

Aula 01 - SEEL 2019

Minicurso de Arduino

Introdução, Portas Digitais e Emulação no Proteus

Adriano Rodrigues

23 de outubro de 2019

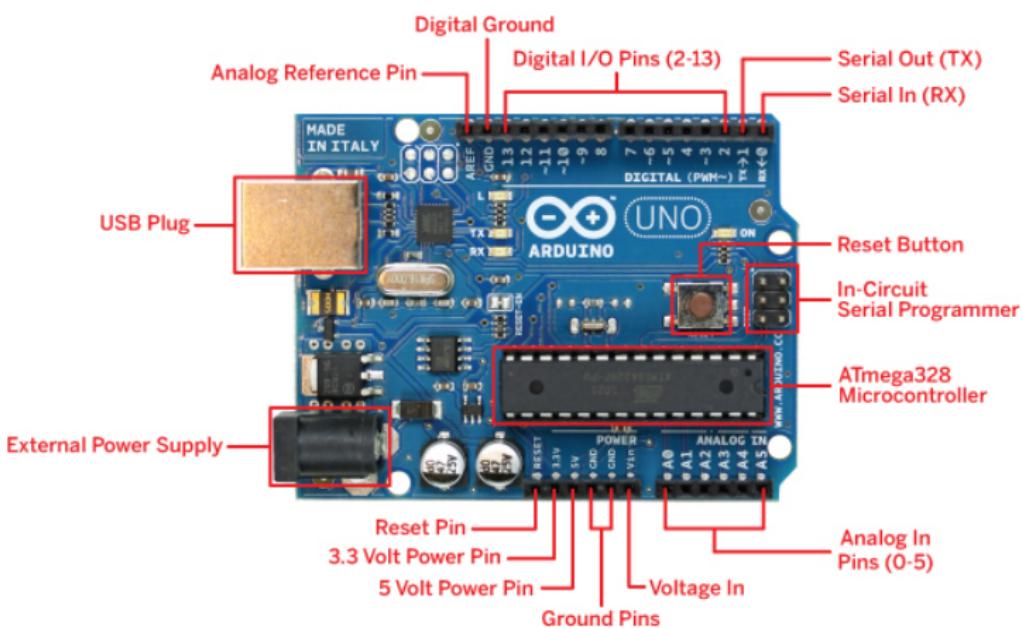
- Introdução;
- Apresentação da Placa;
- Aplicações;
- Introdução à Interface;
- Introdução à Programação;
- Portas Digitais:
 - Saídas Digitais;
 - Entradas Digitais.
- Emulação do Arduino no Proteus.

- 1 Conteúdo
- 2 Índice
- 3 O Arduino
- 4 O Arduino UNO R3
- 5 Aplicações
- 6 Introdução à Interface
- 7 Introdução à Programação
- 8 Funções Digitais
- 9 Exercício 1
- 10 Exercício 2
- 11 Introdução ao Proteus
- 12 Exercício 2 - Proteus
- 13 Tarefa 1
- 14 Exercício 3
- 15 Tarefa 2
- 16 Sugestão 1
- 17 Fim!

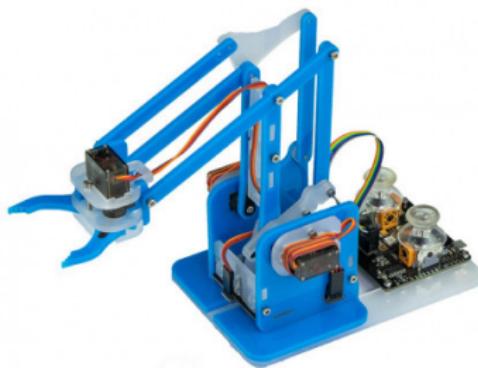
- Site oficial: arduino.cc;
- Plataforma eletrônica de código aberto baseada em hardware e software livres de fácil utilização. Foi pensado para que qualquer pessoa possa fazer seus próprios projetos interativos;
- Modelos: Uno, Mega (2560, ADK), Due, Leonardo etc...

- 14 portas digitais (6 podem ser configuradas PWM);
- 6 entradas analógicas (10 bits);
- Microcontrolador ATmega328P (16 MHz de clock);
- Opera em 5 V;
- Alimentado entre 7-12 V.

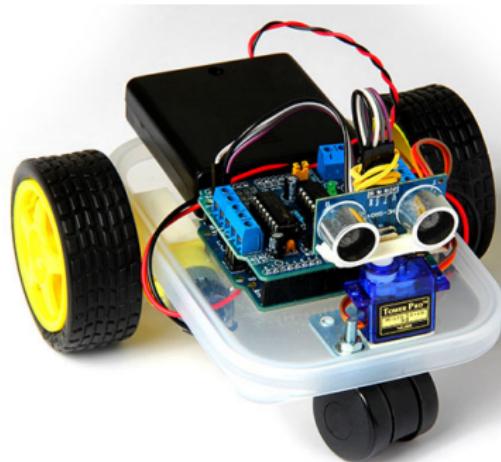
O Arduino UNO R3



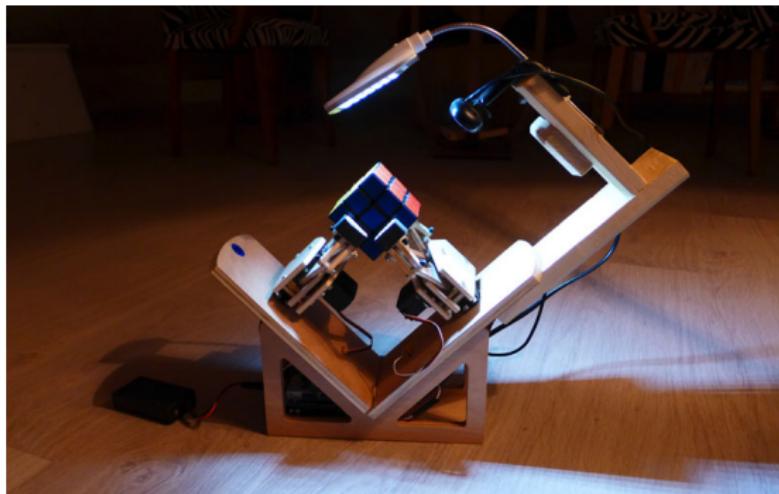
- Robótica:



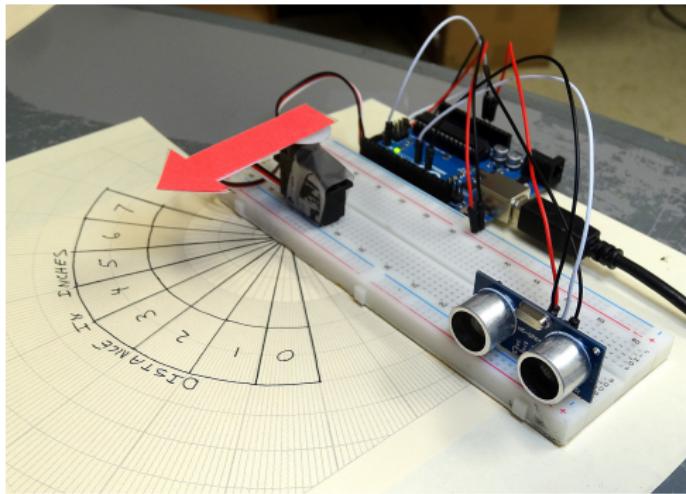
- Robótica:



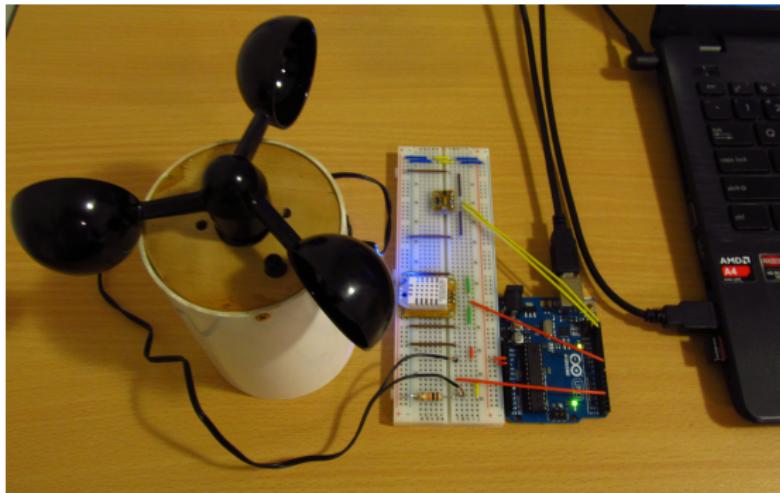
- Robótica:



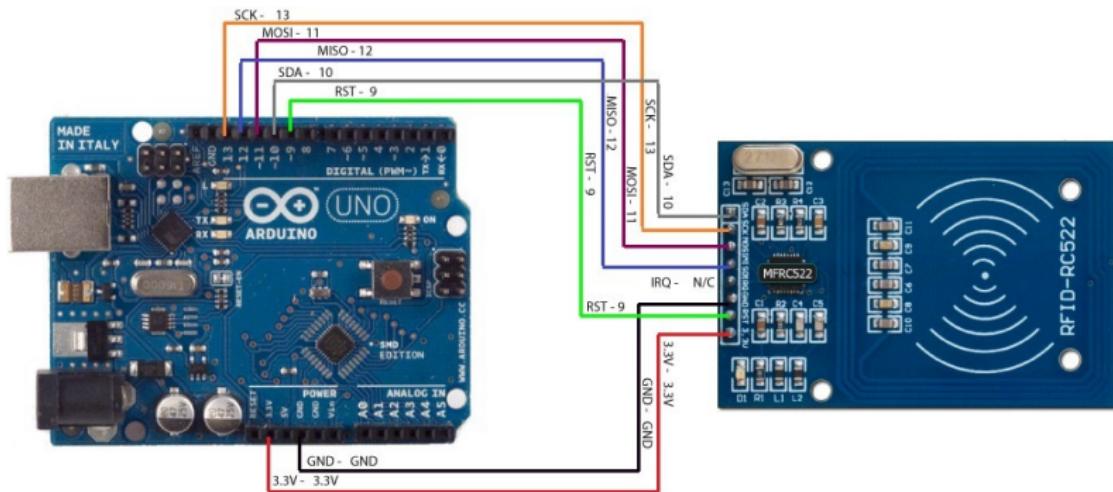
- Sensoriamento:



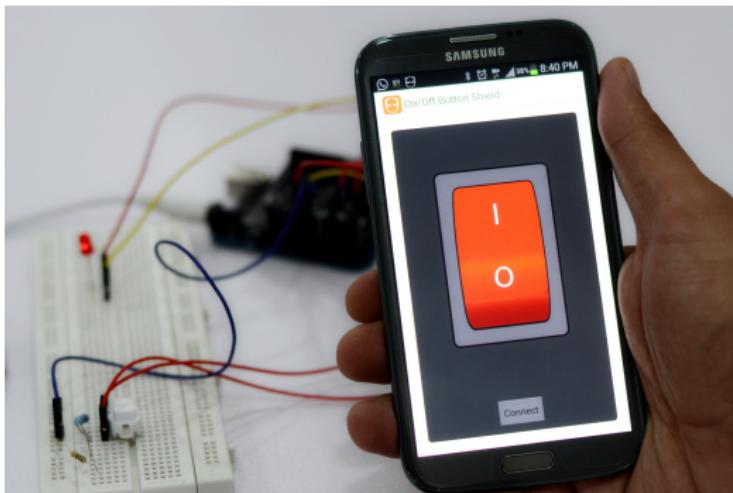
- Sensoriamento:



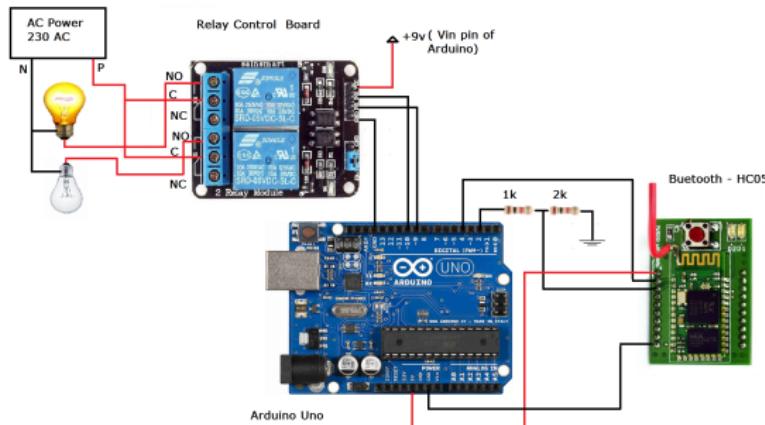
- Sensoriamento:



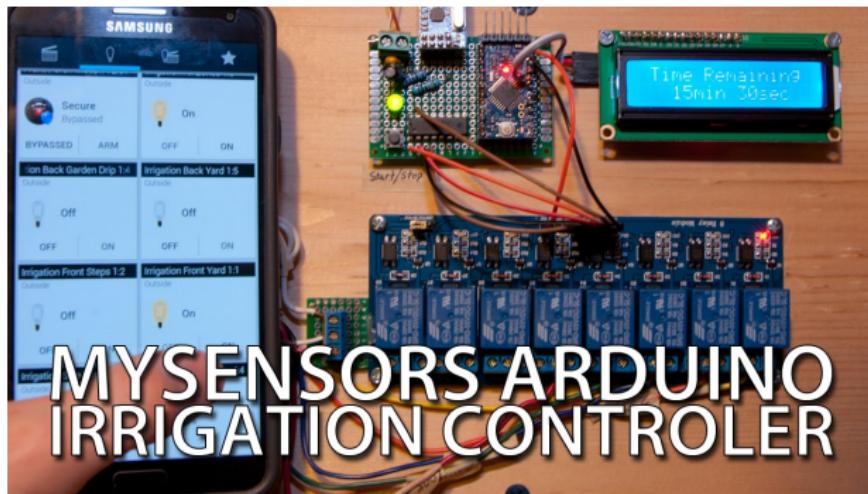
- Automação:



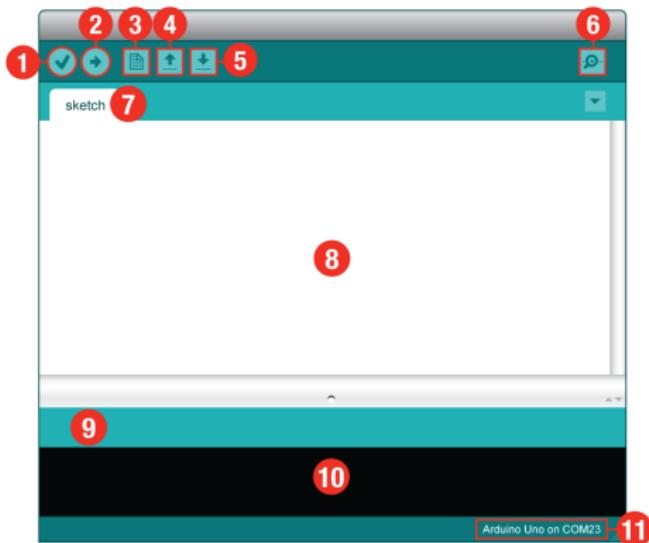
- Automação:



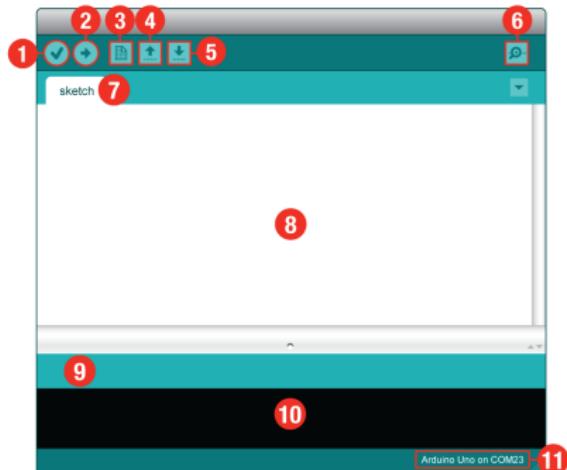
- Automação:



Introdução à Interface



- 1 Verificar e compilar o Código;
- 2 Fazer Upload do Código no Arduino;
- 3 Criar Novo Sketch;
- 4 Abrir outro Sketch;
- 5 Salvar este Sketch;



- 6 Monitor Serial - Tela de monitoramento da serial;
- 7 Título do Sketch;
- 8 Campo do Código;
- 9 Status da Interface;
- 10 Status e Avisos Relacionados ao Código;
- 11 Informação de Conexão com o Arduino.

Tipos de Variáveis

- int - inteiros de -32768 até 32767 ;
 - `int x = 5;`
 - `int x = 5, y = 4, z = 10;`
- long - inteiros de $-2.147.483.648$ até $2.147.483.647$;
- float - números reais de $-3,4028235 \cdot 10^{38}$ até $3,4028235 \cdot 10^{38}$;
- char - caracteres (exemplo: `char 'A'`);
- byte - binários de $d0$ até $d255$ ($b10010 = d18$);
- boolean - `false` ou `true`;
- array:
 - `int myInt[6];` (5 elementos + fim de curso)
 - `int myPins[] = {2,4,8,3,5};`
 - `int mySensors[6] = {1,4,-8,3,2};`
 - `char message[6] = "Hello";`
 - Primeiro elemento é referido como “`myPins[0]`”, por exemplo;

- O Básico

```
1 void setup()
2 {
3     // Esse código roda somente uma vez.
4 }
5 void loop()
6 {
7     // Código principal que roda recursivamente.
8 }
```

- if / else if / else:

```
1  if (variavel < condicao1)
2  {
3      // Fazer A.
4  }
5  else if (variavel >= condicao2)
6  {
7      // Fazer B.
8  }
9  else
10 {
11     // Fazer C.
12 }
```

- for:

```
1 for (inicializacao; condicao; incremento)
2 {
3     // Instrucoes.
4 }
```

