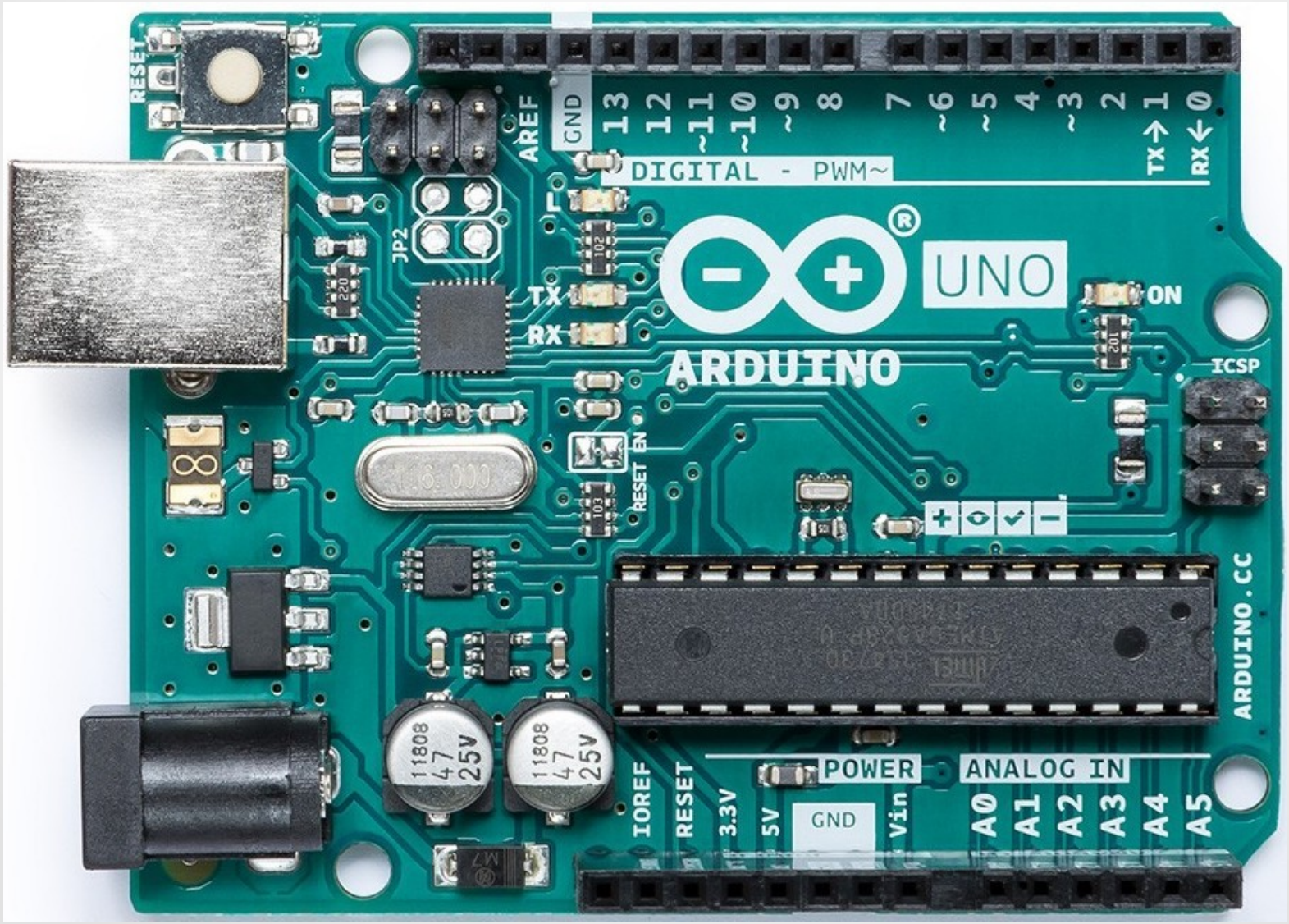
