

Pantalla Nextion 3.2" con Arduino.

NEXTION LCD es una pantalla táctil que proporciona una interfaz de control y visualización entre un humano y un proceso, una máquina, una aplicación o un dispositivo. Esta interfaz de pantalla táctil analógica cuenta con botones de función programables, medidores, barras de proceso, etc.

El editor de Nextion proporciona componentes masivos, como botones, textos, barras de progreso, controles deslizantes, paneles de instrumentos, etc., todo esto para enriquecer el diseño de la interfaz y, con la función de arrastrar y soltar asegura que los usuarios pasarán menos tiempo en la programación, lo que reducirá sus cargas de trabajo de desarrollo del 99%. La pantalla de Nextion es sólo la opción para resolver sus problemas de programación con el mínimo de tiempo y esfuerzo, ya que los usuarios pueden ver sus ventajas competitivas por no mencionar su buen precio

Esta pantalla también es ideal para ocuparse con algún tipo de microcontrolador; ya sea Arduino, microchip, AVR, etc. porque la mayoría de sus funcionalidades y procesos son independientes en la pantalla. La comunicación con un Arduino es a través de un puerto UART serie. La ventaja es que el Arduino no utiliza muchos recursos o pines que se ocupan de una pantalla táctil de alta resolución, simplemente envía comandos en serie a la pantalla o recibe notificaciones de eventos como pulsaciones de botones.

En la figura N° 1 se puede observar una implementación de la pantalla Nextion para el control de sensores o el encendido de leds, entre otras aplicaciones más, cabe mencionar que dicha pantalla es ideal para remplazar el tradicional LCD y el tubo LED NIXIE. Además de que esta pantalla también se puede usar sin algún microcontrolador y se puede programar directo con un cable USB a TTL.

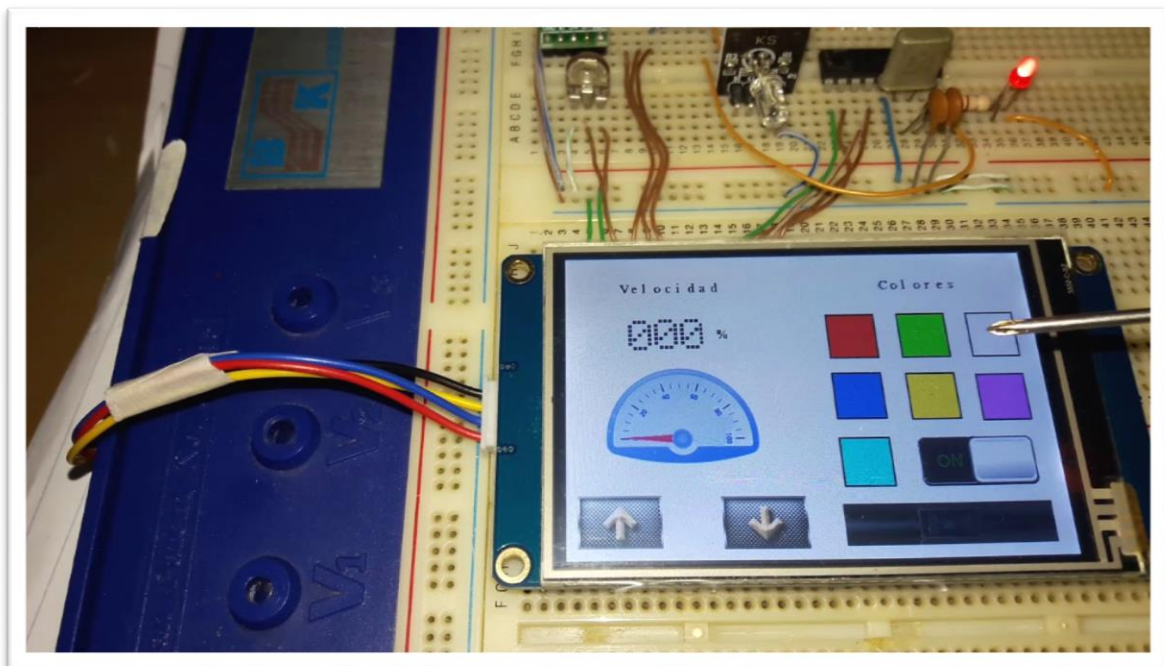


Figura N° 1: Pantalla Nextion e implementación.

