

FATEC

Desenvolvimento de Software Multiplataforma

2º SEMESTRE 2024

IAL-011 - Internet das Coisas e Aplicações

Prof. Me. Eng. Santana

Revisão

Declaração de variaveis

`int ledPin = 13; // Declara uma variável do tipo inteiro`

`float temperature = 24.5; // Declara uma variável do tipo float`

`char myChar = 'A'; // Declara uma variável do tipo char`

`const int ledpin=13; // constante que não poderá ser alterada`

Vetores e Matrizes

- Vetores (Arrays Unidimensionais): Um vetor é uma coleção ordenada de elementos do mesmo tipo, acessados por um índice inteiro.

```
int vetor[5];
```

```
int vetor[5] = {10, 20, 30, 40, 50};
```

Matrizes (Arrays Multidimensionais):

Uma matriz é uma coleção multidimensional de element organizados por combinações de índices.

```
int matriz[3][3];
```

```
int matriz[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
```

Estruturas de Controle

```
if (sensorValue > 500) {  
    // Código a ser executado se a condição for verdadeira  
    digitalWrite(ledPin, HIGH);  
} else {  
    // Código a ser executado se a condição for falsa  
    digitalWrite(ledPin, LOW);  
}
```

Estrutura de Seleção

```
switch (opcao) {  
    case 1:  
        //executar instruções  
        break;  
    case 2:  
        //executar instruções  
        break;  
    default: //executar instruções caso não  
            encontre um case com a opção  
            informada  
        break;  
}
```

Estruturas de Repetição

```
for (int i = 0; i < 10; i++) {  
    // Executa este bloco 10 vezes  
    digitalWrite(ledPin, HIGH);  
    delay(100);  
    digitalWrite(ledPin, LOW);  
    delay(100);  
}
```

```
int contador = 1;  
while (contador <= 10) {  
    Serial.println (contador);  
    contador = contador + 1;  
• }
```

Metodos/Funcoes Especificas Arduino

```
void setup() {  
  // Inicializa o pino 13 como saída  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  // Acende o LED conectado ao pino 13  
  digitalWrite(13, HIGH);  
  delay(1000); // Espera por um segundo  
  // Apaga o LED conectado ao pino 13  
  digitalWrite(13, LOW);  
  delay(1000); // Espera por um segundo  
}
```

Funcoes - Arduino

- Entrada e Saída Digital
- Função pinMode()

```
void setup() {  
    pinMode(13, OUTPUT); // Configura o pino 13 como  
    saída  
    pinMode(7, INPUT); // Configura o pino 7 como  
    entrada  
}
```


Funcoes - Arduino

- Função `digitalWrite()`

`digitalWrite(13, HIGH);` // envia ao pino 13 HIGH (5V)

`digitalWrite(13, LOW);` // envia ao pino 13 LOW (0V)

Funcoes - Arduino

- Função `digitalRead()`

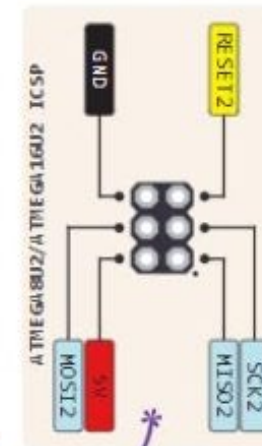
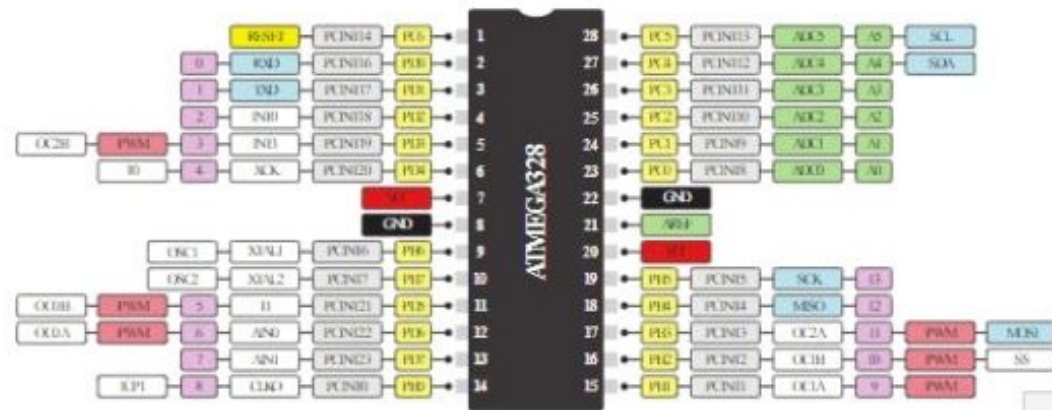
```
int buttonState = digitalRead(7);
```

