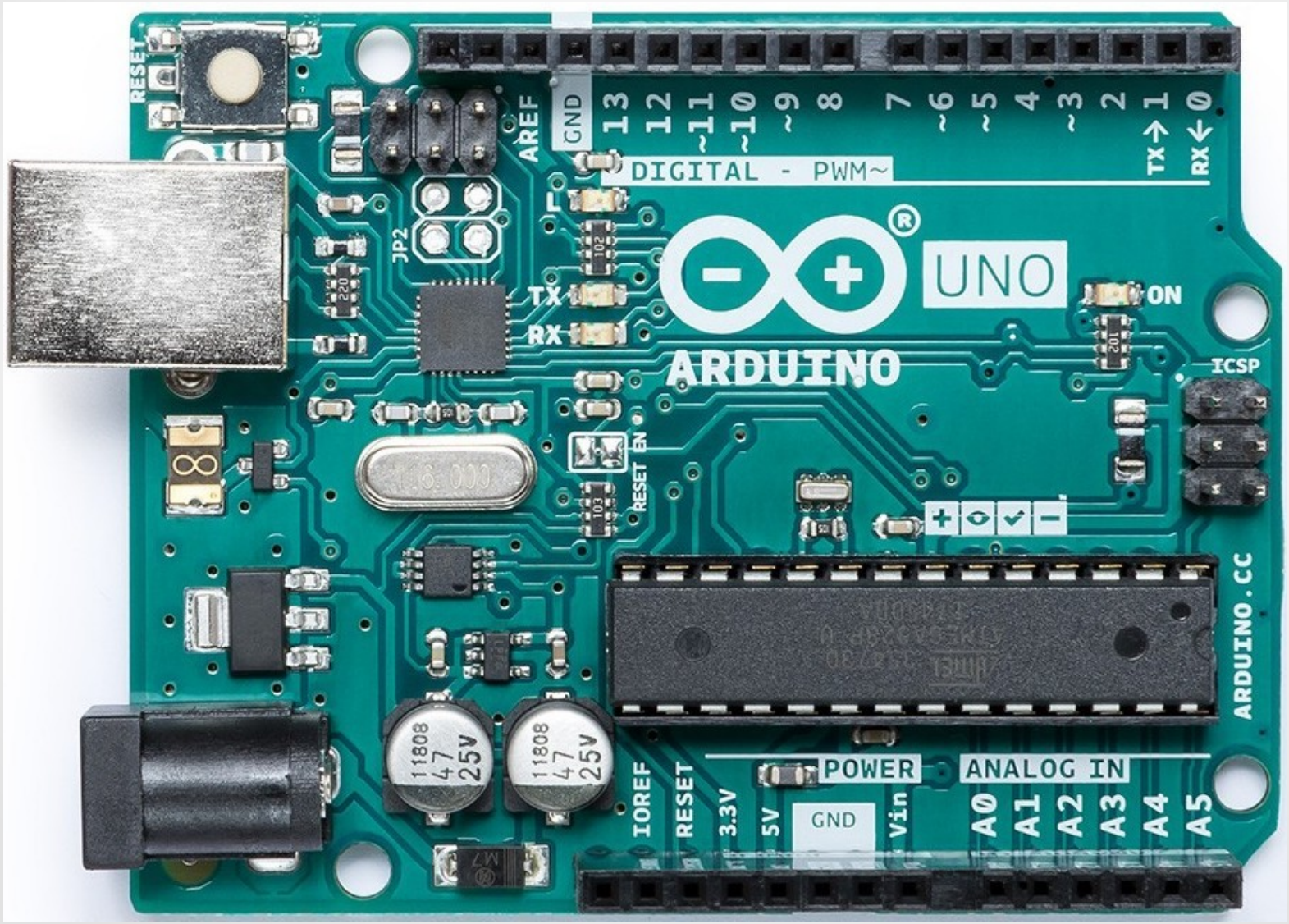