Exemplo:

```
1 for (int i = 0; i <= 255 ; i++)
2 {
3     a[i] = 2*i;
4 }
```

- while:

```
1 while(condicao)
2 {
3     // Comando .
4 }
```

Exemplo:

```
1 int var = 0;
2 while(var < 200)
3 {
4     var++;
5 }
```

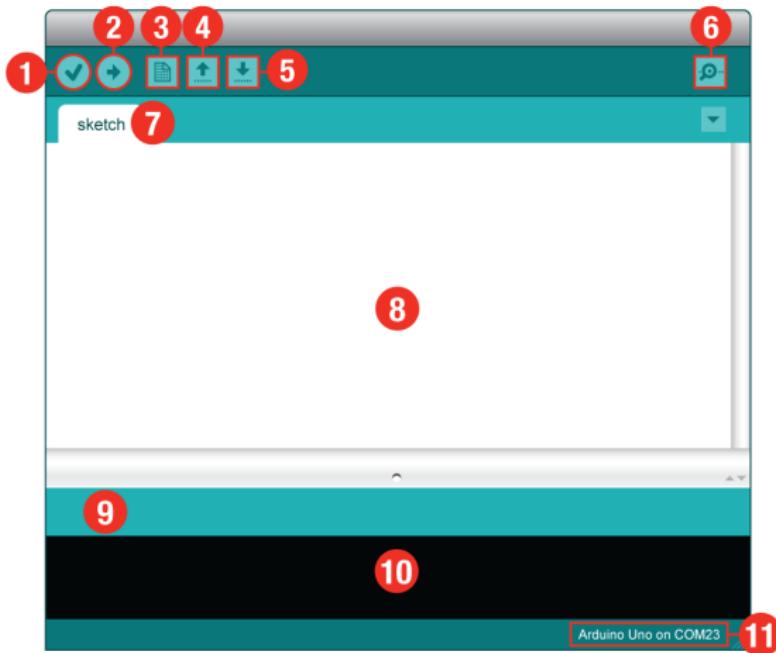
No **setup()**. Definição das portas:

- `pinMode(pino,direção);`
pino = int de 0 a 13
direção = INPUT ou OUTPUT.

No **loop()**:

- `digitalRead(pino);`
pino = int de 0 a 13
Função de **leitura!** Retorna booleano.
- `digitalWrite(pino, estado);`
pino = int de 0 a 13
estado = HIGH ou LOW.

Exercício 1 - Introdução à IDE - Piscar LED

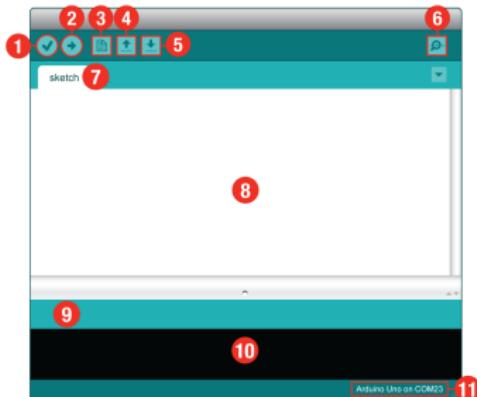


8 Escrever o código aqui.

Exercício 1 - Introdução à IDE - Piscar LED

```
1 // LED piscando - Adriano Rodrigues.  
2  
3 // Porta do LED.  
4 int led = 13;  
5  
6 void setup()  
7 {  
8     // Configuração da porta digital do LED.  
9     pinMode(led, OUTPUT);  
10 }  
11  
12 void loop()  
13 {  
14     digitalWrite(led, HIGH); // Escreve 5V na porta LED.  
15     delay(2000) ; // Espera por 2000 milissegundos.  
16     digitalWrite(led, LOW) ; // Escreve 0V na porta LED.  
17     delay(2000) ; // Espera por 2000 milissegundos.  
18 }
```

Exercício 1 - Introdução à IDE - Piscar LED



1 Verificar Coerência; + Salvar o código:

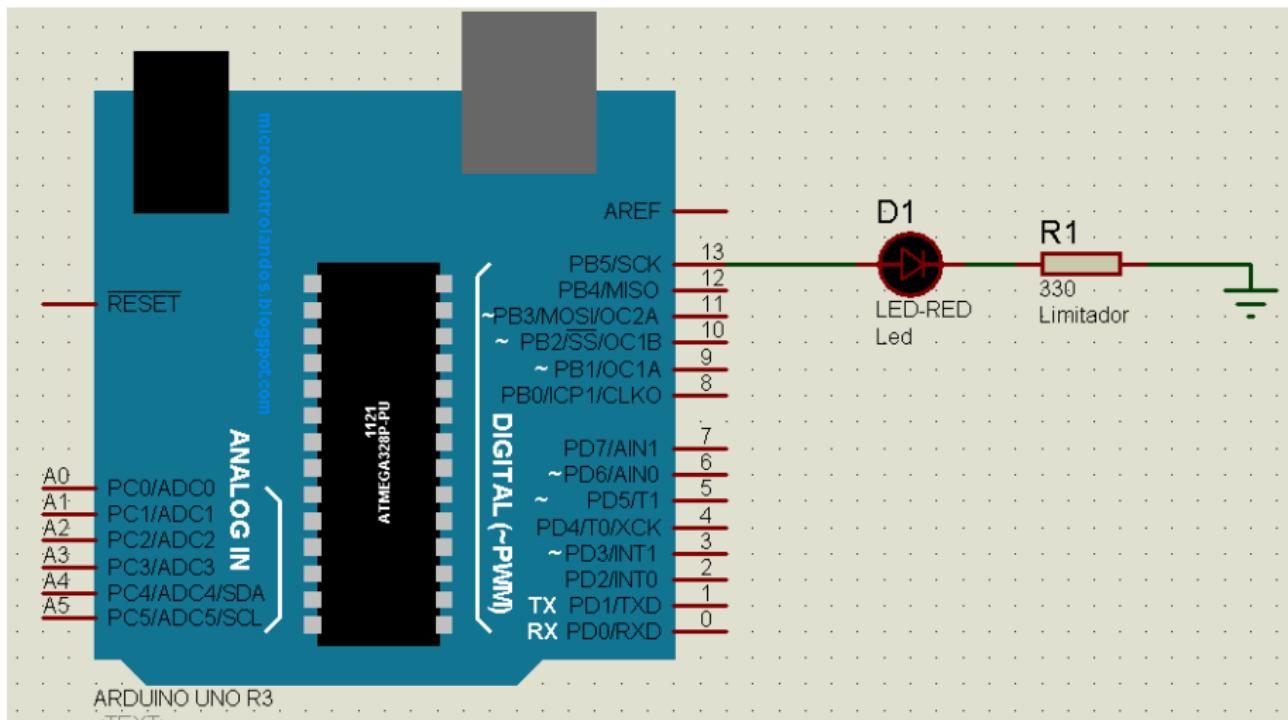
- Abre o browser padrão para novos códigos;
- Não separar nomes, ou iniciar por números, ou por caracteres especiais;
- Esse procedimento cria uma pasta com um arquivo .ino com o texto do sketch.

