Programação em Linguagem C

Unidade 2

Nesta unidade estudaremos:

- ✓ If/else;
- ✓ for;
- ✓ while;
- ✓ switch case;
- ✓ Comunicação serial com PC;
- ✓ “apelidos” para os pinos do Arduino.

Relembrando...

Caso geral da instrução if-else:

```
if( expressão ){
  instrução1;
}else{
  instrução2;
}
```

O if-else funciona do seguinte modo:

1. O valor da expressão é calculado;
2. Se for verdadeiro, a instrução1 será executada, mas a instrução2 não;
   Se for falso, a instrução2 será executada, mas a instrução1 não.

Caso geral da instrução for:

```
for(expressão1; expressão2; expressão3 ){
  instrução;
}
```

A instrução for funciona do seguinte modo:

1. A expressão1 é executada apenas na primeira vez que o programa passa pelo ciclo for.
2. O valor da expressão2 é calculado e se for verdadeiro, o programa executa a expressão3 e as instruções dentro das chaves.
3. O processo repete-se, isto é, o valor da expressão2 é calculado novamente. Se for verdadeiro, o programa executa a expressão3 e as instruções dentro das chaves.
4. O ciclo **for** termina quando a expressão2 for falsa.

Caso geral da instrução while:

```
while(expressão) {
instrução;
}
```

A instrução while funciona do seguinte modo:

A expressão é testada, se for verdadeira, o programa executa a instrução. Depois de executar a instrução, o programa testa novamente a expressão, executa a instrução (caso verdadeira) e continua assim indefinidamente até que a expressão se torne falsa.

Caso geral da instrução switch case:

```
switch(expressão){
  case 3:
    instrução1;
    break;
  case 5:
    instrução2;
    break;
  default:
    instrução10; break;
}
```

A instrução switch case funciona do seguinte modo:

1. Logo após a palavra **switch**, calcula-se o valor expressão.
2. Baseado nesse valor, o programa salta para o caso apropriado. Por exemplo, se o valor for 5: o programa salta para case 5, executa instrução2 e prossegue com as instruções restantes até aparecer a instrução break. Esta instrução faz com que o computador salte para fora do switch.
3. O default é opcional e é executado se nenhum dos outros casos ocorrer.

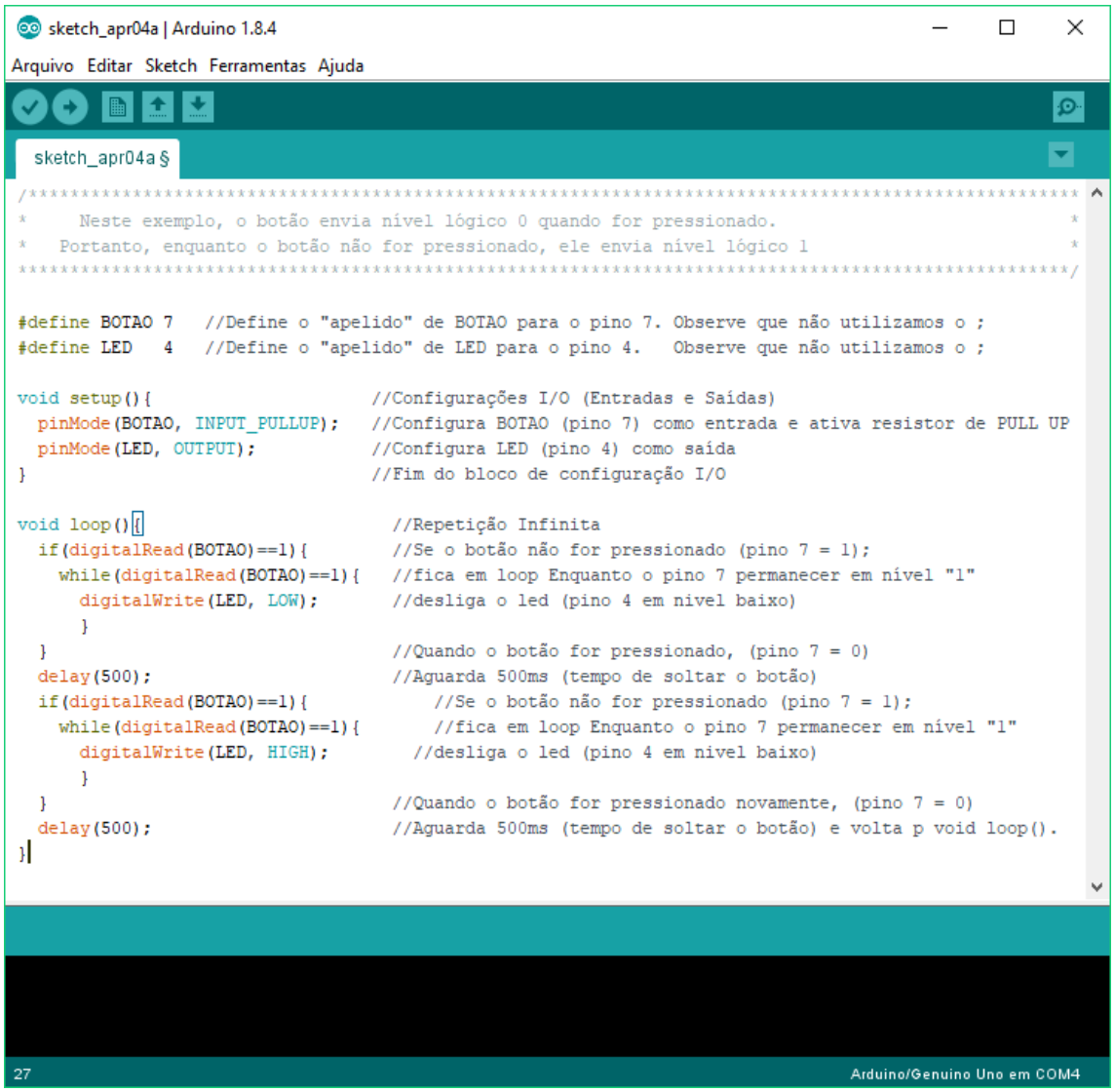
Obs.: O switch case funciona também com caractere (letra).

