

Pantalla Nextion

Una pantalla nextion no deja de ser una pantalla TFT con un panel táctil de la familia denominada como Nextion. Estas pantallas son fabricadas por Itead, un fabricante asentado en Shenzhen.

Existen muchos tipos de pantallas de este fabricante, principalmente están divididas en el grupo de pantallas que tiene GPIO y EEPROM y las que no lo tienen. Dentro de estos dos grupos, existen distintas dimensiones de panel, las más pequeñas de 2.4 pulgadas y las más grandes de 7 pulgadas. En este tutorial usaremos el modelo NX8048T050, es decir un modelo sin GPIO, sin EEPROM y con 5 pulgadas de dimensiones de panel.

Una de las peculiaridades de Nextion es que la interfaz gráfica se carga en la memoria flash de la propia pantalla y no en el micro que controla el sistema, por tanto, el micro principal queda un poco más libre y el refresco de pantalla es mucho más rápido. Como la luz y el día.

Otra de las grandes peculiaridades de la pantalla es que usa una UART para comunicarse y por tanto solo usaremos los 2 pines del puerto serie de nuestro microcontrolador para gestionar la pantalla, tanto lo que se muestra como las lecturas del panel táctil.

Ventajas y desventajas de la pantalla Nextion

Se me ocurren muchas ventajas sobre estas pantallas, para empezar tienen hardware dedicado. Cuentan con un micro basado en ARM 7 y una memoria flash dedicada al control de la pantalla, solo para el control de la pantalla. Y en los modelos más grandes encontramos memoria DRAM y una FPGA de Altera para gestionar en HARDWARE el refresco de pantalla.



Hardware dedicado en pantalla Nextion

Esto es lo que hace que estas pantallas sean muy interesantes, el disponer de hardware dedicado a procesar y almacenar los datos de la pantalla provoca que el refresco de la imagen y la carga de nuevos datos sea muy rápido, quizá no la más «rápida del oeste», pero si es muy rápida.

La comunicación entre nuestro sistema y la pantalla se hace por UART, lo que simplifica el número de líneas necesarias para la comunicación y simplifica el control.