2 Fazer Upload;

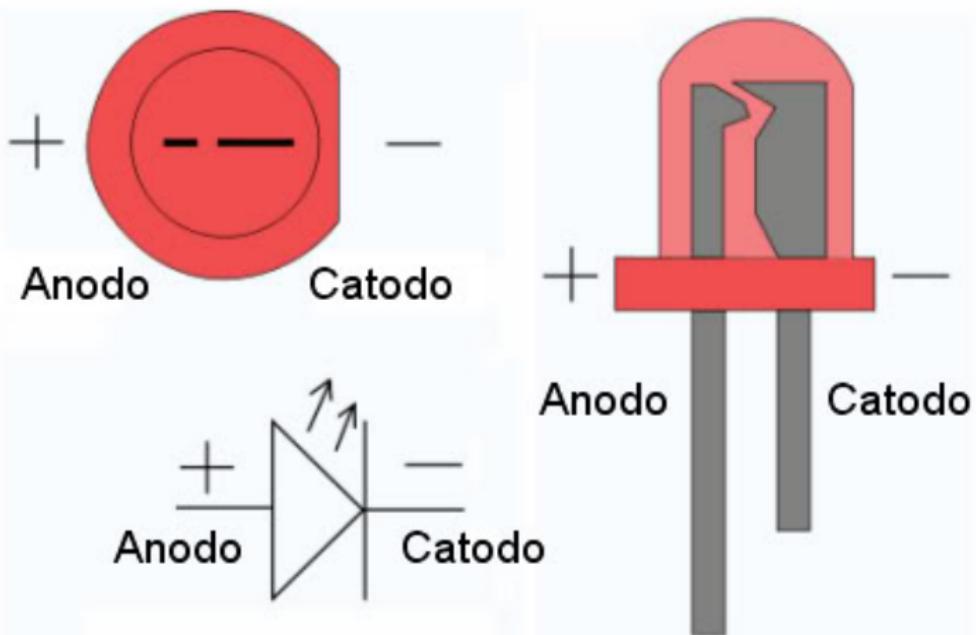
9 Barra de progresso do Upload;

10 Verificar Status do Upload;

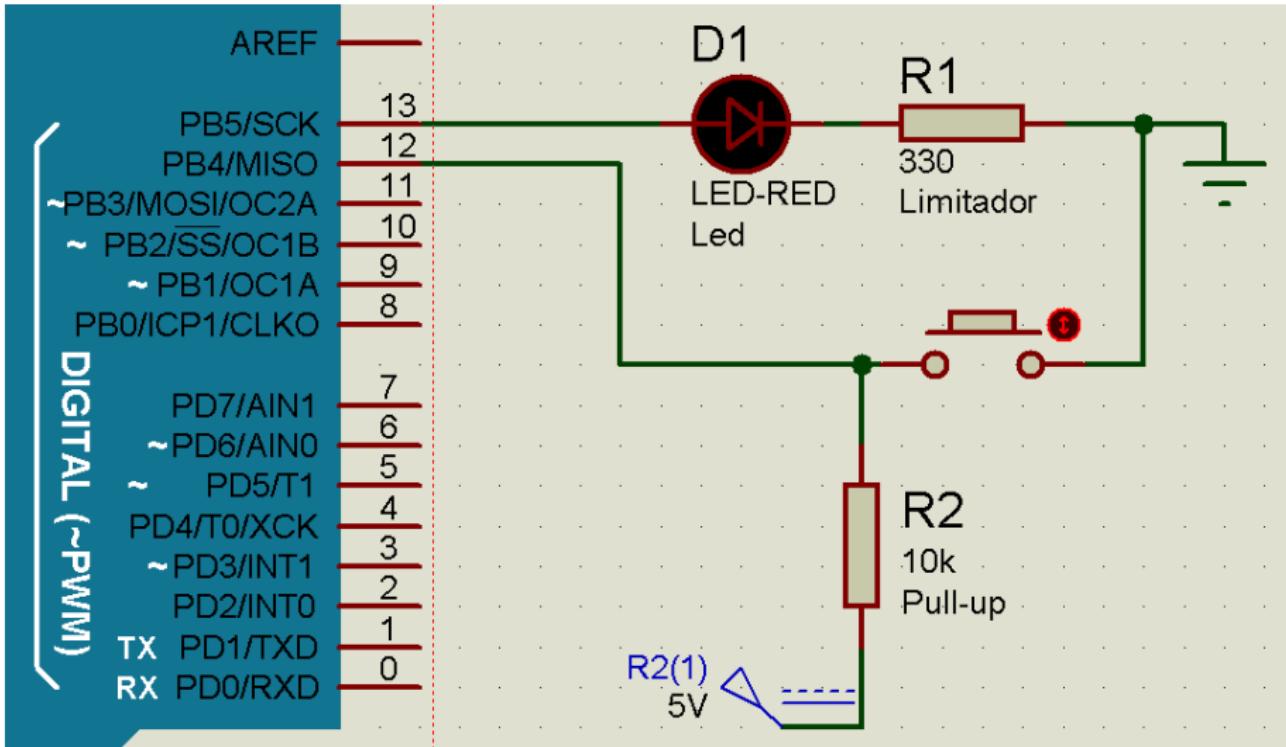
Exercício 1 - Introdução à IDE - Pisca LED



