

## Sistemas Embarcados

# Desenvolvimento de códigos

Tiago Piovesan Vendruscolo

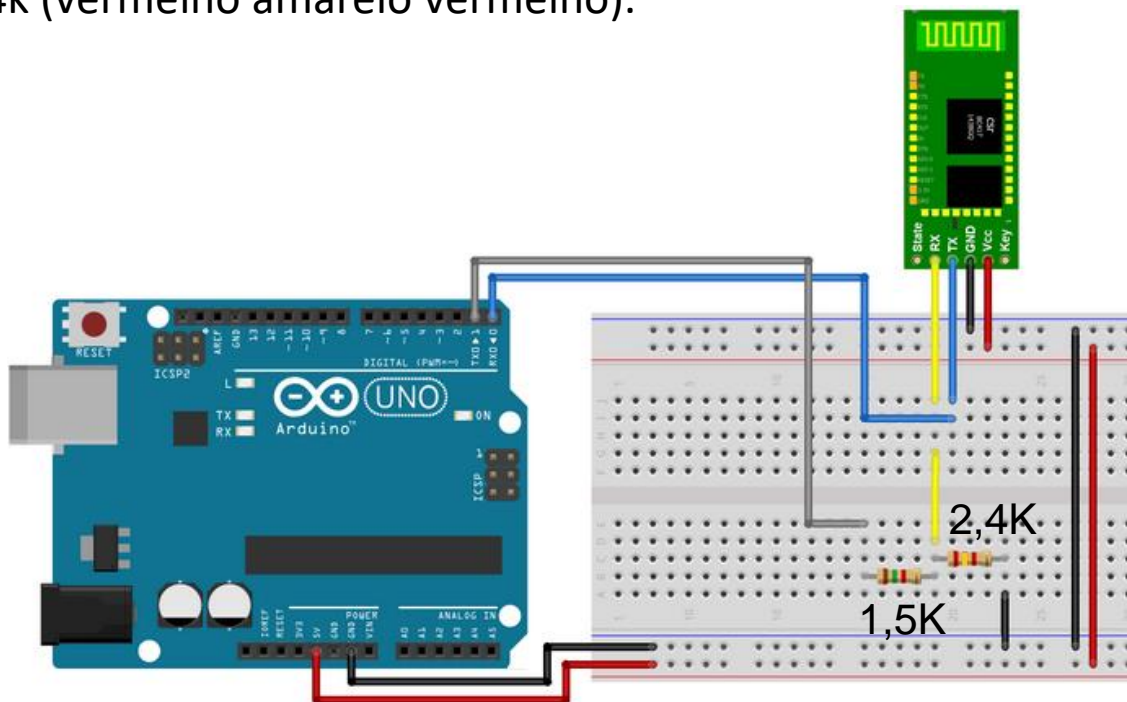
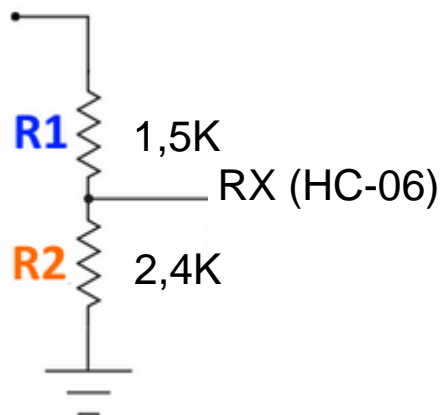


Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito aos autores originais. [4.0 international](https://creativecommons.org/licenses/by-nc-nd/4.0/)

# Módulo bluetooth

- O módulo bluetooth HC-06 trabalha exatamente como uma porta serial, não sendo necessário utilizar bibliotecas. Deve-se tomar cuidado com o pino de RX, que em alguns módulos é 5V e em outros é 3.3V. No nosso caso, é 3.3V, então é necessário utilizar um divisor de tensão, formado por um resistor de 1,5k (marrom verde vermelho) e outro de 2,4k (vermelho amarelo vermelho).

TX (Arduino)



- Como o canal serial utilizado pelo módulo é o mesmo de comunicação com o computador para a gravação, devemos deixar a alimentação do módulo desconectada quando formos gravar o Arduino.