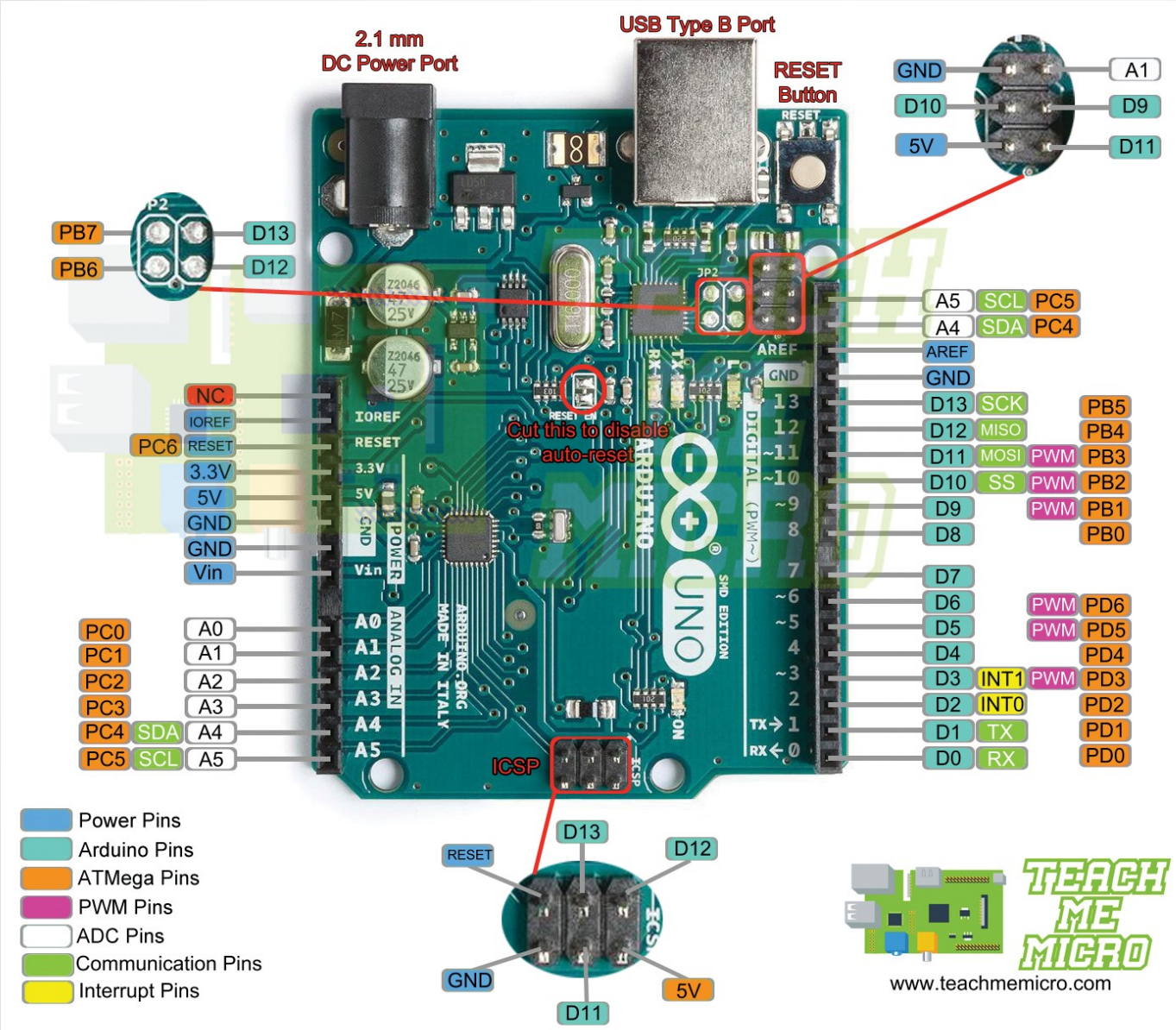
Entradas dixitais



