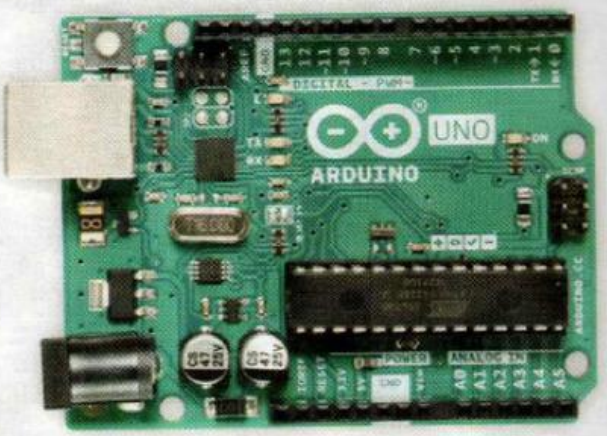
El futuro digital es de todos MinTIC

La educación es de todos Mineducación

¿Cómo funciona?

Programación en Arduino UNO

La programación de Arduino es la programación de un microcontrolador.
Programar Arduino consiste en traducir a líneas de código las tareas automatizadas que queremos hacer leyendo de los sensores y, en función de las condiciones del entorno, programar la interacción con el mundo exterior mediante unos actuadores.



Estructura de Sketch

Un programa de Arduino se denomina sketch o proyecto y tiene la extensión .ino.

Importante: para que funcione, el sketch el nombre del fichero debe estar en un directorio con el mismo nombre que el sketch.

La estructura básica de un sketch de Arduino es bastante simple y se compone de, al menos dos partes. Estas dos partes son obligatorias y encierran bloques que contienen declaraciones, estamentos o instrucciones.

Bli

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Función Setup o función preparar ()

La función setup() se llama cuando comienza un boceto. Es usada para inicializar variables, modos de pin, comenzar a usar bibliotecas, etc. La función setup() sólo se ejecutará una vez después de cada encendido o reinicio de la placa Arduino.

```
int buttonPin = 3;
  // put your setup code here, to run once:

void setup() {
  Serial.begin (9600); // inicio del monitor serial.
  pinMode (buttonPin, INPUT); // declarar un puerto como entrada.
}

void loop() {
  // ...
}
```

Función loop o función lazo ()

Después de crear una función setup(), que inicializa y establece los valores iniciales, la función loop() hace precisamente lo que sugiere su nombre y se repite consecutivamente, lo que permite que su programa cambie y responda. Es usada para controlar activamente la placa Arduino.

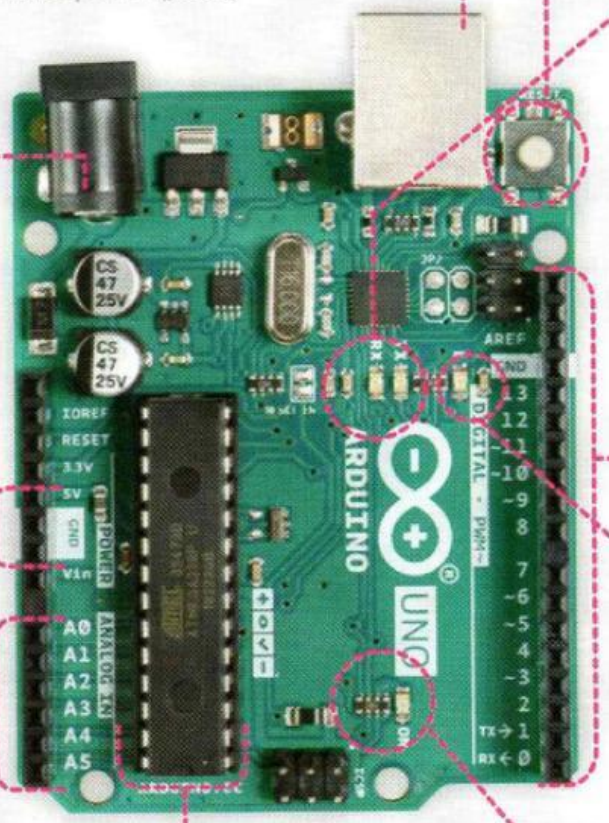
```
int buttonPin = 3;

  // setup initializes serial and the button pin
void setup() {
  Serial.begin (9600);
  pinMode (buttonPin, INPUT);
}

// bucle comprueba el pin del botón cada vez,
///y enviará serial si se presiona
void loop () {
  if (digitalRead (buttonPin) == HIGH) {
    Serial.write('H');
  }
  else {
    Serial.write('L');

  }
  delay(1000);
  }
```

El futuro digital es de todos MinTIC

La educación es de todos Mineducación

Este conector se utiliza para alimentar la placa Arduino cuando no está conectada a un puerto USB. Acepta tensiones entre 7 y 12V.

Usado para alimentar y cargar los programas a su Arduino, y para la comunicación con el programa de Arduino (mediante la instrucción Serial.println() etc.)



Puesta a cero del micro-controlador ATmega.

Estos diodos LEDs indican cuando se realiza una comunicación entre Arduino y el ordenador. Parpadean rápidamente cuando se carga el programa así como durante la comunicación serie. Útil para la depuración.

Usar estos pins con las instrucciones digitalRead(), digitalWrite(), y analogWrite().
analogWrite() solo trabaja con los pins con el símbolo PWM.

El único componente que actúa como dispositivo de salida incorporado a su Arduino Uno. Lo usará cuando ejecute su primer programa. Este LED es muy útil para la depuración.

El corazón de la placa Arduino Uno.

Usar estos pins para proporcionar una tensión de +5V y masa para los circuitos externos a la placa.

Indica que la placa Arduino está siendo alimentada. Útil para la depuración.

Usar estos pins con la instrucción analogRead()

· Led de Encendido

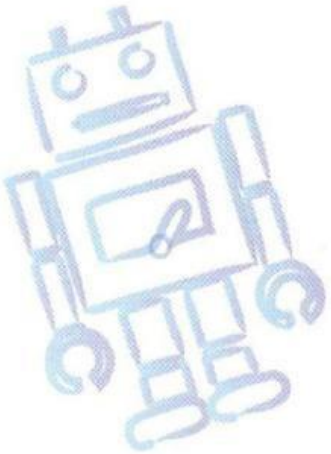
Fue (2016)

Puerto USB · Botón de reset Pin 13 LED · Microcontrolador ATmega LEDs TX y RX

· Pins Digitales Entradas Analógicas Conector de alimentación Pines GND y 5V

¿Para qué sirve la función RANDOM?

La función randomSeed(seed) reinicia el generador de números pseudoaleatorios de Arduino. Aunque la distribución de los números devueltos por random () es esencialmente aleatoria, la secuencia es predecible. Se debería reajustar el generador a algún valor aleatorio.



```
//Variable donde almacenaremos el numero aleatorio
long randomNumber;

//Función de inicialización
void setup() {
  //Inicializamos la comunicación serial
  Serial.begin(9600);

  //Escribimos por el puerto serie mensaje de inicio
  Serial.println("Inicio de sketch - secuencia de numeros aleatorios

  //Establecemos la semilla en un pin analogico
  randomSeed (analogRead (A0));
}

//Bucle principal
void loop() {

  //Genera un número aleatorio entre 1 y 100
  randomNumber = random(1,100);

  //Escribimos el numero aleatorio por el puerto serie
  Serial.print("El numero aleatorio es = ");
  Serial.print(randomNumber);
```

Elaboración propia