

# Curso avanzado sobre Arduino: C y C++ avanzado

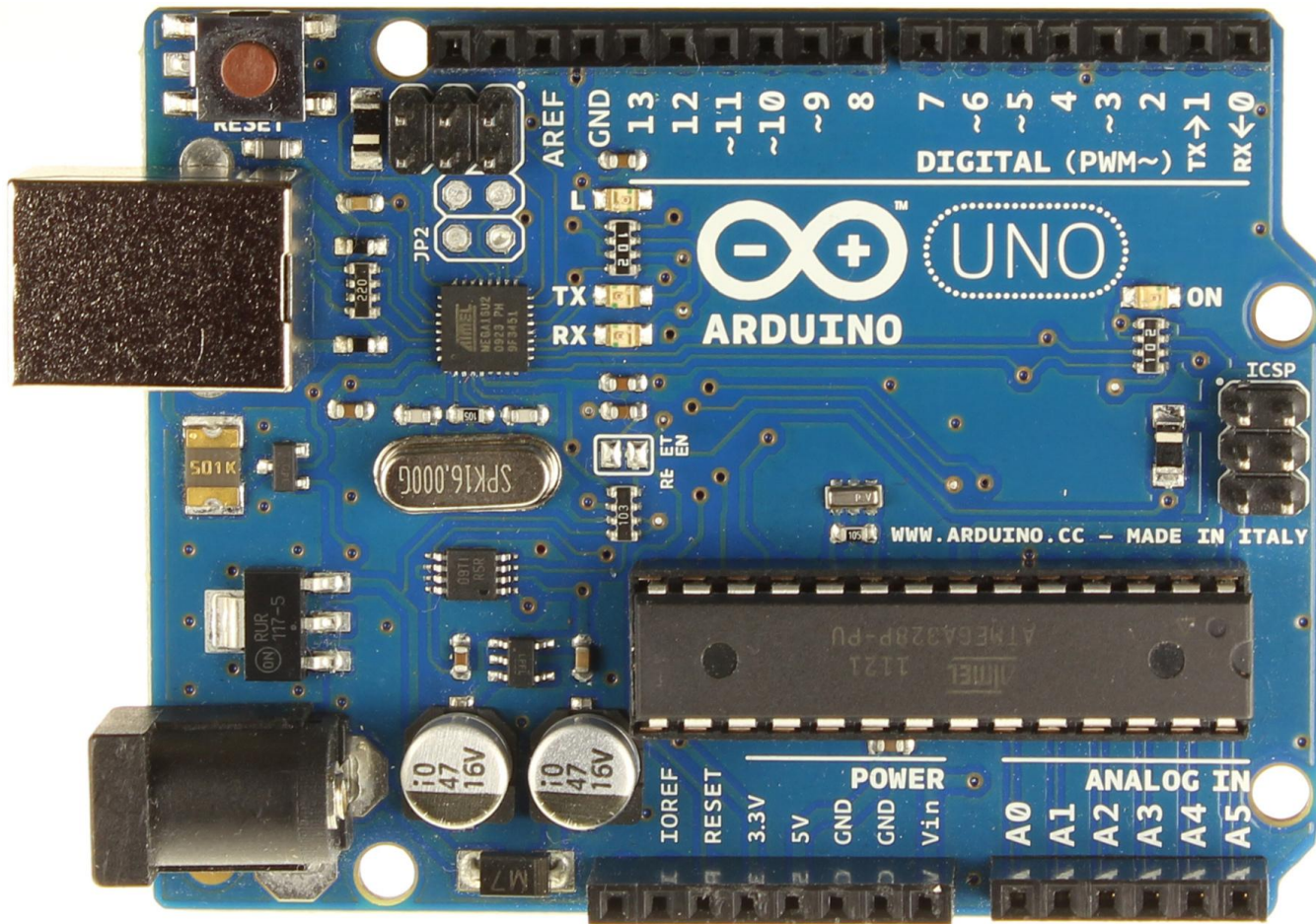
[ElCacharreo.com](http://ElCacharreo.com)



[ElCacharreo.com](http://ElCacharreo.com)



