

Automação

Automação com Arduino

Setembro 2017

Placas de Arduino



Arduino Mega

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Detalhes

- Entrada e Saída:

Cada um dos 54 pinos digitais do Mega2560 pode ser usado como entrada ou saída, usando as funções de `pinMode()`, `digitalWrite()`, e `digitalRead()`. Eles operam a 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e possui um resistor interno (desconectado por default) de 20-50K?.

- Comunicação:

Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL.

- Interruptores:

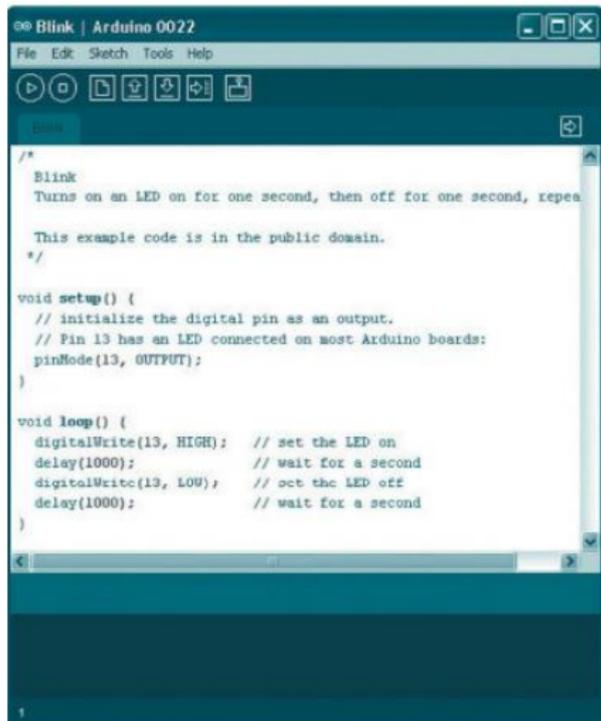
Interruptores externos: 2 (interruptor 0), 3 (interruptor 1), 18 (interruptor 5), 19 (interruptor 4), 20 (interruptor 3), and 21 (interruptor 2).

Detalhes

- PWM:
0 a 13. Fornecem saída analógica PWM de 8-bits com a função `analogWrite()`.
- Porta Analógica de Entrada
16 entradas analógicas, cada uma das quais com 10 bits de resolução (i.e. 1024 valores diferentes)
- LED: 13:
Há um LED conectado ao pino digital 13. Quando o pino está em HIGH o led se acende.

Instalando o Software

- Instalar o software do Arduino

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main text area contains the following code:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repea  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);   // set the LED on  
  delay(1000);              // wait for a second  
  digitalWrite(13, LOW);   // set the LED off  
  delay(1000);              // wait for a second  
}
```

www.arduino.cc

Ligando o Arduino

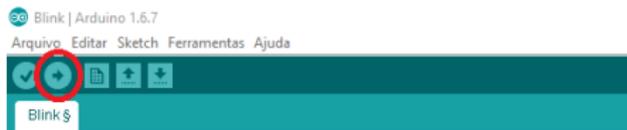
- Conecte o Cabo USB e verifica se o computador o reconhece
- Abra o programa do Arduino como administrador



- Vai em Ferramentas e depois em:
 1. Placa: Selecione Arduino Mega (se for esta sua placa)
 2. Porta: Selecione a porta que o computador reconheceu.

Hello Word

- Vamos executar o primeiro programa. Para isto:
 1. Arquivo > Exemplos > 01.Basics > Blink
 2. Clique em carregar.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

Resultado:

Led do pino digital 13 irá acender e apagar

Funcionamento do Código no Arduino

Setup e Loop

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
}
```

Exercício 1

1. Execute **setup** e faça o pino 13 ser uma saída.
2. Execute **loop** e faça o pino 13 passar para nível HIGH, ou alto (LED acende).
3. Execute um retardo (delay) de meio segundo.
4. Faça o pino 13 passar para nível LOW, ou baixo (LED apaga).
5. Execute novamente **loop**, voltando para o passo 2 e fazendo o nível do pino 13 ser alto, ou HIGH (LED acende).