Material Utilizado.






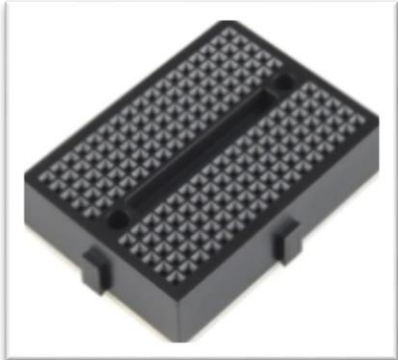
Electrónico		
<p>Pantalla Touch 3.2".</p>  <p>SKU: MO0083</p>	<p>Arduino Uno RV3.</p>  <p>SKU: AR0016</p>	<p>Dupont M/H de 20 cm.</p>  <p>SKU: CS0006</p>
<p>Cable USB/TTL.</p>  <p>SKU: CA0006</p>	<p>Led Morado Ultrabrillante de 5 mm.</p>  <p>SKU: LD0012</p>	<p>Protoboard Mini.</p>  <p>SKU: PT0001</p>

Diagrama de Conexión.

Para probar la pantalla se realizó la conexión con un cable USB a TTL como se puede ver en la figura N° 2 y se describe más adelante. El cable USB cuenta con cuatro conexiones:

- **5V.** Voltaje de alimentación.
- **GND.** Conexión a tierra.
- **TX.** Transmite datos.
- **RX.** Recibe datos.

En la figura N° 3 muestra las conexiones que se hicieron de la pantalla al Arduino uno y del Arduino a una Protoboard mini.

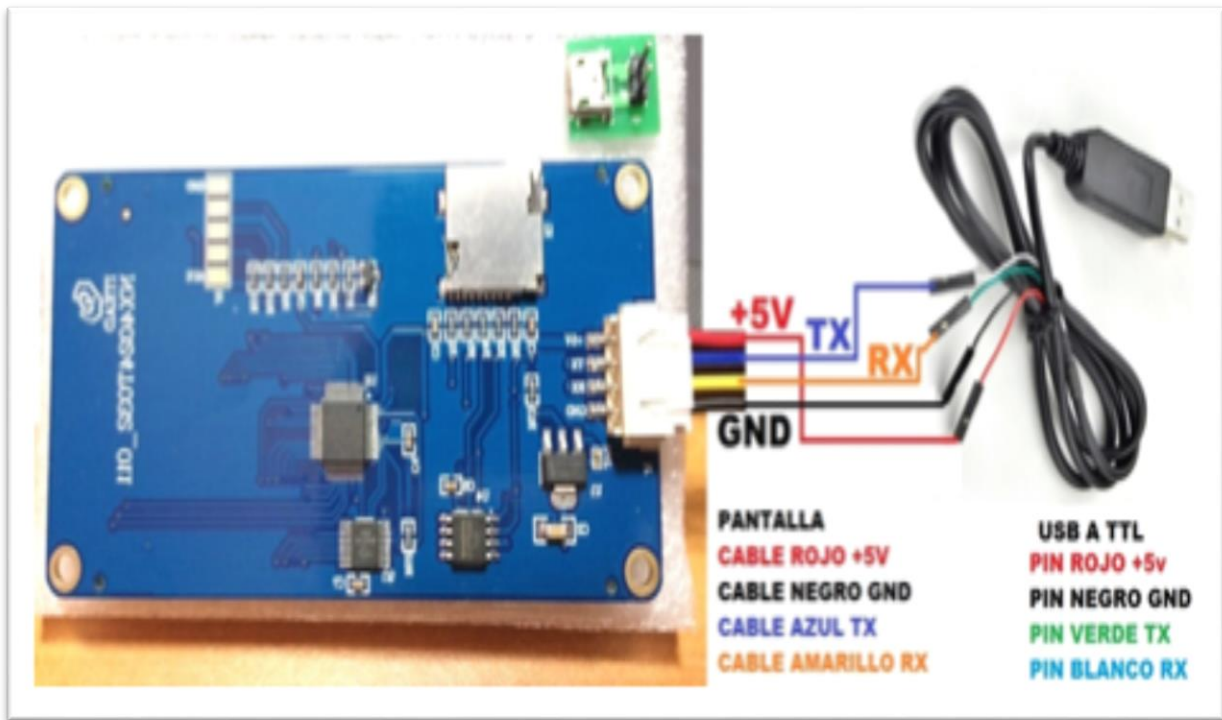


Figura N° 2: Pantalla táctil y USB a TTL con sus conexiones

Conexión Pantalla con Cable USB-TTL.