# Módulo bluetooth

- Exemplo 1: Imprima a palavra “teste” na serial
  - Desconecte o módulo bluetooth
  - Faça a gravação no Arduino
  - Reconecte o módulo

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    delay(500);  
    Serial.println("teste");  
}
```

- Instale no celular o aplicativo Arduino bluetooth controller (Giuming Apps) ou Arduino bluetooth control (Broxcode).
- Conecte no dispositivo HC-06 modulo X (etiqueta), senha 1234.
- Utilize o modo terminal.

- Funções para a leitura da serial:
  - *Utilizado para verificar se possui uma nova informação na serial:*

```
Serial.available()
```

```
if (Serial.available() > 0)
```

- *Função para ler a serial:*

```
Serial.read()
```

- *Função para ler um integer:*

```
Serial.parseInt();  
x = Serial.parseInt();
```

Cuidado ao utilizar esse tipo de função, pois ele ocupa o processador até ser concluído ou estourar o timeout. Não utilize em códigos multitarefas. Ou caso decida utilizar, setar o timeout para um valor adequado.

# Módulo bluetooth

- Exemplo 2: Faça a leitura da serial, de forma que quando você digitar um valor, ele imprima o valor digitado.
  - Desconecte o módulo bluetooth
  - Faça a gravação no Arduino
  - Reconecte o módulo

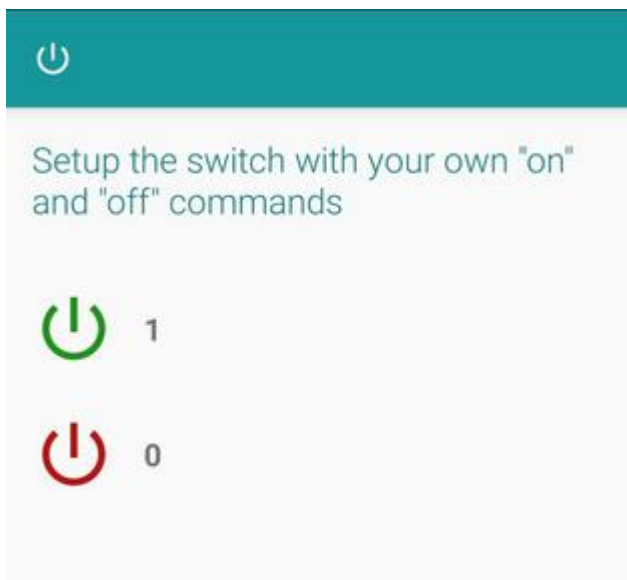
```
int x;

void setup() {
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    x = Serial.parseInt();
    Serial.println(x);
  }
}
```

# Módulo bluetooth

- Exercício 1: Faça a leitura da serial, de forma que quando você digitar “1” irá ligar um LED no pino 7, e quando digitar “0” irá desligar. Utilize o Switch mode do app
  - Desconecte o módulo bluetooth
  - Faça a gravação no Arduino
  - Reconecte o módulo



```
#define LED0 7

int x;

void setup() {
  Serial.begin(9600);
  pinMode(LED0, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    x = Serial.parseInt();
    Serial.println(x);
  }
  if (x == 0) {
    digitalWrite(LED0, LOW);
  }
  if (x == 1) {
    digitalWrite(LED0, HIGH);
  }
}
```

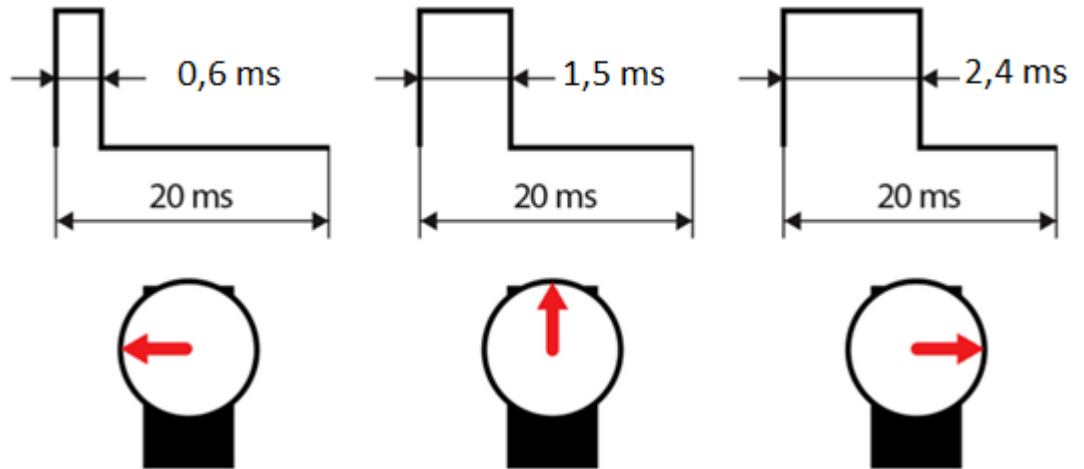
# Utilização de Bibliotecas

- Faça o controle o servo motor, de forma que ele mude a angulação proporcionalmente com o potenciômetro.

Pin Potenciômetro	Função
1	5 V
2 (meio)	A0
3	GND

Cabo Servo	Função
Vermelho	5 V
Marrom	GND
Laranja	Controle (9)

PWM=Orange (┐┌)  
Vcc = Red (+)  
Ground=Brown (-)



# Utilização de Bibliotecas

## ■ Arquivo > Exemplos > Servo > Knob