Exercício 1

Resultado:

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
}
```

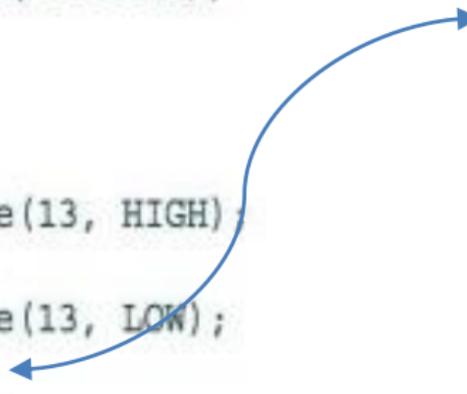
Neste programa o Led vai aparentar estar sempre aceso, pois quando ele apaga a primeira coisa que acontece logo em seguida é que ele é novamente aceso. Isso ocorre muito rapidamente.

Exercício 1 - Concertado

```
// sketch 03-01
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

Solução



Utilizando Variáveis

```
// sketch 03-02
int ledPin = 13;
int delayPeriod = 500;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
}
```

- Para piscar mais rápido:
Simplesmente mude delayPeriod = 100;

Exercícios 2

Faça um Pisca-Pisca utilizando a porta 13 do Arduino que começa a piscar muito rapidamente e aos poucos vai se tornando mais e mais lento

```
// sketch 03-03
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
  delayPeriod = delayPeriod + 100;
}
```

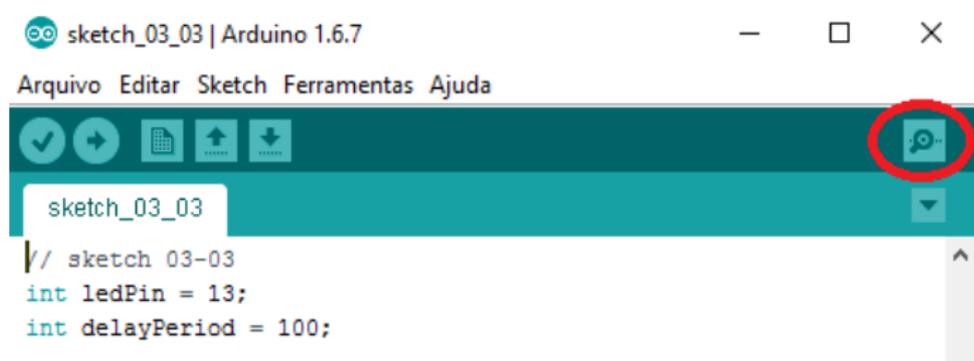


Referência

Monk, Simon; Programação com Arduino, começando com sketches.
Porto Alegre: Bookman, 2013.

Serial Monitor

Sua finalidade é atuar como um canal de comunicação entre o computador e o Arduino



- Função que será usada para isto:

`Serial.println`

Exemplo 3

Faça o Arduino mostrar na tela do computador o resultado de uma soma de 2 variáveis

```
// sketch 03-04
void setup()
{
  Serial.begin(9600);
  int a = 2;
  int b = 2;
  int c = a + b;
  Serial.println(c);
}

void loop()
{}
```



Obs.: Lembre-se de colocar no monitor serial a mesma porta que foi colocado no código.

Operadores

Operador	Significado	Exemplos	Resultado
<	Menor do que	9 < 10 10 < 10	verdadeiro falso
>	Maior do que	10 > 10 10 > 9	falso verdadeiro
<=	Menor do que ou igual a	9 <= 10 10 <= 10	verdadeiro verdadeiro
>=	Maior do que ou igual a	10 >= 10 10 >= 9	verdadeiro verdadeiro
==	Igual a	9 == 9	verdadeiro
!=	Não igual a, diferente de	9 != 9	falso

Comando IF

Se houver uma condição com a necessidade de verificação se uma condição é verdadeira ou falsa, o comando IF é uma boa solução.