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

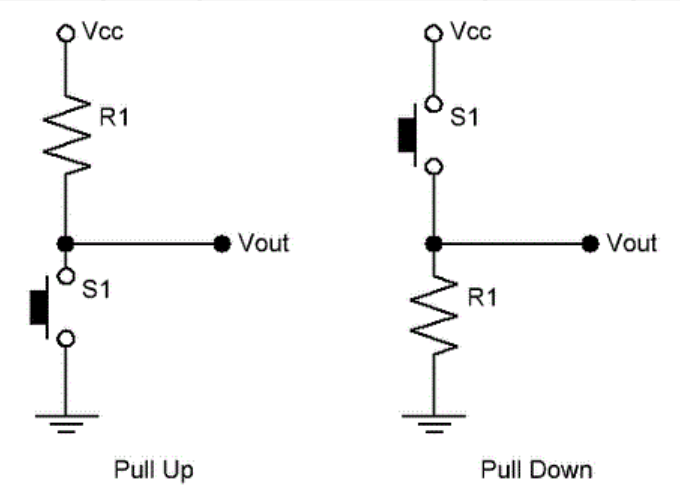
- Pins 0 e 1

ICSP

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

Entradas dixitais – Detección de pulsador

- Para probar as entradas imos manter a montaxe de tres LEDs conectados a tres dos pins I/O de Arduino. Aparte conectaremos un pulsador e en función do estado do mesmo, programaremos os LEDs.
- Debemos ter en conta que para Arduino, o un ‘1’ lóxico corresponde cun valor de entre 3 e 5 V no pin e un ‘0’ lóxico corresponde con entre 0 e 2 V. Se a placa detecta valores na banda de garda (entre 2 e 3 V), o estado do pin queda indeterminado. É dicir, en teoría pode detectar un ‘1’ ou un ‘0’ lóxico indistintamente.
- Para prever esta situación, ao pulsador engádeselle unha resistencia de ‘pull up’ ou ‘pull down’ segundo o valor que queremos no pin ao premer no pulsador.



Víctor . Álvarez 201911

Entradas dixitais – Detección de pulsador

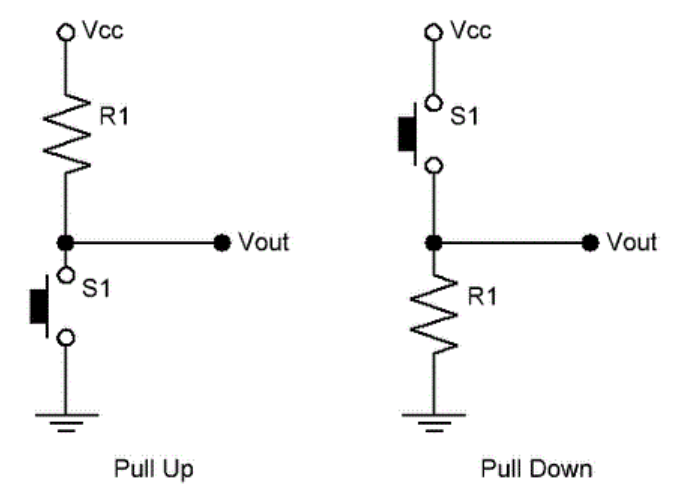
- Para ‘ler’ entradas dixitais, temos que declarar o pin de entrada como lectura, empregando:

```
pinMode(pinPuls, INPUT);
```

- e para obter o valor con:

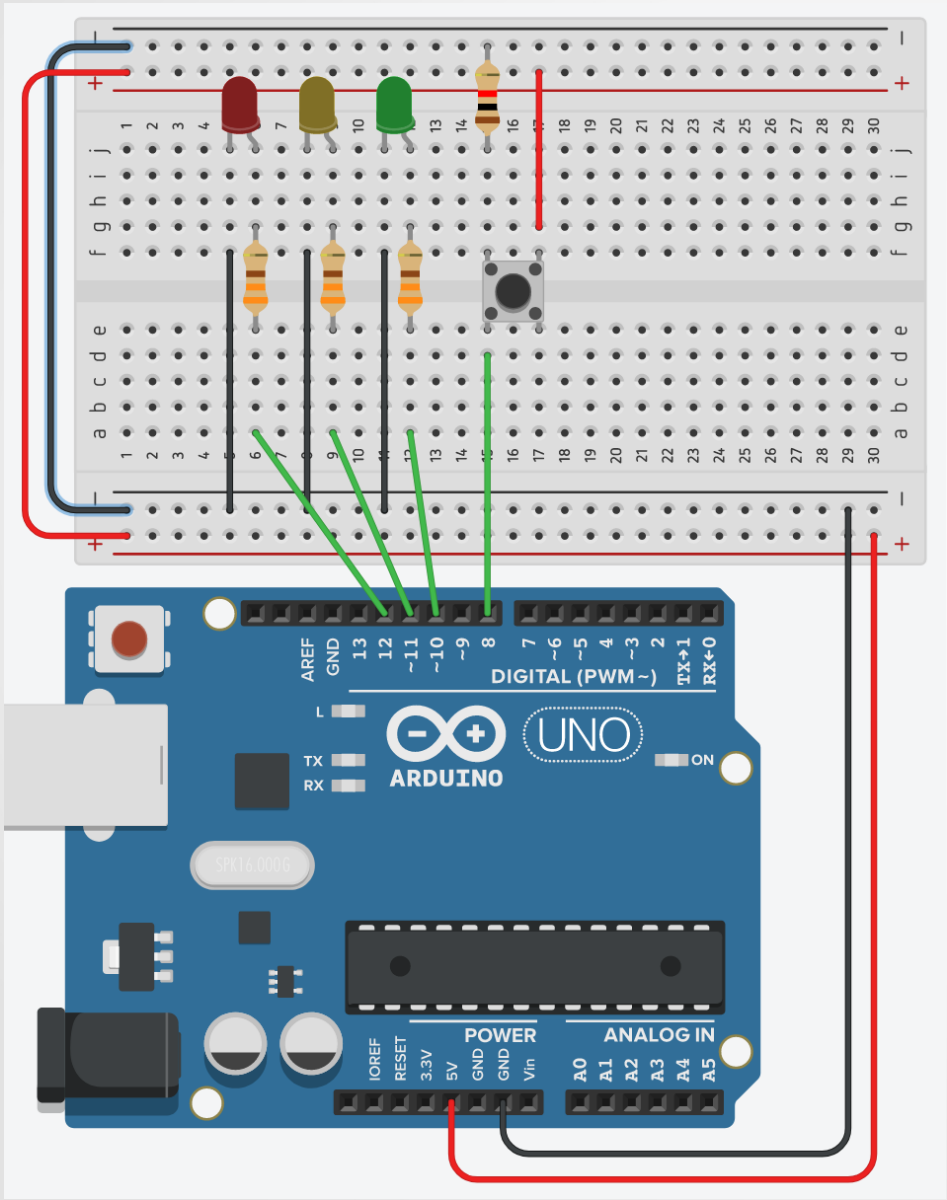
```
digitalRead(pinPuls);
```

- Imos conectar un pulsador como pull down, empregando unha resistencia de 10 KΩ.
- Cada pulsación encenderá ou apagará os LEDs de xeito que codifiquen os números 0 a 7 en binario:



Víctor . Álvarez 201911

Entradas dixitais – Detección de pulsador



- Hai algunha forma de que o texto do programa non se faga enorme?
- Unha vez que consigas un programa funcional, intenta optimizalo para que faga o mesmo empregando menos liñas e a menor cantidade de recursos.
- Aproveita o uso das estruturas condicionais