Exemplo 1: Controlar o acionamento de um LED através de um botão de pulso.

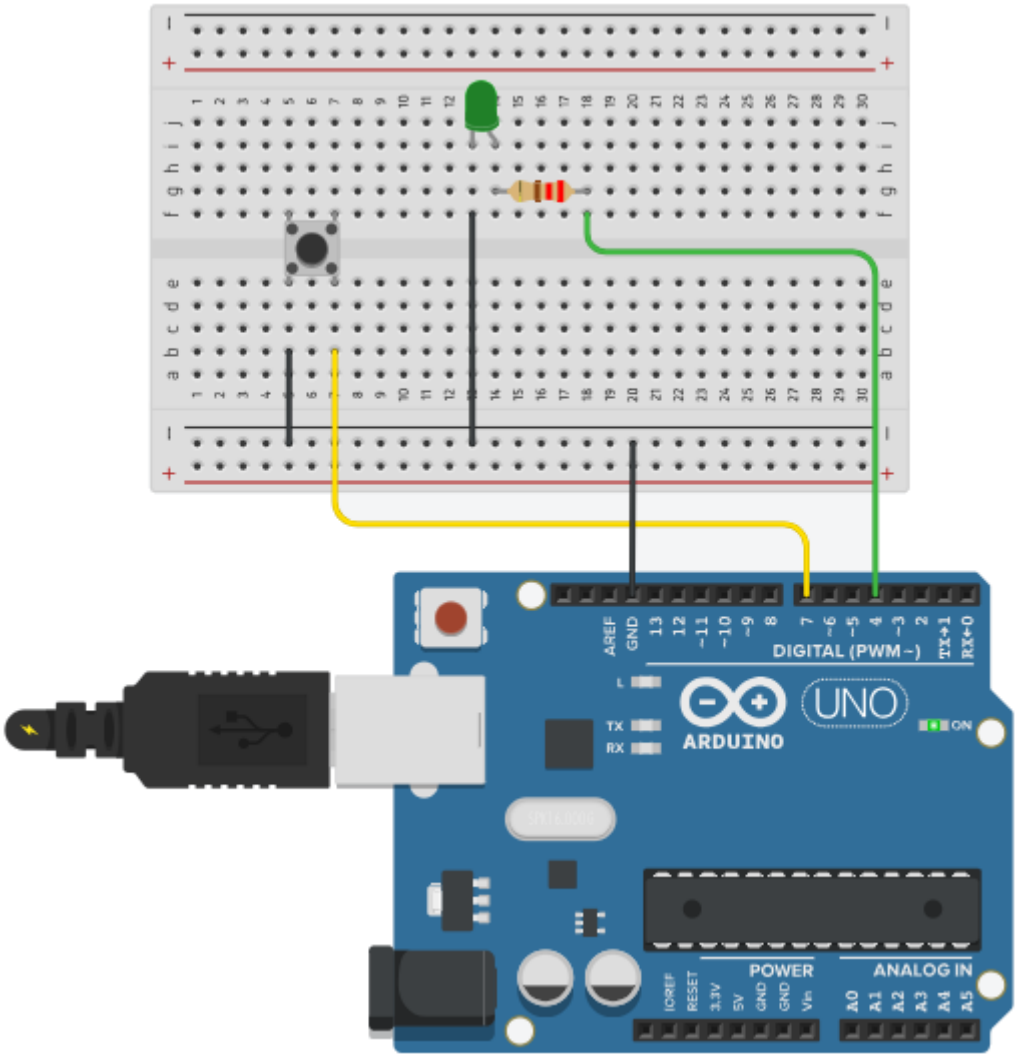
Dados.: Botão de pulo conectado ao pino 7 do Arduino, envia nível lógico 0 quando pressionado. LED conectado ao pino 4 do Arduino, acende com nível lógico 1.

Funcionamento.:

a) Inicialmente o LED deverá estar apagado;
b) O LED deverá alterar seu estado (de ligado para desligado e vice-versa) toda vez que o botão for pressionado.



Montagem



Exemplo 2: Faça um semáforo (com pedestre).

Dados:

