

# Programación para Sistemas Embebidos

## 3. Tipos de datos, variables y constantes

### 3.1. Práctico 03 - Tipos de datos y Variables

#### ¿Qué es una variable?

En los términos más simples, una variable es un lugar para almacenar un valor específico y darle un nombre útil. Pensemos en una variable como un contenedor que almacena un valor.

Las variables pueden contener diferentes tipos de valores. Por ahora, veremos las variables que contienen un valor lógico, verdadero o falso, High o Low.

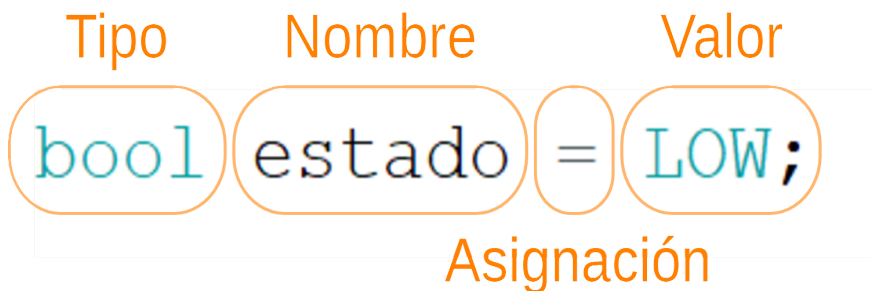
En la siguiente línea de código, declaramos y asignamos una variable.

```
bool estado = LOW;
```

Declarar una variable significa darle un nombre y asignarle un valor.

Podemos declarar variables sin darles un valor, pero no podemos dar un valor a algo que aún no ha declarado.

En el lenguaje para Arduino vamos a tener hasta 4 partes cuando hagamos una declaración de variables:



1. El Tipo de valor que contendrá la variable, en este caso un valor lógico que llamamos bool.
2. El Nombre que le asignamos al contenedor; como queremos almacenar el estado de un interruptor, le damos un nombre significativo, por ejemplo estado.
3. Con el signo = se realiza la asignación de un valor, esto es opcional al momento de declarar la variable, podríamos asignarlo luego.
4. El Valor propiamente dicho. Debe ser coherente con el tipo de dato declarado.

El valor lógico que puede almacenar una variable de tipo "bool" puede tener dos estados posibles:

**HIGH o LOW**

Esta dualidad puede ser pensada de distintas maneras según el contexto en el que estemos:

**HIGH = True = 1 = 5 V = Encendido**

**LOW = False = 0 = 0 V = Apagado**

De esta manera, lo que almacenamos en la variable se presenta de distintas formas en nuestro proyecto de Arduino.

#### La sentencia IF

Podemos usar el contenido de una variable tipo bool, para decidir si ejecutamos una sección de código o no, esto es posible gracias a la sentencia IF.

Veamos el siguiente código

```

8 void loop() {
9   estado = digitalRead(3);
10
11   if(estado == HIGH) {
12     digitalWrite(5, HIGH);
13     delay(1000);
14     digitalWrite(5, LOW);
15   }
16 }

```

dentro de la función loop leemos el estado del pin 3, (recordemos el ejercicio anterior, donde teníamos conectado un pulsador en este pin).

Luego evaluamos SI estado es igual a HIGH, entonces ejecutamos lo que está dentro de las llaves. En este caso, encendemos la salida 5, esperamos un segundo y luego apagamos la salida 5.

*Notemos que usamos dos signos igual seguidos para comparar, si usáramos solo uno, estaríamos asignando el valor HIGH a la variable estado.*

**¡Con esta poderosa instrucción podemos ejecutar de manera selectiva trozos de código de acuerdo al estado de las entradas de Arduino!**

## Sentencia ELSE

Podemos utilizar una instrucción complementaria a IF que es ELSE, se ejecuta cuando la condición que evalúa IF no se cumple.

En el caso del ejemplo anterior se ejecuta cuando estado es LOW.

Podríamos hacer un programa como el siguiente:

```

8 void loop() {
9   estado = digitalRead(3);
10
11   if(estado == HIGH) {
12     digitalWrite(5, HIGH);
13     delay(1000);
14     digitalWrite(5, LOW);
15   }
16   else{
17     digitalWrite(5, HIGH);
18     delay(100);
19     digitalWrite(5, LOW);
20     delay(900);
21   }
22 }

```

La sentencia if evalúa si lo que está dentro de paréntesis es True o False. Por esto, podríamos quitar la evaluación estado == HIGH y sólo incluir el nombre de la variable estado, lo que devuelve el contenido de la misma. Notemos que esto sólo se puede hacer si el tipo de la variable es bool.

De este modo, dependiendo del valor leído en el pin 3, ejecutamos una acción u otra.

## Ejercicios

1. Visitemos la referencia de Arduino e investiguemos que otros tipos de datos existen <https://www.arduino.cc/reference/en/#variables>. ¿Para que pueden ser útiles las otras formas de almacenar información en la memoria de Arduino?
2. Usar el pulsador para encender un LED, cuando se suelta el pulsador el LED se apaga.
3. Usar un pulsador para simular el funcionamiento de un interruptor de luz, cuando presiono el interruptor la primera vez, el led se enciende y permanece encendido hasta que el pulsador sea presionado de nuevo.
4. Escribir un programa que cuando sea pulsado el interruptor, el led parpadee 5 veces con una cadencia de 1 seg.