Víctor . Álvarez 201911

Entradas dixitais – Detección de pulsador

contador.tres.bits.raw | Arduino 1.8.10

contador.tres.bits.raw

```
/*
 * Conta pulsacións do pusador
 * e mostra o resultado en
 * binario de tres bits.
 *
 * 0 -- 000 -- 000    4 -- 100 -- v00
 * 1 -- 001 -- 00v    5 -- 101 -- v0v
 * 2 -- 010 -- 0a0    6 -- 110 -- va0
 * 3 -- 011 -- 0av    7 -- 111 -- vav
 *
 */

#define VERDE 10
#define AMARELO 11
#define VERMELLO 12
#define PULSADOR 8

int tempo = 200;
int conta = 0;
int pulsacion = false;

void setup() {
  pinMode(VERDE, OUTPUT);
  pinMode(AMARELO, OUTPUT);
  pinMode(VERMELLO, OUTPUT);
  pinMode(PULSADOR, INPUT);
}

void loop() {
  pulsacion = digitalRead(PULSADOR);
  if(pulsacion) {
    conta++;
```

Compilação Terminada

```
O rascunho usa 1218 bytes (3%) do espaço de armazenamen
Variáveis globais usam 11 bytes (0%) de memória dinâmic
```

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

sketch_nov10a | Arduino 1.8.10

sketch_nov10a §

```
void loop() {
  pulsacion = digitalRead(PULSADOR);
  if(pulsacion) {
    conta++;
    if(conta > 7) {
      conta = 0;
    }
    if(conta == 0) {
      digitalWrite(VERDE, LOW);
      digitalWrite(AMARELO, LOW);
      digitalWrite(VERMELLO, LOW);
    }
    else if(conta == 1) {
      digitalWrite(VERDE, HIGH);
      digitalWrite(AMARELO, LOW);
      digitalWrite(VERMELLO, LOW);
    }
    else if(conta == 2) {
      digitalWrite(VERDE, LOW);
      digitalWrite(AMARELO, HIGH);
      digitalWrite(VERMELLO, LOW);
    }
    else if(conta == 3) {
      digitalWrite(VERDE, HIGH);
      digitalWrite(AMARELO, HIGH);
      digitalWrite(VERMELLO, LOW);
    }
    else if(conta == 4) {
      digitalWrite(VERDE, LOW);
      digitalWrite(AMARELO, LOW);
      digitalWrite(VERMELLO, HIGH);
```