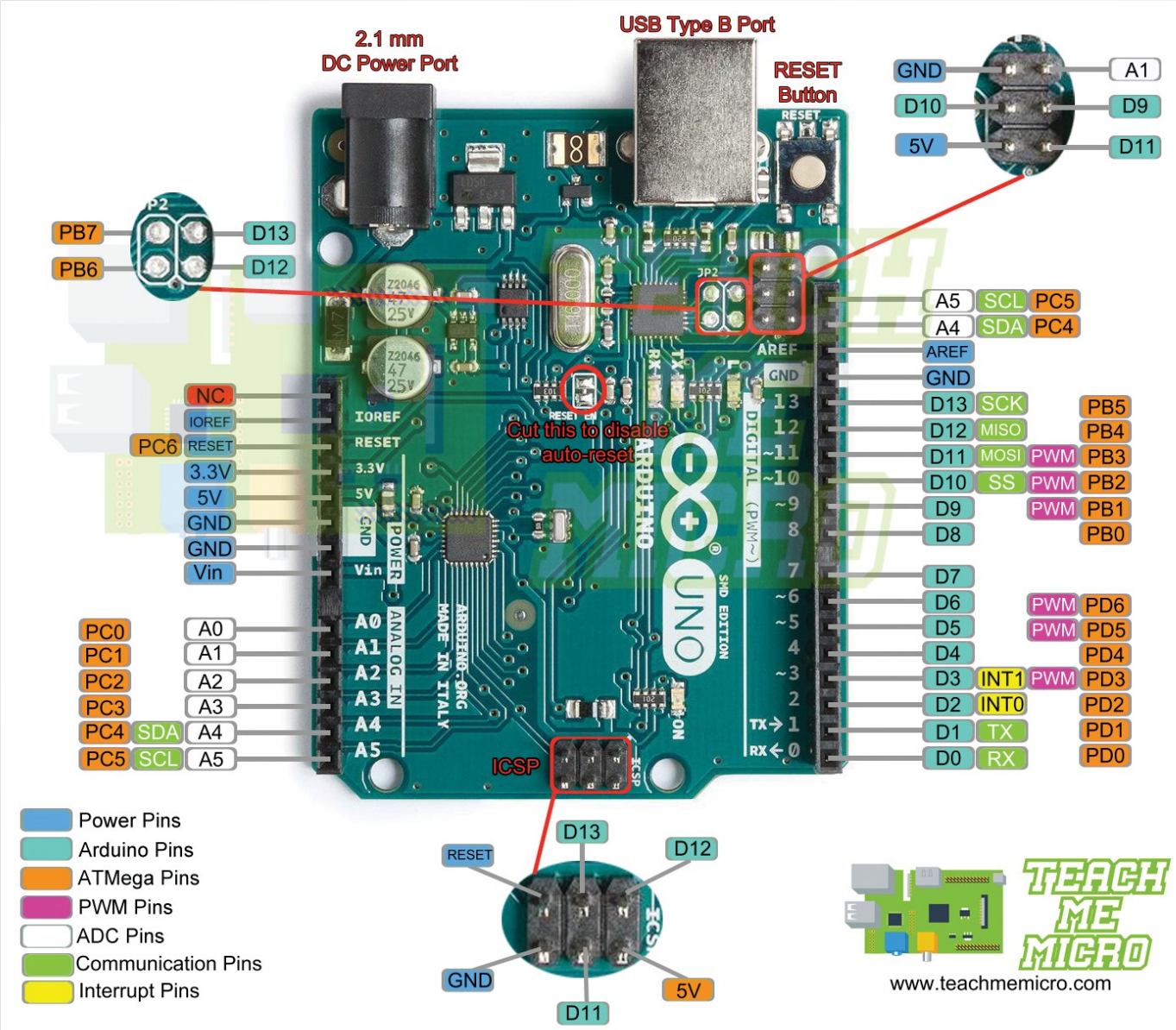
Saídas analóxicas



