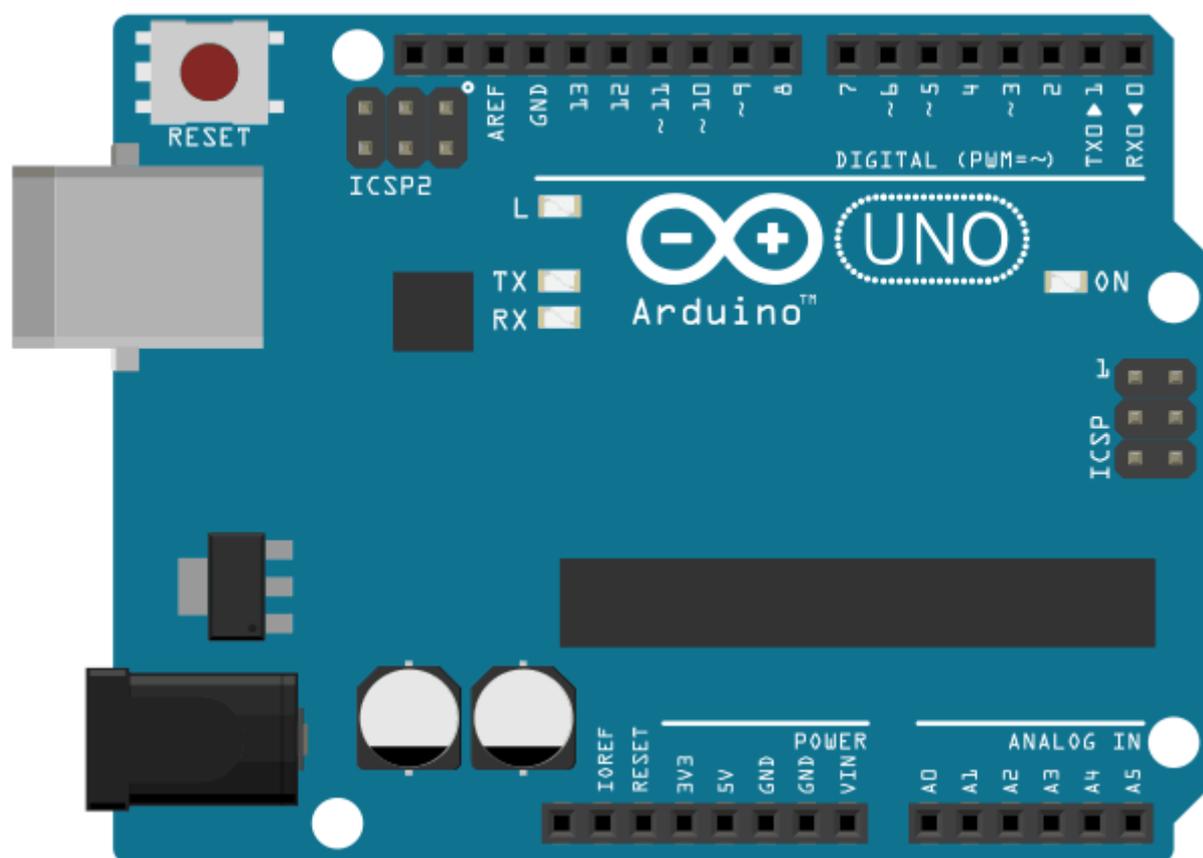


Arduino - Entradas e Saídas digitais

Por [Fábio Souza](#) - 09/12/2013



ÍNDICE DE CONTEÚDO

1. Entradas e Saídas digitais
2. Funções para Entradas e Saídas

1. Entradas e Saídas digitais

Quando você adquire uma placa Arduino Uno, a primeira coisa que vem em mente é como ligar e desligar um equipamento ou eletrodoméstico, ou como ler uma tecla do computador para enviar comandos para a placa.

Este artigo visa apresentar as funções de entrada e saída digital e exibir a correta configuração de um pino digital antes da utilização pela sua aplicação.