Exercício 1 - Introdução à IDE - Pisca LED



Exercício 2 - Esquemático



Exercício 2.0 - Botão de Comando

```
1 // LED com if (botao) - Adriano Rodrigues.  
2 int led    = 13; // Porta do LED.  
3 int botao = 12; // Porta do Botao.  
4 void setup()  
5 {  
6     pinMode(led, OUTPUT) ; // Configuracao da porta digital do LED.  
7     pinMode(botao, INPUT); // Configuracao da porta digital do BOTAO.  
8 }  
9 void loop()  
{  
11     if (digitalRead(botao) == HIGH)  
12     {  
13         digitalWrite(led, HIGH); // Escreve 5V na porta LED.  
14     }  
15     else // ou 'else if (digitalRead(botao) == LOW)' .  
16     {  
17         digitalWrite(led, LOW); // Escreve 0V na porta LED.  
18     }  
19 }
```

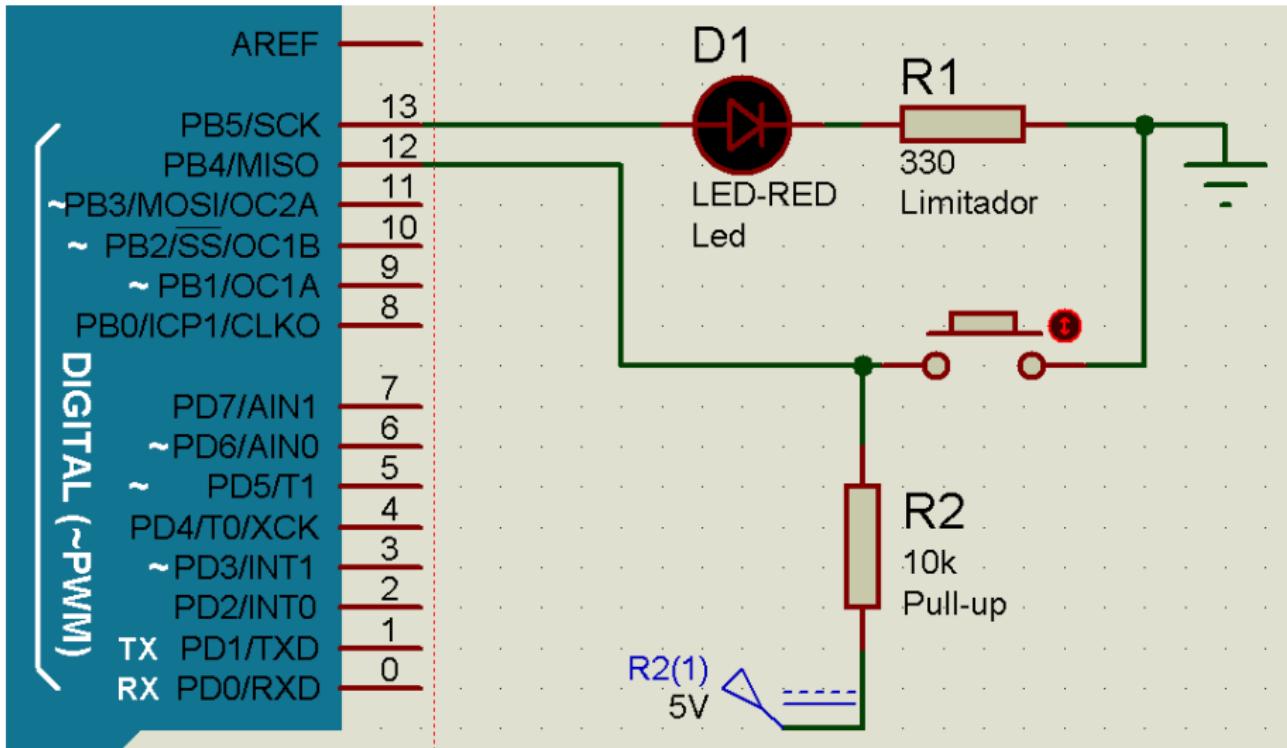
Exercício 2.1 - Botão de Comando - Simplificado

```
1 // LED com if (botao) - Adriano Rodrigues.  
2 int led = 13, botao = 12; // Porta do LED - Porta do botao.  
3 void setup()  
4 {  
5     pinMode(led, OUTPUT); pinMode(botao, INPUT); // Configurar PDs.  
6 }  
7 void loop()  
8 {  
9     if (digitalRead(botao) == HIGH)  
10         digitalWrite(led, HIGH); // Escreve 5V na porta LED.  
11     else  
12         digitalWrite(led, LOW); // Escreve 0V na porta LED.  
13 }
```

Exercício 2.2 - Botão de Comando - Simplificado Menor

```
1 // LED com if (botao) - Adriano Rodrigues.  
2 int led = 13, botao = 12; // Porta do LED - Porta do botao.  
3 void setup()  
4 {  
    pinMode(led, OUTPUT); pinMode(botao, INPUT); // Configurar PDs.  
6 }  
7 void loop()  
8 {  
    if (digitalRead(botao) == HIGH) digitalWrite(led, HIGH);  
    else digitalWrite(led, LOW);  
11 }
```

Exercício 2 - Botão de Comando





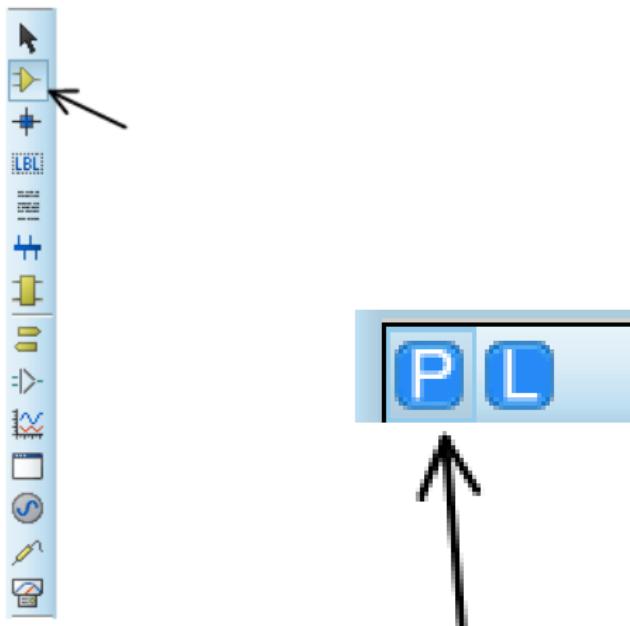
Programa que auxilia na simulação de circuitos e componentes elétricos integrados.

- Simular o Arduino no esquemático do Proteus Isis.



