



# PINGUINO : Implementación De Un Modulo Básico Para Comunicación Y Desarrollo

por : Julio Fabio De La Cruz G. -<http://integradorelectronica.blogspot.com>

El puerto paralelo del computador, es un puerto comúnmente usado en los proyectos con lógica digital en los que se requiera enviar datos desde el computador para su control. También este puerto es empleado para la captura y visualización de datos. Sin embargo este puerto es poco frecuente encontrarlo en un equipo normal de computo actual, aun mas en un equipo portátil. Es aquí donde la tarjeta pingüino viene al rescate, configurándose como un modulo básico de comunicación y control digital de gran utilidad.

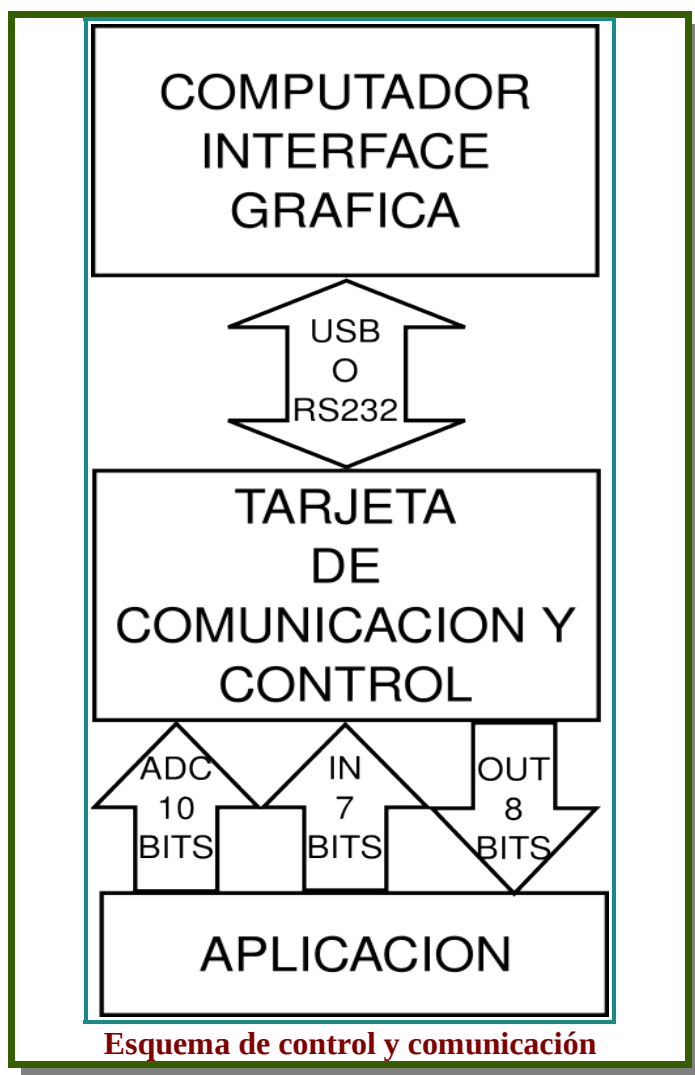
## Modulo Pingüino configurado para aplicaciones de lógica digital básica.

Este modulo preconfigurado tiene como fin el de ser una tarjeta de comunicación serial con el computador para reemplazar el puerto paralelo como puerto de comunicación para la realización de aplicaciones

El microcontrolador de pingüino contara con un programa ya establecido, siendo el modulo como una caja negra para el usuario, donde este solo debe saber que pines serán la entradas y que pines 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, ya sea por medio del puerto serial o USB.

El usuario contara una API para el uso de este modulo, con funciones como `usb_send(dato)` para enviar un dato digital a la tarjeta por el puerto USB y una función como `dato = usb_read()` para leer un dato por medio del puerto USB o serial.



## Características del modulo para aplicaciones con lógica digital básica.

Para la realización de los proyectos el modulo tendrá las siguientes entradas y salidas ya definidas.