Víctor . Álvarez 201911

Introdución – pinout Arduino



E/S

Dixitais:

- Pins 0 a 13
  0 a 5V, 20 mA
  Low: 0 a 2V
  High: 3 a 5V

Entradas Analóxicas

- Pins A0 – A5
  0 a 5V, 20 mA prec 1024
  0 a 3V, 50 mA prec 1024

Saídas PWM

- ~ Pins 3, 5, 6, 9, 10, 11
  0 a 5 V, 20 mA prec 256

Comunic. Serie TX/RX

- Pins 0 e 1

ICSP

- 6 pins para comunicarse directamente co proc. Atmega328
- 6 pins para programar o USB

Saídas analóxicas – brillo dun LED

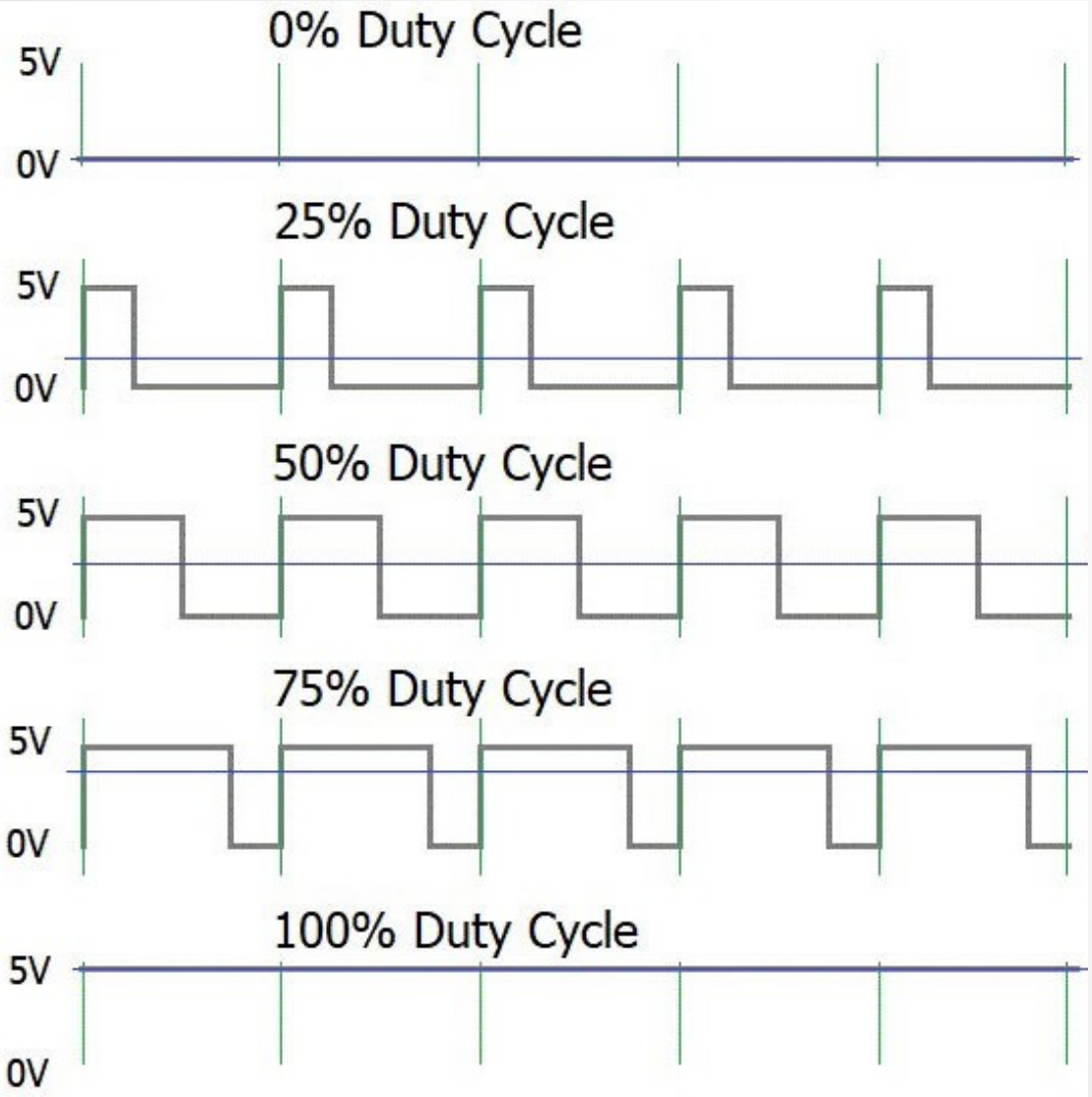
- En Arduino non hai saídas analóxicas, pero pódense emular mediante Pulse Width Modulation (PWM).
- Os pins que habilitan PWM están sinalados con ‘~’ e permiten saídas entre 0 e 5 V cunha precisión de 256 valores (19.5 mV).
- NON é necesario declarar os pins PWM con pinMode(), ponse directamente un valor analóxico no pin empregando analogWrite().
- Por exemplo:

```
analogWrite(6, 128);
```