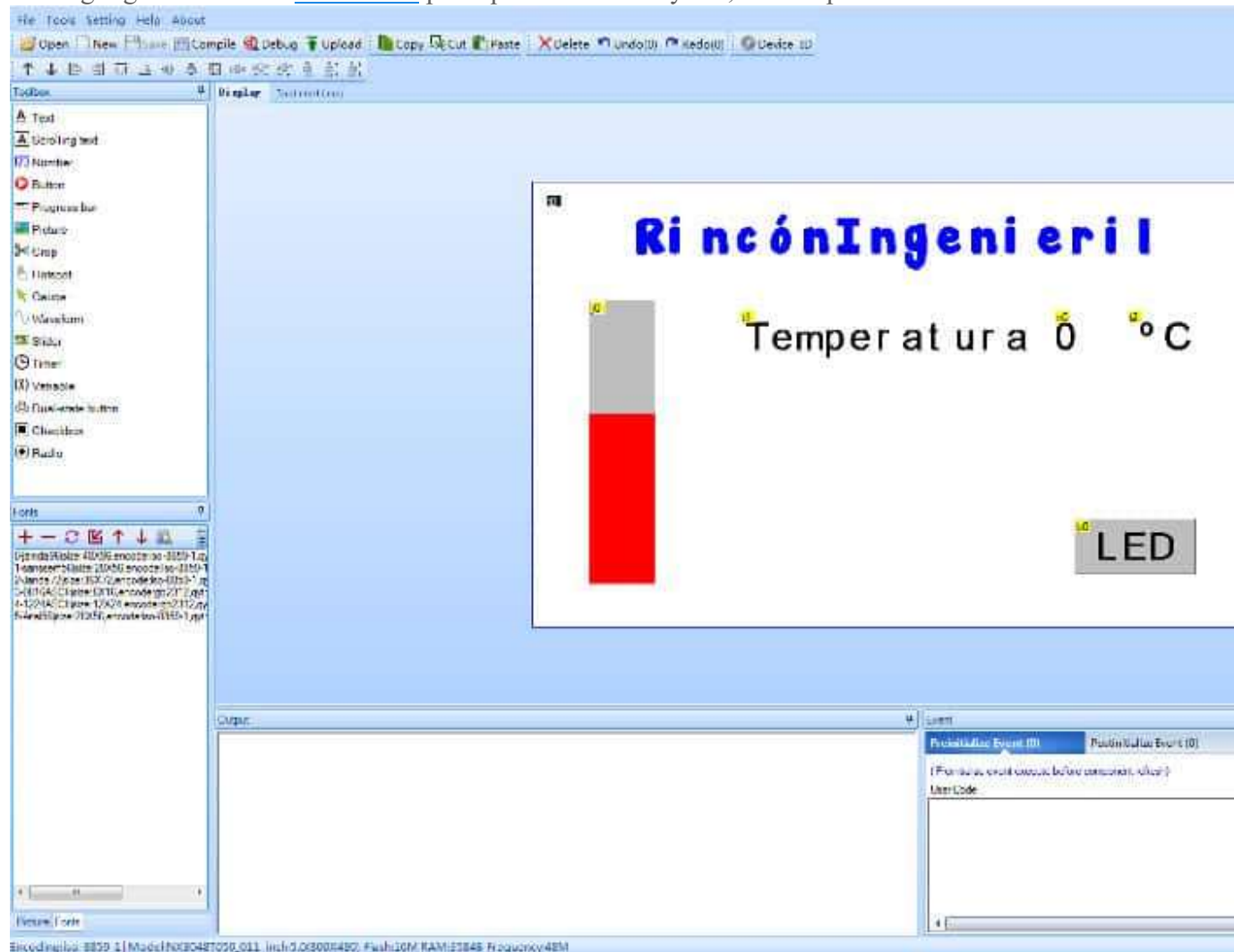
La interfaz se diseña desde un software específico y por tanto, aunque no se te de bien el diseño gráfico, puedes conseguir grandes resultados.

Todas las pantallas son táctiles, con panel resistivo. Y existen muchos tamaños (y precios) para ajustarse a las necesidades de tu proyecto.

Si eres de los míos, es decir, del equipo Open Source, seguro que ya te han chirriado un par de cosas, entre ellas que Nextion Editor no es Open Source. Es gratuito, pero no es Open Source.

Nextion editor para crear la interface gráfica

Nextion tiene su propio software diseñado para diseñar, probar y subir a la memoria de la pantalla la interface que usaremos. Este software se llama **Nextion Editor** el cual podemos descargar gratis desde su [web oficial](#) pero que hasta donde yo se, no es Open Source.



Interface Nextion Editor para pantallas Nextion

El editor Nextion está dividido en seis secciones, tres columnas y dos filas. En la columna de la izquierda nos encontramos con dos secciones, arriba se encuentra una lista con todos los elementos que podemos introducir en pantalla denominada toolbox. Debajo encontramos la lista de fuente y de imágenes.

En el toolbox encontramos, textos, números, botones, pulsadores, imágenes, barras de progreso, sliders y muchos elementos más.

La tabla de fuentes, no es otra cosa que el conjunto de tipografías y de tamaños de texto que vamos a usar y en la tabla de imágenes tendremos todas las imágenes que queramos mostrar en pantalla..

La columna de la izquierda tiene un primer cuadro con todas las distintas páginas que vayamos a mostrar en pantalla y un poco más abajo está la tabla de atributos con la cual podemos configurar todos los elementos de pantalla.

En la columna central tenemos una representación visual de como quedarán los elementos en pantalla y códigos de diálogo para configurar código propio de la pantalla.

Colocar elementos en pantalla es tan fácil como arrastrar desde el toolbox hasta la pantalla en la columna central y luego acudir a la sección de atributos para configurar texto, tamaño, posición, etc.

En el vídeo configuramos la pantalla que hay en la imagen de arriba, puedes echarle un vistazo para aprender el proceso.