A placa Arduino UNO possui 14 pinos que podem ser configurados como entrada ou saídas digitais conforme a necessidade de seu projeto. Estes pinos são numerados de 0 a 13, conforme destacado na figura a seguir:

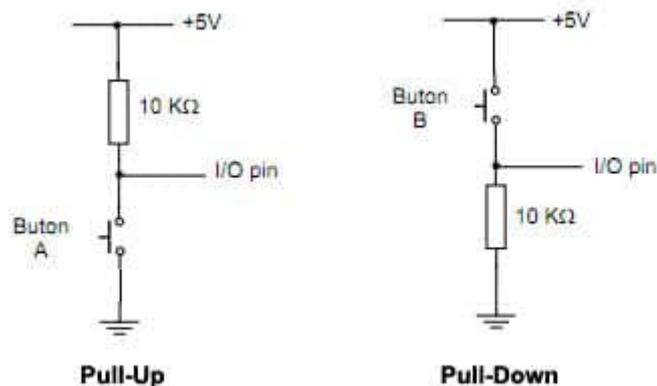
Entradas / Saídas Digitais



Figura 1 - Entradas e Saídas Digitais da placa Arduino

Antes de utilizar cada um desses pinos em sua aplicação, você deve configurá-lo como entrada ou saída digital, conforme a necessidade. Por exemplo, para acionar um LED você deve configurar o pino como saída e para ler uma tecla você deve configurar o pino como entrada.

Por padrão os pinos no Arduino estão configurados como entradas digitais, porém, para ficar mais explícito na programação, deve-se configurar o pino como entrada. Dessa forma o pino é colocado em um estado de alta impedância, equivalente a um resistor de 100 Megohms em série com o circuito a ser monitorado. Dessa forma, o pino absorve uma corrente muito baixa do circuito que está monitorando. Devido a essa característica de alta impedância, quando um pino colocado com entrada digital encontrasse flutuando (sem ligação definida), o nível de tensão presente nesse pino fica variando não podendo ser determinado um valor estável devido a ruído elétrico e até mesmo capacitância de entrada do pino. Para resolver esse problema é necessário colocar um resistor de pull up (ligado a +5V) ou um resistor de pull down (ligado a GND) conforme a necessidade. Esses resistores garantem nível lógico estável quando por exemplo uma tecla não está pressionada. Geralmente utiliza-se um resistor de 10K para esse propósito. A seguir é exibida a ligação desses resistores no circuito para leitura de tecla:



O microcontrolador ATmega328, da placa Arduino UNO, possui resistores de pull-up internos (20 Kiloohms) que facilitam a ligação de teclas, sensores sem a necessidade de conectar externamente um resistor de pull-up. A habilitação desses resistores é feita de maneira simples via software.

Quando um pino é configurado com saída, ele se encontra em estado de baixa impedância. Dessa forma, o pino pode fornecer ou drenar corrente para um circuito externo. A corrente máxima que um pino pode fornecer ou drenar é de 40 mA, porém a soma das correntes não pode ultrapassar 200 mA. Deve-se ficar atento a corrente maiores que este limite e a curto-circuitos que podem danificar o transistor de saída danificando o pino e até mesmo queimar o microcontrolador. Essa é uma característica perigosa para a placa Arduino e seria interessante se tivessem resistores ou algum tipo de proteção em todos os pinos utilizados como saída para limitar a corrente em uma situação anormal.

2. Funções para Entradas e Saídas digitais

A plataforma Arduino possui funções para trabalhar com entradas e saídas digitais que abstraem toda a configurações dos registradores que configuram e acessam os pino de I/O. Isso torna a programação do Arduino realmente fácil e esse é seu encanto. Essas funções são:

- `void pinMode();`

Essa função é utilizada para configurar um pino como entrada ou saída digital. Ela geralmente é utilizada dentro da função `setup()`. Apresenta as seguintes características:

Sintaxe:

```
pinMode(pino, modo);
```

Parâmetros:

pino: Número correspondente ao pino que se deseja configurar, conforme a placa que está trabalhando. No caso da Arduino UNO pode ser de 0 a 13;

modo: Modo que deseja configurar o pino. INPUT, INPUT_PULLUP, OUTPUT.