- Conecte el pin 5V de la pantalla a 5V del cable USB/TTL
- Conecte el pin GND de la pantalla a GND del cable USB/TTL
- Conecte el pin TX de la pantalla a TX del cable USB/TTL
- Conecte el pin RX de la pantalla a RX del cable USB/TTL
- Conecte el USB a la computadora.

Conexión Pantalla con Arduino.

- Conecte el pin 5V de la pantalla a 5V del Arduino Uno.
- Conecte el pin GND de la pantalla a GND del Arduino Uno.
- Conecte el pin RX de la pantalla a TX del Arduino Uno.
- Conecte el pin TX de la pantalla a RX del Arduino Uno.
- Conecte el pin digital 8 del Arduino uno al cátodo (+) del Led.
- Conecte el pin GND del Arduino uno a una terminal de una resistencia de 330Ω la cual va conectada al ánodo (-) del led.

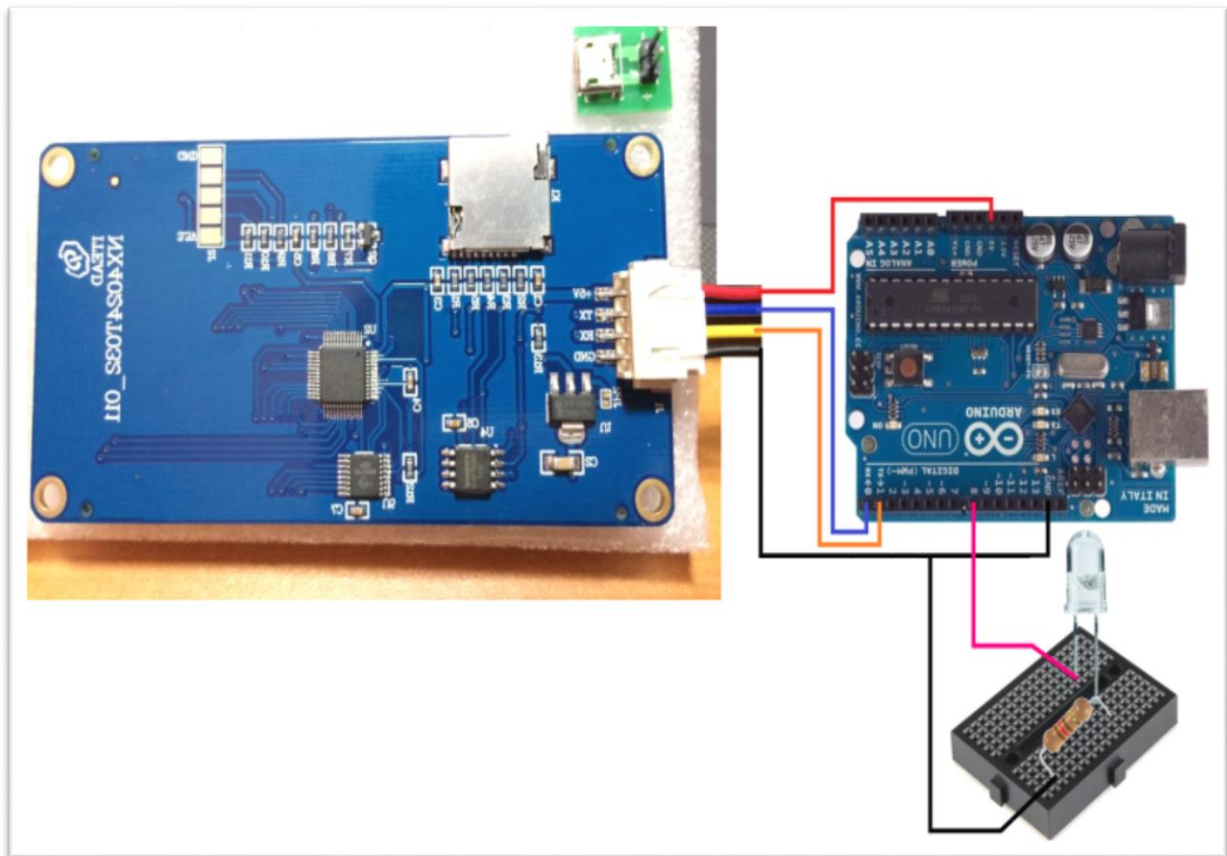


Figura N° 3: Pantalla táctil al Arduino Uno a Protoboard mini con led.