# Programando Arduino: Presente



# Programando Arduino:: Presente



José Antonio Vacas Martínez

blog  
javacasm@elcacharreo.com  
twitter  
linkedin



# Programando Arduino: 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: Operadores

- Operadores Booleanos: &&, ||, !
- Operadores de acceso a punteros: \*, &
- Operadores de bits: &, |, ^, ~, <<, >>
- Operadores compuestos:
  - Incremento/decremento de variables: ++, --
  - Asignación y operación: +=, -=, \*=, /=, &=, |=

<http://arduino.cc/en/Reference/HomePage>



# Arduino Avanzado: Operadores

## Operadores

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

## Bits and Bytes

- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()



# Programando Arduino: Conversiones de tipos

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

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



# Programando Arduino: Matemáticas

## Math

- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

## Trigonometría

- sin()
- cos()
- tan()

## Números aleatorios

- randomSeed()
- random()



# Programando Arduino: Objetos

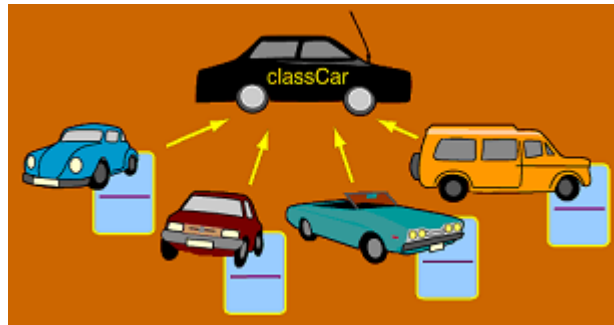
- Un objeto encapsula
- Un objeto simplifica
  
- Ejemplo: Serial
  - `Serial.println()`
  - `Serial.write( )`
- Ejemplo: Servo



# Programando Arduino: Objetos

Diferencia entre clase y Objeto

Ejemplo coches



# Programando Arduino: String

Además del array de caracteres existe el objeto **String**

- String()
- charAt()
- compareTo()
- concat()
- endsWith()
- equals()
- equalsIgnoreCase()
- getBytes()
- indexOf()
- lastIndexOf()
- length()
- replace()
- setCharAt()
- startsWith()
- substring()
- toCharArray()
- toLowerCase()
- toUpperCase()
- trim()



# Arduino Avanzado: Preprocesador

#include <ficheroSistema.h>

#include "ficheroPropio.h"

#define etiqueta valor

**Preprocesador** #ifndef etiqueta

// codigo si definida

#else

// codigo si no definida

#endif





# Programando Arduino: Internamente...

```
#include <Arduino.h>
void main()
{
    setup();

    while(1)
    {
        loop();
    }
}
```



# Arduino Avanzado: memoria

char \*

malloc

calloc

pun(ñe)teros

Casting

\* dereference operator

& reference operator



# Arduino Avanzado: enum

enum

```
{ monday, tuesday, wednesday, thursday,  
friday, saturday, sunday  
} day;
```

<http://www.yolinux.com/TUTORIALS/C++Enum.html>

<http://playground.arduino.cc/Code/Enum>



# Arduino Avanzado: estructuras

```
struct data
```

```
{
```

```
char id;
```

```
unsigned long time;
```

```
int an0;
```

```
int an1;
```

```
int dig0;
```

```
int dig1;
```

```
};
```

```
data datos;
```

```
datos.time=millis();
```

```
datos.id=10;
```

```
datos.an0=analogRead(A0);
```

```
.....
```

```
datos.dig1=digitalRead(13);
```

<http://playground.arduino.cc/Code/Struct>



# Arduino Avanzado: stdio.h

stdio = Librería básica de c

<http://www.cplusplus.com/reference/cstdio/>

sprintf <http://www.cplusplus.com/reference/cstdio/sprintf/>

scanf <http://www.cplusplus.com/reference/cstdio/scanf/>

atoi

itoa

str....



# Arduino Avanzado: enum

Ejemplo máquina de estados:

semáforo

semáforo con peatones

semáforo con peatones y con botón



# Conclusiones

Gracias por vuestra atención