- Função `analogRead()`

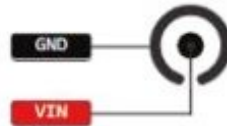
```
int SensorValue = analogRead(7);
```

THE DEFINITIVE ARDUINO UNO PINOUT DIAGRAM

- ⚠ Absolute max per pin 40mA recommended 20mA
- ⚡ Absolute max 200mA for entire package

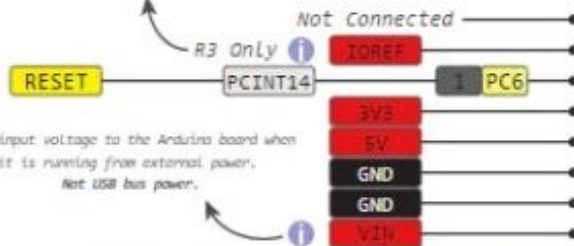


7-12V Depending on current drawn

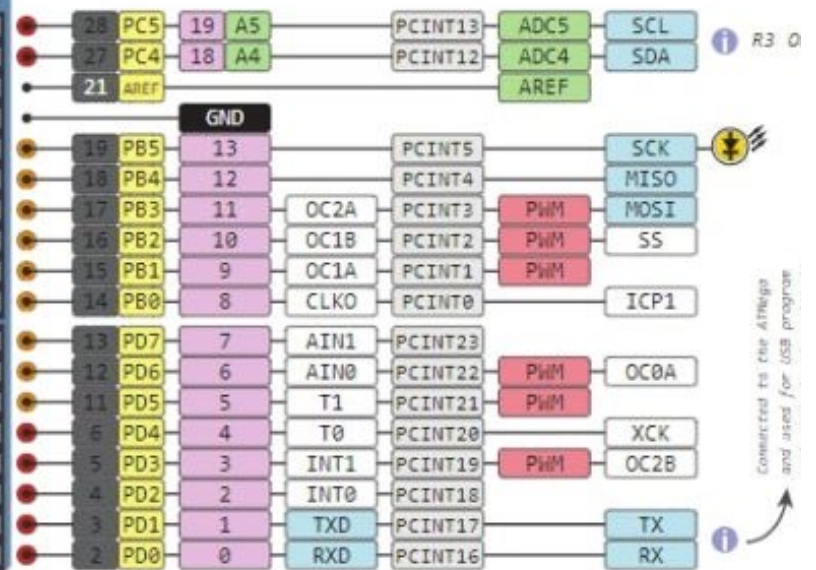
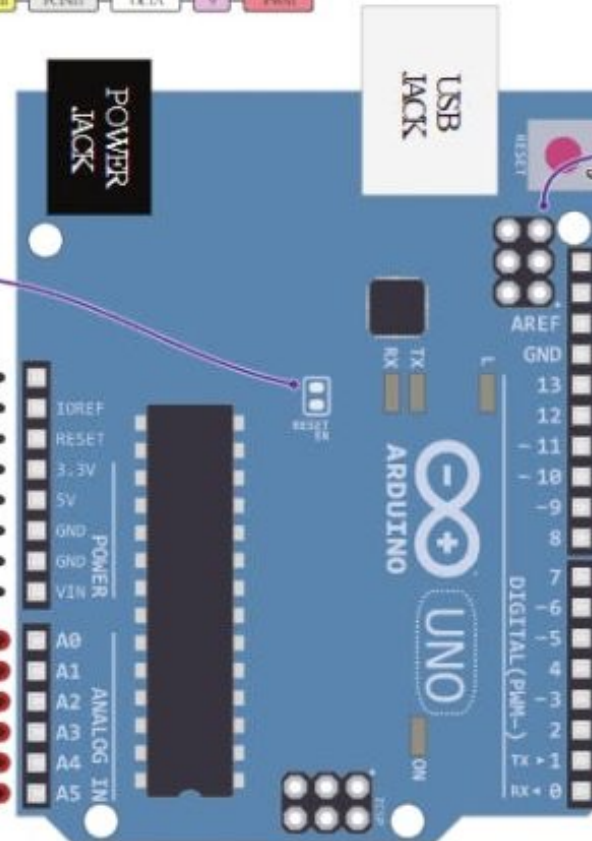
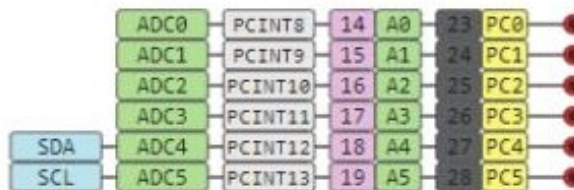