Código Usado.

Este es el código que se usó en la tarjeta Arduino para el funcionamiento la pantalla Nextion de 3.2". Para programar es necesario contar con el programa IDE de Arduino.

```
int led=13;
int char1, char2;
int charflag=0;
int pinLed = 8;

void setup()
{
  Serial.begin(9600);           //configuración del puerto serial
  pinMode(led, OUTPUT);
  pinMode(pinLed, OUTPUT);
}

void loop()
{
  if(Serial.available()>0)
  {
    if(charflag)
    {
      char2= Serial.read();      //lee el puerto serial y almacena el dato en char2
    }
  }
}
```

```

if(char2==1)
{
    digitalWrite(13, HIGH);
    digitalWrite(pinLed, HIGH);
}
else if(char2==0x00)
{
    digitalWrite(13, LOW);
    digitalWrite(pinLed, LOW);
}
charflag=0;
Serial.print(char2);
}

else
{
    char1=Serial.read();
    if(char1==0x65)
    {
        charflag=1;
    }
}
}
}

```

Programación Pantalla Táctil.

Al igual que Arduino, la pantalla cuenta con una plataforma de programación, una vez instalada la plataforma, ésta se visualizará como en la figura N° 4.

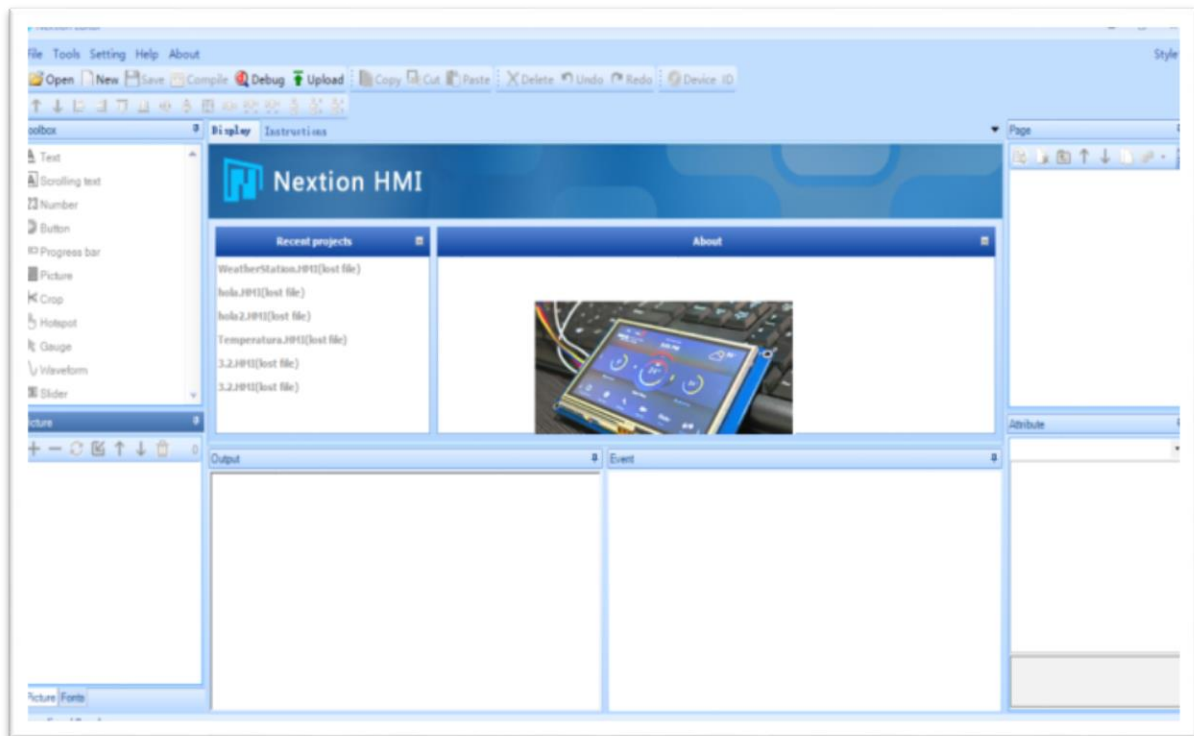


Figura N° 4: Plataforma de programación.

