millis()-lastdebouncetime)<debouncedelay)
    {
      if (leitura != estadobotao)
      {
        estadobotao=leitura;
        if (estadobotao==LOW)
        {
          flag=false; //apos o primeiro clique, o pr
        }
      }
    }
    ultimoestadobotao=leitura;
  }
}
```

```
int numverde=random(0,16);
int numvermelho=random(0,16);
digitalWrite(13,(numverde & n1)); //
digitalWrite(12,(numverde & n2));
digitalWrite(11,(numverde & n3));
digitalWrite(10,(numverde & n4));
digitalWrite(9,(numvermelho & n1));
digitalWrite(8,(numvermelho & n2));
digitalWrite(7,(numvermelho & n3));
digitalWrite(6,(numvermelho & n4));
```

← Ao sair deste while, o programa vai gerar dois números aleatórios entre 0 e 15(pois o numero máximo que 4 leds podem mostrar é 1111 que corresponde ao numero 15) e vai ligar ou manter desligados os leds consoante cada número.

A seguir o programa vai gerar um número aleatório de 1 a 3 e consoante este número, por exemplo:

Através da condição IF:

Caso o numero gerado seja 1, o RGB acende a luz vermelha, indicando assim que o operador bitwise é **and** e que o numero que o utilizador quer acertar é (numverde **and** numvermelho), e assim acontece com as outras operações ligando a luz azul caso seja a operação bitwise **OR** e acendendo a luz verde caso seja a operação bitwise **XOR**.

```
int operadorbitwise=random(1,4);
int numerocorreto;
if (operadorbitwise==1) //Liga o rgb consoa
{
  digitalWrite(RED, HIGH);
  numerocorreto = numverde & numvermelho ;
}
else if (operadorbitwise==2)
{
  digitalWrite(GREEN, HIGH);
  numerocorreto = numverde ^ numvermelho ;
}
else if (operadorbitwise==3)
{
  digitalWrite(BLUE, HIGH);
  numerocorreto = numverde | numvermelho ;
}
```

Usei a condição while, para que o utilizador tenha apenas 7 segundos para adivinhar o número correto transmitido a partir dos leds e do RGB.

O tempoderesposta foi definido para 7000ms que corresponde a 7 segundos.

Dentro da condição while, a variável tempo não é atualizada mas a função millis sim, ou seja, o tempo é um valor fixo, e millis é um valor que vai aumentando, logo quando a diferença entre o millis e o tempo for maior que o tempoderesposta(7segundos), a condição já não se verifica, saindo assim deste while e continuando o programa.

```
unsigned long tempo=millis();  
while((millis()-tempo)<tempoderesposta)
```

```
unsigned long tempo=millis();  
while((millis()-tempo)<tempoderesposta)  
{  
  if ((millis()-tempo)==5250) //75% do tempo (7segundos)  
  {  
    Serial.println("Ja passou 75% do tempo! ");  
  }  
  if ((millis()-tempo)==3500) // 50% do tempo (7segundos)  
  {  
    Serial.println("Ja passou metade do tempo! ");  
  }  
}  
  
int leitura = digitalRead(botao);  
if (leitura != ultimoestadobotao)  
{  
  lastdebouncetime = millis();  
}  
if ((millis()-lastdebouncetime)<debouncedelay)  
{  
  if (leitura != estadobotao)  
  {  
    estadobotao=leitura;  
    if (estadobotao == LOW)  
    {  
      Serial.println("NUMERO - "+String(num));  
      num=num+1;  
    }  
  }  
}  
ultimoestadobotao=leitura;
```

É impresso no ecrã quando passa 75% do tempo e quando passa 50% do tempo.

O numero que o utilizador insere enquanto tenta ganhar o jogo, é impresso no ecrã e é contabilizado corretamente a partir do botão (usando novamente o debounce).

Chamei a função setup que usei para configurar novamente os pinos e também para imprimir no ecrã a mensagem inicial “Jogo BIT MATH - Boa sorte!” e “Clica no botão para dar inicio ao jogo!” e chamei a função loop, onde o programa volta ao inicio do loop de modo a que seja permitido jogar varias vezes seguidas. Resumidamente, tentei fazer com que o programa seja reiniciado.

Durante os 7 segundos o utilizador pode reiniciar o jogo, através de um clique prolongado com a duração de mais de um segundo no botão (usei o debounce mas em relação aos anteriores, em que usei um debouncedelay de 50 ms, neste defini tempobotaoresetar como 1000ms que corresponde a 1segundo). Após mostrar a mensagem “Jogo a ser reiniciado” faz o varrimento dos leds 3 vezes e ao acabar, chama as funções setup e loop.