- 8 salidas digitales
- 7 entradas digitales
- 1 entrada ADC
- comunicación serial RS232
- comunicación USB.

Los gráfico ilustran que pines se disponen como salidas indicadas por OUT0 a OUT7, como entradas los pines IN0 a IN6, como entrada ADC se indica IN

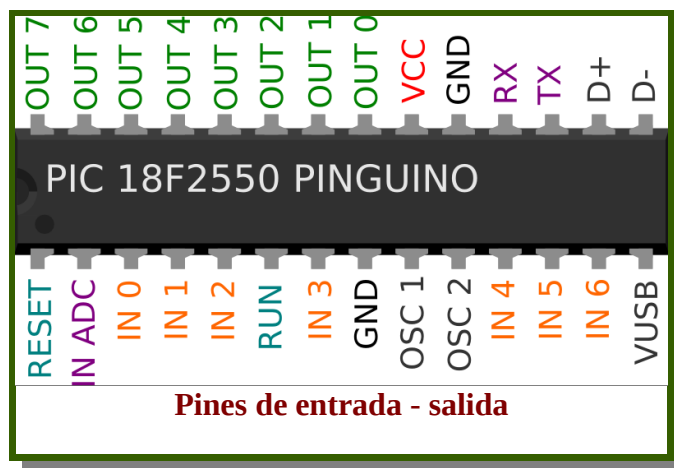
ADC, la comunicación serial por RX, TX y la USB por D+ y D-

## Reprogramación del modulo para aplicaciones con lógica digital.

El usuario tiene la posibilidad si desea de hacer su propio programa y reprogramar el microcontrolador para una necesidad especifica, esto es fácil de hacer por que el lenguaje para programar la tarjeta pingüino utiliza una sintaxis similar al lenguaje C como el visto en los cursos de básicos de programación y por que solo necesita los conocimientos básicos de lógica digital hacer su propio programa modificando el propuesto para este modulo preconfigurado.

Otra ventaja de este modulo es que es consecuente con las materias de de programación vistas y a futuro con un curso de microcontroladores, así que la realización de una

aplicaciones mas compleja se iniciarían desde un hardware y software ya construidos.



*“Tiene como fin el de ser una tarjeta de comunicación serial con el computador para reemplazar el puerto paralelo como puerto de comunicación para la realización de aplicaciones ”*

El microcontrolador quedaría configurado tal como se muestra en la gráfica, los terminales que están configurados como salidas indicados como OUT, las entradas digitales como IN, la entrada al conversor analogo digital como IN ADC y los terminales RX y TX para comunicación serial.

## Ejemplo de activación de las salidas de la tarjeta.

El propósito sera el de enviar un dato binario de 8 bits por medio del puerto **USB** del computador y visualizar este dato por medio de 8 leds conectados en los pines de salida de la tarjeta pingüino.

Para realizar esto primero debemos tener un programa en el computador que envíe un dato por el puerto **USB**. Los pasos a realizar en el programa son los siguientes:

- Verificar que la tarjeta esta conectada al puerto USB, para esto se utiliza la función `pinguino_init()`.

- Enviar dato por medio del puerto USB utilizando la función `usb_send(dato)`.
- Liberar la tarjeta pinguino del puerto USB, utilizando la función `pinguino_release()`.
- Cerrar el puerto USB, con la función `pinguino_close()`.

En este ejemplo la variable `leds` se carga con el dato binario 11111111 (en hexadecimal 0xFF) para encender todos los 8 leds