pon un valor de 2.5 V no pin PWM ~6.

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED



- Que valores de voltaxe corresponden con cada un dos pulsos da figura adxunta?
- Como se poden calcular valores intermedios?

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED

- Reutiliza un dos LEDs da bredboard para usar cos dous seguintes scripts.
- Selecciona seis niveis de brillo para o LED, exprésaos en valores entre 0 e 255 e realiza dous scripts en Sketch de maneira que:
  - en primeiro lugar o LED brille de menos a máis e así sucesivamente,
  - en segunto lugar o LED brille de menos a máis e de máis a menos sucesivemente.
- Fai o mesmo incrementando o brilo cun valor fixo e detectando cando sae do intervalo [0, 255].

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED

brillo.led.raw | Arduino 1.8.10

brillo.led.raw

```
#define LED 6
#define NIVEL00 0
#define NIVEL01 51
#define NIVEL02 102
#define NIVEL03 153
#define NIVEL04 204
#define NIVEL05 255

int tempo = 150;

void setup() {
  // Non é necesario declarar as
  // saídas analóxicas.
}

void loop() {
  analogWrite(LED, NIVEL00);
  delay(tempo);
  analogWrite(LED, NIVEL01);
  delay(tempo);
  analogWrite(LED, NIVEL02);
  delay(tempo);
  analogWrite(LED, NIVEL03);
  delay(tempo);
  analogWrite(LED, NIVEL04);
  delay(tempo);
  analogWrite(LED, NIVEL05);
  delay(tempo);
  analogWrite(LED, NIVEL04);
  delay(tempo);
  analogWrite(LED, NIVEL03);
  delay(tempo);
  analogWrite(LED, NIVEL02);
```

```
int tempo = 150;

void setup() {
  // Non é necesario declarar as
  // saídas analóxicas.
}

void loop() {
  analogWrite(LED, NIVEL00);
  delay(tempo);
  analogWrite(LED, NIVEL01);
  delay(tempo);
  analogWrite(LED, NIVEL02);
  delay(tempo);
  analogWrite(LED, NIVEL03);
  delay(tempo);
  analogWrite(LED, NIVEL04);
  delay(tempo);
  analogWrite(LED, NIVEL05);
  delay(tempo);
  analogWrite(LED, NIVEL04);
  delay(tempo);
  analogWrite(LED, NIVEL03);
  delay(tempo);
  analogWrite(LED, NIVEL02);
  delay(tempo);
  analogWrite(LED, NIVEL01);
  delay(tempo);
  analogWrite(LED, NIVEL00);
  delay(tempo);
}
```

Carregamento completo

O rascunho usa 1272 bytes (3%) do espaço de armazenamento
Variáveis globais usam 9 bytes (0%) de memória dinâmica,

9 Arduino/Genuino Uno em /dev/cu.wchusbserial1420

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED

- O script do lado modifica o brillo do LED aumentando paulatinamente a tensión aplicada no pin 6, entre 0 e 5 V.
- Estas tensións mapéanse a valores enteiros entre 0 e 255 en incrementos de 5 unidades.
- Para detectar os límites do rango empregamos unha nova estructura, o condicional if().

