

Grado en Ingeniería Informática

Dispositivos Hardware e Interfaces

DHI

Práctica 0. Kit de prácticas Arduino UNO

Profesores de la asignatura



Grupo de Tecnología Electrónica y
Comunicaciones



UNIVERSIDADE DA CORUÑA

UNIVERSIDADE DA CORUÑA

Objetivos

- Instalación y puesta en marcha del kit de Arduino UNO que se va a utilizar en las prácticas de la asignatura
- Familiarización con el entorno de desarrollo
- Pruebas

Introducción

- Características del Arduino UNO

<http://arduino.cc/en/Main/arduinoBoardUno>

- Descarga del IDE Arduino

<http://arduino.cc/en/Main/Software>

- Guía de inicio del Arduino

<http://arduino.cc/en/Guide/HomePage>

- Tutorial en e.g. Windows

<http://arduino.cc/en/Guide/Windows>

- Seguir las instrucciones paso a paso para instalar el entorno de desarrollo y probar algunos ejemplos.
- Descargar la versión actual del entorno (e.g. 1.8.11)

■ Leer y escribir pin digital

```
// Leer estado de pin y sacar a led
const int pinPin = 3; // Pin a entrada digital 3
const int ledPin= 13; // Led a salida digital 13

// setup inicializa entrada pin con pullup y salida led
void setup(){
    pinMode(ledPin, OUTPUT);
    pinMode(pinPin, INPUT_PULLUP);
}

// Cada bucle lee el estado del pin y saca a led
void loop() {
    digitalWrite(ledPin, digitalRead(pinPin));
}
```

- Dímer con entrada por potenciómetro y salida PWM a led

```
// Leer valor de pin y sacar a PWM
const int anaPin = 0; // Pin a entrada analógica 0
const int PWMPin= 10; // Salida PWM 10

void setup(){
}

// Cada bucle lee el valor de pin y saca a PWM
void loop() {
    analogWrite(PWMPin, analogRead(anaPin)/4);
}
```

■ Contador de pulsaciones de pulsador con entrada con resistencia de pulldown sin usar filtrado SW

```
// Cada pulsación incrementa una unidad un contador
// de pulsaciones y es enviado por el puerto serie.
// Utiliza polling y no filtra rebotes.
// José J. Lamas. UDC. 2019.
```

```
const int pulsador = 2;
int counter = 0;
```

```
void setup()
{
  Serial.begin(115200);
  pinMode(pulsador, INPUT);
  Serial.println("Start");
}
```

```
void loop() {
  leePulsador();
  //delay(50);
}
```

```
void leePulsador() {
  static boolean puls_enable= false;
  if (digitalRead(pulsador)) {
    if (puls_enable) {
      puls_enable= false;
      Serial.print ("Count: ");
      Serial.println(++counter);
    }
  }
  else puls_enable= true;
}
```

■ Contador de pulsaciones de pulsador con entrada con resistencia de pulldown y filtrado SW con millis()

```
// Cada pulsación incrementa una unidad un contador
// de pulsaciones y es enviado por el puerto serie.
// Utiliza polling y filtrado de
// rebotes (debouncing) usando millis().
const int pulsador = 2;
const int debounce_time_ms = 50;
int counter = 0;
void setup() {
  Serial.begin(115200);
  pinMode(pulsador, INPUT);
  Serial.println("Start");}
void loop() {
  leePulsador();
  delay(debounce_time_ms + random(-10,10));}
```

```
void leePulsador() {
  static unsigned long starttime = millis();
  static boolean puls_enable= false;
  if (digitalRead(pulsador)) {
    if (puls_enable) {
      puls_enable= false;
      if (millis() - starttime > debounce_time_ms) {
        Serial.print ("Count: ");
        Serial.println(++counter);
      }
      starttime = millis();}
  }
  else puls_enable= true;
}
```

■ Escribir número aleatorio a puerto serie

```
// Número aleatorio a puerto serie USART

// Inicializa puerto serie y semilla random
void setup(){
    Serial.begin(9600);
    randomSeed(analogRead(0));
}
// Cada bucle genera valor aleatorio de 0 a 9
// y lo envía por el puerto serie.
void loop(){
    Serial.println(random(10));
    delay(100);
}
```