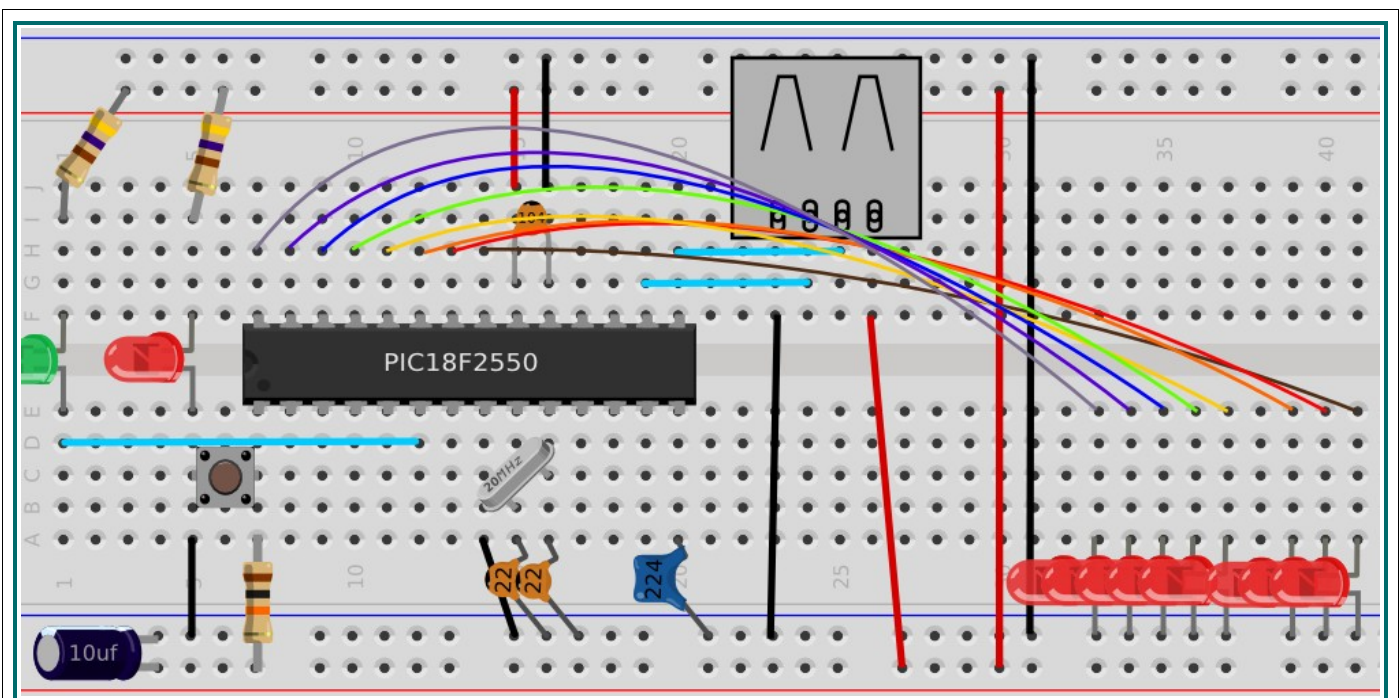
En los gráficos se puede observar el montaje y el hardware empleados para este ejemplo de envío de datos binarios

## Ejemplo de lectura de las entradas de la tarjeta.

El propósito será el de leer un dato binario de 4 bits por medio del puerto **USB** del computador y visualizarlo en pantalla. El dato será enviado por medio de pulsadores tal como se ilustra en el circuito.

Para realizar esto primero debemos tener un programa en el computador que envíe un dato por el puerto **USB**. Los pasos a realizar en el programa son los siguientes:

- Verificar que la tarjeta está conectada al



**Montaje básico para mostrar las salidas digitales**

```

1  #include <iostream>
2  #include<usb.h>
3  #include<pinguino.h>
4  using namespace std;
5
6  int main(){
7      char leds;
8      if( pinguino_init() < 0 ){
9          cout << "error al iniciar pinguino";
10         return -1;
11     }
12     leds=0xFF;
13     usb_send(leds);
14     pinguino_release();
15     pinguino_close();
16     return 0;
17 }

```

**Envío de dato**

puerto USB, para esto se utiliza la función `pinguino_init()`.

- Leer dato por medio del puerto USB utilizando la función `usb_read()`. La lectura se almacena en la variable tipo `char` entrada.
- Mostrar el dato leído en la consola.
- Liberar la tarjeta pinguino del puerto USB, utilizando la función `pinguino_release()`.

- Cerrar el puerto USB, con la función `pinguino_close()`.

