

Internet das Coisas

Aula 09 – Porta digital comum como saída



Portas digitais comuns

O Arduino possui uma série de portas digitais. Algumas delas comuns (as que veremos neste material) e algumas delas com funções especiais como as PWM (que veremos no próximo material). As portas digitais comuns, como já vimos, são capazes de trabalhar com apenas 2 estados na entrada ou na saída: o estado baixo (LOW, equivalente ao 0 binário e à tensão de 0V) e o estado alto (HIGH, equivalente ao 1 binário e à tensão de 5V).

Portas digitais comuns

Deste modo, seja funcionando como porta de entrada (recebendo valores de sensores) ou funcionando como porta de saída (enviando informações ou ativando e desativando dispositivos externos), as portas digitais comuns trabalham apenas como essa variação binária. Essa variação pode ser única (ou está em estado baixo ou está em estado baixo) ou pode ser sequencial (alterna entre o estado alto e o baixo com uma determinada frequência e em um determinado tempo, lendo ou gerando uma sequência binária).

Definindo uma porta como entrada ou saída

Para definir uma porta como de entrada ou de saída (seja ela digital ou analógica), utilizamos o método abaixo (geralmente dentro do método setup):

```
pinMode( <porta>, <modo> );
```

<porta>

número da porta que queremos definir o modo

<modo>

modo de funcionamento da porta, que pode receber os valores.

Pode receber os valores:

- **INPUT**: porta funciona como entrada (leitura)
- **OUTPUT**: porta funciona como saída (controle)
- **INPUT_PULLUP**: funciona com porta com pullup (botões)

Alterando o estado da porta digital

Definida como uma porta de saída, a porta digital pode ter seu estado alterado para baixo (LOW, 0 binário, 0V) ou para alto (HIGH, 1 binário, 5V) com o método abaixo:

```
digitalWrite( <porta>, <estado> );
```

<porta>

número da porta que queremos definir o estado de saída

<estado>

Estado lógico da porta. Pode receber os valores:

- LOW: estado baixo (0V, desligada)
- HIGH: estado alto (5V, ligada)

Criando paradas temporizadas

Outro método muito útil para garantir o estado de uma porta por um determinado tempo é o método `delay`, que para a execução do código pelo tempo determinado:

```
delay( <tempo> );
```

<tempo>

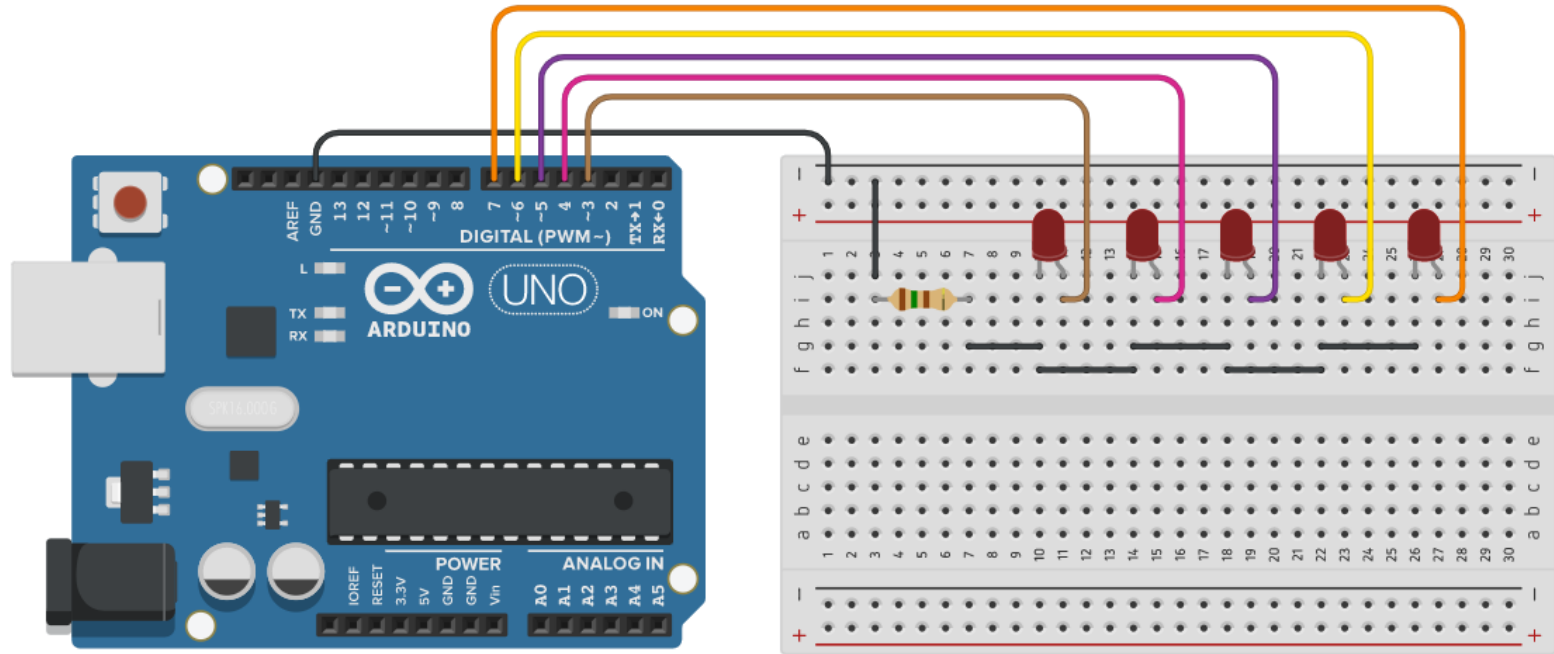
tempo de parada, informado em milissegundos (ou seja, cada 1000 equivale a 1 segundo)