Elementos en pantalla

Colocaremos 3 textos a fijos:

- Rincón Ingenieril en la parte superior de la pantalla.
- Temperatura
- °C

Y los elementos modificables

- Un número, que en este caso recibe la referencia **n0**
- Una barra de progresión con el nombre **j0**
- Un botón con el nombre **b0**

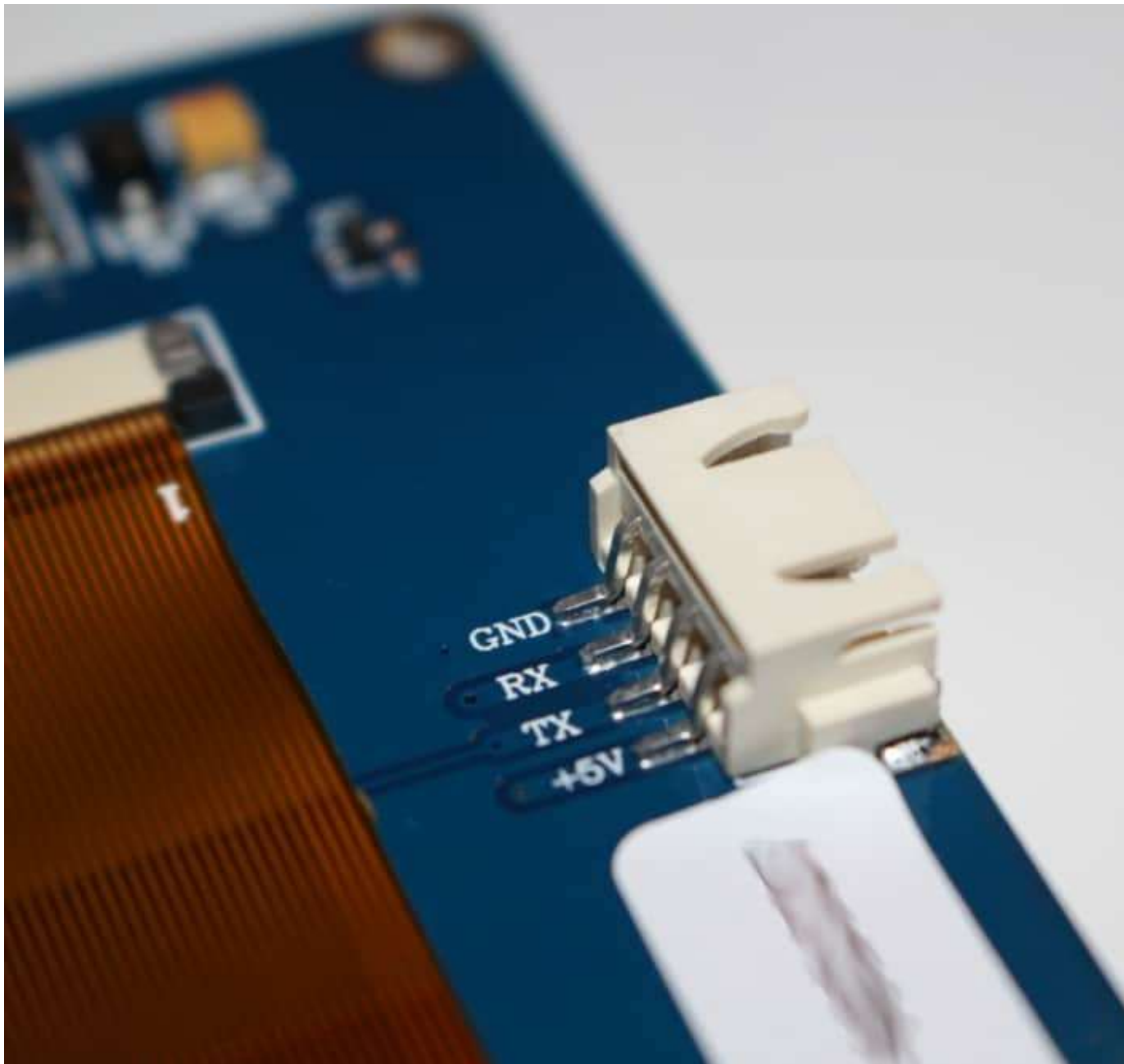
El número y la barra de progresión ayudaran a mostrar la temperatura y el botón servirá para controlar el LED de la placa Arduino en el pin 13.

