

Arduino - Generaliades

Autor:

Juan David Argüello Plata - Ingeniero Mecánico

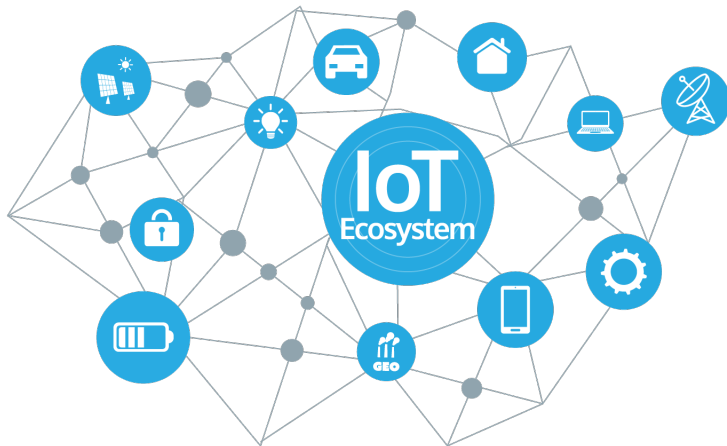
Profesor tutor:

Jairo René Martínez Morales - Químico PhD

CENIVAM

Universidad Industrial de Santander

Introducción



Objetivos de hoy

- Conocer la lógica básica del lenguaje.
- Tips de programación en Arduino.
- Prueba de conexiones electrónicas.

Instalación de Arduino I

Accede a la página de Arduino, en la sección Software - Descargas (o has click [aquí](#)).

[HOME](#) [STORE](#) [SOFTWARE](#) [EDU](#) [RESOURCES](#) [COMMUNITY](#) [HELP](#)

Download the Arduino IDE




ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

[Windows](#) Installer, for Windows XP and up
[Windows](#) ZIP file for non admin install

Windows app Requires Win 8.1 or 10


Mac OS X 10.8 Mountain Lion or newer

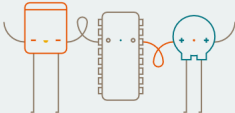
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Instalación de Arduino II

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **34,582,765** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 **\$5** **\$10** **\$25** **\$50** **OTHER**

[JUST DOWNLOAD](#)

CONTRIBUTE & DOWNLOAD

Software - Layout I

Arduino usa `C++` como lenguaje de programación. A diferencia de Python, `C++` es un lenguaje **estático**: debes declarar la naturaleza de las variables y tener en cuenta el tamaño que ocuparán éstas en la memoria RAM.

Tipo variable (TV)	Memoria usada [<i>bits</i>]	Aplicación
void	-	Declaración de funciones
byte	8	Número entero entre 0 y 255.
int	16	Número entero entre -32767 y 32767.
long	32	Número entero entre -2,147,483,648 y 2,147,483,648.
float	32	Número real entre -3.4028325E+38 y 3.4028325E+38.
boolean	8	Variable booleana: true/false.
char	8	Caracteres ASCII.
String	-	Cadena de texto.

Software - Layout II

Un *Sketch* común de Arduino se compone de:

- *void* setup: acciones que se desarrollan **una** sola vez.
- *void* loop: acciones que se repiten una y otra vez (similar al ciclo while).

Los comentarios dentro del algoritmo se hacen de la siguiente manera:

- `//` Comentario: Comenta una sola línea de código.
- `/*` Comentario `*/`: Comenta una o varias líneas.

IMPORTANTE: La sintaxis de *C++* demanda que el final de cualquier línea de código sea con `“;”`.

Condicionales

Las condicionales en Arduino presentan la siguiente sintaxis:

```
if (condicion1) {  
    ...  
} else if (condicion2) {  
    ...  
} else {  
    ...  
}
```


for - while

Las funciones cíclicas se usan de la siguiente manera:

```
for (inicio; fin; incremento) {  
    ...  
}
```

```
while (condicion) {  
    ...  
}
```

Switch

Es una función propia de C++ que cumple un propósito similar al de if - else.

```
switch (var) {  
    case label1:  
        ...  
        break;  
    case label2:  
        ...  
        break;  
    default:  
        ...  
        break;  
}
```

Funciones

Una *función* es un segmento de código que desarrolla una tarea específica. Las funciones se escriben comúnmente al principio o al final del *Sketch*, o *Script*, y se **llaman** dentro de otras funciones.

Las ventajas de usar funciones son:

- Ayudan a mantener la **organización** del código (conceptualiza el algoritmo escrito).
- Permiten probar “tareas” **una sola vez**.
- Disminuye probabilidades de errores de programación.
- Permite reutilizar código.

```
TV nombre_funcion (TV var1 , TV var2 , ...) {  
    ...  
    return resultados;           // Opcional ...  
}
```

Prueba de conexiones

Existen tres maneras de probar conexiones eléctricas - electrónicas:

- 1 Pruebas de **continuidad**.
- 2 A ciegas... funciona o no funciona.
- 3 Si no funciona: pruebas de diferencia de potencial - **voltaje** (Arduino maneja **5V**).

