Microcontrolador Arduino UNO

Raul Ramires[1]

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brazil

ra82293@uem.br

1. Introdução

O arduino é uma placa microcontroladora baseada no microcontrolador ATMega e pode ser integrada com uma grande diversidade de componentes e sensores. O projeto a ser desenvolvido propõe a configuração do ambiente de trabalho e a realização de diversos experimentos com diferentes sensores.

2. Ambiente de Trabalho

O arduino possui uma IDE *Integrated Development Environment* chamada de "Arduino IDE", que é onde serão escritos os códigos para que depois sejam enviados ao arduino e executado.

2.1. Download da IDE

Para fazer o download da IDE basta ir ao site do arduino e escolher a versão de acordo com o sistema operacional utilizado, estando disponível para Windows, Linux e MacOS.

3. Programação

A programação do arduino é feita na linguagem C++ e possui uma grande quantidade de bibliotecas prontas para a utilização de sensores.

3.1. Estrutura do código

A Figura 1 mostra a estrutura básica de um código para arduino. Esse código é dividido em duas funções: a função *setup* e a função *loop*.

```
void setup() {
  //Executa apenas uma vez
}

void loop() {
  //Executa indefinidamente
}
```

Figura 1. Estrutura do código.

A função *setup* será executada apenas uma vez no início do programa, geralmente é aqui que ficam as definições de pinos de entradas e saídas e algumas variáveis globais. A função *loop* será executada infinitamente enquanto o arduino estiver ligado, é aqui que serão feitas as leituras de sensores e decisões de ações com bases nos dados lidos dos sensores.

3.2. Upload do código

Para realizar o *upload* do código para a placa do arduino, basta clicar no botão “carregar”, então o código será compilado e carregado para o arduino e então começar a execução.

4. Experimentos

Nessa seção serão realizados vários experimentos com o arduino, utilizando uma grande diversidade de componentes e sensores.

Todos os experimentos realizados nesse trabalho estão disponíveis no repositório no *GitHub* e os esquemáticos dos circuitos estão disponíveis no *Tinkercad*.

4.1. Piscar um LED

Para realizar esse experimentos será necessário um LED, uma *protoboard* e dois *jumpers*.

A *protoboard* é construída como uma matriz, onde cada coluna possui 5 pontos de contato que são interligados entre si, porém uma coluna é isolada de sua vizinha, sendo necessário a utilização de um *jumper* para interligar duas colunas.

Alguns modelos de *protoboard* possuem dois barramentos laterais, um negativo e um positivo que percorrem a *protoboard* inteira. O padrão de utilização desses barramentos é ligar o 5V no barramento vermelho e o GND no barramento preto, ficando assim mais simples de realizar a alimentação dos componentes utilizados.

A Figura 2 mostra o esquema de uma *protoboard* de 360 pontos.

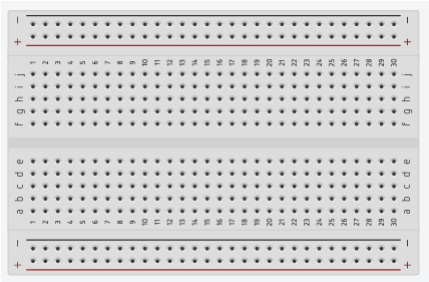

Figura 2. *Protoboard* de 360 pontos

A Figura 3 mostra o esquema do circuito para piscar um LED, utilizando o arduino.

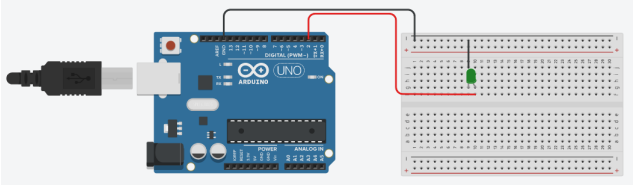

Figura 3. Esquemático do circuito do experimento 1.

Nesse circuito o pino digital 2 do arduino está ligado na coluna 10 da protoboard, onde, nessa mesma coluna está ligado o terminal positivo do LED. O terminal negativo do LED está ligado na coluna 9, que possui um *jumper* para o barramento preto, que por sua

vez está ligado ao GND do arduino. Sendo assim o LED irá acender quando o arduino enviar o sinal de 5V no pino 2.

A Figura 4 mostra o código que fará com que o arduino pisque o LED.

```
void setup() {
  //Executa apenas uma vez

  //Definição do pino positivo que será ligado ao LED
  int pinoLED = 2

  //Definição do pino pinoLED como saída
  pinMode(pinoLED, OUTPUT);
}

void loop() {
  //Executa indefinidamente

  //Envia um sinal alto (5V) para o pinoLED
  digitalWrite(pinoLED, HIGH);

  //Espera 1 segundo
  delay(1000);

  //Envia um sinal baixo (0V) para o pinoLED
  digitalWrite(pinoLED, LOW);

  //Espera 1 segundo
  delay(1000);
}
```

Figura 4. Código do experimento 1.

Referências