

# Curso avanzado sobre Arduino

elCacharreo



ElCacharreo.com



# Arduino avanzado: Presente



# Arduino avanzado: Presente



José Antonio Vacas Martínez

blog  
javacasm@elcacharreo.com  
twitter  
linkedin



# Arduino avanzado: Programando

¿Qué sabemos hasta ahora?

pinMode

digitalWrite

digitalRead

analogRead

analogWrite

delay



# Programando Arduino: Comentarios

Es muy importante comentar nuestro código

```
/* .....  
*/
```

ó

```
// .....
```



# Programando Arduino: Variables I

Una variables es una forma de etiquetar y guardar un valor que puede ser de diferentes tipos. Podemos modificar este valor.

```
int a=10;  
float c=10.2;  
int b; ← Declaración  
b=15; ← Asignación
```



# Programando Arduino: Variables II

El alcance (ámbito, scope) de la variable nos dicta donde ésta existe.

```
int a=10;
```

```
void funcion prueba()  
{  
    float b=10.2;  
}
```



# Programando Arduino: Tipos

byte: 8bits      0 a 255

int: 16bits      -32768 a 32767

long: 32bits    -2x10<sup>6</sup> a 2x10<sup>6</sup>

float: guarda decimales -3.4x10<sup>34</sup> a 3.4x10<sup>34</sup>

!!!Cuidado con los números negativos!!!





# Programando Arduino: Aritmética

Operadores aritméticos: +, -, \*, /, % (módulo)

```
a=a+3;
```

```
b=a/3;
```

```
c=b%3;
```



# Programando Arduino: Arrays

Un array es una agrupación de variables con un tipo y un tamaño determinados

```
int miArray[5];  
int miOtroArray[]={1,23,2}
```

Accedemos al elemento i con array[i]



# Programando Arduino: Constantes

Son variables a las que no podemos modificar al valor:

true/false

HIGH/LOW

INPUT/OUTPUT



# Programando Arduino: Funciones

Una función es un conjunto de instrucciones agrupadas para un nombre, al que le pasaremos unos argumentos y devolverá un valor

```
tipo funcion(tipo argumento 1, tipo argumento 2)
{
}
```



# Arduino Intermedio: Control

if else

switch

while

do while

goto

break

continue



# Programando Arduino: Estructuras de control I: Condicionales

```
if (pinFiveInput < 500)
{
  // action A
}
else
{
  // action B
}
```

```
if ((pinFiveInput < 500) && (pinFiveInput > 100))
{
  // action A
}
else
{
  // action B
}
```



# Programando Arduino: Estructuras de control I: Condicionales

```
switch (var) {  
  case 1:  
    //hacemos algo si var es 1  
    break;  
  case 2:  
    //hacemos algo si var es 2  
    break;  
  default:  
    // si nada concuerda, default  
    // default es opcional  
}
```



# Programando Arduino: Estructuras de control II: Bucles

```
for (initialization; condition; increment) {  
  //sentencia(s);  
}
```

```
for (int i=0;i<20;i=i+1) {  
  //sentencia(s);  
}
```

```
while(expresion){  
  // sentencia(s);  
}
```

```
while(i<20){  
  // sentencia(s);  
  i=i+1;  
}
```

```
do  
{  
  //sentencia(s)  
} while (test condicion);
```

```
do  
{  
  //sentencia(s)  
  i=i+1;  
} while (i<20);
```





# Programando Arduino: Estructuras de control III: Saltos

- break: sale del bucle
- continue: salta el paso actual del bucle
- return: sale de la función
- goto: salta a la etiqueta

```
for(int i=0;i<10;i++)  
{  
    if (bsalto>0)  
        continue;  
    else  
        break;  
}
```



# Programando Arduino: Primer programa

```
void setup()  
{}
```

```
void loop()  
{}
```



# Arduino avanzado: Programando

## Conversiones ¿por qué?

- char()
- byte()
- int()
- word()
- long()
- float()