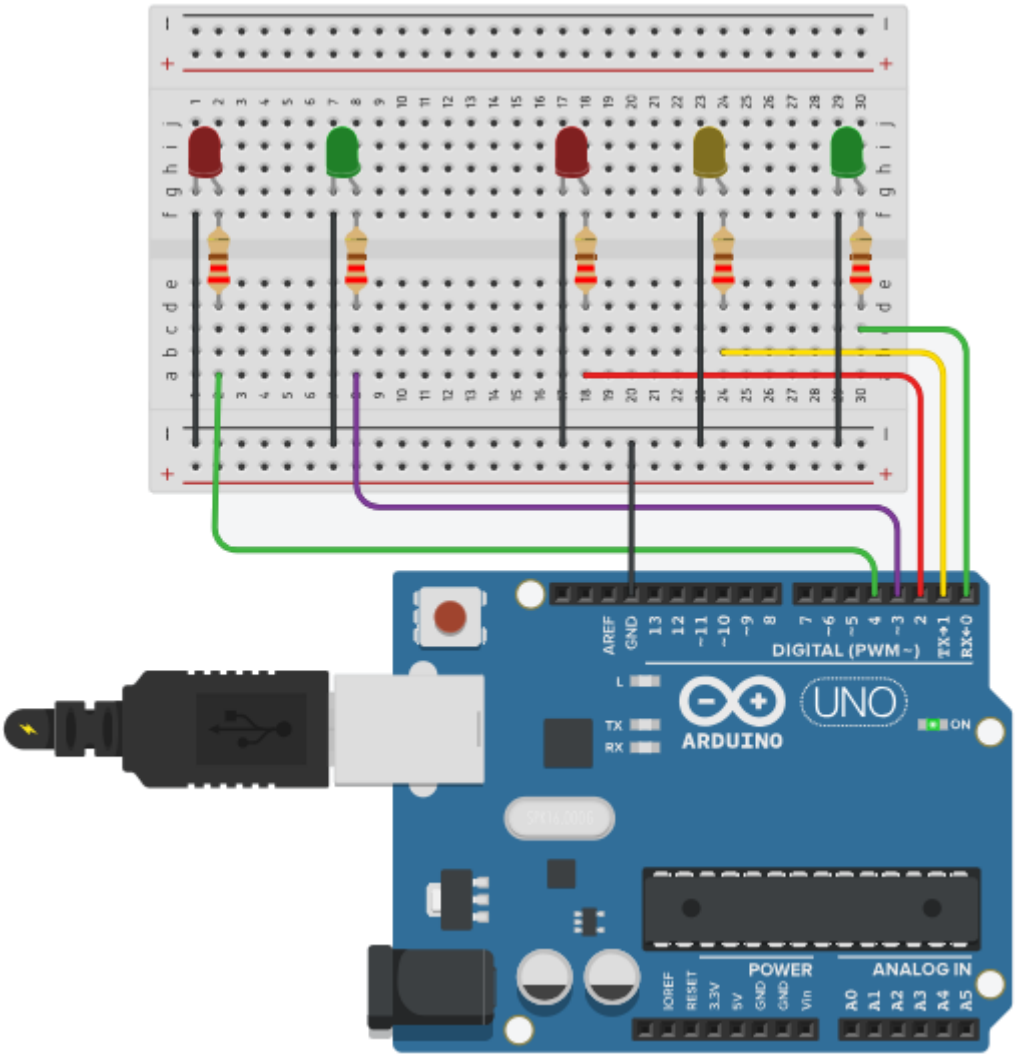
Veicular: VERDE pino 0, AMARELO pino 1, VERMELHO pino 2;
Pedestre: VERDE_P pino 4, VERMELHO_P pino 5.

Funcionamento.:

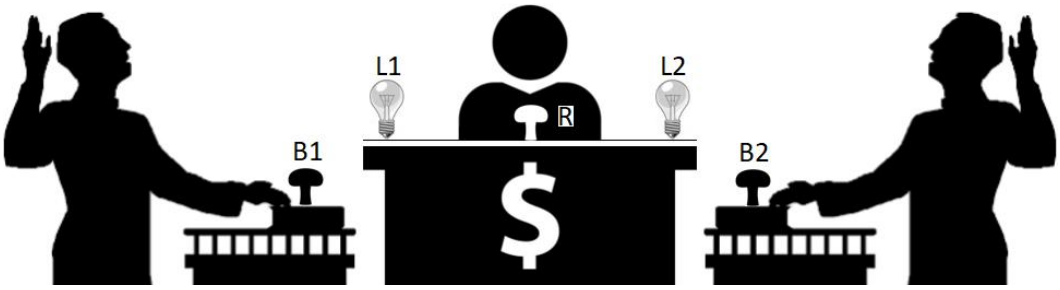
a) Os vermelhos devem permanecer ligados por 2s;
b) Ligue o VERDE por 10s e depois ligue o AMARELO por 3s;
c) Ligue o VERMELHO; Ligue o VERDE_P por 2s;
d) O VERMELHO_P deve piscar 4x com intervalos de 1s.
e) Retorne para o item b.


```
#define VERDE       0   //Define "apelido" VERDE para pino 0.      Observe que não utilizamos o ;
#define AMARELO     1   //Define "apelido" AMARELO para pino 1.    Observe que não utilizamos o ;
#define VERMELHO    2   //Define "apelido" VERMELHO para pino 2.   Observe que não utilizamos o ;
#define VERDE_P     4   //Define "apelido" VERDE_P para pino 4.    Observe que não utilizamos o ;
#define VERMELHO_P  5   //Define "apelido" VERMELHO_P para pino 5. Observe que não utilizamos o ;

void setup() {                    //Configurações I/O (Entradas e Saídas)
  pinMode(VERDE,      OUTPUT);    //Configura VERDE      (pino 0) como saída
  pinMode(AMARELO,    OUTPUT);    //Configura AMARELO    (pino 1) como saída
  pinMode(VERMELHO,   OUTPUT);    //Configura VERMELHO   (pino 2) como saída
  pinMode(VERDE_P,    OUTPUT);    //Configura VERDE_P    (pino 4) como saída
  pinMode(VERMELHO_P, OUTPUT);    //Configura VERMELHO_P (pino 5) como saída
}                                 //Fim do bloco de configuração I/O

void loop() {   //Inicio do bloco de loop (repetição)
  digitalWrite(VERMELHO, HIGH);   //liga vermelho veicular (pino 2)
  digitalWrite(VERMELHO_P, HIGH); //liga vermelho pedestre
  delay(2000);                    //Aguarda 2 segundos

  while (true) {  //while = Enquanto; true = Verdade;   (loop infinito)
    digitalWrite(VERMELHO_P, HIGH);     //liga vermelho pedestre (pino 5)
    digitalWrite(VERMELHO, LOW);        //desliga vermelho veicular (pino 2)
    digitalWrite(VERDE, HIGH);          //liga verde veicular (pino 0)
    delay(10000);                       //Aguarda 10 segundos
    digitalWrite(VERDE, LOW);           //desliga verde veicular (pino 0)
    digitalWrite(AMARELO, HIGH);        //liga amarelo veicular (pino 1)
    delay(3000);                        //Aguarda 3 segundos
    digitalWrite(AMARELO, LOW);         //desliga amarelo veicular (pino 1)
    digitalWrite(VERMELHO, HIGH);       //liga vermelho veicular (pino 2)
    digitalWrite(VERMELHO_P, LOW);      //desliga vermelho pedestre (pino 5)
    digitalWrite(VERDE_P, HIGH);        //liga verde pedestre (pino 4)
    delay(2000);                        //Aguarda 2 segundos
    digitalWrite(VERDE_P, LOW);         //desliga verde pedestre (pino 4)

    for(int i=0; i<4; i++){   //Repete este laço 4x.
       digitalWrite(VERMELHO_P, HIGH);  //liga vermelho pedestre (pino 5)
       delay(1000);                     //aguarda 1 segundos
       digitalWrite(VERMELHO_P, LOW);   //desliga vermelho pedestre (pino 5)
       delay(1000);                     //aguarda 1 segundo
    }                       //fim do laço da repetição 4x
  }
}
```