Una vez dentro se procede a abrir un nuevo archivo el cual se le asignará un nombre y en qué lugar se desea guardar, después saldrá una ventana que permitirá elegir la pantalla Nextion que se vaya a utilizar: 3.2”, 2.8”, 2.4”, etc. Ver figura N° 5.

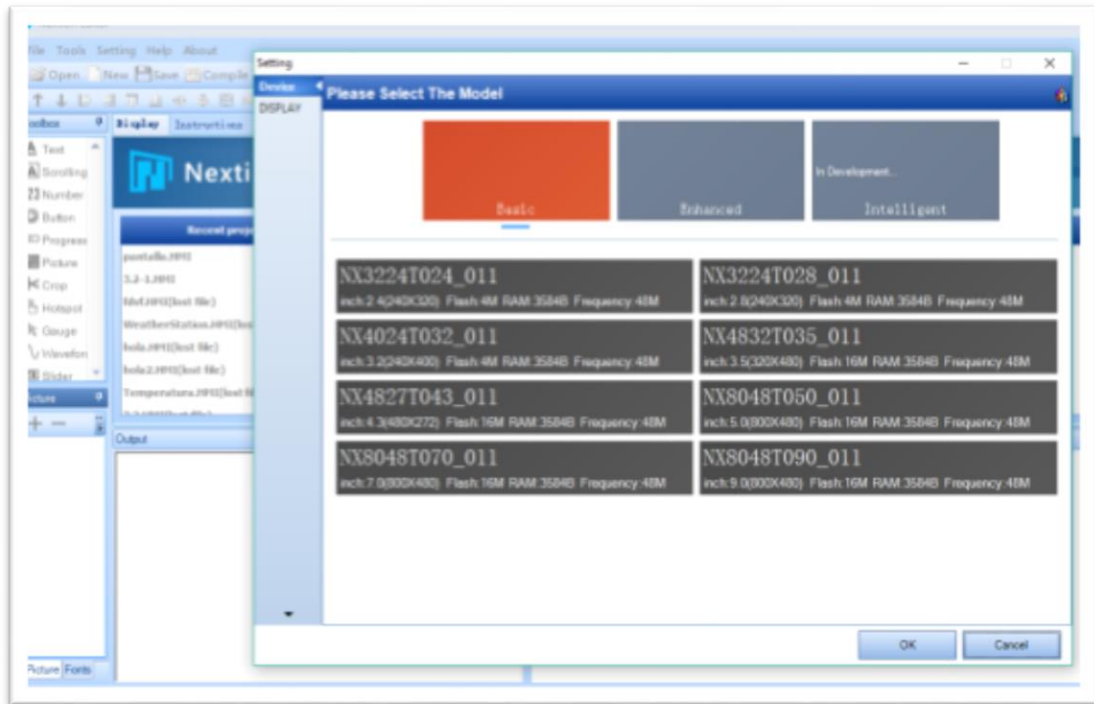


Figura N° 5: Selección de tipo de pantalla.

Después de haber seleccionado el tipo de pantalla que se va a utilizar se despliega una ventana con la orientación en la cual se trabajara la pantalla, como se muestra en la figura N° 6.

