



PINGUINO : Implementación De Un Modulo Básico Para Comunicación Con El Pc

Por : Julio Fabio De La Cruz Gomez. -<http://micropinguino.blogspot.com>

La comunicación entre dispositivos electrónicos y el PC es de uso como en nuestros días, nada tan fácil como conectar nuestro dispositivo en el puerto USB y empezarlo a utilizar. En electrónica un dispositivo conectado al PC puede ser empleado para la captura y visualización de datos, así, ¿de por que no disponer de una tarjeta electrónica para tal fin ?. Es aquí donde la tarjeta pinguino viene al rescate, configurándose como un dispositivo básico de comunicación y control digital de gran utilidad para la realización de nuestros proyecto electrónicos.

Modulo Pinguino configurado para aplicaciones de lógica digital básica.

Este modulo preconfigurado tiene como fin el de ser una tarjeta de comunicación con el computador y nuestras aplicaciones electrónicas.

El microcontrolador de pinguino cuenta con un programa ya establecido, siendo este modulo como una caja negra para el usuario, donde este solo debe saber que terminales serán las entradas, que terminales serán las salidas digitales y como se acceden por medio de un programa en el PC.

El modulo esta configurado de manera que desde el PC se pueda enviar y recibir datos, por medio del puerto USB.

El usuario cuenta con un conjunto de funciones en el lenguaje de programación **Processing** para el uso de este modulo, permitiendo enfocarse en programa que controlara la aplicación y su interfaz gráfica de usuario. También este conjunto de funciones de alto nivel pueden ser llevadas a cualquier otro lenguaje de programación dada la versatilidad de **Pinguino**.

Características del modulo para aplicaciones con lógica digital básica con el PIC 18F4550.

Para la realización de los proyectos aquí propuestos el modulo tendrá las siguientes entradas y salidas predefinidas, según sean los requerimientos se puede cambiar esta configuración reprogramando el dispositivo.

- 12 salidas digitales
- 9 entradas digitales
- 8 entradas ADC
- 1 salida análoga

En la Figura 1 se ilustran que pines se disponen como salidas digitales, como entradas digitales y conversión análogo a digital ADC.

Programación en Processing.

Processing es un lenguaje de programación basado en Java, con la característica de facilitar el desarrollo de interfaces gráficas y animaciones además de contar con la facilidad que proporciona Java en la programación web.

