

# Cap 2 – Recebendo dados

Paula Laysa Amorim Silva

Anotações:

Variáveis em C++ possuem uma propriedade chamada de escopo. Uma variável local é visível apenas onde é declarada, já uma variável local pode ser vista por todas as funções.

Const é uma abreviação de constante, é uma variável que não se pode dar um valor.

PinMode() configura um pino específico para entrada ou saída.

DigitalRead() lê o valor de um pino digital específico, podendo ser high ou low.

DigitalWriter() aciona o valor de high ou low em um pino digital.

! (não lógico) resulta em verdadeiro se o operador é falso.

&& (lógico) resulta em verdadeiro, apenas se ambos os operadores forem verdadeiros.

|| (lógico) resulta em verdadeiro se pelo menos um dos operadores forem verdadeiros.

Entradas analógicas só podem ter 2 estados, ou high ou low, ou 0 V ou 5 V.

analogRead() lê o valor de um pino analógico específico.

Float é um número racional.

O TMP36 é um componente eletrônico digital para projetos que necessitam de monitoramento de temperatura, basicamente é um sensor para medir temperatura no ambiente.

Map() são valores dentro de uma faixa para valores dentro de outra faixa.

PWM é a variação do valor médio de onda periódica.

AnalogWriter() escreve um valor PWM em um pino digital que possui função PWM

Exercício:

1- Uma variável local é visível apenas onde é declarada, já uma variável local pode ser vista por todas as funções.

2-a)

```
for (byte i = 0; i < 4; i++) {  
    bool bt_atual = digitalRead(BOTAO[i]);
```

```

if (bt_atual == 1) {
    digitalWrite (LED[i], 1);
} else {
    digitalWrite (LED[i], 0);
}

```

b)

```

for (byte i = 0; i < 4; i++) {
    bool bt_atual = digitalRead(BOTAO[i]);
    if (bt_atual == 1) {
        digitalWrite (LED[i], 0);
    } else {
        digitalWrite (LED[i], 1);
    }
}

```

3- a)

```

const byte LED = 13;
const byte BOTAO[] = {5, 4};

void setup() {
    for (byte i = 0; i < 2; i++) {
        pinMode ( LED, OUTPUT );
        pinMode ( BOTAO[i], INPUT );
    }
}

void loop() {
    delay (10);
    for (byte i = 0; i < 2; i++) {
        bool bt_atual = digitalRead(BOTAO[i]);
        if (bt_atual == 1 && BOTAO[i] == 5) {
            digitalWrite (LED, 1);
        } else if (bt_atual == 1 && BOTAO[i] == 4) {
            digitalWrite(LED, 0);
        }
    }
}

```

b)

```

const byte LED = 13;
const byte BOTAO[] = {5, 4};

void setup() {
    Serial.begin(9600);
    pinMode ( LED, OUTPUT );
    for (byte i = 0; i < 2; i++) {
        pinMode ( BOTAO[i], INPUT );
    }
}

```

```

}

void loop() {
  delay (100);
  bool bt1 = digitalRead(BOTA0[0]);
  bool bt2 = digitalRead(BOTA0[1]);
  Serial.print(bt1);
  Serial.print(" | ");
  Serial.println(bt2);

  if (bt1 && bt2) {
    digitalWrite(LED, HIGH);
  } else if (!bt1 && !bt2) {
    digitalWrite(LED, LOW);
  }
};
c)
const byte LED = 13;
const byte BOTA0[] = {5, 4};

void setup() {
  Serial.begin(9600);
  pinMode ( LED, OUTPUT );
  for (byte i = 0; i < 2; i++) {
    pinMode ( BOTA0[i], INPUT );
  }
}

void loop() {
  delay (100);
  bool bt1 = digitalRead(BOTA0[0]);
  bool bt2 = digitalRead(BOTA0[1]);
  Serial.print(bt1);
  Serial.print(" | ");
  Serial.println(bt2);

  if (bt1 && bt2) {
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
};

4-
const byte LED = 13;

void setup() {
  Serial.begin(9600);
  pinMode ( LED, OUTPUT );
}

void loop() {
  delay (10);
  char c;
  byte n;

  if (Serial.available() > 0) {
    c = Serial.read();
    n = c;

    if (n == 65 && digitalRead(LED) == LOW) {
      digitalWrite(LED, HIGH);
    }
  }
}

```

```

        Serial.println("LED ligado!");
    } else if (n == 97 && digitalRead(LED) == HIGH) {
        digitalWrite(LED, LOW);
        Serial.println("LED desligado!");
    }
}
};

```

5-

```
const byte LED = 13;
```

```

void setup() {
    Serial.begin(9600);
    pinMode ( LED, OUTPUT );
}

```

```

void loop() {
    delay (10);
    char c;
    byte n;

    if (Serial.available() > 0) {
        c = Serial.read();
        n = c;

        if (n >= 65 && n <= 90 && digitalRead(LED) == LOW) {
            digitalWrite(LED, HIGH);
            Serial.println("LED ligado!");
        } else if (n >= 97 && n <= 122 && digitalRead(LED) == HIGH) {
            digitalWrite(LED, LOW);
            Serial.println("LED desligado!");
        }
    }
};

```

6- O valor máximo é 1023 e o valor mínimo é 0.