Montagem

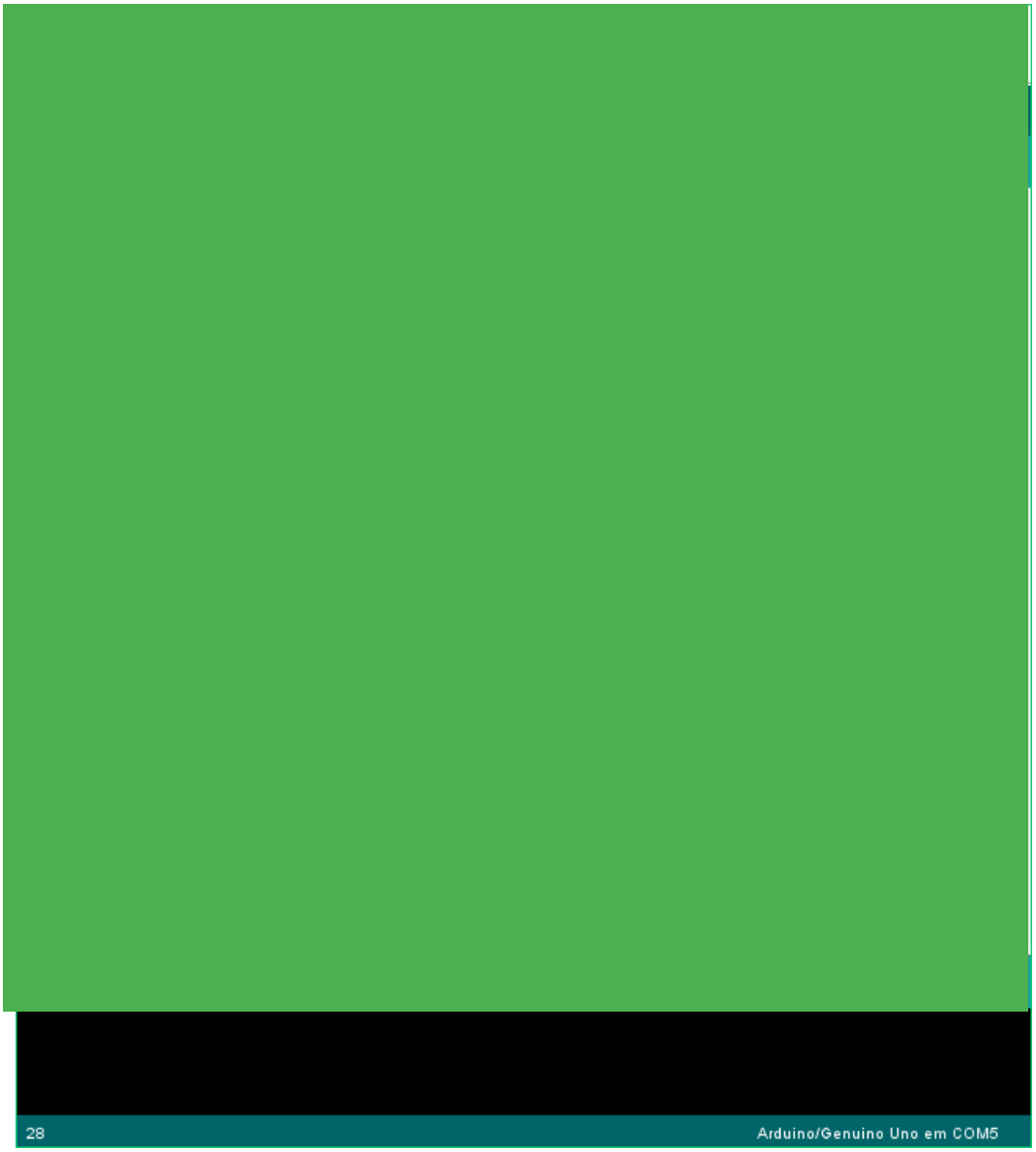


Exemplo 3: A imagem abaixo trata-se de um jogo de perguntas e respostas.

