

Computación Física: Introducción al Arduino

Programando Arduino /
Processing

Programación

- Se programa en un dialecto de C
- Se requiere (forzosamente) una función setup y una función loop
- Setup es la preparación
- Loop es la ejecución

Referencia - Estructura

- **Control Structures**

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return

- **Further Syntax**

- : (semicolon)
- { } (curly braces)
- // (single line comment)
- /* */ (multi-line comment)

Referencia - Estructura

- **Arithmetic Operators**

- plus (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

- **Comparison Operators**

- === (equal to)
- !== (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Referencia - Estructura

- **Boolean Operators**
- && (and)
- || (or)
- ! (not)
- **Compound Operators**
- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)

Referencia - Variables

- **Constants**
- Constants are particular values with specific meanings.
- HIGH | LOW
- INPUT | OUTPUT
- true | false
- Integer Constants
- **Data Types**
- Variables can have various types, which are described below.
- boolean
- char
- byte
- int
- unsigned int
- long
- unsigned long
- float
- double
- string
- array

Referencia - Funciones

- **Digital I/O**
- pinMode(pin, mode)
- digitalWrite(pin, value)
- int digitalRead(pin)
- **Analog I/O**
- int analogRead(pin)
- analogWrite(pin, value) -
PWM
- **Advanced I/O**
- shiftOut(dataPin, clockPin, bitOrder, value)
- unsigned long pulseIn(pin, value)
- **Time**
- unsigned long millis()
- delay(ms)
- delayMicroseconds(us)

Referencia - Funciones

- **Math**
- min(x, y)
- max(x, y)
- abs(x)
- constrain(x, a, b)
- **Random Numbers**
- randomSeed(seed)
- long random(max)
- long random(min, max)
- **Serial Communication**
- Serial.begin(speed)
- int Serial.available()
- int Serial.read()
- Serial.flush()
- Serial.print(data)
- Serial.println(data)

Librerías

- **ARDUINO**
- [SoftwareSerial](#) - for serial communication on any digital pins
- [Stepper](#) - for controlling stepper motors
- [EEPROM](#) - reading and writing to "permanent" storage
- **WIRING**
- [Matrix](#) - Basic LED Matrix display manipulation library
- [Sprite](#) - Basic image sprite manipulation library for use in animations with an LED matrix
- [Wire](#) - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors. On the Arduino, SDA is on analog input pin 4, and SCL on analog input pin 5.

Librerías no oficiales

- [Simple Message System](#) - send messages between Arduino and the computer
- [OneWire](#) - control devices (from Dallas Semiconductor) that use the One Wire protocol.
- [LCD](#) - control LCDs (using 8 data lines)
- [LCD 4 Bit](#) - control LCDs (using 4 data lines)
- [LedControl](#) - for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- [TextString](#) - handle strings
- [Metro](#) - help you time actions at regular intervals
- [Servo](#) - provides software support for Servo motors on any pins.
- [Servotimer1](#) - provides hardware support for Servo motors on pins 9 and 10
- [X10](#) - Sending X10 signals over AC power lines
- [LedControl](#) - an alternative to the Matrix library for driving multiple LEDs with Maxim chips.

Ejemplos que veremos

- Blink, hacer parpadear un LED
- Ejercicios con LEDs
- AnalogRead, leer una señal del ADC
- AnalogWrite, escribir PWM
- Ejercicios con Processing
- Sonido piezoeléctrico
- Display LCD
- Motor CD

Blink

- Ya hay un LED SMD conectado al pin 13
- También se puede conectar un LED y observarlos simultáneamente

Blink - Código

```
int ledPin = 13; // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000); //waits for a second
}
```

Ejercicio

- Hacerlo parpadear con un tiempo de encendido de 500 ms y de apagado de 2000 ms

Ejercicio 2

- Conectar 2 LEDs y hacer que parpadeen simultáneamente
- OJO: utilizando puertos diferentes
- Con esos 2 LEDs hacer que parpadeen alternadamente (ambos tiempos de 500 ms)
- Hacer lo mismo con 3 LEDs (tiempo de ms)

AnalogRead

- Conectamos un potenciómetro como divisor de voltaje
- Leemos ese voltaje
- Un LED parpadea proporcionalmente al voltaje leído

AnalogRead - Código

```
int potPin = 2; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED

int val = 0; // variable to store the value coming from the sensor

void setup()
{
  pinMode(ledPin, OUTPUT); // declare the ledPin as an OUTPUT
}

void loop()
{
  val = analogRead(potPin); // read the value from the sensor
  digitalWrite(ledPin, HIGH); // turn the ledPin on
  delay(val); // stop the program for some time
  digitalWrite(ledPin, LOW); // turn the ledPin off
  delay(val); // stop the program for some time
}
```

Ejercicios

- Invertir el sentido del potenciómetro sin cambiar las conexiones eléctricas
- Con Processing graficar el valor leído del potenciómetro

Para graficar con Processing

- <http://www.arduino.cc/en/Tutorial/Graph>
- Leeremos el puerto análogo 0
- Primero programar Arduino con el siguiente código (sí, es realmente pequeño):