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED

- Un condicional if() ten a estrutura que aparece abaixo. A sentencia else non leva condición e é opcional.
- Cando a condición evalúa a TRUE, execútase o primeiro bloque, en caso contrario o segundo bloque (se existe).

```
if( condicion ) {
  // Tarefas que se realizan
  // cando condicion == TRUE
}
else {
  // Tarefas que se realizan
  // cando a condicion != TRUE
}
```

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED

- A condición é calquera expresión que se avalíe a TRUE, o que significa que toma un valor maior ou igual a 1.
- Pode ser unha variable ou ben unha expresión que inclúa variables, comparacións e operadores lóxicos.
- A seguir tes un exemplo de expresión condicional con varias cláusulas compostas (usando comparacións) e unidas por operadores lóxicos).

```
if(dia == lectivo && dia != festivo && (hora >= inicRecreo || hora <= finRecreo) && !garda) {
  // Pausa para café
}
else {
  // Outras tarefas
}
```

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED

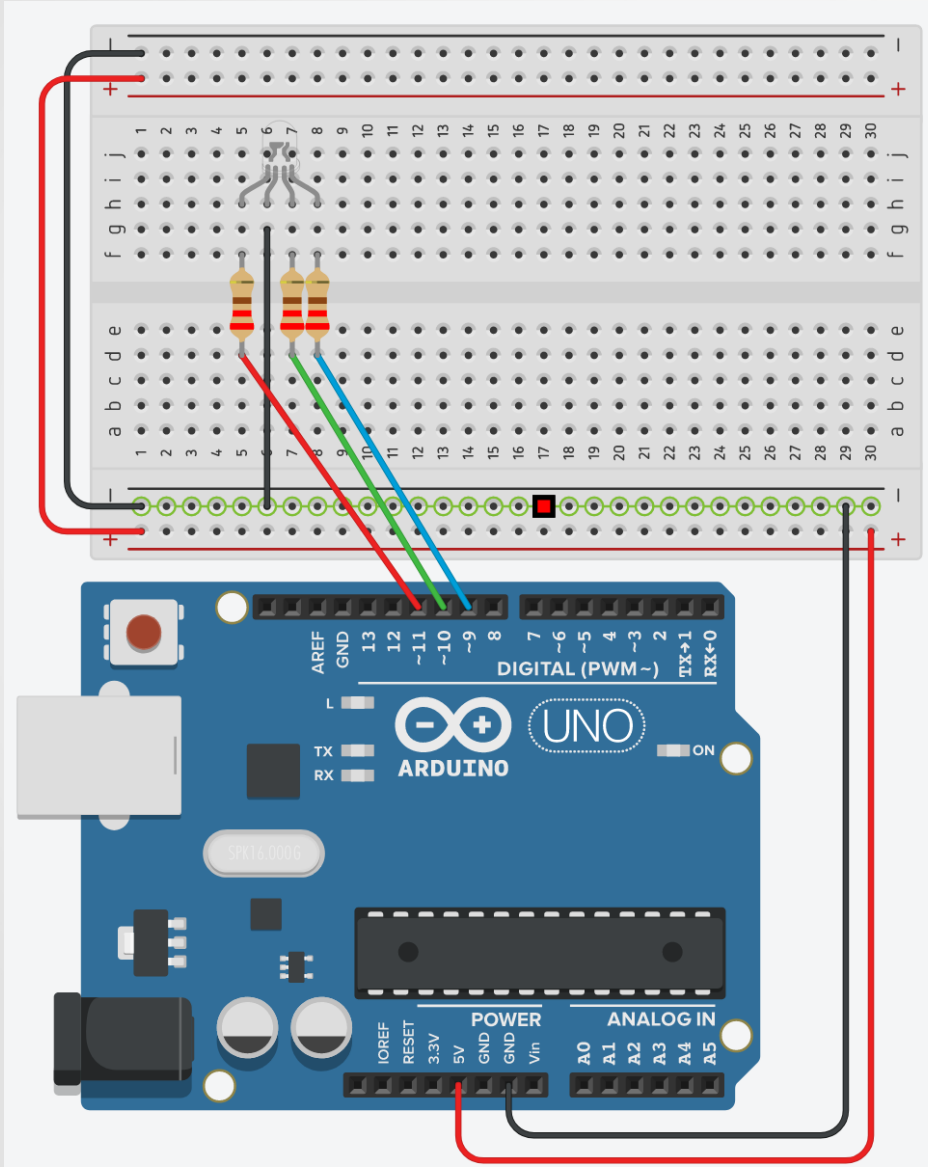
sketch_nov07a §

```
//Operadores lóxicos
  &&   //Operador AND lóxico     (nota > 5) && (nota < 3)   FALSE
  ||   //Operador OR lóxico      (nota > 5) || (nota <= 5)  TRUE
  !    //Operador NOT lóxico     !aprobado

//Operadores de comparación
  ==   //Igualdade (non confundir    nota == 5
       //coa asignación: '=')
  !=   //Desigualdade               nota != 0
  <    //Menor que                  nota < 3
  >    //Maior que                  nota > 7
  <=   //Menor ou igual que         nota <= 10
  >=   //Maior ou igual que         nota >= 5

//Operadores aritméticos unarios
  ++   //Incremento nunha unidade e asignación          i++
  --   //Decremento nunha unidade e asignación          i--
  +=   //Incrementa a cantidade a seguir e asigna       i += 1, i += 8
  -=   //Decrementa a cantidade a seguir e asigna       i -= 1, i -= 3
  *=   //Multiplica pola cantidade a seguir e asigna    i *= 4
  /=   //Divide pola cantidade a seguir e asigna        i /= 9
```