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

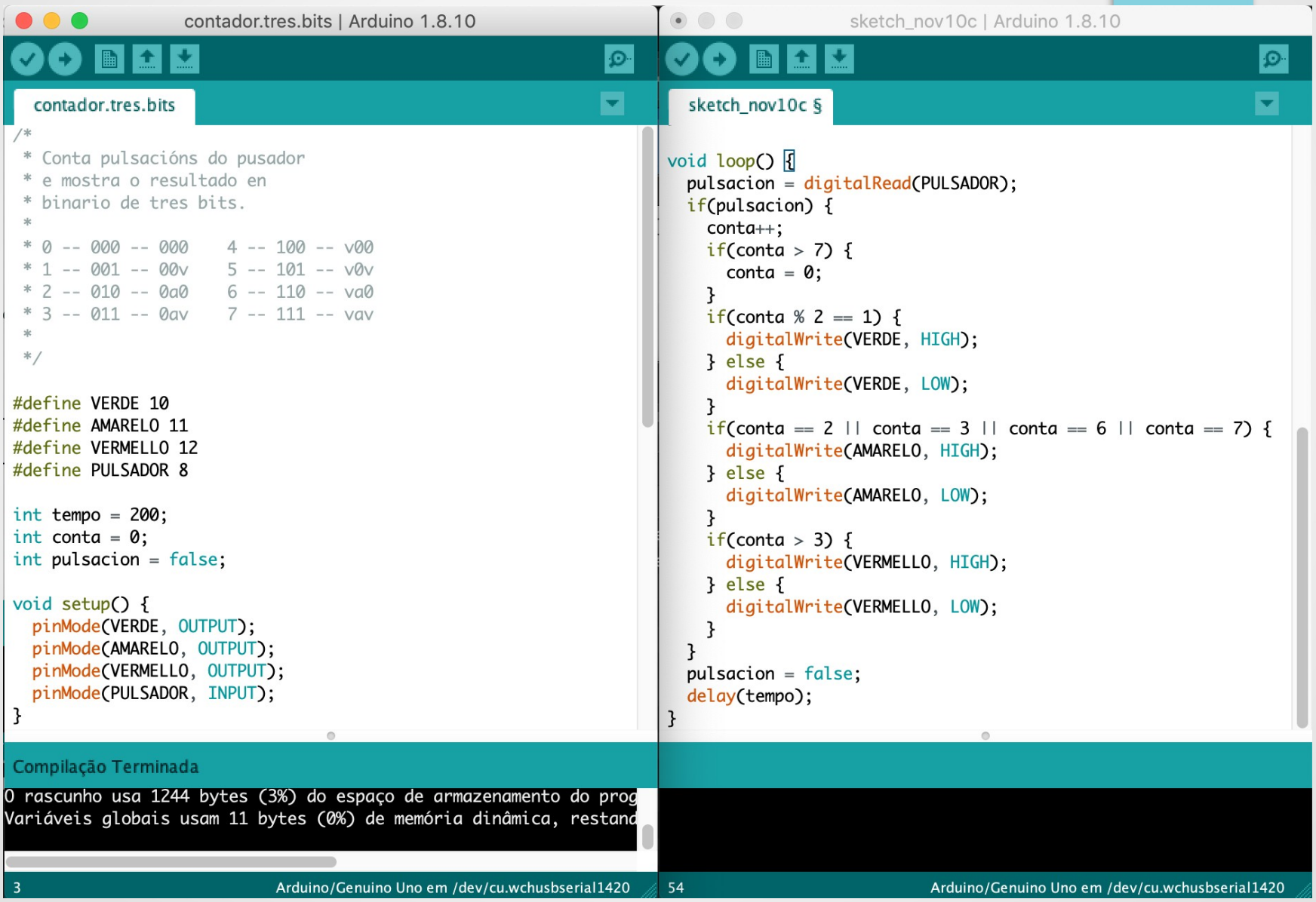
sketch_nov10b | Arduino 1.8.10

sketch_nov10b §

```
      digitalWrite(AMARELO, HIGH);
      digitalWrite(VERMELLO, LOW);
    }
    else if(conta == 3) {
      digitalWrite(VERDE, HIGH);
      digitalWrite(AMARELO, HIGH);
      digitalWrite(VERMELLO, LOW);
    }
    else if(conta == 4) {
      digitalWrite(VERDE, LOW);
      digitalWrite(AMARELO, LOW);
      digitalWrite(VERMELLO, HIGH);
    }
    else if(conta == 5) {
      digitalWrite(VERDE, HIGH);
      digitalWrite(AMARELO, LOW);
      digitalWrite(VERMELLO, HIGH);
    }
    else if(conta == 6) {
      digitalWrite(VERDE, LOW);
      digitalWrite(AMARELO, HIGH);
      digitalWrite(VERMELLO, HIGH);
    }
    else if(conta == 7) {
      digitalWrite(VERDE, HIGH);
      digitalWrite(AMARELO, HIGH);
      digitalWrite(VERMELLO, HIGH);
    }
  }
  pulsacion = false;
  delay(tempo);
}
```

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

Entradas dixitais – Detección de pulsador



contador.tres.bits | Arduino 1.8.10

contador.tres.bits

```
/*
 * Conta pulsacións do pusador
 * e mostra o resultado en
 * binario de tres bits.
 *
 * 0 -- 000 -- 000    4 -- 100 -- v00
 * 1 -- 001 -- 00v    5 -- 101 -- v0v
 * 2 -- 010 -- 0a0    6 -- 110 -- va0
 * 3 -- 011 -- 0av    7 -- 111 -- vav
 *
 */

#define VERDE 10
#define AMARELO 11
#define VERMELLO 12
#define PULSADOR 8

int tempo = 200;
int conta = 0;
int pulsacion = false;

void setup() {
  pinMode(VERDE, OUTPUT);
  pinMode(AMARELO, OUTPUT);
  pinMode(VERMELLO, OUTPUT);
  pinMode(PULSADOR, INPUT);
}
```