Figura 1. Configuración de los terminales

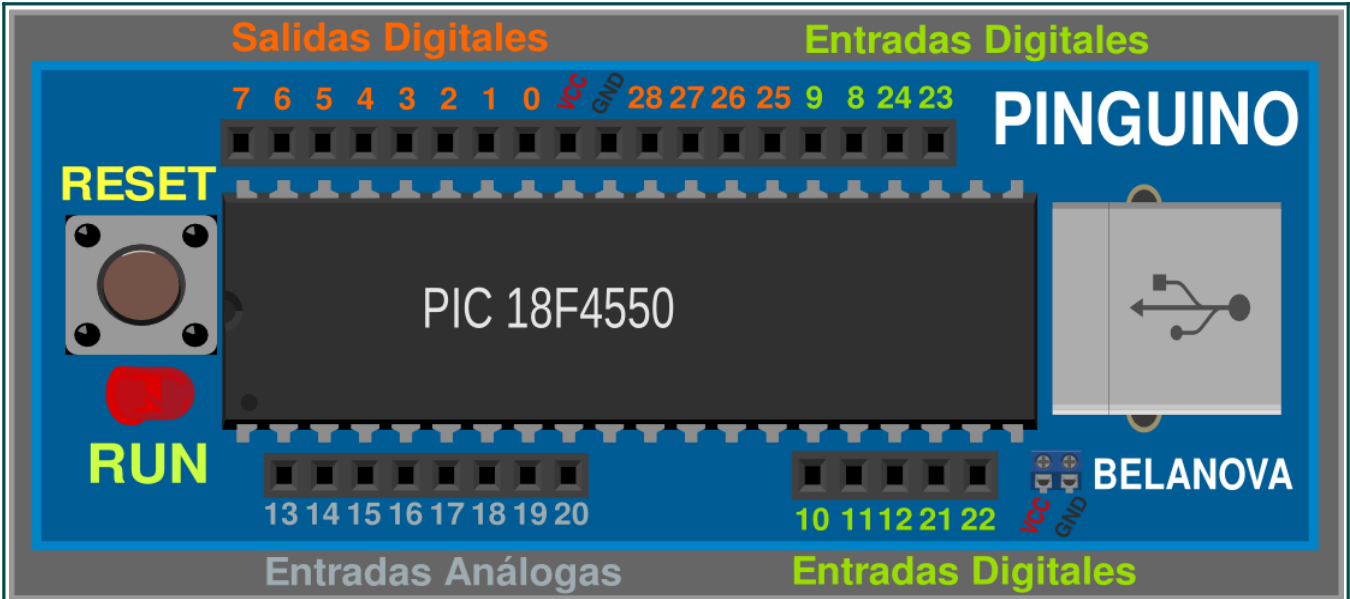
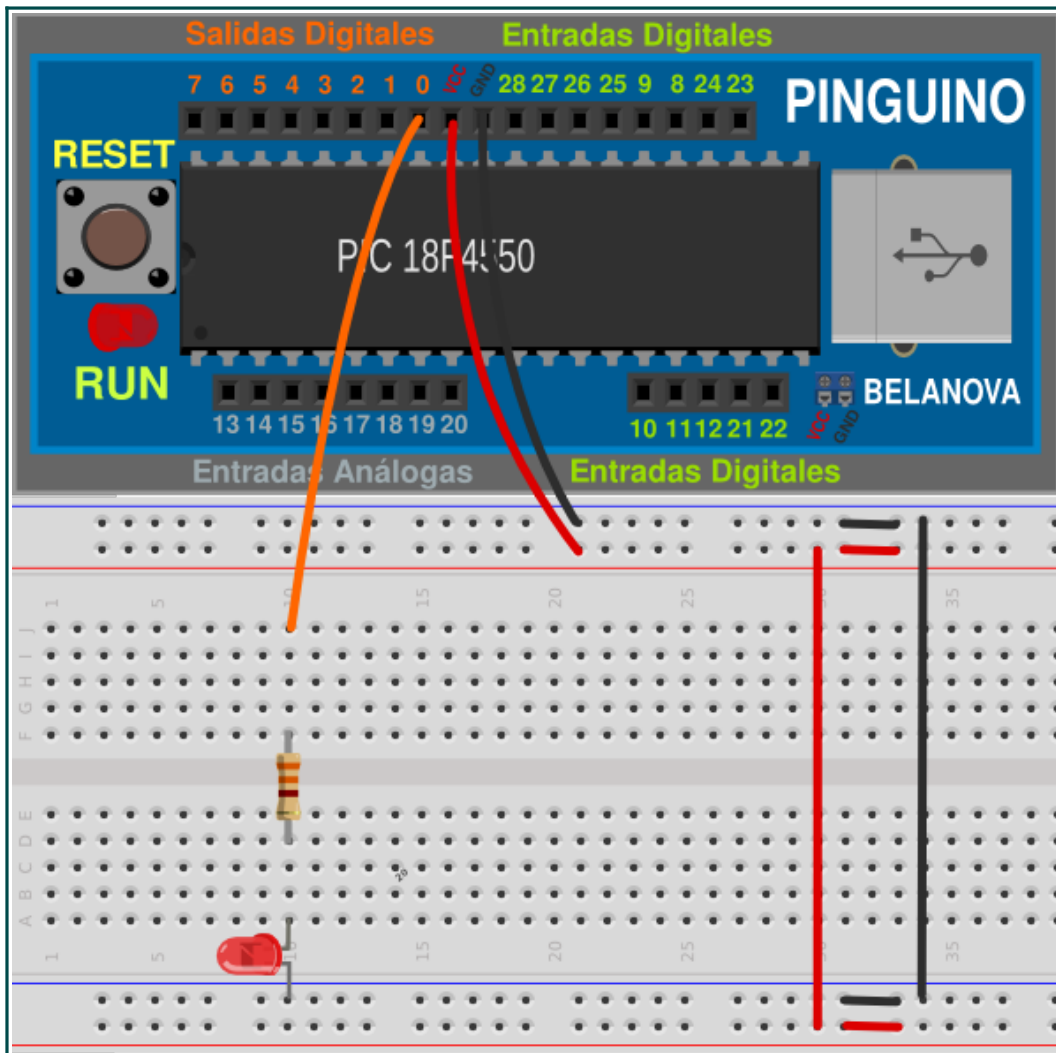


Figura 2. Encendido y apagado de un led.