Exemplo 4

```
// sketch 03-06
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
  delayPeriod = delayPeriod + 100;
  if (delayPeriod > 3000)
  {
    delayPeriod = 100;
  }
}
```

o pino do LED para que seja uma saída.

p.



d.

que 3.000, torne-o novamente igual a 100.

FOR

É uma estrutura de repetição, parecido com o loop do Arduino, porém com este comando é possível prender o programa em um determinado laço até que uma contagem tenha sido atingida

Exemplo 5

Escreva um programa em um sketch que faz um LED piscar 20 vezes, em seguida fazer uma pausa de 3 segundos e então começar tudo de novo.

Resolução

Forma Simples

```
// sketch 03-07
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);

  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);

  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);

  // repita as 4 linhas anteriores mais 17 vezes
  delay(3000);
}
```

Forma Eficiente

```
// sketch 03-08
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  for (int i = 0; i < 20; i ++){
    digitalWrite(ledPin, HIGH);
    delay(delayPeriod);

    digitalWrite(ledPin, LOW);
    delay(delayPeriod);
  }
  delay(3000);
}
```

Problema que podem ocorrer na forma Eficiente

A função loop vai precisar de muito tempo para executar o laço FOR. Desta forma pode ocorrer de:

- Não conseguir ler um botão se ele for pressionado e o programa estiver dentro de um laço FOR;
- Não receber nenhuma comunicação serial, se ela for enviada no momento que o programa estiver dentro do laço FOR;
- Qualquer outra coisa que tenha sido acionado no Arduino e estiver dentro do laço FOR pode ocorrer de o programa não executar.

Solução:

A função loop deve ser executada tão rapidamente quanto possível para que ela seja executada o maior número possível de vezes.

Resolução Melhorada

Forma Eficiente

```
// sketch 03-08
int ledPin = 13;
int delayPeriod = 100;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  for (int i = 0; i < 20; i ++){
    digitalWrite(ledPin, HIGH);
    delay(delayPeriod);
    digitalWrite(ledPin, LOW);
    delay(delayPeriod);
  }
  delay(3000);
}
```



Forma Eficiente Melhorada

```
// sketch 03-09
int ledPin = 13;
int delayPeriod = 100;
int count = 0;
void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
  count ++;
  if (count == 20)
  {
    count = 0;
    delay(3000);
  }
}
```



While

Também é um laço de repetição, que pode ser usada no lugar do comando FOR.

- A diferença que a verificação da contagem é feita antes de entrar no programa
- Para que o programa permaneça dentro do laço de repetição, a expressão do comando while deve ser verdadeira.

```
int i = 0;
while (i < 20)
{
    digitalWrite (ledPin, HIGH);
    delay(delayPeriod);
    digitalWrite (ledPin, LOW);
    delay(delayPeriod);
    i ++;
}
```

Funções (Estruturando seus sketches)

Desenvolva Funções no algoritmo para facilitar sua organização.

Exemplos de Funções:

- digitalWrite
 - Delay
 - Setup
 - loop
- } Funções interna

O que é uma Função:

Uma Função é algo um pouco parecido com um programa dentro de um programa. Usado para “empacotar” algumas coisas que você deseja realizar.

Como Funciona:

A função poderá ser chamada de qualquer lugar do seu sketch e contém as suas próprias:

- Variáveis
- Lista de comando

Quando os seus comandos terminam de ser efetuados, a execução do sketch continua no ponto imediatamente após a linha no código em que a função foi chamada.

Exemplo 6

Faça um LED piscar 20 vezes utilizando para isto uma função com a atribuição de acender e apagar o LED.

Resolução

```
// sketch 04-01
int ledPin = 13;
int delayPeriod = 250;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  for (int i = 0; i < 20; i ++)
  {
    flash();
    delay(3000);
  }
}

void flash()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
}
```

Flash()

20X

Depois de 20X entrando em flash, finalmente o código vai para a próxima linha em delay(3000).