Compilação Terminada

```
O rascunho usa 1244 bytes (3%) do espaço de armazenamento do prog
Variáveis globais usam 11 bytes (0%) de memória dinâmica, restand
```

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

sketch_nov10c | Arduino 1.8.10

sketch_nov10c §

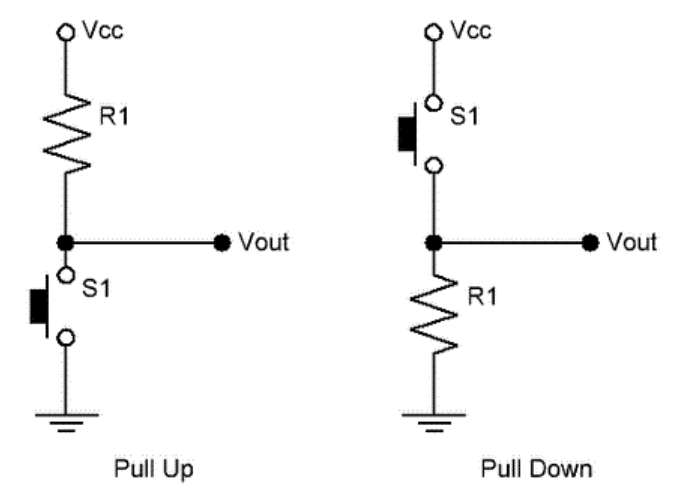
```
void loop() {
  pulsacion = digitalRead(PULSADOR);
  if(pulsacion) {
    conta++;
    if(conta > 7) {
      conta = 0;
    }
    if(conta % 2 == 1) {
      digitalWrite(VERDE, HIGH);
    } else {
      digitalWrite(VERDE, LOW);
    }
    if(conta == 2 || conta == 3 || conta == 6 || conta == 7) {
      digitalWrite(AMARELO, HIGH);
    } else {
      digitalWrite(AMARELO, LOW);
    }
    if(conta > 3) {
      digitalWrite(VERMELLO, HIGH);
    } else {
      digitalWrite(VERMELLO, LOW);
    }
  }
  pulsacion = false;
  delay(tempo);
}
```