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println(analogRead(0));
  delay(20);
}
```

Para graficar con Processing

- Ahora cerrar la aplicación de Arduino y abrir Processing
- Volver a checar en Device Manager qué puerto tenemos asignado al COM USB
- Pegar el código de la página poniendo atención en la sección que aparece en **negrita** en la siguiente diapositiva:

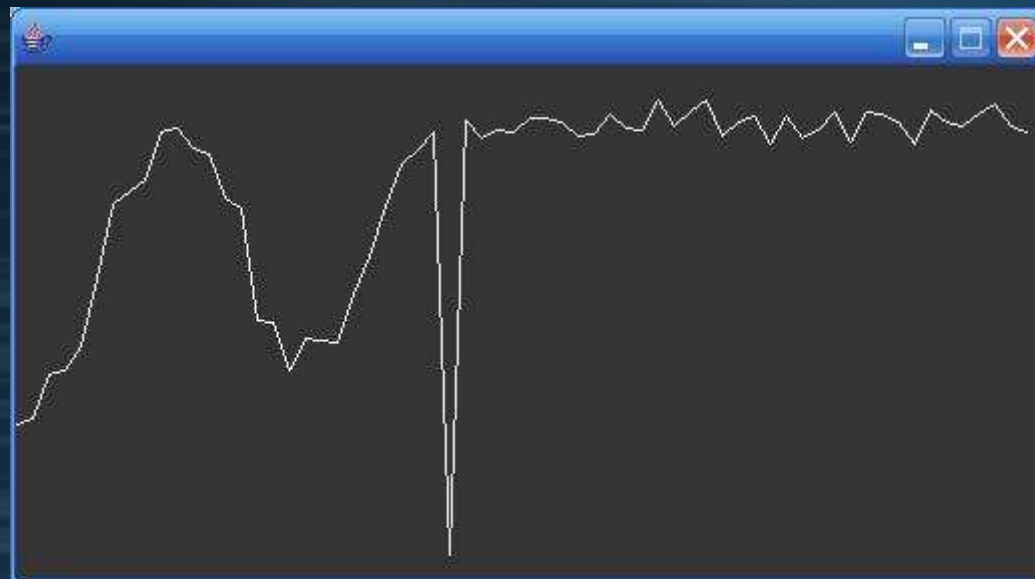
```
import processing.serial.*;
Serial port; String buff = "";
int NEWLINE = 10; // Store the last 64 values received so
    we can graph them.
int[] values = new int[64];
void setup() { size(512, 256);

println("Available serial ports:");
println(Serial.list());

port = new Serial(this, Serial.list()[0], 9600);
// If you know the name of the port used by the Arduino
    board, you
// can specify it directly like this.
//port = new Serial(this, "COM1", 9600); }

void draw() { background(53); stroke(255);
for (int i = 0; i < 63; i++)
line(i * 8, 255 - values[i], (i + 1) * 8, 255 - values[i +
    1]);
```

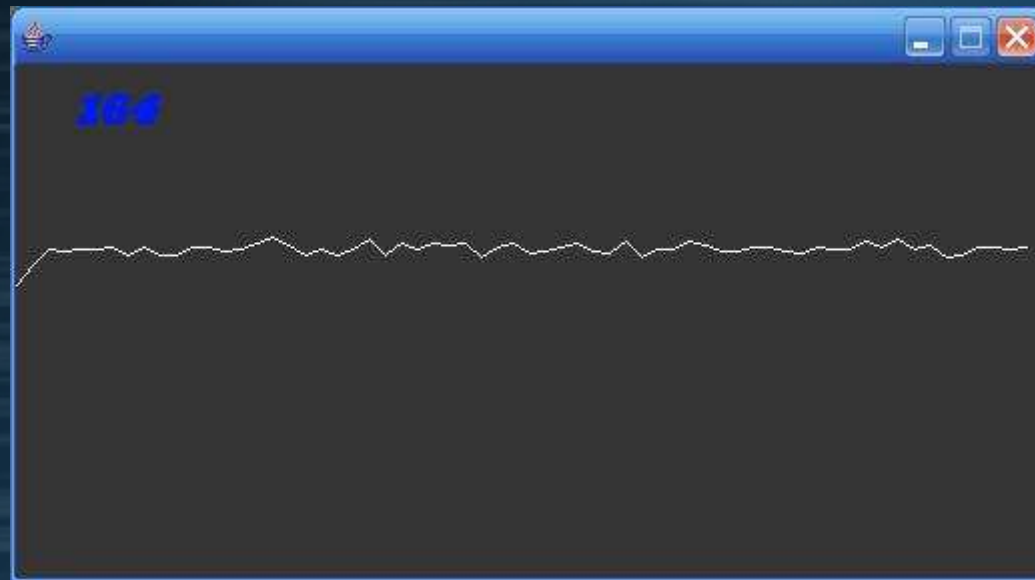
Debemos obtener esto:



Ejercicio

- Lograr escribir en la misma ventana de la gráfica el valor de la altura de la línea (el rango es de 0 a 255)
- Se sugiere promediar algunos valores para que el valor no oscile
- Se sugiere leer y entender el ejemplo *examples > basics > typography > words*

El resultado debe ser así:



Trabajo Independiente

- Investigar e implementar demostraciones de las siguientes técnicas:
- Switch debouncing
- Habilitar resistencia de Pull-Up para entradas digitales

Ejercicios Sugeridos

- Programar y mostrar un contador binario de 4 bits
- Programar secuencias de LED's (incrementos lineales, El Auto Increíble, etc.)
- Reducir y aumentar el brillo de un LED con PWM
- Programar un vúmetro con una entrada análoga y 8 LED's

Ejercicios Sugeridos

- Cargar la librería LCD para enviar mensajes a una pantalla de cristal líquido
- Leer el valor de temperatura de un sensor LM35
- Escribir ese valor en el display LCD

Ejercicios Sugeridos

- Utilizando Processing lograr variar la intensidad de un LED moviendo el mouse