Pantalla Nextion de Itead y Arduino

De Nextion Editor a la pantalla Nextion

Una vez terminado el diseño de las distintas pantallas de nuestra interface, debemos subir esos datos a la memoria flash. Para hacer esto usaremos un conversor USB a serie TTL de esos que estoy seguro que tienes en algún cajón del laboratorio o de casa.

Conectaremos los cables de la pantalla nextion respetando lo que indica la PCB, Vcc a 5v, GND a GND y Tx y Rx con sus respectivos en el conversor USB a serie.



Nonector UART de pantalla

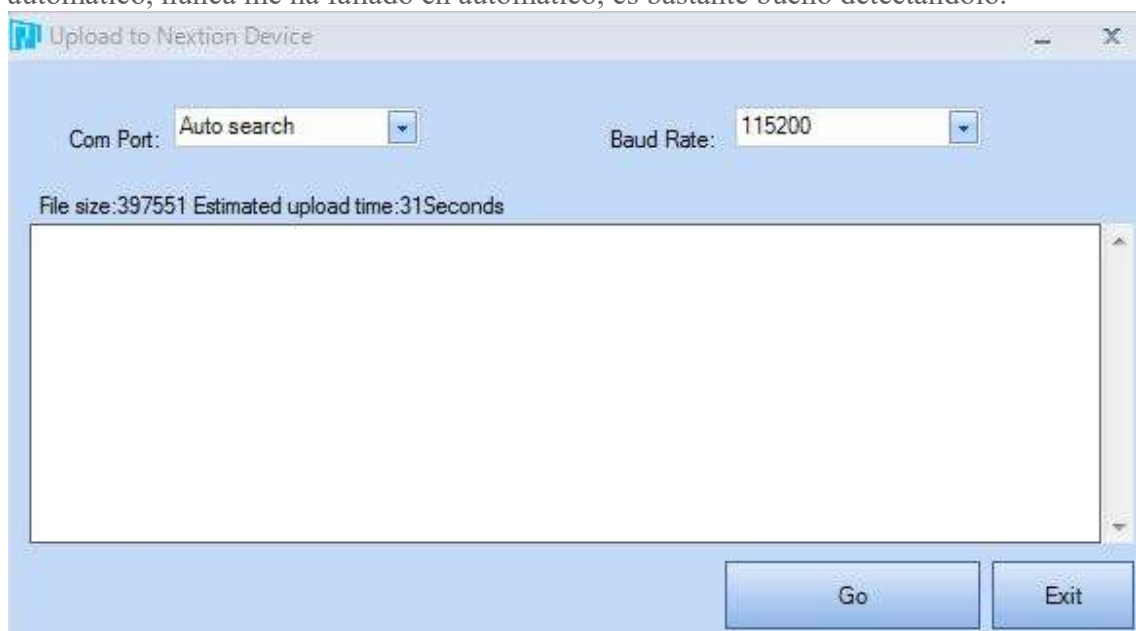
Antes de conectar la pantalla al USB, debemos asegurar que la interfaz que hemos montado funciona correctamente y esto lo hacemos con el *debuguer* que incorpora el editor. Al hacer

click sobre el botón Debug se abrirá el simulador con el cual verificaremos que todo funciona correctamente. Si el editor no se abre es porque existe algún error, este último lo podremos conocer si nos fijamos en el apartado Output (en la parte inferior de Nextion Editor).



Menu del editor

Ahora si que podemos subir el código a la pantalla. Pulsa en el botón **Upload** del menú y se abrirá un cuadro de diálogo en el cual podrás seleccionar en que puerto está conectada la pantalla y a que velocidad está configurada. Si no lo tienes claro, puedes dejar en e automático, nunca me ha fallado en automático, es bastante bueno detectandolo.



Upload a Nextion Editor

Por último dale a **Go** y deja que el puerto serie haga su magia.

La carga puede tardar unos minutos y si tienes imágenes aun más, ten en cuenta que es un puerto serie, normalmente a 115200 baudios.