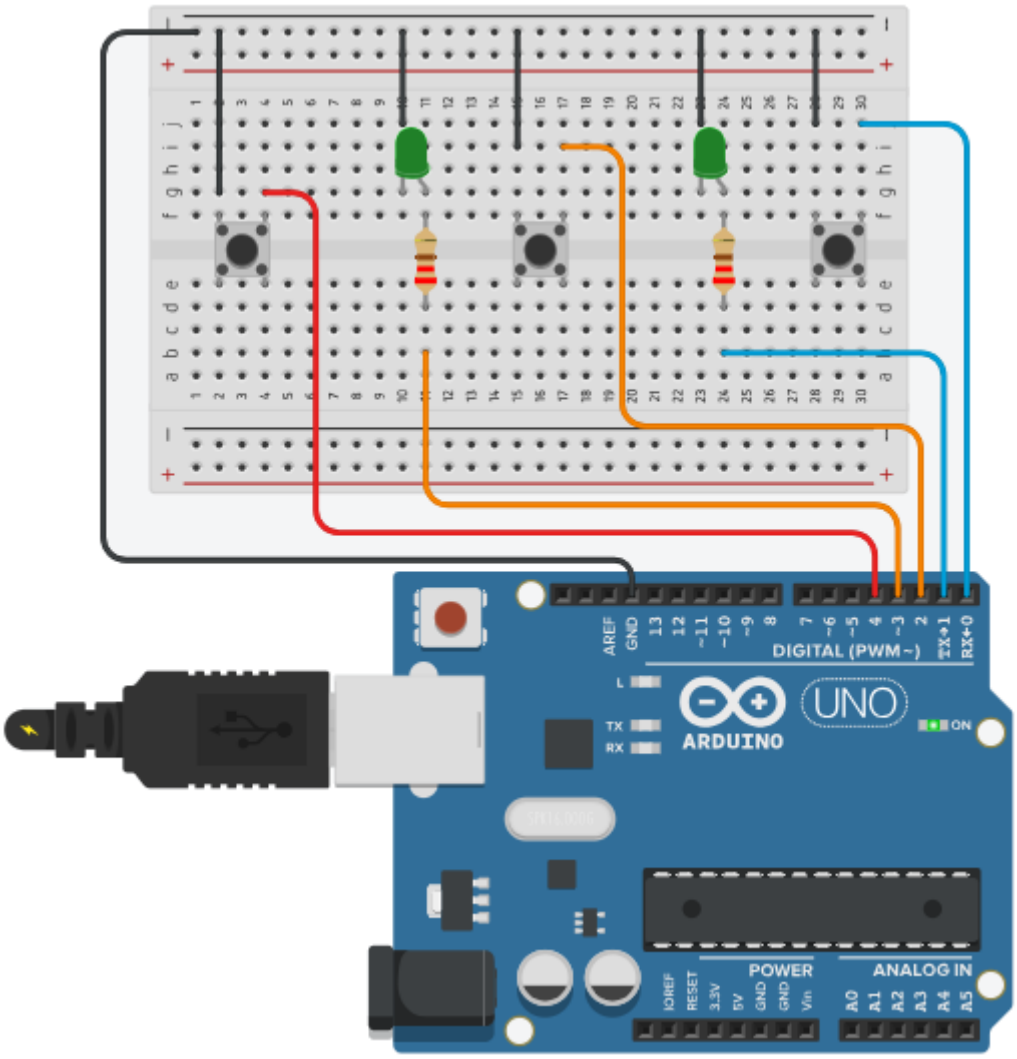


Deve ser feito o seguinte programa:

a) Se o botão B1 for pressionado primeiro que B2, a lâmpada L1 deverá acender e somente deverá apagar quando o botão R for pressionado pelo apresentador;
b) Se o botão B2 for pressionado primeiro que B1, a lâmpada L2 deverá acender e somente deverá apagar quando o botão R for pressionado pelo apresentador;
c) Se uma lâmpada estiver acesa a outra não poderá acender.