7-

```
const byte POT = A0;
```

```

void setup() {
    Serial.begin(9600);
    pinMode(13, OUTPUT);
}

```

```

void loop() {
    delay (10);
    float valor_pot = analogRead(POT);
    Serial.println( valor_pot );
    if ( valor_pot >= 1023/2 ) {
        digitalWrite(13, HIGH);
    } else {
        digitalWrite(13, LOW);
    }
}

```

```
}
```

8-

```
const byte POT = A0;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  delay (10);  
  int valor_pot = analogRead(POT);  
  Serial.println( valor_pot );  
  if (valor_pot <= 341) {  
    digitalWrite(13, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(11, LOW);  
  } else if (valor_pot >= 342 && valor_pot <= 682) {  
    digitalWrite(12, HIGH);  
    digitalWrite(13, LOW);  
    digitalWrite(11, LOW);  
  } else if (valor_pot >= 683) {  
    digitalWrite(11, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(13, LOW);  
  }  
}
```

9-

```
const byte POT = A0;  
const byte LDR = A1;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  delay (10);  
  float valor_pot = analogRead(POT);  
  float valor_ldr = analogRead(LDR);  
  
  Serial.print( valor_pot );  
  Serial.print( " | " );  
  Serial.println( valor_ldr );  
  if ( valor_ldr < valor_pot ) {  
    digitalWrite(13, HIGH);  
  } else {  
    digitalWrite(13, LOW);  
  }  
}
```

10- a) Dados: -40, 20 e 124, 358.

```

b)
const byte POT = A0;
const byte LDR = A1;
const byte TMP = A2;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {
  delay (10);
  int valor_pot = analogRead(POT);
  int valor_ldr = analogRead(LDR);
  int valor_tmp = analogRead(TMP);
  int ctmp = map(valor_tmp, 20, 358, -40, 125);

  Serial.print( valor_pot );
  Serial.print( " | " );
  Serial.print( valor_ldr );
  Serial.print( " | " );
  Serial.print( valor_tmp );
  Serial.print( " / " );
  Serial.println( ctmp );
  if ( valor_ldr < valor_pot ) {
    digitalWrite(13, HIGH);
  } else {
    digitalWrite(13, LOW);
  }
}

```

```

c)

const byte POT = A0;
const byte LDR = A1;
const byte TMP = A2;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {
  delay (10);
  int valor_pot = analogRead(POT);
  int valor_ldr = analogRead(LDR);
  int valor_tmp = analogRead(TMP);
  int ctmp = map(valor_tmp, 20, 358, -40, 125);
  float mapa = (((double)valor_tmp/1024) * 5) - 0.5) * 100;

  Serial.print( valor_pot );
  Serial.print( " | " );
  Serial.print( valor_ldr );
  Serial.print( " | " );
  Serial.print( valor_tmp );
  Serial.print( " / " );
  Serial.print( ctmp );
  Serial.print( " / " );
  Serial.println( mapa );
  if ( valor_ldr < valor_pot ) {

```

```

    digitalWrite(13, HIGH);
} else {
    digitalWrite(13, LOW);
}
}

```

12- O pino não é PWM, pois quando chega na metade da contagem ele acende e fica assim até o reiniciar a contagem (255).

13-

```

const byte LED = 11;
const byte BTN1 = 5;
const byte BTN2 = 4;

```

```

void setup() {
  pinMode ( LED, OUTPUT );
  pinMode ( BTN1, INPUT );
  pinMode ( BTN2, INPUT );
  Serial.begin ( 9600 );
}

```

```

void loop() {
  delay (50);
  static byte n = 1;
  bool p1 = digitalRead(BTN1);
  bool p2 = digitalRead(BTN2);
  analogWrite(LED, n);
  Serial.println(n);
  if (n > 0 && n < 255) {
    if (p1) {
      n++;
    } else if (p2) {
      n--;
    }
  } else {
    Serial.println("Limite atingido!");
    if (n == 0) { n = 1; }
    if (n == 255) { n = 254; }
  }
}

```

14- 6 outros pinos.

```

const byte LED = 10;
const byte POT = A0;

```

```

void setup() {
  pinMode ( LED, OUTPUT );
  pinMode ( POT, INPUT );
  Serial.begin ( 9600 );
}

```

```

void loop() {
  delay (50);
  int v = analogRead(POT);
  int praOrganizar = map(v, 0, 1023, 0, 255);
  analogWrite(LED, praOrganizar);
  Serial.print(v);
  Serial.print(" | ");
  Serial.println(praOrganizar);
}

```