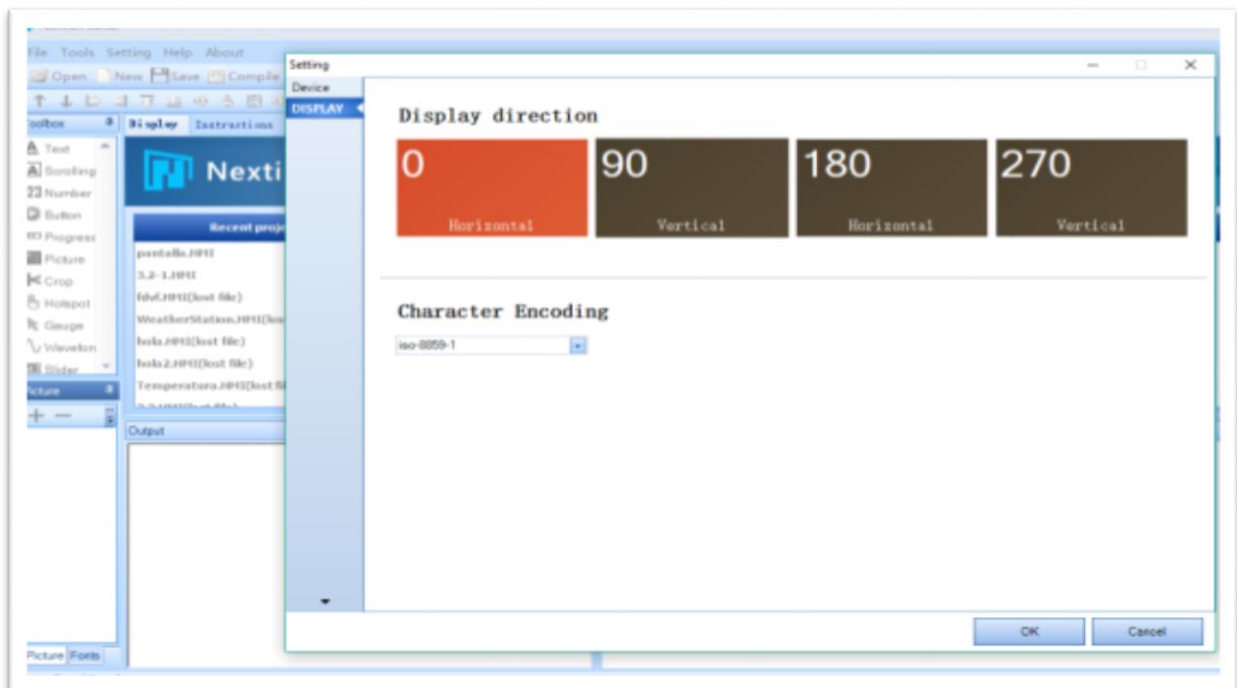


Figura N° 6: Selección de la orientación de la pantalla.

Una vez elegido los parámetros en los que se trabajara la pantalla se colocan las paginas que se usaran; por ejemplo, para realizar la función de encendido y apagado de un led se usaran 2 paginas: la página cero será el botón de encendido mientras que la página uno será el botón de apagado; para ello habrá que configurar datos importantes como el fondo de pantalla, botones, etc. Para colocar el fondo de pantalla se precargan las imágenes a usar y se colocan como se tenía previsto. En la figura N° 7 se puede ver la configuración de las paginas que se crean.

Nota: Las imágenes que se agreguen deberán estar en un tamaño de 400*240.