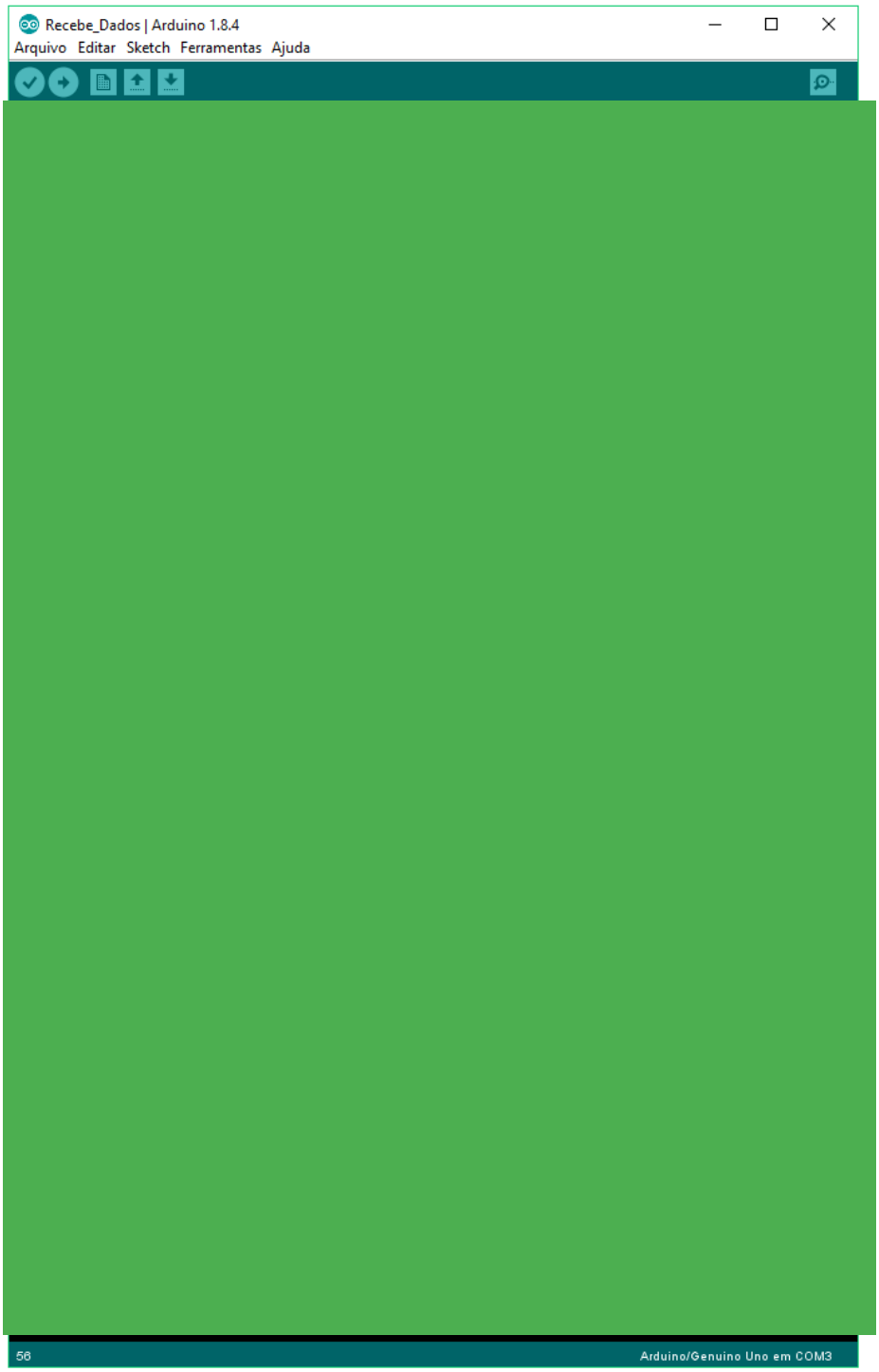
Montagem



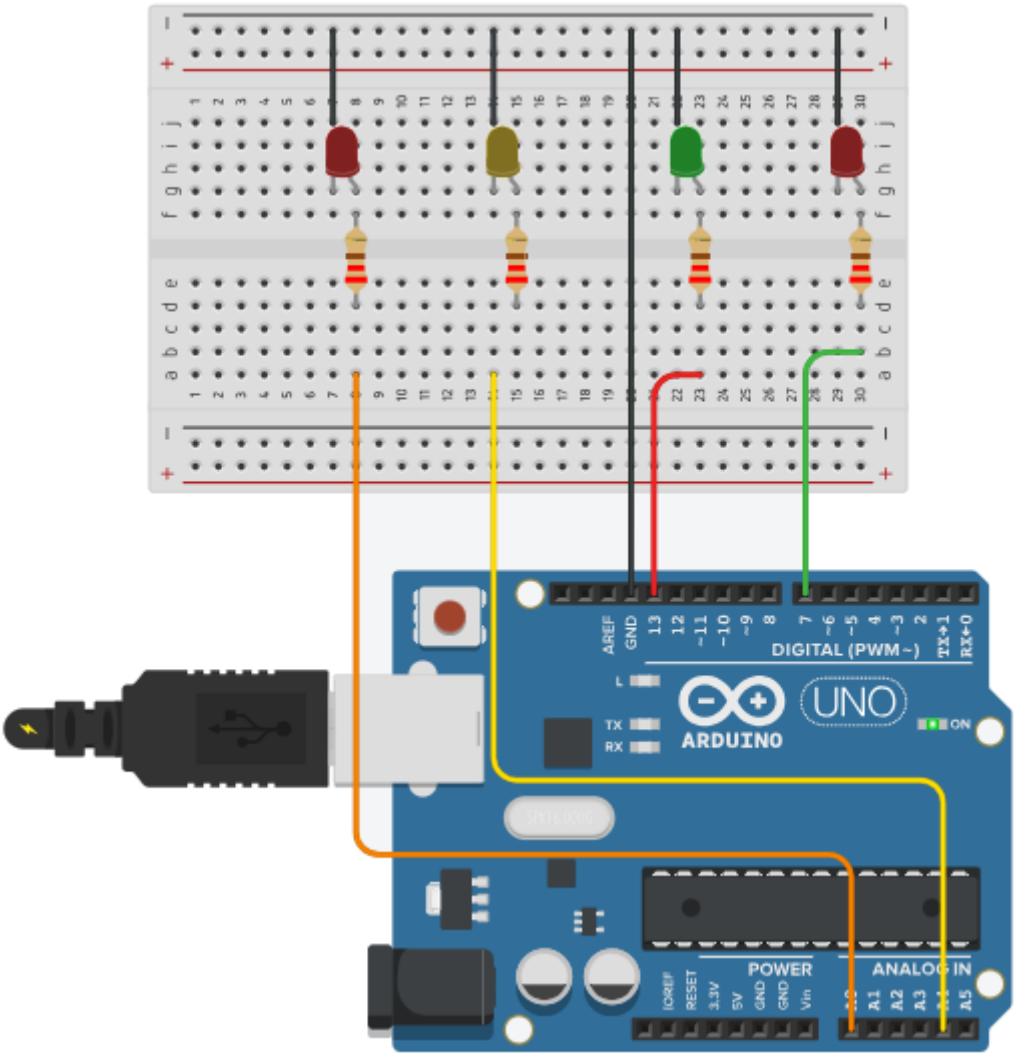
Exemplo 4: Recebendo dados do PC e Tratando com o switch case