Exemplo: ligando e desligando Leds

A combinação destes 3 métodos nos permite criar um circuito com o Arduino, programando-o para controlar o ligamento e o desligamento de leds em uma sequência pré-determinada. Em nosso exemplo, vamos controlar 5 leds. Para isso, os pinos da entrada positiva (os mais compridos ou os com joelho), devem ser conectados, cada um, à uma porta digital diferente (no nosso caso, nas portas 3, 4, 5, 6 e 7), para que ao alterar os estados para baixo e alto, os leds possam ligar e desligar. Já os pinos da saída negativa devem ser ligados entre si e passar por um resistor (no nosso exemplo, de 560Ω) ligado à linha negativa da protoboard (sendo esta ligada ao GND do Arduino). Assim, baixamos a corrente para proteger os leds.

Exemplo: ligando e desligando Leds

Nosso esquema eletrônico deverá ficar como o apresentado abaixo:



Exemplo: ligando e desligando Leds

Agora vamos ao nosso código. Primeiramente dentro do setup, devemos utilizar o método pinMode, para definir que as 5 portas são portas de saída. Depois disso, ainda dentro do setup, podemos utilizar o método digitalWrite utilizando o valor LOW para garantir que todas as 5 portas iniciem desligadas. Teremos então no setup, o código ao lado:

```
void setup(){  
  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, LOW);  
}
```



Exemplo: ligando e desligando Leds

Finalmente, dentro do loop, com o método `digitalWrite` (alternando o valor LOW e HIGH) e com o método `delay`, podemos criar sequências de ligamento e desligamento dos leds, gerando assim efeitos rítmicos nesse processo. Veja três exemplos diferentes a seguir:

Exemplo: ligando e desligando Leds

```
void loop() {  
  
    delay(500);  
    digitalWrite(3, HIGH);  
    delay(500);  
    digitalWrite(4, HIGH);  
    delay(500);  
    digitalWrite(5, HIGH);  
    delay(500);  
    digitalWrite(6, HIGH);  
    delay(500);  
    digitalWrite(7, HIGH);  
  
    delay(500);  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, LOW);  
    delay(1000);  
  
}
```

```
void loop() {  
  
    delay(500);  
    digitalWrite(3, HIGH);  
    digitalWrite(7, HIGH);  
    delay(500);  
    digitalWrite(4, HIGH);  
    digitalWrite(6, HIGH);  
    delay(500);  
    digitalWrite(5, HIGH);  
  
    delay(500);  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, LOW);  
    delay(1000);  
  
}
```

```
void loop() {  
  
    delay(500);  
    digitalWrite(3, HIGH);  
    digitalWrite(7, HIGH);  
    delay(500);  
    digitalWrite(3, HIGH);  
    digitalWrite(7, HIGH);  
  
    delay(500);  
    digitalWrite(5, HIGH);  
    delay(500);  
    digitalWrite(5, LOW);  
  
    delay(500);  
    digitalWrite(4, HIGH);  
    digitalWrite(6, HIGH);  
    delay(500);  
    digitalWrite(4, HIGH);  
    digitalWrite(6, HIGH);  
  
}
```

Exemplo: ligando e desligando Leds

Foto do circuito montado e funcional:

