

ARDUINO BÁSICO

Jimmy Soares - jimmytsoares@gmail.com

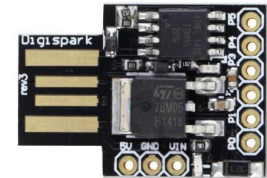
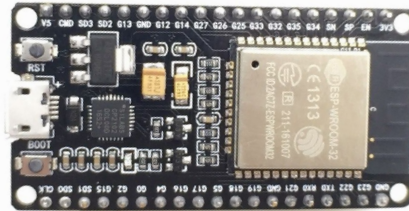
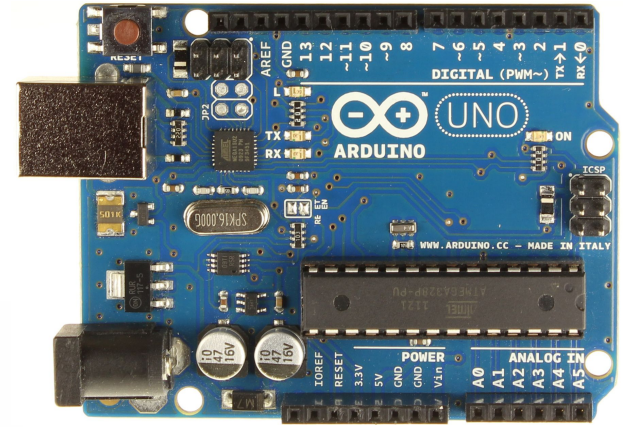
<https://github.com/jimmytsoares/oficina-arduino>

O QUE É ARDUINO?

- É um microcontrolador, como se fosse um mini-computador
- Fácil de programar e fazer seus próprios projetos
- Sua IDE simples de utilizar e o sucesso das placas fez com que fabricantes de outras placas desenvolvessem uma maneira de utilizar o Arduino IDE com seus respectivos microcontroladores

PLACAS

- Arduino: Uno, Micro, Nano, Mini, Mega
- DigiStump: DigiSpark, DigiSpark Pro
- ESP8266: NodeMCU, WemosD1
- ESP32



MADE
IN ITALY

AREF

GND

13

12

11

10

9

8

7

6

5

4

3

2

1

0

TX0 →

RX0 ←

TX3 14

RX3 15

TX2 16

RX2 17

TX1 18

RX1 19

SDA 20

SCL 21

22

24

26

28

30

32

34

36

38

40

42

44

46

48

50

52

53

GND

TX

RX

WWW.ARDUINO.CC

ATMEL
ATMEGA2560
16AU 1126

ICSP

RESET

ON



ARDUINO

MEGA

2560

IOREF

RESET

3.3V

5V

GND

GND

Vin

POWER

ANALOG IN

A0

A1

A2

A3

A4

A5

6

A7

A8

A9

A10

A11

A12

A13

A14

A15

DIGITAL

46

47

48

49

50

51

52

53

GND

PINOS

A0-A15: entrada analógica, o conversor do Arduino é de 10 bits então 0-5V -> 0-1023