Faça um programa que:

a) Após receber a letra “T” enviada pelo computador, ligue o LED conectado ao pino 13 do Arduino;
b) Após receber a letra “a”, pisque um LED conectado ao pino 7 três vezes;
c) Após receber a letra “t”, ligue o LED conectado ao pino A0;
d) Após receber a letra “A”, pisque um LED conectado ao pino A4 cinco vezes;
e) Desligue todos os LED’s quando o Arduino receber a letra (P) enviada pelo computador;



Montagem



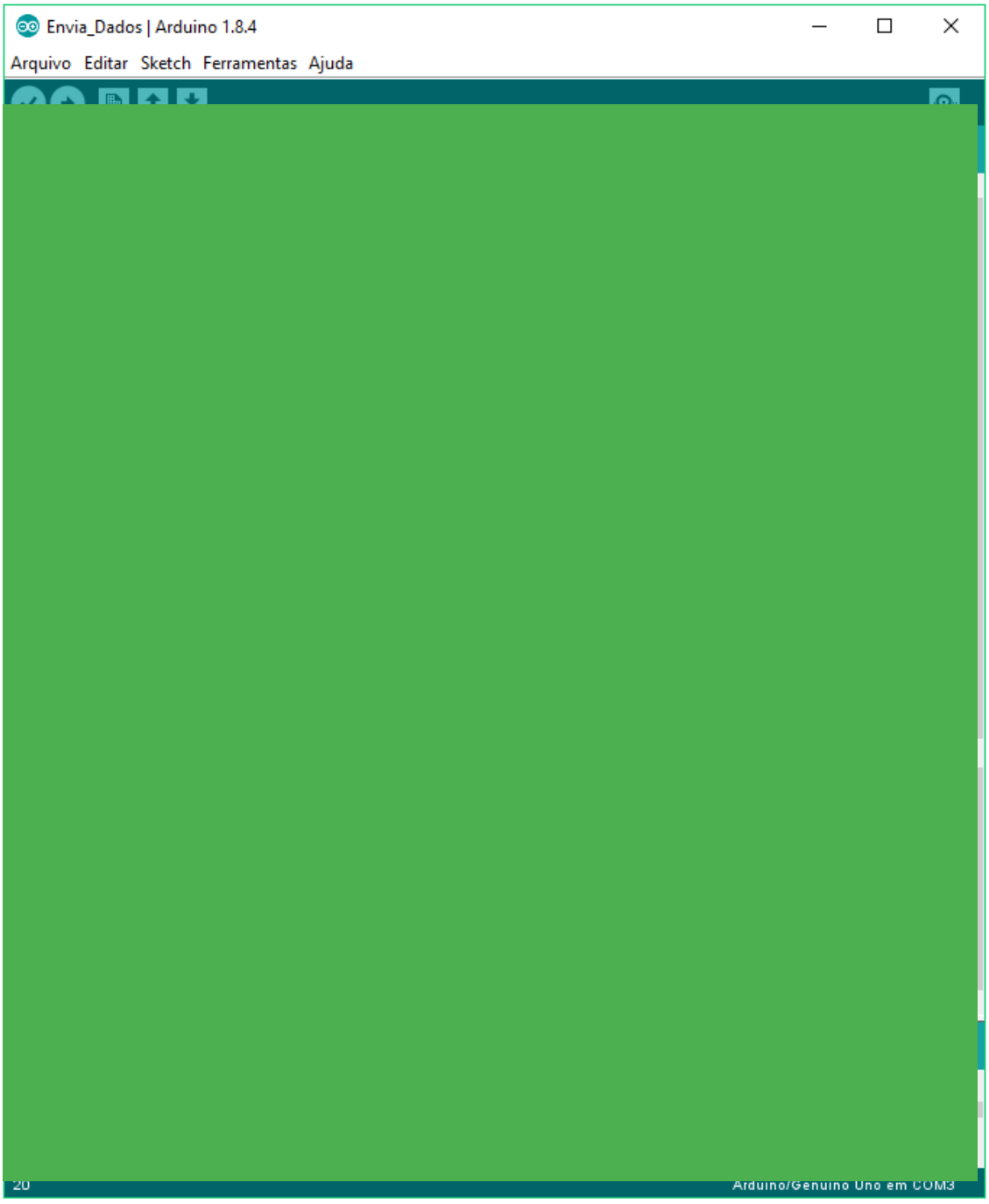
Para enviar dados do computador para o Arduino, utilize o monitor serial. No final desta unidade, é demonstrado como utilizar o monitor serial.

Exemplo 5: Enviando dados para o PC

Utilizando botões com retenção (trava), faça um programa que:

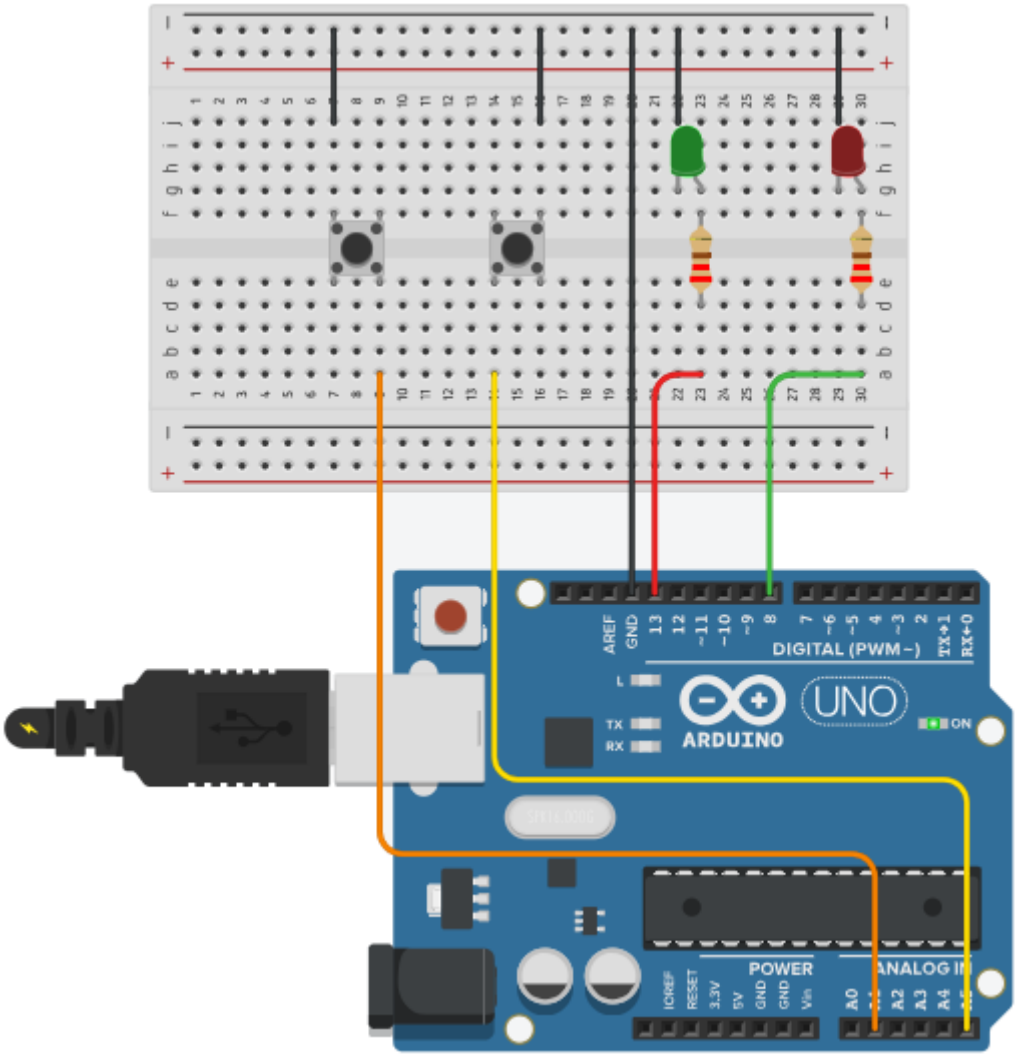
a) Se o botão BT_1 (pino A1) estiver fechado, ligue o LED_1 (pino 13) e envie a mensagem: Saída 1 Ativada. Se BT_1 aberto, desligue LED_1 e envie a mensagem: Saída 1 Desativada.

b) Se o botão BT_2 (pino A5) estiver fechado, ligue o LED_2 (pino 8) e envie a mensagem: Saída 2 Ativada. Se BT_2 aberto, mensagem: Saída 2 Desativada.



NOTA.: Neste exemplo, de 1 em 1 segundo será escrito o estado das saídas.

Montagem



Para enviar dados do Arduino para o computador, utilize o monitor serial. No final desta unidade, é demonstrado como utilizar o monitor serial.

Monitor Serial, comunicação entre Arduino e PC

Introdução

O Arduino consegue se comunicar com o PC através da porta USB. Essa comunicação se dá através de um processo chamado “comunicação serial”, que podemos acessar pelo “Monitor Serial” no IDE do Arduino. Essa comunicação é feita em duas vias, ou seja, enviando e recebendo dados.

Materiais utilizados neste tutorial

- ✓ 01 Arduino UNO;
- ✓ 01 Cabo USB.

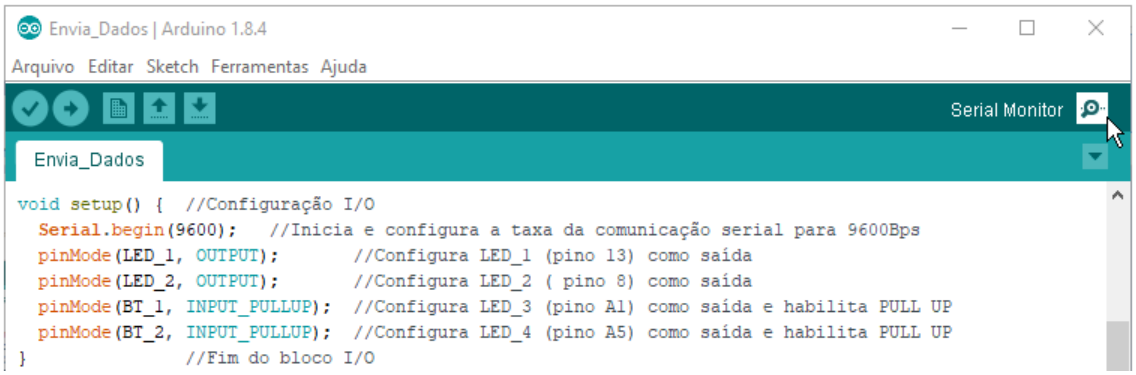
Montagem

1) Conecte o Arduino ao PC por meio de um cabo USB;
2) Abra a IDE (software utilizado para digitar nossos programas do Arduino);
3) Selecione a placa e a COM. Caso não lembre como fazer, leia a unidade 1.

Monitor Serial

Após fazer “Upload” do código no Arduino, abra o Monitor Serial.

(Para abri-lo clique em Serial Monitor assim como mostrado na imagem a baixo).



O Monitor Serial tem a seguinte aparência:

Selecione a mesma velocidade de comunicação inserida no programa. Para um melhor entendimento, observe as duas imagens.

Para enviar dados do PC ao Arduino, escreva na janela e clique em Send. Utilizando esta imagem como exemplo, se clicarmos em Send, será enviado a letra A.

Se o Arduino enviar dados, os mesmos serão exibidos no espaço em branco.