```
if ((millis()-lastdebouncetime)>tempobotaoresetar) /  
{  
  if (estadobotao == LOW)  
  {  
    num=0;  
    Serial.println("|-----|");  
    Serial.println("|JOGO A SER REINICIADO!");  
    Serial.println("|-----|");  
    while(nvarrimento<4)  
    {  
      for (int i=0 ; i<=3 ; i++)  
      {  
        digitalWrite(10+i, HIGH);  
        digitalWrite(9-i, HIGH);  
        delay(100);  
        digitalWrite(10+i, LOW);  
        digitalWrite(9-i, LOW);  
        delay(100);  
        if (operadorbitwise==1)  
        {  
          digitalWrite(RED, HIGH);  
          delay(150);  
          digitalWrite(RED, LOW);  
          delay(150);  
        }  
        if (operadorbitwise==2)  
        {  
          digitalWrite(GREEN, HIGH);  
          delay(150);  
          digitalWrite(GREEN, LOW);  
          delay(150);  
        }  
        if (operadorbitwise==3)  
        {  
          digitalWrite(BLUE, HIGH);  
          delay(150);  
          digitalWrite(BLUE, LOW);  
          delay(150);  
        }  
      }  
      nvarrimento = nvarrimento + 1;  
    }  
    num=0;  
    setup(); //serve para chamar a funcao setup, princip  
    loop();  
  }  
}
```

```
if ((num-1) == numerocorreto) //o num-1 inclui o zero, pois o numerocorreto pode ser 0  
{  
  Serial.println("Parabens! Acertaste no numero correto! ");  
  unsigned long tempo=millis();  
  while ((millis()-tempo) < tempodosmodosWeL)  
  {  
    ledsdavitoria();  
  }  
}
```

Após os 7 segundos (como a condição dentro do while já não se verifica, o programa sai do while e continua), e se o número que o utilizador inseriu for igual ao numero correto (através da condição IF) é chamada a função ledsdavitoria, em que todos os leds piscam durante 5 segundos (usei o while com a variável tempodosmodosWeL para definir os 5 segundos (5000ms)).

Função ledsdavitória:

```
void ledsdavitória() //Se
{
  digitalWrite(13, HIGH);
  digitalWrite(12, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(6, HIGH);
  delay(200);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
  digitalWrite(10, LOW);
  digitalWrite(9, LOW);
  digitalWrite(8, LOW);
  digitalWrite(7, LOW);
  digitalWrite(6, LOW);
  delay(200);
  return;
}
```

LEDS LIGAM

LEDS DESLIGAM

Após piscarem os leds durante os 5 segundos, os leds ligam alternadamente, os verdes ligam da direita para a esquerda (varrimento) e os vermelhos da esquerda para a direita (varrimento) e o led RGB liga a cor do operador que foi usado no jogo. Após fazer este procedimento 3 vezes, este programa sai da condição, onde logo a seguir o número que o utilizador coloca para adivinhar o número volta ao valor 0 e são chamadas novamente as 2 funções: setup e loop.

```
else
{
  Serial.println("Nao acertaste no numero correto");
  unsigned long tempol=millis();
  for (int pin=6 ; pin<=13 ; pin++)
  {
    digitalWrite(pin, LOW);
  }
  while ((millis()-tempol) < tempodosmodosWeL)
  {
    if (operadorbitwise==1)
    {
      digitalWrite(RED, HIGH);
      delay(150);
      digitalWrite(RED, LOW);
      delay(150);
    }
    if (operadorbitwise==2)
    {
      digitalWrite(GREEN, HIGH);
      delay(150);
      digitalWrite(GREEN, LOW);
      delay(150);
    }
    if (operadorbitwise==3)
    {
      digitalWrite(BLUE, HIGH);
      delay(150);
      digitalWrite(BLUE, LOW);
      delay(150);
    }
  }
  num=0;
  setup();//serve para chamar a funcao setup, pri
  loop();// voltamos ao inicio do loop
}
}
```

```
while (nvarrimento<4)
{
  for (int i=0 ; i<=3 ; i++)
  {
    digitalWrite(10+i, HIGH);
    digitalWrite(9-i, HIGH);
    delay(100);
    digitalWrite(10+i, LOW);
    digitalWrite(9-i, LOW);
    delay(100);
    if (operadorbitwise==1)
    {
      digitalWrite(RED, HIGH);
      delay(150);
      digitalWrite(RED, LOW);
      delay(150);
    }
    if (operadorbitwise==2)
    {
      digitalWrite(GREEN, HIGH);
      delay(150);
      digitalWrite(GREEN, LOW);
      delay(150);
    }
    if (operadorbitwise==3)
    {
      digitalWrite(BLUE, HIGH);
      delay(150);
      digitalWrite(BLUE, LOW);
      delay(150);
    }
  }
  nvarrimento = nvarrimento + 1;
}
num=0;
setup(); //serve para chamar a fun
loop();
```

Se o número que o utilizador inseriu for diferente do número correto, os leds são desligados e o led RGB pisca durante durante 5 segundos (através da condição while para definir os 5 segundos) a cor do operador, e ao acabar o programa chama as funções setup e loop.

#### Dificuldades e comentários na realização desta tarefa:

O meu programa contém ainda alguns erros na parte do código, não consegui fazer corretamente o varrimento e não consegui fazer com que os leds dos números que foram gerados piscassem, por isso coloquei a piscar todos os leds. Senti que o random não funciona muito bem no Arduino, pois os números são quase sempre os mesmos e o operador bitwise também, fazendo desta forma a mesma operação entre os números. Na montagem física no Arduino, não montei corretamente o RGB pois ficou em falta uma resistência (no tinkercad está montado corretamente).