22-53: entrada/saída digital, apenas nível alto ou baixo

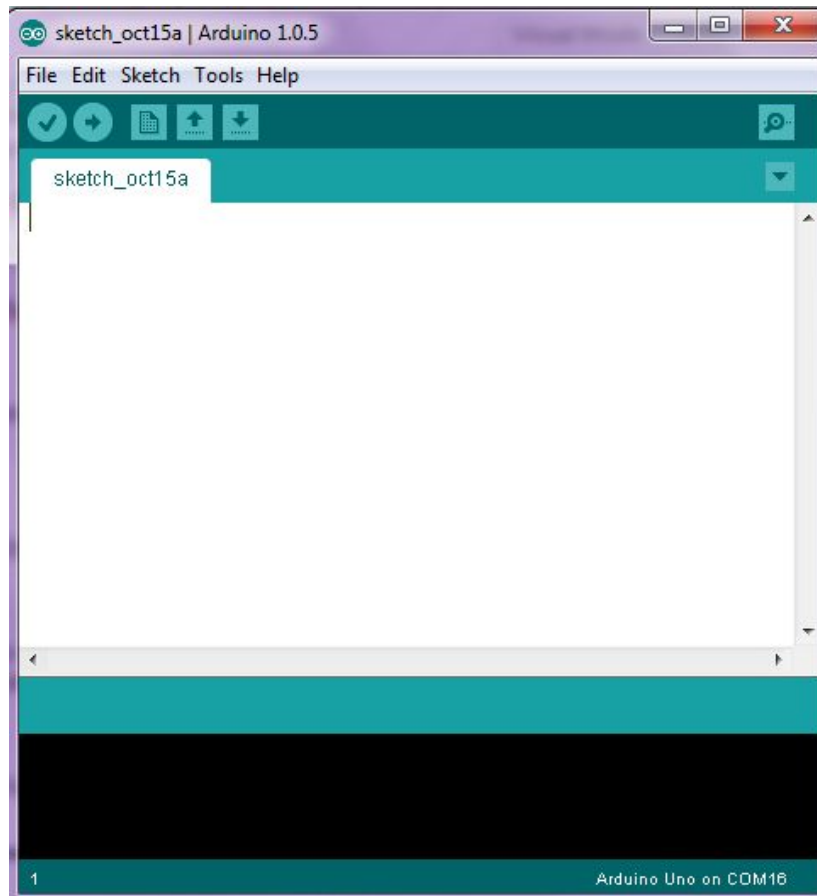
0,1,14-19: comunicação serial

20-21: I2C, comunicação com periféricos (sensores)

2-13: entrada/saída digital com PWM

COMO PROGRAMAR?

- Antes de começar
 - Conectar o Arduino no Computador
 - Ir em Tools -> Board -> Arduino Mega 2560
 - Tools -> Processor -> ATmega2560
 - Tools -> Port -> Port que o Arduino está conectado
- A IDE reconheceu sua placa!
:)



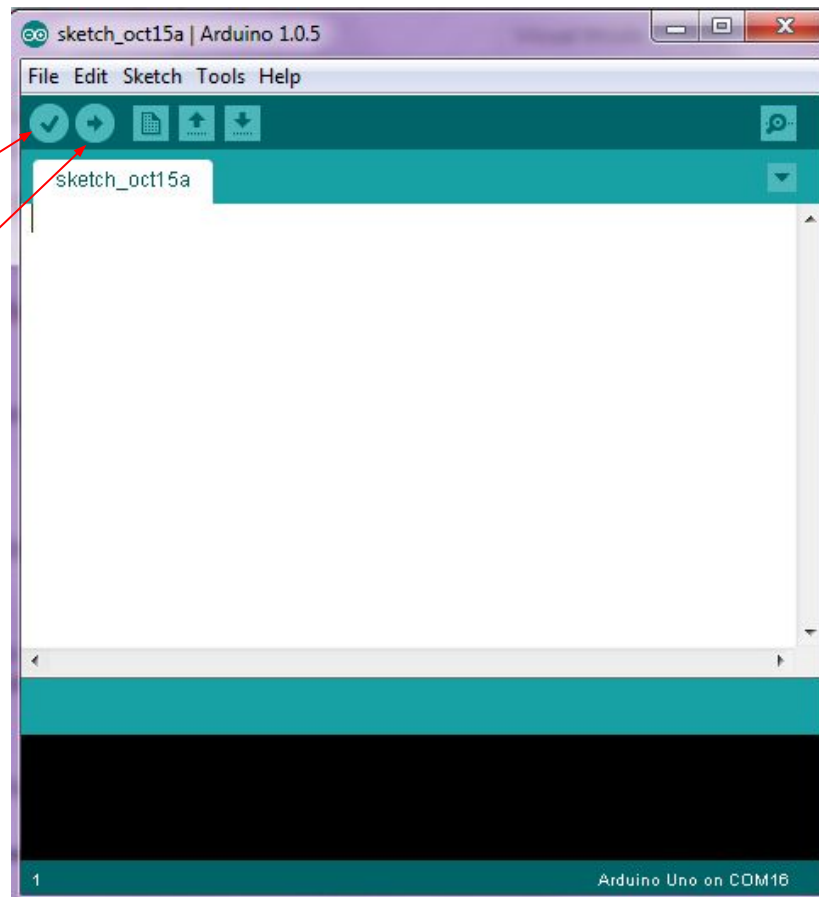
PISCAR O LED DA PLACA

```
void setup() {  
    //Executa uma única vez para arrumar a configuração do Arduino  
    //Define um pino como saída  
    //LED_BUILTIN corresponde ao pino 13, e está conectado a um led na placa  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    //É a 'main' do Arduino, só que ela executa várias vezes!  
    //Como escrever algo no pino  
    digitalWrite(LED_BUILTIN, HIGH); //Liga  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW); //Desliga  
    delay(1000);  
}
```

Não se esqueça que o Arduino Mega funciona a uma velocidade de 16MHz! Sem os delays, não dá para ver o LED piscar!

COMPILAÇÃO / UPLOAD

- Então como fazer pro Arduino funcionar de verdade?
 - Clique em Verify
 - Compila o Código e vê se não tem nada escrito errado
 - Clique em Upload



ADICIONANDO MONITOR SERIAL

```
void setup() {  
    //Executa uma única vez para arrumar a configuração do Arduino  
    //Define um pino como saída  
    //LED_BUILTIN corresponde ao pino 13, e está conectado a um led na placa  
    pinMode(LED_BUILTIN, OUTPUT);  
    Serial.begin(9600); //Inicia o monitor serial Ctrl+Shift+M para abrir  
}
```

```
void loop() {  
    //É a 'main' do Arduino, só que ela executa várias vezes!  
    //Como escrever algo no pino  
    digitalWrite(LED_BUILTIN, HIGH); //Liga  
    Serial.println("LED Ligado!");  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW); //Desliga  
    Serial.println("LED Desligado!");  
    delay(1000);  
}
```

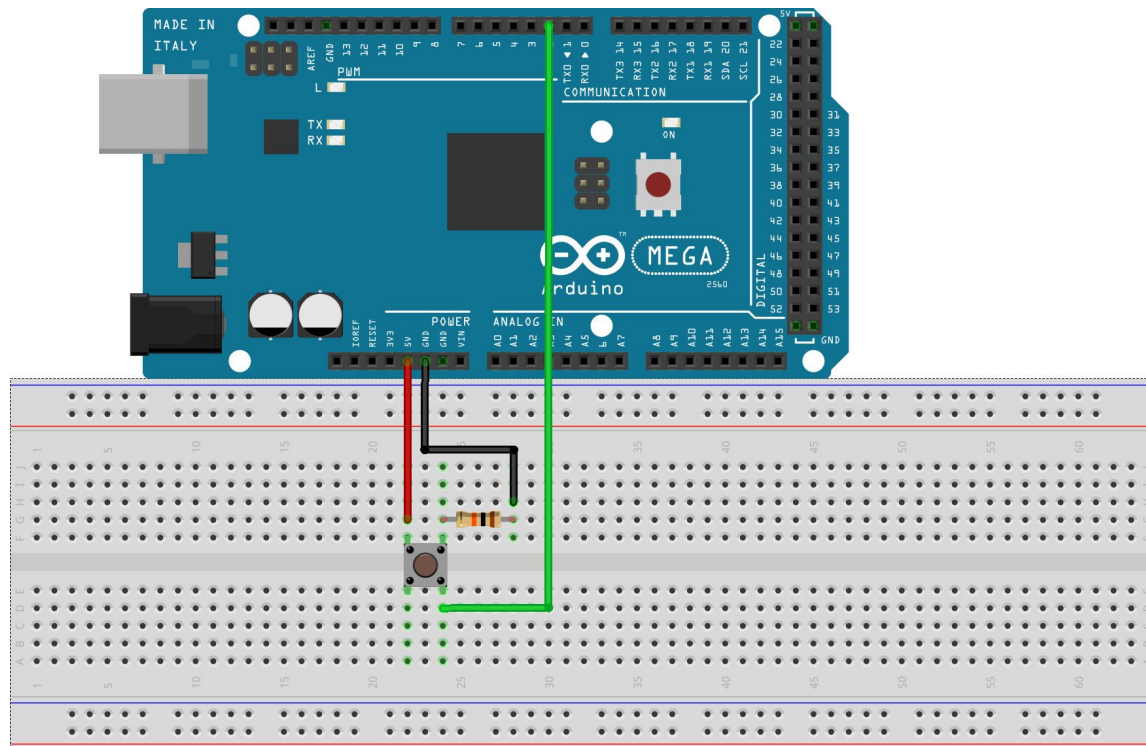
ADICIONANDO BOTÃO

```
int botao = 0;
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(2, INPUT);
    Serial.begin(9600); //Inicia o monitor serial Ctrl+Shift+M para abrir
}

void loop() {
    botao = digitalRead(2);
    if(botao == HIGH) { //é possível usar apenas if(botao)
        digitalWrite(LED_BUILTIN, HIGH); //Liga
        Serial.println("LED Ligado!");
        delay(1000);
    } else {
        digitalWrite(LED_BUILTIN, LOW); //Desliga
        Serial.println("LED Desligado!");
        delay(1000);
    }
}
```