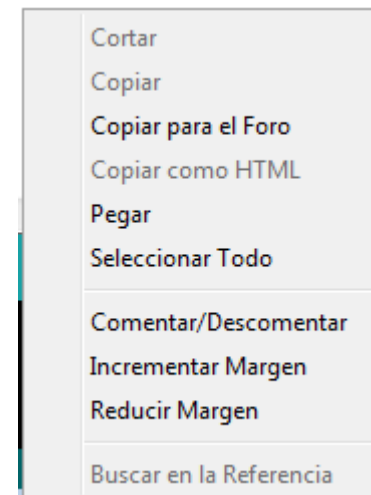
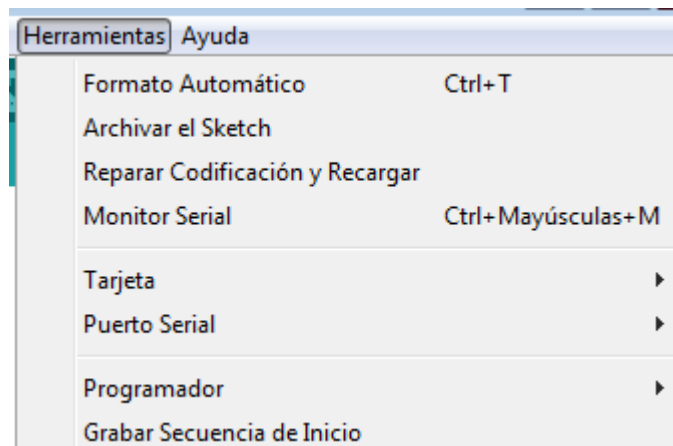
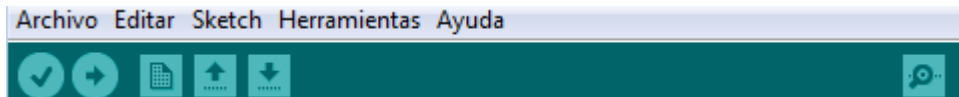
char(), byte(), int(),  
word(), long(), float()

float a=2.4;  
int b=int(a);



# Arduino avanzado: Entorno

## Menús



# Arduino avanzado: Entorno

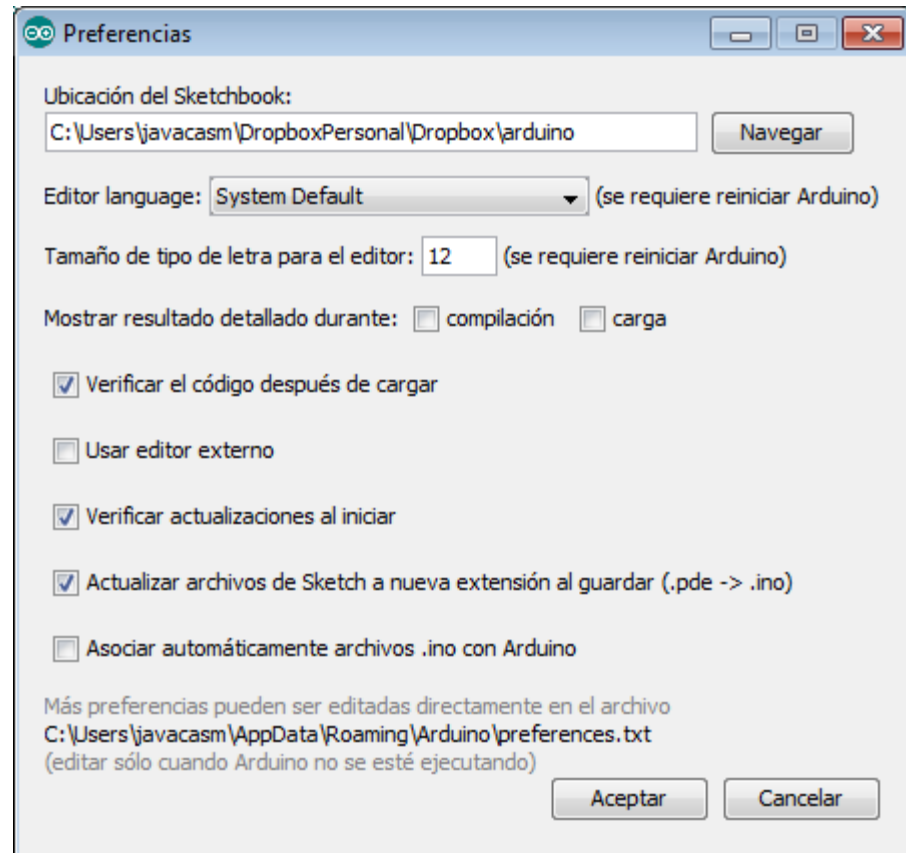
## Estructura interna

Nombre	Tamaño
tools	8 609
reference	1 992 079
libraries	1 083 990
lib	3 431 006
java	112 692 925
hardware	132 047 193
examples	209 604
drivers	3 387 650
rtxSerial.dll	77 759
revisions.txt	37 256
libusb0.dll	43 520
cygwin1.dll	1 872 821
cygiconv-2.dll	969 728
arduino.exe	860 160



# Arduino avanzado: Entorno

## Preferencias



Truco: ¿cómo evitar problemas con código? : DrobBox, Git



# Conclusiones

Gracias por vuestra atención