```
#include <Servo.h>
```

Biblioteca para o controle de um Servo motor

```
Servo myservo; // cria o objeto
```

```
int pot_valor;
```

Pino em que o servo está conectado

```
void setup() {  
    myservo.attach(9);  
}
```

$$\text{Map} = (\text{val} - \text{in\_min}) * (\text{out\_max} - \text{out\_min}) / (\text{in\_max} - \text{in\_min}) + \text{out\_min};$$

```
void loop() {  
    pot_valor = analogRead(A0);  
    pot_valor = map(pot_valor, 0, 1023, 0, 180);  
    myservo.write(pot_valor);  
    delay(15);  
}
```



# Módulo bluetooth

- Exercício 2: Utilizando o terminal do celular, faça o controle do servo motor, digitando o ângulo em que deseja posicionar o servo.
  - Desconecte o módulo bluetooth
  - Faça a gravação no Arduino
  - Reconecte o módulo

Cabo Servo	Função
Vermelho	5 V
Marrom	GND
Laranja	Controle (9)

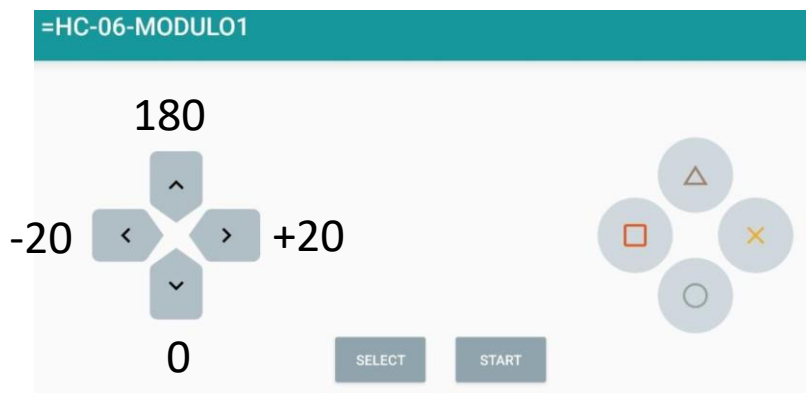
```
#include <Servo.h>
int x;
Servo myservo; // cria o objeto

void setup() {
  myservo.attach(9);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    x = Serial.parseInt();
    Serial.println(x);
  }
  myservo.write(x);
  delay(500);
  Serial.println(x);
}
```

# Módulo bluetooth

- Exercício 3: Utilizando o terminal do celular, no modo joystick, faça o controle do servo motor, de forma que:



```
#include <Servo.h>
int x,posicao;
Servo myservo; // cria o objeto

void setup() {
  myservo.attach(9);
  Serial.begin(9600);
}
```

Cabo Servo	Função
Vermelho	5 V
Marrom	GND
Laranja	Controle (9)

Arquivo no MOODLE