CONECTANDO UM BOTÃO

- O resistor pode ser de 10K



CONECTANDO UM POTENCIÔMETRO

$$R = (5V - V_{led}) / 20mA,$$

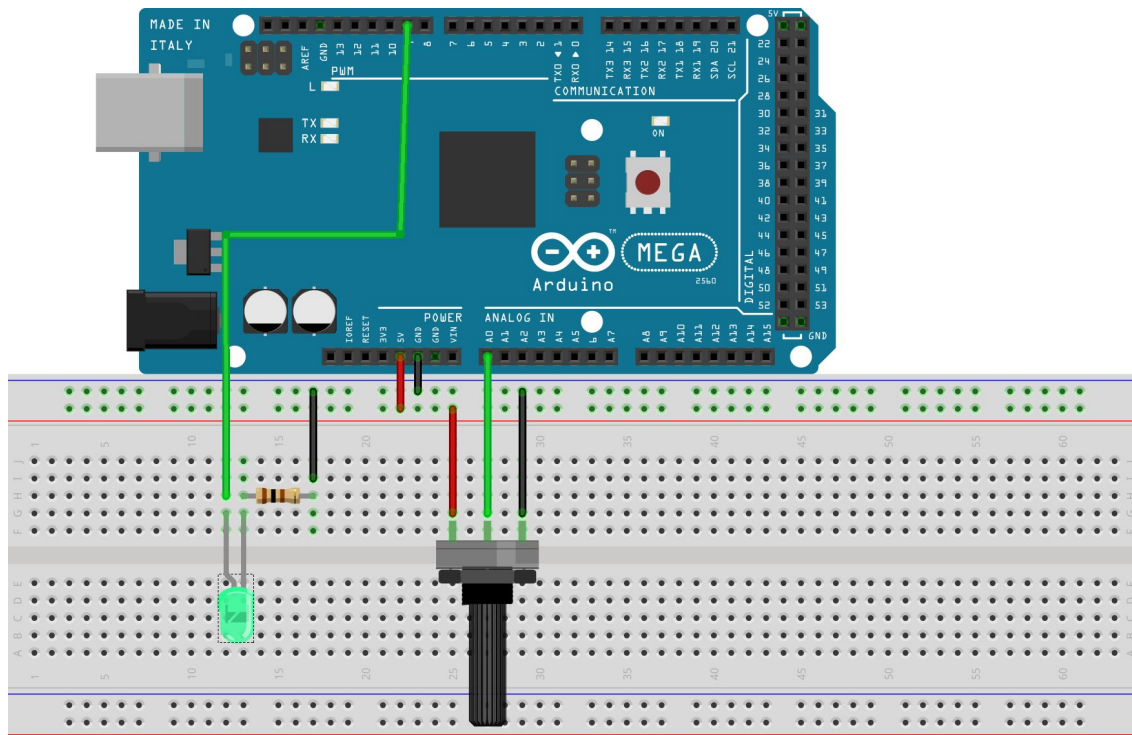
$V_{led} = 2V$ para **vermelho**,
amarelo e **laranja**

