

[Arduino] Referencia: Declaración y conversión de tipos de variables

Tipos de Datos	Memoria que ocupa	Rango de valores
boolean	1 byte	0 o 1 (True o False)
byte / unsigned char	1 byte	0 – 255
char	1 byte	-128 – 127
int	2 bytes	-32.768 – 32.767
word / unsigned int	2 bytes	0 – 65.535
long	2 bytes	-2.147.483.648 – 2.147.483.647
unsigned long	4 bytes	0 – 4.294.967.295
float / double	4 bytes	-3,4028235E+38 - 3,4028235E+38
string	1 byte + x	Array de caracteres
array	1 byte + x	Colección de variables

```
//tipo vacio (solo para funciones)
void
```

Booleanos

```
//booleano, false o true
boolean variable = false;
```

Enteros

```
//entero, 16 bits, de -32,768 a 32,767
int variable = 100;

//entero, 16 bits, de 0 a 65535 (excepto en Due, donde son 32 bits)
unsigned int variable = 100;

//entero, 16 bits, de 0 a 65535
short variable = 100;

//entero, 32 bits, de -2,147,483,648 a 2,147,483,647
long variable = 100000L;

//entero, 32bits, de 0 a 4,294,967,295
unsigned long variable = 100000L;
```

Coma flotante

```
//coma flotante, 32 bits, de -3.4028235E+38 a 3.4028235E+38. Precision 6 digitos
float variable = 1.117;

//idéntico a float, excepto en Arduino Due donde es flotante de 64 bits
double variable = 1.117;
```

Bytes

```
//8 bits, de 0 a 255
byte variable = B10010;

//16bits, sin signo, de 0 a 65535
word variable = 10000;
```

Caracteres

```
//8 bits, de -128 a 127
char variable = 'A';

//8 bits, de 0 a 255
unsigned char variable = 240;
```

Conversión entre variables

```
//convierte a char
char(variable);

//convierte a byte
byte(variable);

//convierte a int
int(variable);

//convierte a word
word(variable);

//convierte a long
long(variable);

//convierte a float
float(variable);
```

Cualificadores de variables

```
//STATIC
//Variables visibles únicamente en el interior de una función,
//y cuyo valor se mantiene entre llamadas a la misma.
static int variable;

//CONST
//Variables cuyo valor no puede ser redefinido tras la inicialización
const float pi = 3.14;

//VOLATILE
//Variables en las que se indica al compilador que no guarde en los registros
//del microprocesador, sino que fuerza la actualización en memoria. Esto se
//hace cuando existe la posibilidad de que el valor de la variable sea
//modificado por otro proceso que se ejecuta concurrentemente con el actual
//(por ejemplo cuando se emplean hilos o interrupciones)
```

```
volatile int variable = LOW;
```

From:

<https://euloxio.myds.me/dokuwiki/> - **Euloxio wiki**

Permanent link:

https://euloxio.myds.me/dokuwiki/doku.php/doc:tec:elo:uc_arduino:arduino_c:variables

Last update: **2025/03/18 10:24**