En los gráficos se puede observar el código empleado y el montaje respectivo.

### Ejemplo de lectura de la entrada ADC de la tarjeta.

El propósito será el de leer un dato binario de 8 bits correspondiente al valor digital de una entrada analógica por medio del puerto **USB** del computador y visualizarlo en pantalla. El dato será enviado por medio de la variación de un potenciómetro.

Para realizar esto primero debemos tener un programa en el computador que envíe un dato por el puerto **USB**. Los pasos a realizar en el programa son los siguientes:

- Verificar que la tarjeta está conectada al puerto USB, para esto se utiliza la función `pinguino_init()`.
- Enviar una instrucción indicando que se realizará una lectura del ADC por medio del puerto USB utilizando la función `usb_send('A')`.
- Leer dato por medio del puerto USB utilizando la función `usb_read()`. La lectura se almacena en la variable tipo `char` `entrada`.
- Mostrar el dato leído en la consola.
- Liberar la tarjeta pinguino del puerto USB, utilizando la función `pinguino_release()`.
- Cerrar el puerto USB, con la función `pinguino_close()`.

En los gráficos se puede observar el código empleado y el montaje respectivo.

```

1  #include <iostream>
2  #include<usb.h>
3  #include<pinguino.h>
4  using namespace std;
5
6  int main(){
7      char entrada;
8      if( pinguino_init() < 0 ){
9          cout << "error al iniciar pinguino";
10         return -1;
11     }
12     entrada=usb_read();
13     cout << "Dato leído" << leer << endl;
14     pinguino_release();
15     pinguino_close();
16     return 0;
17 }
```