```
void loop() {
  if (Serial.available()>0){
    x = Serial.parseInt();
  }
  if(x==1){
    posicao = posicao+20;
    if (posicao>180) {posicao=180;}
    Serial.println(posicao);
  }
  if(x==2){
    posicao = posicao-20;
    if (posicao<0) {posicao=0;}
    Serial.println(posicao);
  }
  if(x==3){
    posicao = 0;
    Serial.println(posicao);
  }
  if(x==4){
    posicao = 180;
    Serial.println(posicao);
  }
  x=0;
  myservo.write(posicao);
  delay(100);
}
```

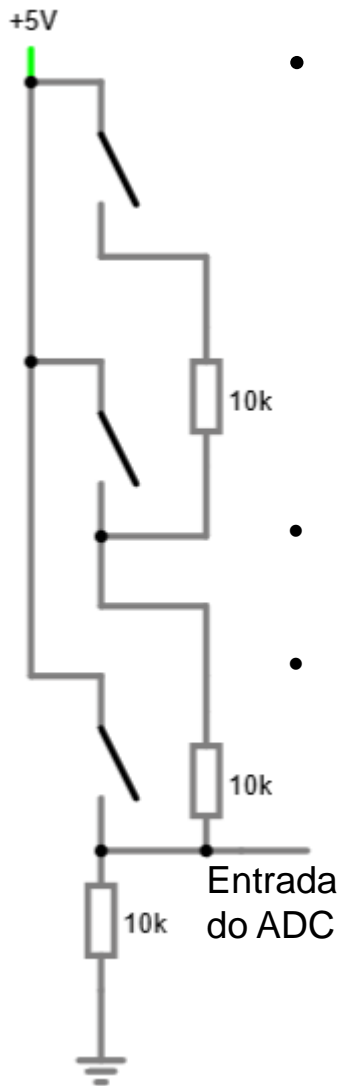
# Ligando mais de um botão em uma entrada analógica

# Ligando mais de um botão em uma entrada analógica

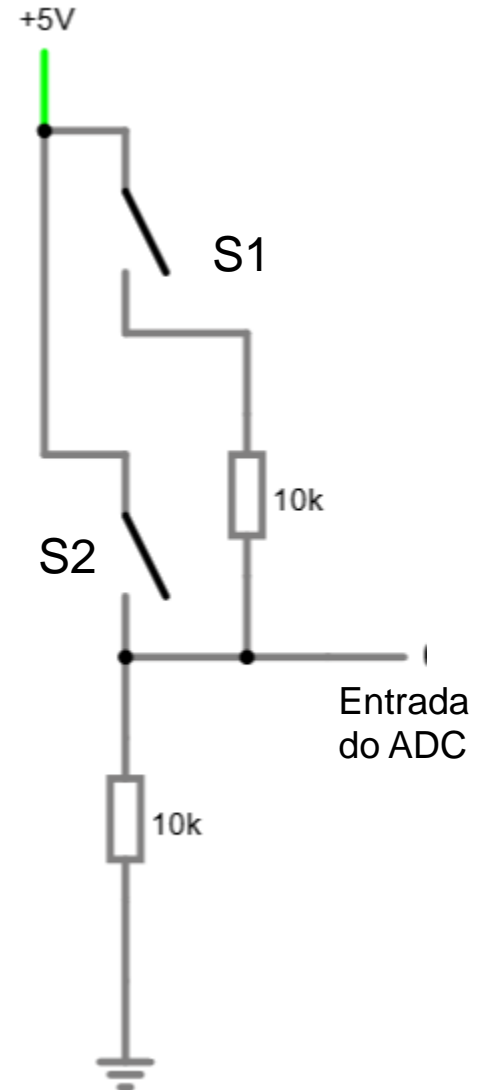
- Caso precisamos ligar vários botões e o microcontrolador não tenha mais entradas digitais disponíveis, porém possui uma entrada A/D, é possível fazer um teclado analógico. Esse teclado pode ser feito de duas formas:
  - Divisor resistivo
  - Rede R-2R

# Ligando mais de um botão em uma entrada analógica

- Divisor resistivo:
  - Nesse modelo teremos a queda de tensão equivalente à chave pressionada. Nesse exemplo, teremos 2.5V pressionando a chave S1 e 5V pressionando a chave S2. Diversas chaves podem ser ligadas nessa configuração.
- Vantagem
  - Simplicidade
- Desvantagem
  - Não é possível detectar quando mais de uma chave foi pressionada. A leitura será sempre equivalente à chave mais próxima da saída.



<https://tinyurl.com/2pq7kfun>



<https://tinyurl.com/2htcckpv>

# Ligando mais de um botão em uma entrada analógica

- Exercício 4: Faça o exemplo com 2 botões e imprima qual botão está sendo pressionado. Primeiro leia a entrada analógica pura, para escolher os ranges corretamente.