Librería para Arduino

Este tipo de pantallas se pueden controlar desde Arduino y por tanto Itead se ha encargado de crear una librería que nos proporciona acceso a los parámetros de la pantalla nextion. Con ella podemos modificar textos, números, colores, leer botones y otras muchas cosas.

Puedes descargar la librería desde le Github de [Itead](#).

!!!Ojo!!! Cuidado

Es importante conocer que la mejor opción si estas desarrollando con la pantalla es usar un Arduino Mega ya que la librería está preparada para usar el puerto serie 0 de Arduino para depurar la comunicación con la pantalla y el puerto serie 2 para comunicarse con la pantalla.

Si quieres usar otro Arduino como el Arduino Uno, deberás modificar un par de parámetros en la librería.

Si decides usar un Arduino Uno, accede al archivo **NexConfig.h** en la carpeta de la librería, comenta la línea

```
#define DEBUG_SERIAL_ENABLE
```

y cambia la línea

```
#define nexSerial Serial2
```

por la línea

```
#define nexSerial Serial
```

De esta manera desactivarás la depuración por puerto serie y podrás conectar la pantalla Nextion al puerto serie de los pines 0 y 1 de Arduino Uno.

Nextion y Arduino mega

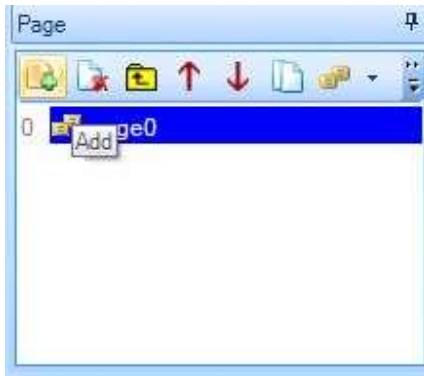
Una vez instalada y configurada la librería, podemos conectar la pantalla a Arduino y de esa manera empezar a controlarla. Para este ejemplo usaré un Arduino Mega 2560. La pantalla la conectaremos en los pines del puerto serie 2.

debemos controlar **n0**, **j0**, y **b0** por eso creamos los siguientes tres objetos, cada uno con su clase correspondiente.

```
NexButton b0 = NexButton(0, 5, "b5");
```

```
NexNumber n0 = NexNumber(0, 3, "n0");
```

```
NexProgressBar j0 = NexProgressBar(0, 2, "j0");
```



Paginas en Nextion Editor

Es muy importante que configuremos bien los constructores de los objetos, es decir, es muy importante que pongamos los parámetros adecuados en los paréntesis.

El primer número corresponde al identificador de la página en la que se encuentra el objeto. En este caso todos los objetos se encuentran en la página 0 y por esto su valor es 0. Puedes ver cual es el identificador de la página fijándote en el cuadro de páginas.

Attribute	
n0(Number)	
id	3
objname	n0
type	54
vscope	local
sta	solid color
style	flat
key	None
bco	<input type="checkbox"/> 65535
pco	<input checked="" type="checkbox"/> 0
font	5
xcen	Left
ycen	Center
val	0
lenth	0
isbr	False
spax	0
spay	0
x	561
y	138
w	80
h	58

Atributos en Nextion Editor

El segundo valor corresponde con el identificador del objeto y este lo podemos encontrar con el nombre **id** en el cuadro de atributos tras seleccionar el objeto haciendo clic sobre él. Por ejemplo en la imagen vemos los atributos del objeto **n0**.

Por último encontramos el tercer parámetro que corresponde con el nombre que se proporcionamos al objeto. Nextion siempre trabaja con identificadores, sin embargo, a los seres humanos nos resulta más fácil acordarnos de nombres, por esto tenemos la posibilidad de darle uno.

Te recomiendo que el nombre del objeto en Arduino sea el mismo que el nombre del objeto en Nextion Editor, de esta manera evitaras confusiones a la hora de trabajar con la pantalla y con el código.