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

Víctor . Álvarez 201911

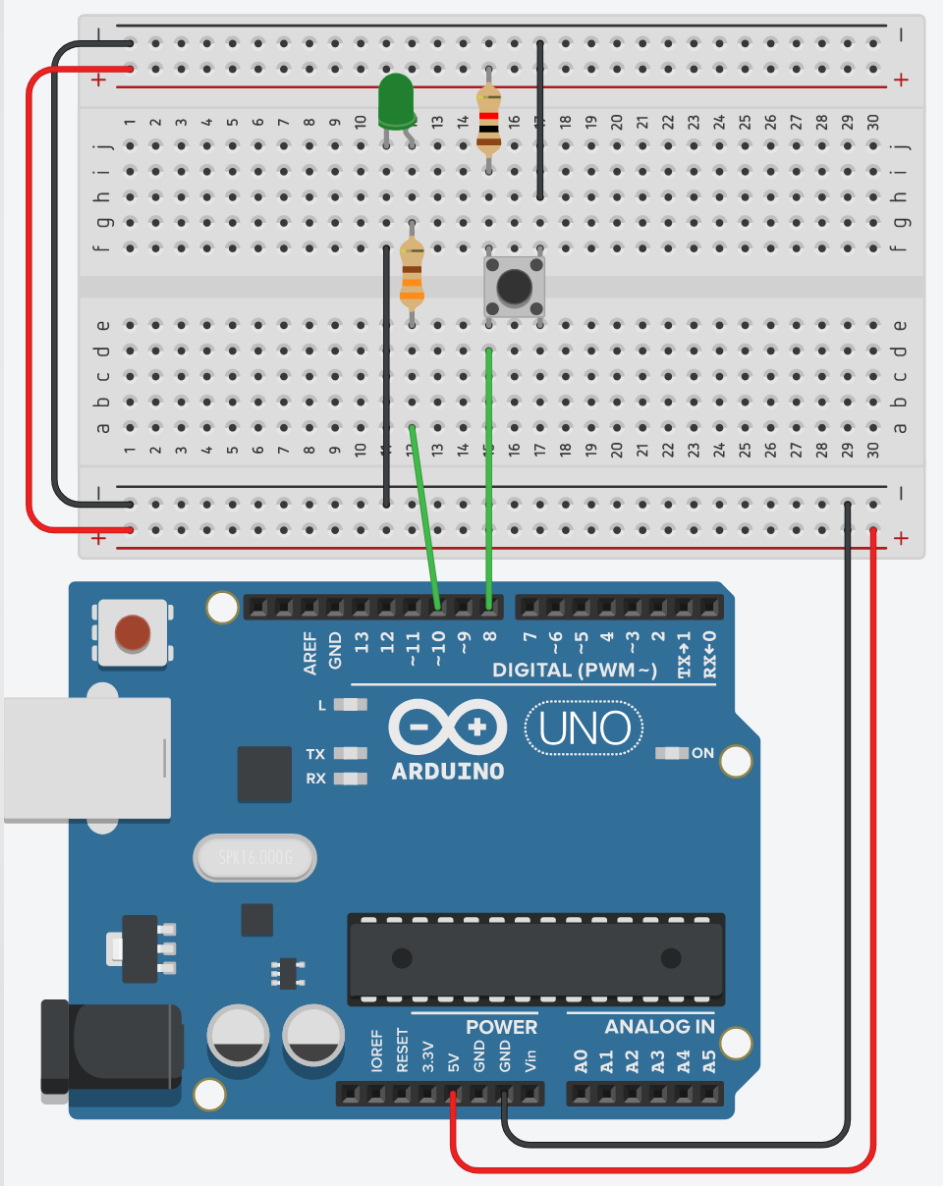
Entradas dixitais – Detección de pulsador

- Imos facer unha montaxe de xeito que cada vez que se prema un pulsador, o LED se apague.
- Para iso, imos reciclar un dos LEDs da montaxe anterior e conectar o pulsador empregando a mesma resistencia de 10 kΩ, esta vez como pull up.
- Ademais imos ter unha restricción: non podemos empregar ‘digitalWrite()’ en todo o programa máis que unha única vez.



Víctor . Álvarez 201911

Entradas dixitais – Detección de pulsador



antipulsador.LED | Arduino 1.8.10

antipulsador.LED

```
/*
 * Ao premer o pulsador
 * o LED encende.
 *
 * Só se pode empregar un
 * único digitalWrite()
 *
 */


#define VERDE 10
#define PULSADOR 8

int tempo = 100;
bool pulsacion = false;

void setup() {
  pinMode(VERDE, OUTPUT);
  pinMode(PULSADOR, INPUT);
}

void loop() {
  pulsacion = digitalRead(PULSADOR);
  digitalWrite(VERDE, !pulsacion);
  delay(tempo);
}
```

Compilação Terminada

O rascunho usa 1032 bytes (3%) do espaço de armazenamento do
Variáveis globais usam 9 bytes (0%) de memória dinâmica, res

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

Víctor . Álvarez 201911

Entradas dixitais

- Nesta unidade aprendemos a:
  - Declarar un pin como entrada dixital -pinMode()- e ler nel un valro HIGH ou LOW -digitalRead()-
  - prever situacións de alta impedancia, empregando resistencias de pull up ou pull down
  - declarar e usar valores de tipo bool
  - mellorar un sketch empregando estructuras predefinidas

Víctor . Álvarez 201911