Encender y apagar un led.

Este primer ejemplo mostrara como se controla el encendido y apagado de un led desde **Processing**, para este ejemplo se realiza el montaje que se muestra en la **Figura 2**.

En el cuadro **Código 1** se muestran las lineas de correspondientes en **Processing**.

Ejemplo de lectura de las entradas digitales.

El ejemplo a continuación muestra como se controla el encendido y apagado de 4 leds por medio de activación de alguno de los 4 pulsadores tal como se muestra en la Figura 3.

En el cuadro Código 2 se muestran las lineas de correspondientes en **Processing**.

Ejemplo de lectura de la entrada ADC de la tarjeta.

El ejemplo a continuación muestra el valor capturado por el conversor análogo digital cuando se varia el potenciómetro el montaje correspondiente se muestra en la **Figura 3**.

En el cuadro **Código 3** se muestran las lineas de correspondientes en **Processing**.

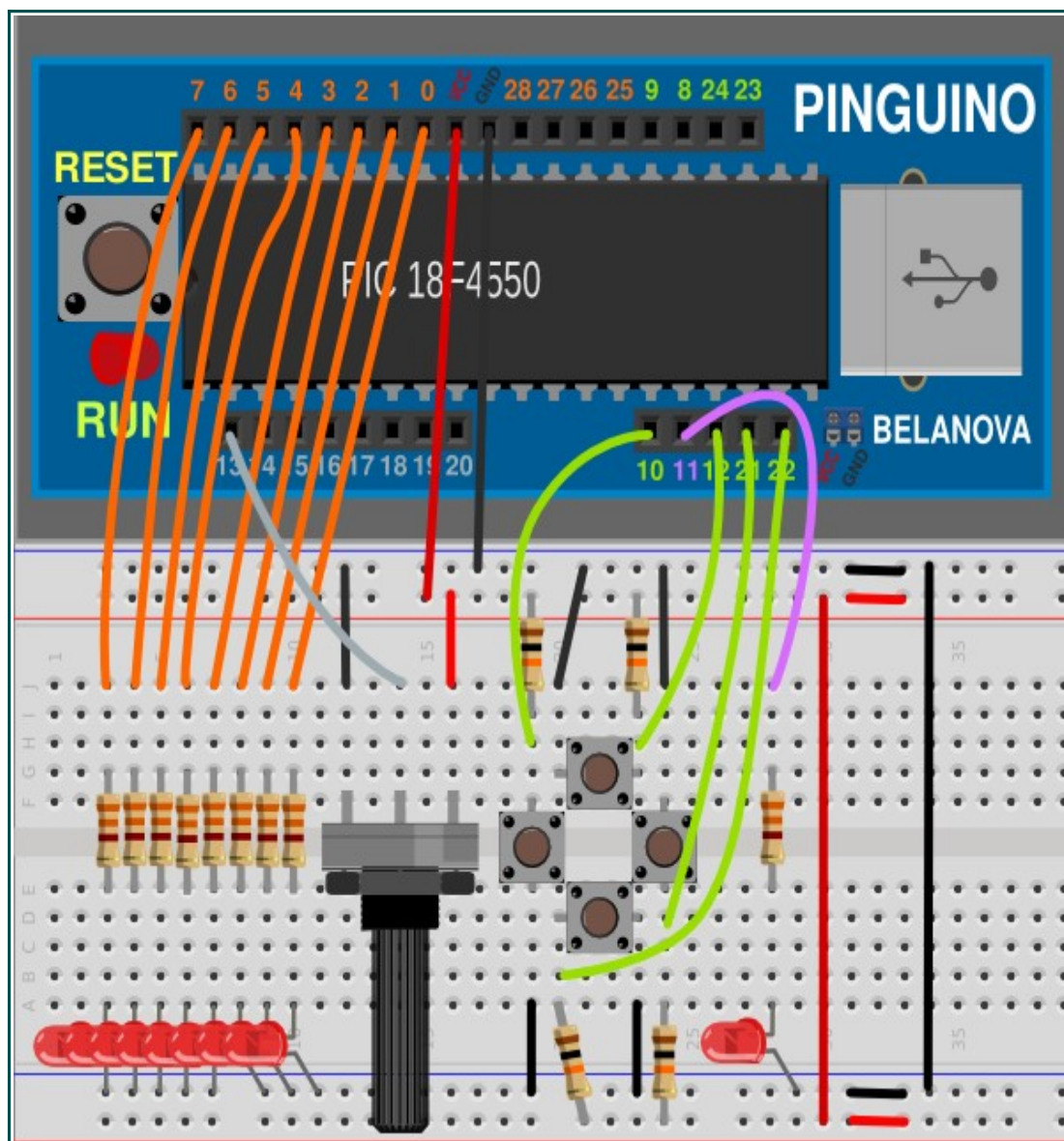
Código 1. Encender y apagar un led

```
import hypermedia.pinguino.*;
Pinguino pinguino;
void setup(){
  Pinguino.log( true );
  pinguino = new Pinguino( this );
}
void draw(){
  pinguino.digitalWrite(0,Pinguino.LOW);
  delay(1000);
  pinguino.digitalWrite(0,Pinguino.HIGH);
  delay(1000);
}
```