- Operadores máis usados e exemplos

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED RGB



- Un LED RGB integra tres LEDs no mesmo encapsulado (red, green, blue), cun pin común (normalmente o cátodo).
- Simula o ciclo dun semáforo: verde, amarelo e vermello, de maneira que a cor amarela escintile antes de dar paso ao vermello.
- Recomendación: define unha función para por a cor, p.ex.:

  void setColor(int r, int g, int b)

DataSheet LED RGB

Víctor . Álvarez 201911

Saídas analóxicas – brillo dun LED RGB

```
void setColor(int red, int green, int blue) {
  //Para o caso en que o LED RGB sexa
  //con ánodo común
  #ifdef ANODO_COMUN
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  //Execútase esta parte en todos os casos
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

- A directiva #ifdef é unha sentencia do preprocesador que se executa en tempo de compilación
- Se a constante a seguir de #ifdef está definida na cabeceira (ou noutro sitio), inclúe na compilación o código posterior até a directiva #endif, en caso contrario esa parte é ignorada polo compilador.
- A vantaxe destas directivas é que permiten modificar o sketch ao cambiar as características do hardware, sen ocupar tempo de programa e aforrando memoria na placa.

DataSheet LED RGB

Víctor . Álvarez 201911

Saídas analóxicas

- Nesta unidade aprendemos a:
  - usar os pins PWM (~) para simular saídas analóxicas, usando a función analogWrite()
  - usar estructuras condicionais if() e if()-else; non practicamos as concatenación de condicións: if()-else if()-else.
  - Elaborar condicións complexas cos operadores lóxicos e condicionais
  - usar outra directiva do preprocesador: #ifdef (seguido de #endif) para evitar o condicional por programación
  - practicar o uso de funcións definidas polo usuario, para evitar código repetitivo e evitar mantementos innecesarios

Víctor . Álvarez 201911