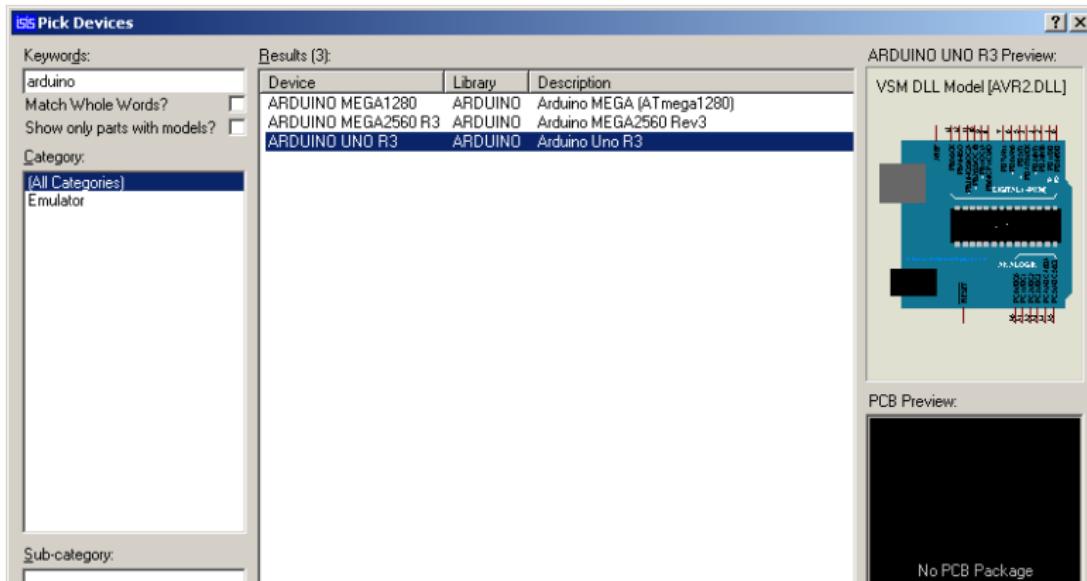
- Inserir Componentes:

- (Barra de Tarefas) Library > Pick parts from libraries.
- (Ícones) Amplificador Operacional Dourado > P.



- (Atalho de teclado) P.

- Inserir Componentes:
 - Arduino (ARDUINO UNO R3);

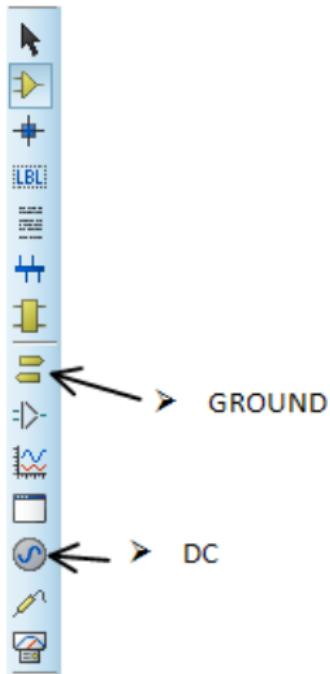


- Inserir Componentes:

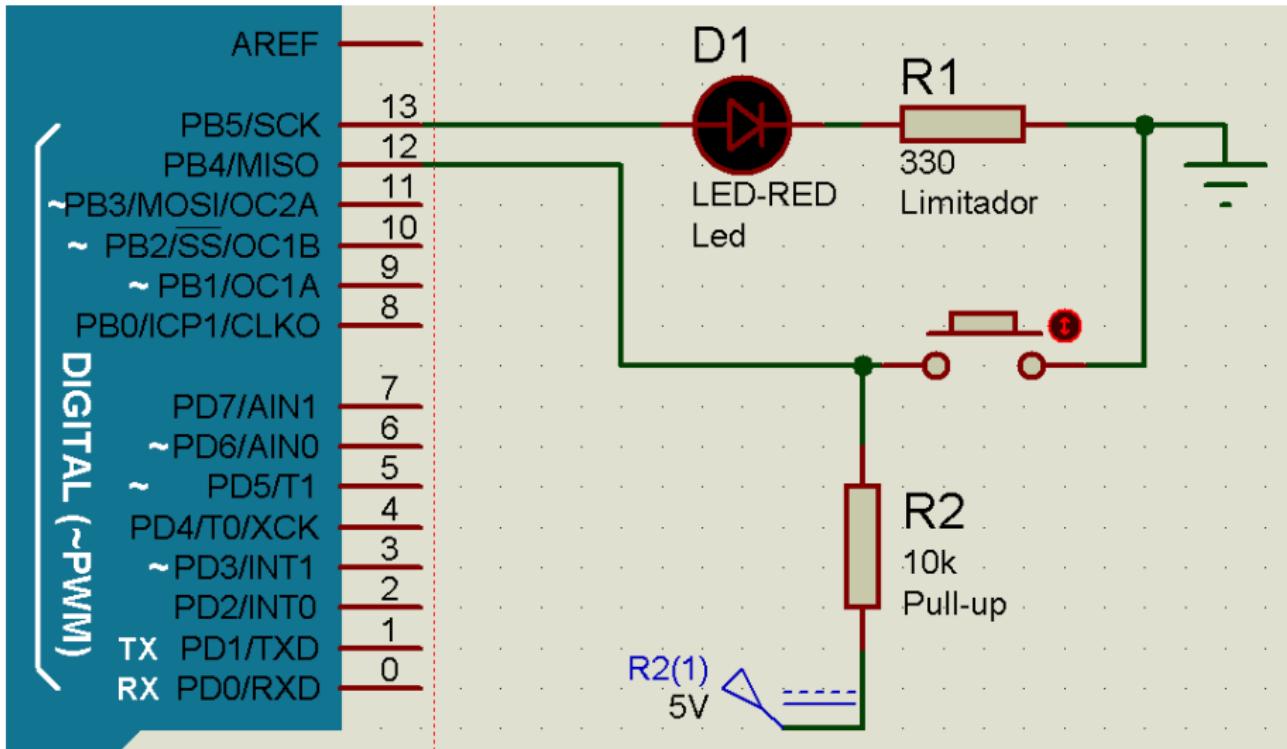
- Arduino (ARDUINO UNO R3);
- Resistor (RES);
- Chave (SWITCH);
- Led (LED-GREEN, LED-RED, LED-BLUE, LED-YELLOW);
- Capacitor (CAP);
- Motor DC (MOTOR);
- etc...