Código 2. Lectura de entradas digitales.

```
import hypermedia.pinguino.*;
Pinguino pinguino;
int nPulsador1;
int nPulsador2;
int nPulsador3;
int nPulsador4;
void setup(){
  Pinguino.log( true );
  pinguino = new Pinguino( this );
}
void draw(){
  nPulsador1=pinguino.digitalRead(10);
  nPulsador2=pinguino.digitalRead(12);
  nPulsador3=pinguino.digitalRead(21);
  nPulsador3=pinguino.digitalRead(22);
  pinguino.digitalWrite(0,Pinguino.LOW);
  pinguino.digitalWrite(1,Pinguino.LOW);
  pinguino.digitalWrite(2,Pinguino.LOW);
  pinguino.digitalWrite(3,Pinguino.LOW);
  if(nPulsador1==0){
    println("Pulsador izquierdo esta presionado ");
    pinguino.digitalWrite(0,Pinguino.HIGH);
  }
  if(nPulsador2==0){
    println("Pulsador arriba esta presionado ");
    pinguino.digitalWrite(1,Pinguino.HIGH );
  }
  if(nPulsador3==0){
    println("Pulsador derecho esta presionado ");
    pinguino.digitalWrite(2,Pinguino.HIGH );
  }
  if(nPulsador4==0){
    println("Pulsador abajo esta presionado ");
    pinguino.digitalWrite(3,Pinguino.HIGH );
  }
}
```

Figura 3. Montaje multiproposito.



Código 3. Lectura del conversor análogo digital

```
import hypermedia.pinguino.*;
Pinguino pinguino;
int nAdc;
void setup(){
  Pinguino.log( true );
  pinguino = new Pinguino( this );
}
void draw(){
  nAdc=pinguino.analogRead(13);
  println("El Valor es: " + nAdc);
}
```

Funciones implementadas en Processing.

Al final de esta guía se encuentra una tabla que enumera y hace una descripción de las diferentes funciones empleadas para manejar la tarjeta **Pinguino** desde **Processing**.

Referencias.