```
#define entrada_analogica A0

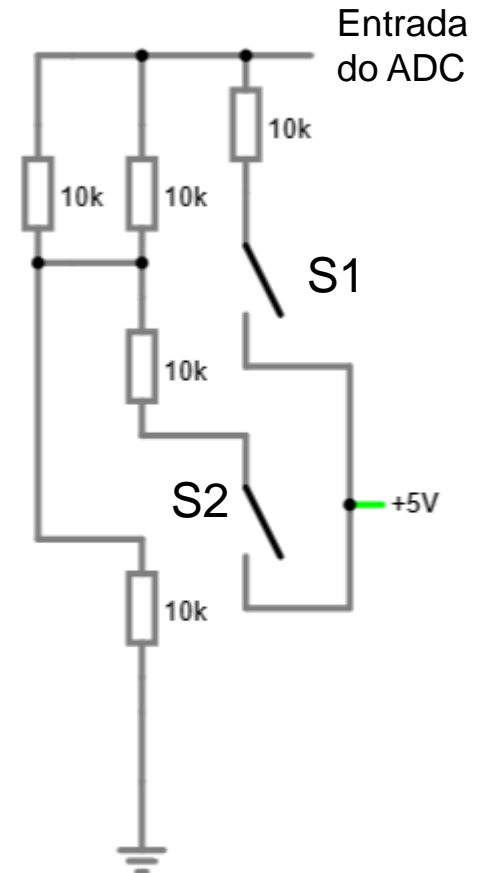
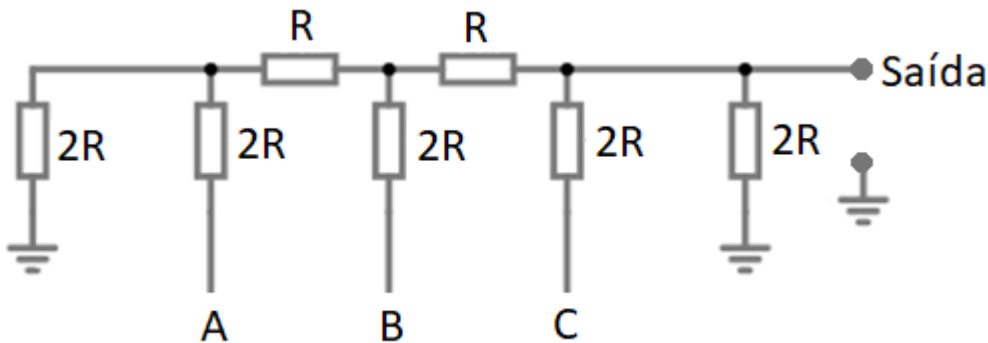
int leitura = 0;
int x = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    leitura = analogRead(entrada_analogica);
    if (leitura>470 && leitura<550 && x==0){
        Serial.println("Botão 1 pressionado");
        x=1;
    }
    else if (leitura>950 && x==0){
        Serial.println("Botão 2 pressionado");
        x=1;
    }
    else if (leitura<50 && x==1){
        x=0;
    }
}
```

# Ligando mais de um botão em uma entrada analógica

- R-2R:
  - Esse método utiliza 2 valores de resistores e também é bastante utilizado como conversor DAC (digital-analógico) que vimos nas aulas de sistemas digitais. Nesse caso está sendo usado 2 resistores em paralelo para utilizar apenas um valor de resistor.
  - Vantagem:
    - É possível saber quando mais de um botão está pressionado e identifica-los.
  - Desvantagem:
    - Montagem um pouco maior.



<https://tinyurl.com/2kzb25xq>

# Ligando mais de um botão em uma entrada analógica

- Exercício 5: Faça o exemplo com 2 botões e imprima qual botão está sendo pressionado e quando ambos estão pressionados.

```
void loop() {  
  leitura = analogRead(entrada_analogica);  
  if (leitura>720 && leitura<800 && x==0){  
    Serial.println("Ambos botões pressionados");  
    x=1;  
  }  
  else if (leitura>580 && leitura<640 && x==0){  
    Serial.println("Botão 2 pressionado");  
    x=1;  
  }  
  else if (leitura>470 && leitura<510 && x==0){  
    Serial.println("Botão 1 pressionado");  
    x=1;  
  }  
  else if (leitura<50 && x==1){  
    x=0;  
  }  
  delay(100);  
}
```

Colocar delay senão ele sempre vai identifica um botão primeiro (nesse caso que imprime apenas uma vez)



# Ligando mais de um botão em uma entrada analógica

- Outra forma para reduzir o debounce e detectar o pressionamento de ambos os botões é fazendo a média de N leituras

```
void loop() {  
    long leitura = 0;  
    for(int i=0;i<100;i++)  
    {  
        leitura = leitura+analogRead(entrada_analogica);  
    }  
    leitura = leitura/100;  
  
    if (leitura>720 && leitura<800 && x==0){  
        Serial.println("Ambos botões pressionados");  
        x=1;  
    }  
    else if (leitura>580 && leitura<640 && x==0){  
        Serial.println("Botão 2 pressionado");  
        x=1;  
    }  
    else if (leitura>470 && leitura<510 && x==0){  
        Serial.println("Botão 1 pressionado");  
        x=1;  
    }  
    else if (leitura<50 && x==1){  
        x=0;  
    }  
}
```