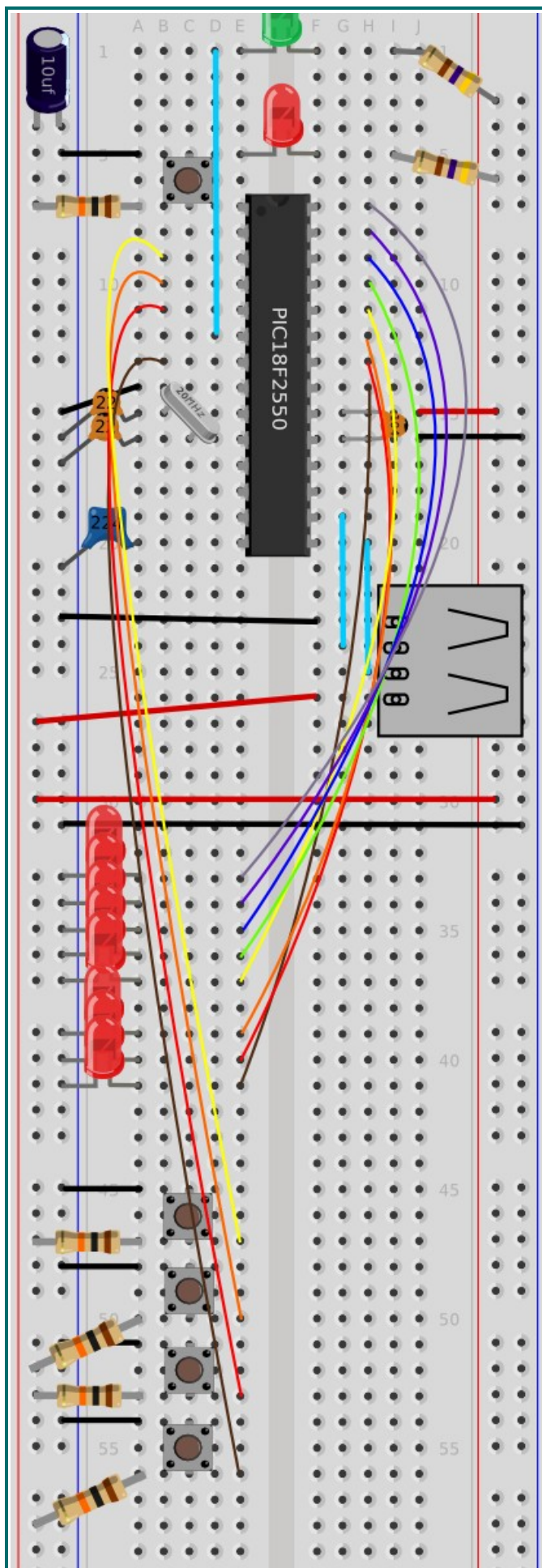
#### Lectura de dato

```

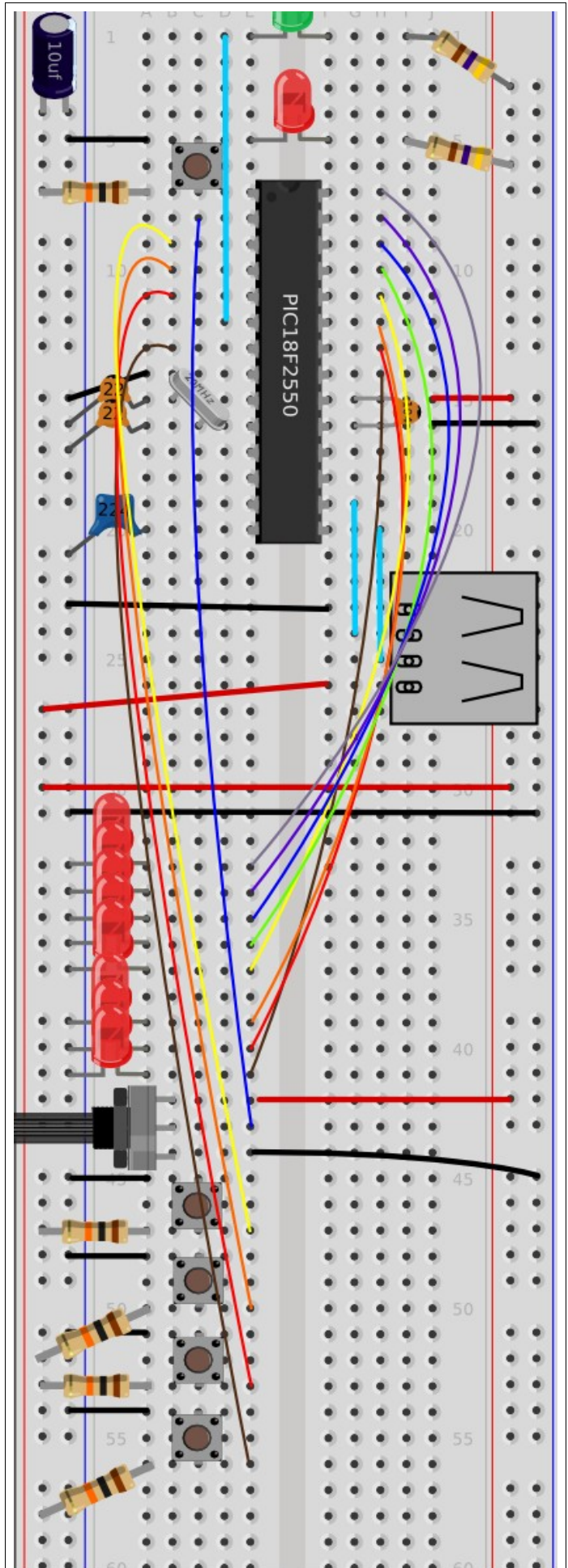
1  #include <iostream>
2  #include <usb.h>
3  #include <pinguino.h>
4  using namespace std;
5
6  int main(){
7      char lectura_adc;
8      if(pinguino_init()<0){
9          cout << "error al iniciar pinguino";
10         return -1;
11     }
12     usb_send('A');
13     lectura_adc=usb_read();
14     cout << "Lectura ADC " << lectura_adc << endl;
15     pinguino_release();
16     pinguino_close();
17     return 0;
18 }
```

#### Lectura ADC





## Entradas digitales



## Entrada análoga