$= 2,5V$ para **verde**

$= 3V$ para **azul** e branco

Abrir o código

`analogReadWriteSerial`

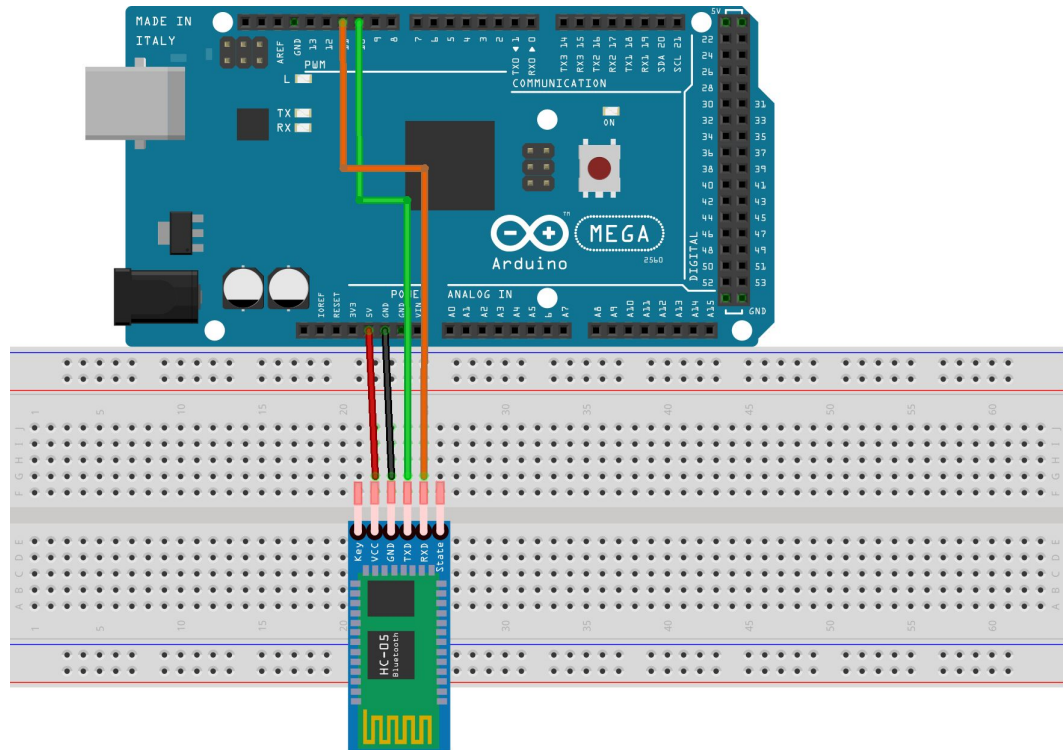


CONECTANDO O MÓDULO HC-05

Abrir o código
btSoftwareSerial

!!!ATENÇÃO!!!

Deixar o Vcc do BT
desligado ao fazer
upload do programa na
placa!



DESAFIO

- Modificar o código `btSoftwareSerial` com as seguintes funções
- Conectar o LED RGB no Arduino e alterar a cor dele conforme mensagens enviadas por Bluetooth
- Caso mensagem seja
 - `'red'` -> `vermelho`
 - `'green'` -> `verde`
 - `'blue'` -> `azul`
- Uma imagem de como conectar um anodo comum está no github
- LED RGB anodo comum é ativado em baixa
- Adicionar o monitor serial e imprimir qual cor está ligada no momento

FIM!

- Obrigado por participarem!