- INPUT: Entrada digital;
- INPUT_PULLUP: Entrada digital com resistor de pull-up (ligado ao VCC) interno habilitado;
- OUTPUT: Saída digital;

Retorno:

Essa função não tem retorno algum.

- `int digitalRead();`

Essa função lê o valor presente em um pino digital. Este valor pode ser HIGH ou LOW. Apresenta as seguintes características:

Sintaxe:

```
digitalRead(pino);
```

Parâmetros:

pino: valor correspondente ao pino que se deseja ler.

Retorno:

HIGH ou LOW.

- `void digitalWrite();`

A função `digitalWrite()` coloca um nível lógico Alto (HIGH, 5V) ou baixo (LOW, 0V) em um pino configurado como saída digital.

Sintaxe:

```
digitalWrite(pino, valor)
```

Parâmetros:

pino: Número correspondente ao pino;

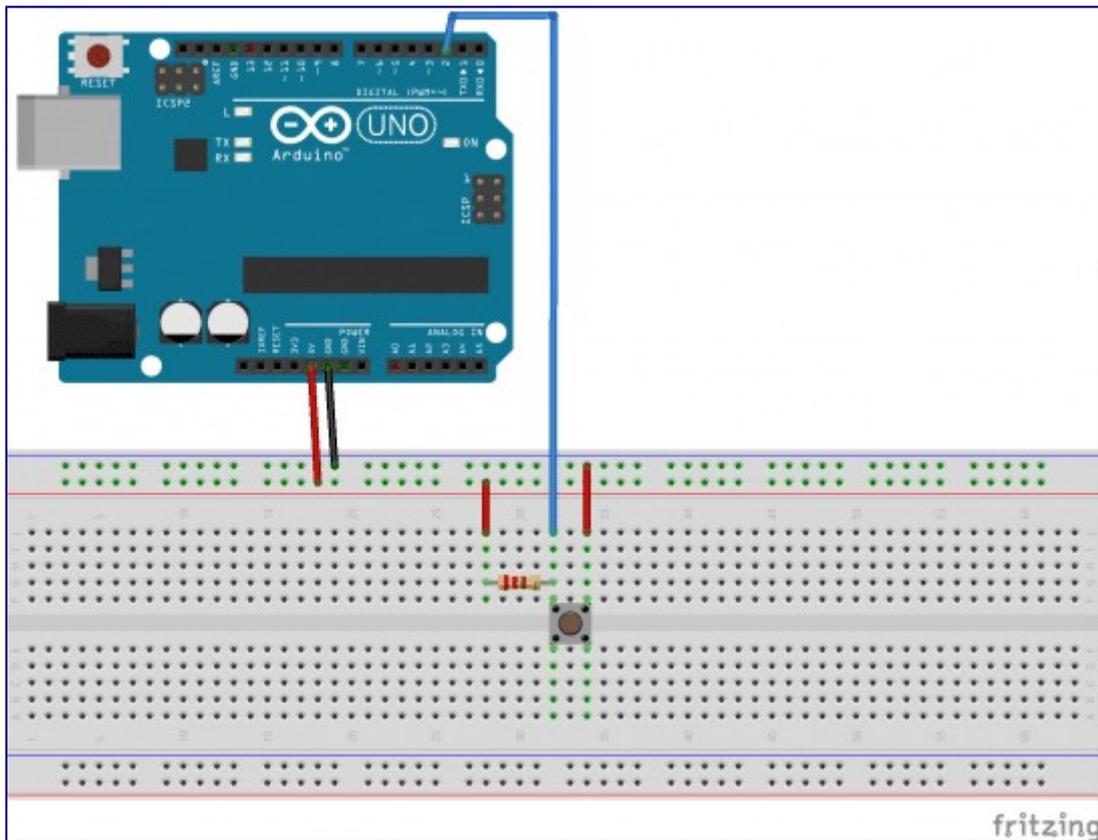
valor: HIGH OU LOW

Retorno:

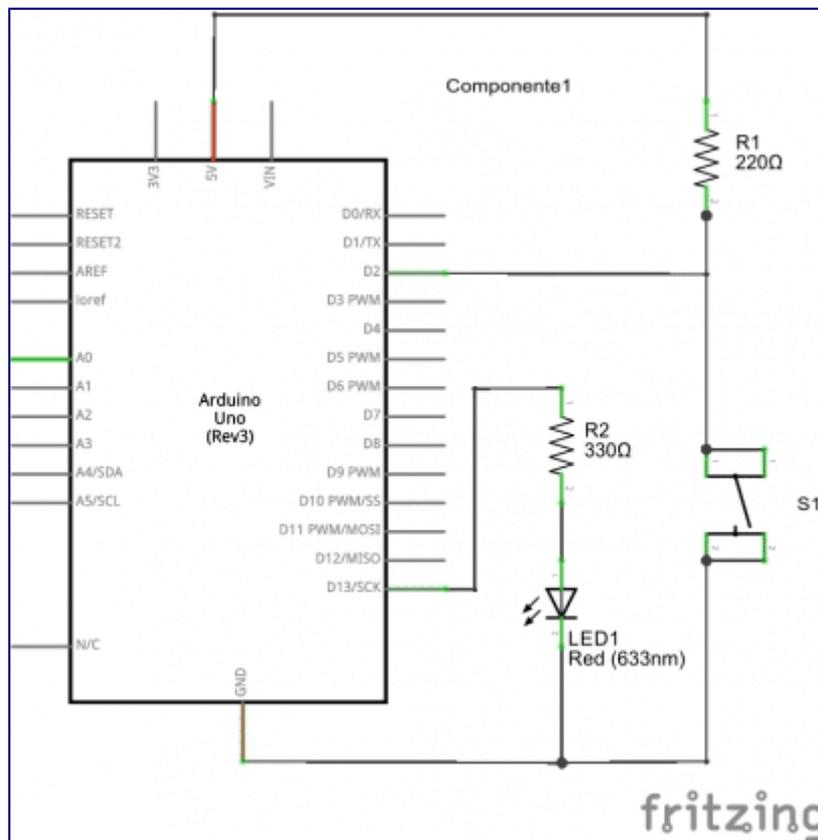
Essa função não tem retorno algum.

Exemplo

Para exemplificar a utilização de pinos de I/O digitais, vamos desenvolver uma aplicação de leitura de tecla e acionamento de LED, conforme o a montagem feita no software [Fritzing](#) exibida a seguir:



O esquema elétrico obtido a partir do Fritzing é exibido abaixo, note que no circuito há um resistor de pull-up que garante nível lógico alto quando tecla não está pressionada:



O exemplo consiste em ler a tecla S1 e ligar o LED caso a mesma estiver pressionada. Caso não esteja sendo pressionada, o LED deve permanecer desligado. O Sketch a seguir exibe a programação:

```
1  /*
2  Leitura de tecla
3  O exemplo le uma tecla conectada ao pino 2 e aciona um led conectado ao pino 13
4  */
5
6  const int ledPin = 13; // cria uma constante com o numero do pino ligado ao LED
7  const int inputPin = 2; // cria uma constante com o numero do pino conectado a tecla
8
9  void setup()
10 {
11   pinMode(ledPin, OUTPUT); // declara o pino do led como saída
12   pinMode(inputPin, INPUT); // declara o pino da tecla como entrada
13 }
14
15 void loop()
16 {
17   int val = digitalRead(inputPin); // le o valor na entrada
18   if (val == LOW) // se valor está em zero( tecla pressionada)
19   {
20     digitalWrite(ledPin, HIGH); // Liga LED indicando tecla pressionada
21   }
22   else
23   {
24     digitalWrite(ledPin, LOW); // Desliga LED indicando tecla solta
25   }
26 }
```

A programação apresentada acima possui uma estrutura bem simples e serve de início para a manipulação de pinos de I/O digitais. Com as três funções apresentadas é possível aplicar em diversos projetos que necessitem de acionamento e leitura de sinais digitais. Agora para fixar os conceitos apresentados é necessário colocar a mão na massa!!! Fica como exercício o desenvolvimento de um sketch para leitura de tecla com resistor de pull up interno habilitado para o pino onde é conecta a tecla S1.

<https://www.embarcados.com.br/arduino-entradasaidas-digitais/>