<http://micropinguino.blogspot.com>

<http://ubaa.net/shared/processing/pinguino/>

<http://jpmandon.blogspot.com/>

Funciones

Función	Descripción	Sintaxis	Ejemplo
analogRead(terminal)	Retorna el dato leído de un terminal específico del microcontrolador, los valores están comprendidos en el rango de 0 y 1023.	<p>terminal indica el terminal del cual se desea leer el dato análogo.</p> <p>Esta función retorna un valor comprendido entre 0 y 1023, si hay un error retorna -1</p>	<pre>import hypermedia.pinguino.*; // conectar a la tarjeta Pinguino pinguino = new Pinguino(this); // lee el valor en el terminal 13 int value = pinguino.analogRead(13); // imprimir valor leído en consola println(value);</pre>
analogWrite(terminal,valor)	Envía un valor análogo (PWM) a un terminal específico del microcontrolador, los valores están comprendidos entre 0 y 1023.	<p>terminal indica el terminal del microcontrolador al que se enviara el dato.</p> <p>valor indica que valor sera enviado, los valores están comprendidos entre 0 y 1023.</p> <p>Esta función no retorna ningún valor.</p>	<pre>import hypermedia.pinguino.*; // conectar a la tarjeta Pinguino pinguino = new Pinguino(this); // envia el dato 100 al terminal 11 pinguino.analogWrite(11, 100);</pre>
clear()	Coloca en 0 todos los terminales del microcontrolador.	esta función no retorna ningún valor	
digitalRead(terminal)	Lee el estado digital de un terminal específico del microcontrolador	terminal indica el numero del terminal del pic que se va a leer su estado digital 0 (LOW) o 1(HIGH). -1 en caso de un error.	<pre>import hypermedia.pinguino.*; // conectar a la tarjeta Pinguino pinguino = new Pinguino(this); // lee el valor digital del terminal 0 int value = pinguino.digitalRead(0); // imprime el valor leído en consola println(value);</pre>
digitalWrite(terminal,valor)	Escribe un dato digital 0 (BAJO) o 1 (ALTO) a un terminal específico del microcontrolador.	<p>terminal indica a terminal se escribirá el dato digital.</p> <p>valor indica que estado lógico se escribirá en el terminal, para indicar un estado bajo se envía 0 o LOW y un estado alto 1 o HIGH.</p> <p>Esta función no retorna ningún valor</p>	<pre>import hypermedia.pinguino.*; // conectar a la tarjeta Pinguino pinguino = new Pinguino(this); //escribir un alto a el terminal 0 pinguino.digitalWrite(0, Pinguino.HIGH);</pre>
log(habilitar)	Sirve para indicar el estado de conexión con la tarjeta pinguino.	habilitar indica si se habilita o no los mensajes de estado por consola, para habilitarlos se coloca true y para deshabilitar se coloca false. Esta función no retorna ningún valor.	<pre>import hypermedia.pinguino.*; // habilita mensajes de consola Pinguino.log(true); // conectar a la tarjeta Pinguino pinguino = new Pinguino(this);</pre>

Programa que se descarga en pinguino

```
int i;
uchar todo,mode,pin,value;
unsigned char buffer[2];
int temp;
int endstring;
void clear();
void setup() {
    for( i=0; i<8; i++ ) pinMode( i, OUTPUT );
    for( i=21; i<=24; i++ ) pinMode( i, INPUT );
    for( i=25; i<=28; i++ ) pinMode( i, OUTPUT );
    clear();
}
void loop() {
    if ( USB.available() ) {
        if (USB.read()=='+'){
            todo =USB.read();
            if ( todo=='C' ) clear();
            if ( todo=='W' ) {
                mode = USB.read();
                pin = USB.read();
                value = USB.read();
                if ( mode=='D' ) digitalWrite( pin, value );
                if ( mode=='A' ) {
                    temp=value+(USB.read()*256);
                    analogWrite( pin, temp );
                }
            }
            if ( todo=='R' ) {
                mode = USB.read();
                pin = USB.read();
                if ( mode=='D' ) {
                    buffer[0] = digitalRead( pin );
                    USB.send( buffer, 1 );
                }
                if ( mode=='A' ) {
                    temp = analogRead( pin );
                    buffer[0]=temp;
                    buffer[1]=temp/256;
                    USB.send( buffer, 2 );
                }
            }
            endstring=USB.read();
        }
    }
}
```

```
void clear() {  
    for( i=0; i<8; i++) {  
        digitalWrite( i, LOW );  
    }  
    for( i=25; i<29; i++) {  
        digitalWrite( i, LOW );  
    }  
}
```