Cut to disable the auto-reset

This provides a logic reference voltage for shields that use it. It is connected to the 5V bus.



The input voltage to the Arduino board when it is running from external power, not USB bus power.



Connected to the Atmega and used for USB program



www.pineapple.com



18 FEB 2013

ver 2 rev 2 - 05.03.2013

Entradas Digitais

- As entradas e saídas digitais são fundamentais no controle e monitoramento de dispositivos eletrônicos com o Arduino. As entradas digitais permitem que o Arduino receba sinais binários (0 ou 1) de sensores ou dispositivos externos, como botões, chaves ou sensores de movimento.
- Quando um sensor envia um sinal de alta tensão (geralmente 5V), o Arduino interpreta isso como um valor 1 (HIGH); quando o sinal é de baixa tensão (0V), o valor é 0 (LOW). Isso permite que o Arduino tome decisões com base em estados simples, como ligar uma luz quando um botão é pressionado.

Entradas Analógicas

- Entradas analógicas no Arduino permitem a leitura de sinais que variam continuamente dentro de um intervalo específico de tensões, geralmente entre 0V e 5V. Esses sinais são provenientes de sensores analógicos, como potenciômetros, sensores de temperatura ou luz, que geram valores que não são apenas 0 ou 1, mas qualquer valor dentro desse intervalo.
- O Arduino converte esse valor de tensão em um valor digital correspondente, que varia de 0 a 1023, utilizando um conversor analógico-digital (ADC). Isso permite ao Arduino interpretar uma vasta gama de valores, possibilitando maior precisão e controle em projetos que requerem sensibilidade a mudanças sutis em variáveis como temperatura, luz ou pressão.

Saídas Digitais

- As saídas digitais, por outro lado, permitem que o Arduino controle dispositivos externos, enviando sinais binários. Por exemplo, o Arduino pode acionar LEDs, motores ou relés, enviando sinais de 0V ou 5V para os pinos de saída.
- Um exemplo comum é o controle de um LED, onde o Arduino envia um sinal HIGH para acender o LED e um sinal LOW para apagá-lo. Essas operações são simples, mas essenciais para a criação de sistemas interativos e automatizados, onde o Arduino atua como o cérebro que processa entradas e aciona saídas conforme programado.

Componentes Comuns/Básicos

- Resistores



Componentes Comuns/Básicos

- LED (Diodo Emissor de Luz)