- Inserir Alimentação:

- 5V (VCC)
- 0V (GND)



Exercício 2 - Botão de Comando

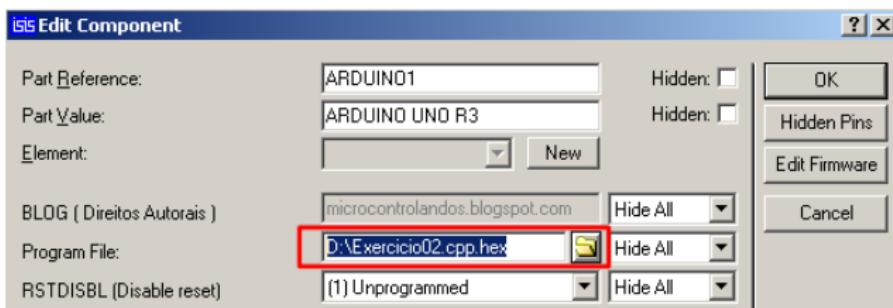


Exercício 2 - Programar Arduino do Proteus

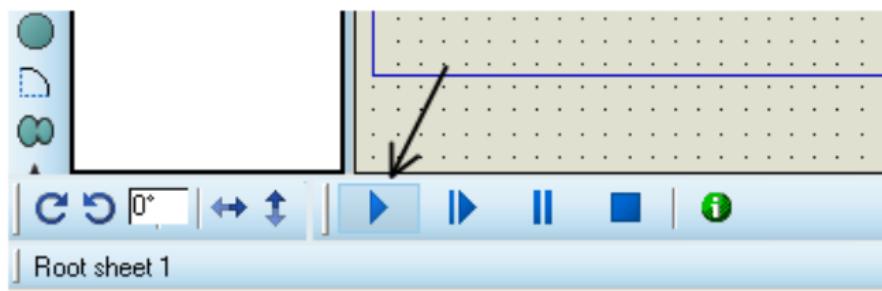
Copiar:

```
Done compiling.  
/tmp/built05681152206486607265.tmp/Exercicio02.cpp.elf /tmp/built05681152206486607265.tmp/Exercicio02.cpp.o  
/tmp/build5681152206486607265.tmp/core.a -L/tmp/build5681152206486607265.tmp -lm  
/home/adriano/arduino-1.6.5/hardware/tools/avr/bin/avr-objcopy -O ihex -j .eeprom --set-section=PROGMEM  
--change-section-lma .eeprom=0 /tmp/build5681152206486607265.tmp/Exercicio02.cpp.elf /tmp/build5681152206486607265.tmp/Exercicio02.cpp.hex  
Sketch uses 994 bytes (3%) of program storage space. Maximum is 32,256 bytes.  
Global variables use 13 bytes (0%) of dynamic memory, leaving 2,035 bytes for local variables.
```

Clicar duas vezes no Arduino (Proteus), Colar:



Play:



- Criar um código que acenda três LEDs que simulem o funcionamento de um semáforo;
- Verificar o funcionamento do código no Proteus.

Tarefa 1 - Semáforo no Proteus

```
1 // Semaforo - Proteus 8 - Adriano Rodrigues.
2 int LEDg = 11, LEDy = 12, LEDr = 13;
3 void setup()
4 {
5     pinMode(LEDg,OUTPUT); pinMode(LEDy,OUTPUT); pinMode(LEDr,OUTPUT);
6     // Inicia os tres LEDs apagados.
7     digitalWrite(LEDg,LOW); digitalWrite(LEDy,LOW); digitalWrite(LEDr,LOW);
8 }
9 void loop()
10 {
11     // Acende o verde por 5s, depois apaga.
12     digitalWrite(LEDg,HIGH); delay(5000); digitalWrite(LEDg,LOW);
13     // Acende o amarelo por 1s, depois apaga.
14     digitalWrite(LEDy,HIGH); delay(1000); digitalWrite(LEDy,LOW);
15     // Acende o vermelho por 5s, depois apaga.
16     digitalWrite(LEDr,HIGH); delay(5000); digitalWrite(LEDr,LOW);
17 }
```

Exercício 3 - Frequência Modificada

```
1 // LED e laco for - Adriano Rodrigues.  
2  
3 // Porta do LED.  
4 int led = 13;  
5  
6 void setup()  
7 {  
8     // Configuracao da porta digital do LED.  
9     pinMode(led, OUTPUT);  
10 }  
11  
12 void loop()  
13 {  
14     for (int i = 0; i < 5; i++)  
15     {  
16         digitalWrite(led, HIGH); // Escreve 5V na porta LED.  
17         delay(100*i) ; // Espera por 100*i milissegundos.  
18         digitalWrite(led, LOW) ; // Escreve 0V na porta LED.  
19         delay(100*i) ; // Espera por 100*i milissegundos.  
20     }  
21 }
```

Aumentar o tempo em que cada ciclo é apresentado.

Cada ciclo estava sendo exibido apenas uma vez por loop. Propõe-se que cada ciclo seja mostrado por uma quantidade maior de períodos antes de ser alterado.

Faça o ajuste para que cada ciclo apareça por 5 períodos antes de ser atualizado.

Tarefa 2 - Frequência Modificada

```
1 // LED e laco for - Adriano Rodrigues.  
2  
3 void loop()  
4 {  
    for (int i = 0; i < 5; i++)  
    {  
        for (int j = 0; j < 5; j++)  
        {  
            digitalWrite(led, HIGH); // Escreve 5V na porta LED.  
            delay(100*i) ; // Espera por 100*i milissegundos.  
            digitalWrite(led, LOW) ; // Escreve 0V na porta LED.  
            delay(100*i) ; // Espera por 100*i milissegundos.  
        }  
    }  
}
```

- Criar um código simule a operação de dois semáforos perpendiculares de um cruzamento;
- Verificar o funcionamento no Proteus.

Até a próxima aula!

Aula 01 - SEEL 2019 Minicurso de Arduino

Introdução, Portas Digitais e Emulação no Proteus

Adriano Rodrigues

23 de outubro de 2019