Como puedes ver, usamos una clase para cada tipo de objeto, una clase *NexButton* para los botones, *NexNumber* para los números y *NexProgressBar* para las barras de progresión. Si accedes a la carpeta «Doc» que encontraras dentro de la librería y abres el archivo «index.html» podrás ver toda la documentación sobre la librería y sobre que clases y

métodos existen para comunicarse con cada uno de los elementos que podamos usar en pantalla.

Hay que reconocer que la documentación podría ser un poco más visible, pero, al menos está disponible.

No se te olvide que siempre debes inicializar la pantalla y para eso debes usar `nexInit()`; en el void `setup()`;

```
nexInit();
```

Escribir un número en pantalla

Empezaremos por enviar un número desde Arduino hacia la pantalla para que se muestre en el objeto de tipo número que hemos puesto en la interfaz HMI.

En este caso, como el número que queremos mandar a la pantalla es un valor de temperatura, lo primero que hacemos es leer el valor de la entrada analógica de Arduino y calcular la temperatura.

```
temperatura = (analogRead(A0) * 0.48828)-273;
if(temperatura != temperatura_salida) {
  cuenta++;
} else {
  cuenta = 0;
}
```

Tras calcular el valor que demos enviar, llamaremos al método «`setValue`» del objeto a modificar y le pasaremos el valor a mostrar. En este caso queremos cambiar el valor del objeto de tipo número `n0` por tanto usamos:

```
n0.setValue(temperatura);
```

Ya que estamos también vamos a actualizar el valor de la barra de progreso vertical, para ello, primero calculamos el valor que debemos enviar. Este número debe estar comprendido entre 0 y 100.

Tras tener el valor llamamos al método «`setValue`» pero del objeto correspondiente, es decir `j0`.

```
progress = map(temperatura, -10, 50, 0, 100);
j0.setValue(progress);
```

Leer cambios de un botón.

Este proceso es un poco más largo. Primero debemos crear (antes del setup) una lista con todos los botones de los que el sistema debe estar pendiente. Para ellos introducimos los nombres de los botones en una lista como esta:

```
NexTouch *nex_listen_list[] =
{
    &b0,
    NULL
};
```

Luego pasaremos a crear una función la cual se ejecutará cuando el botón sea pulsado. En este caso la función se llama *cambiarLED* y lo que hace es cambiar el estado del LED, si estaba encendido lo apaga y si estaba apagado lo enciende.

```
void cambiarLED(void *ptr) {
    digitalWrite(13, !digitalRead(13));
}
```

En el **void setup** debemos asociar cada botón con su función correspondiente, en este caso, el botón **b0** con la función **cambiarLED**.

```
b0.attachPop(cambiarLED, &b0);
```

Y por último, debemos indicar a Arduino que esté atento a los cambios que se produzcan en el panel táctil de la pantalla. Esto lo haremos con la función *nexLoop()*; dentro del **void loop**

```
nexLoop(nex_listen_list);
```

Con todo esto, cuando se presione el botón en la pantalla táctil, Arduino ejecutará la función que hemos asociado a ese botón.

CODIGO COMPLETO:

```
#include "Nextion.h"

NexButton b0 = NexButton(0, 5, "b5");

NexNumber n0 = NexNumber(0, 3, "n0");

NexProgressBar j0 = NexProgressBar(0, 2, "j0");

float temperatura = 0.0;
```

```
float temperatura_salida = 0.0;

int cuenta = 0;

int progress = 0;


NexTouch *nex_listen_list[] =

{

&b0,

NULL

};


void cambiarLED(void *ptr) {

digitalWrite(13, !digitalRead(13));

}


void setup() {

nexInit();

b0.attachPop(cambiarLED, &b0);

pinMode(13, OUTPUT);

}


void loop() {
```

```
nexLoop(nex_listen_list);

temperatura = (analogRead(A0) * 0.48828)-273;

if(temperatura != temperatura_salida) {

cuenta++;

} else {

cuenta = 0;

}

if(cuenta == 5){

temperatura_salida = temperatura;

cuenta = 0;

n0.setValue(temperatura);

progress = map(temperatura, -10, 50, 0, 100);

j0.setValue(progress);

}

delay(100);

}
```