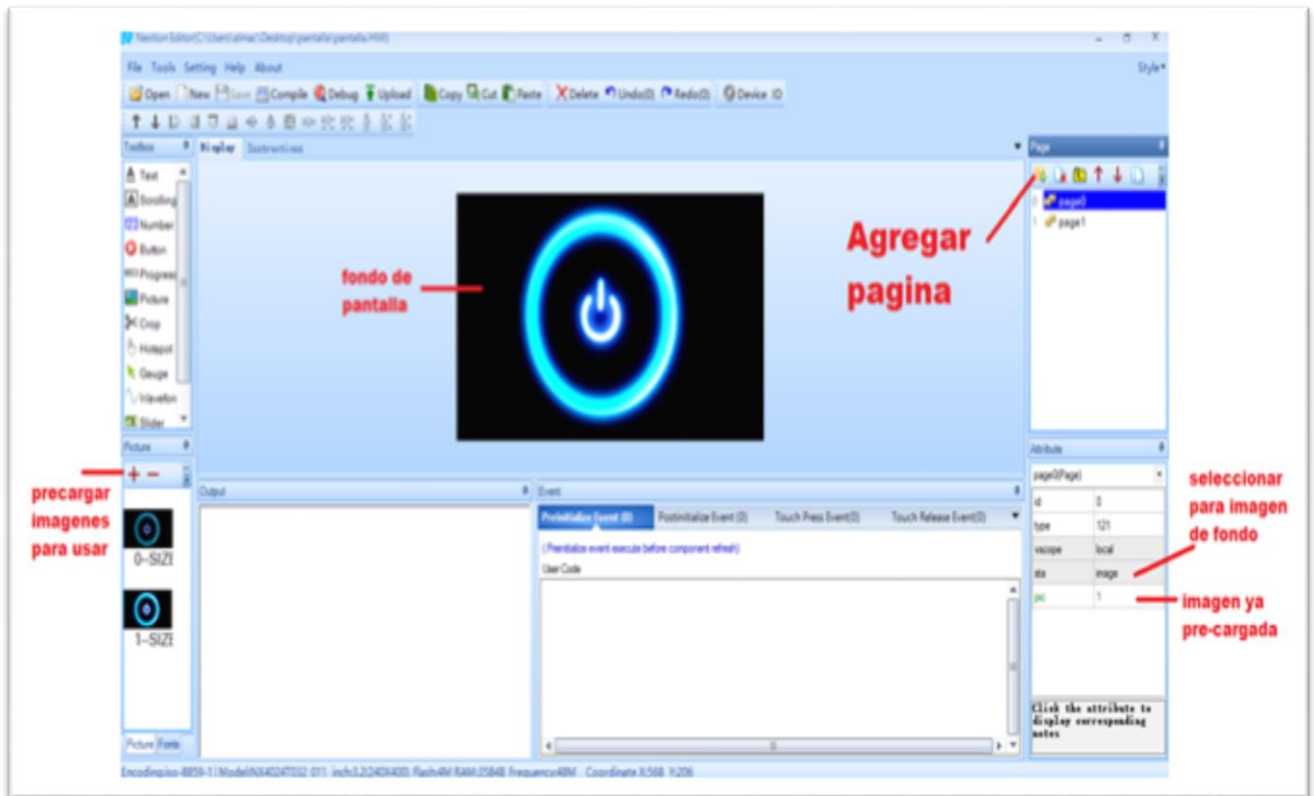


Figura N° 7: Configuración de páginas.

Ya que se han configurados los fondos de pantallas de las dos páginas a utilizar, se procede a agregar los botones en ambas páginas. Se agrega un botón en la página cero y se configuran las características del botón, en la figura N° 8 se muestran las características del botón sin modificar.

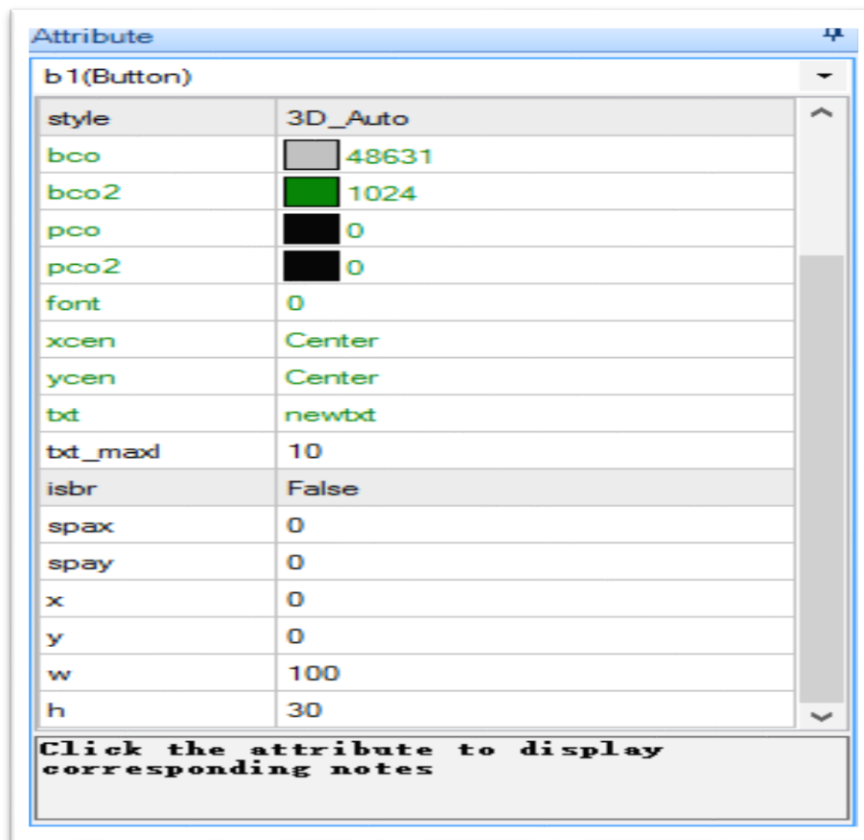


Figura N° 8: Características del botón recién agregado

Una vez agregado el botón se configuran las características, se agregan las imágenes que dan el efecto de encendido y apagado en la pantalla, también se ajusta el tamaño. Es importante que en las características de PIC se seleccione la imagen de botón encendido y en la característica PIC2 se seleccione la imagen del botón apagado.

Para poder mandar datos al Arduino nos vamos a la pestaña “Touch Release Event()” en la cual seleccionaremos “Send component ID” y en la sección “User code” se agrega el código “page 1” para que al momento de presionar el botón se cambie a la página correspondiente. En la figura N° 9 se puede visualizar los cambios que se acaban de describir.

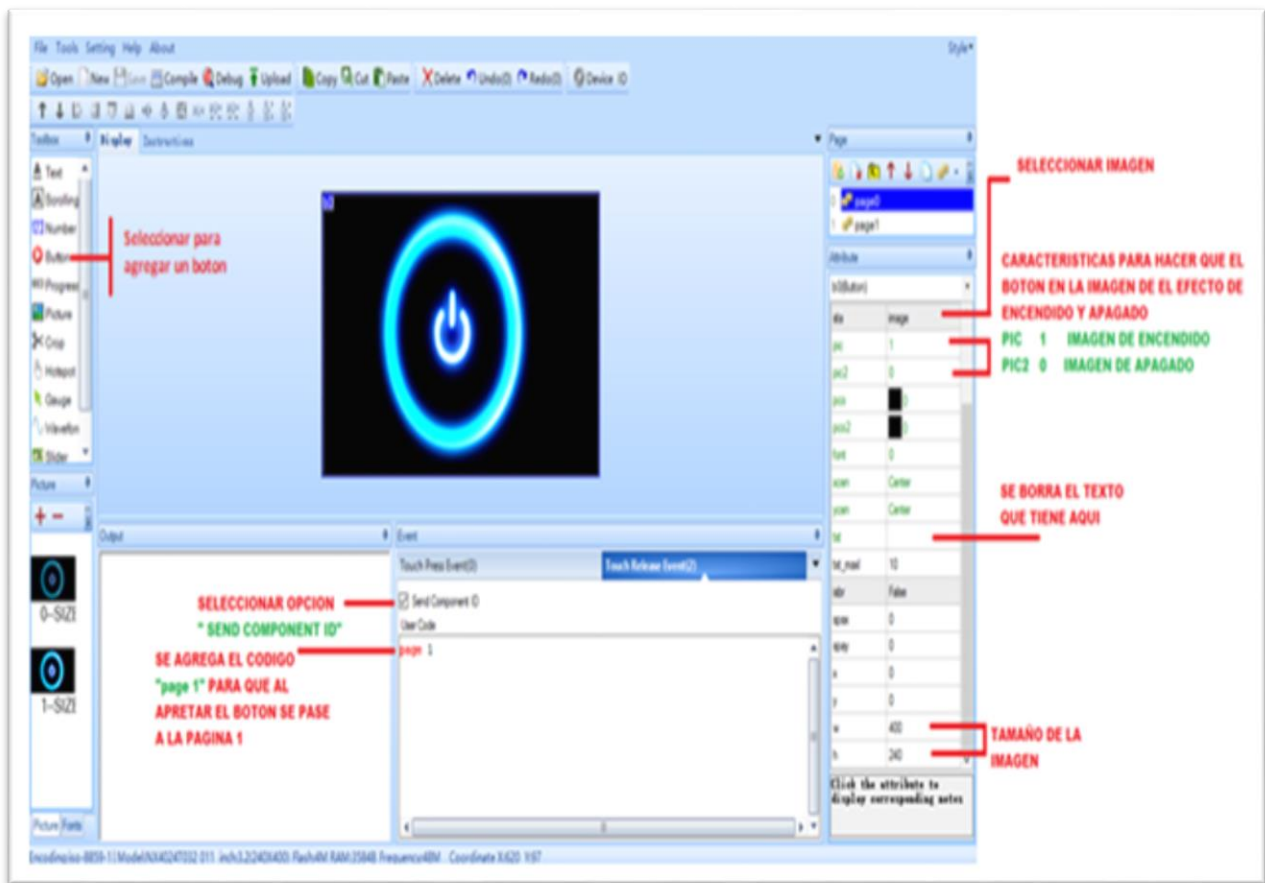


Figura N° 9: Configuración del botón en la Página 0.

Se procede a ir a la página uno y se agrega otro botón y se configuran las características, se agregan las imágenes que dan el efecto de apagado y encendido en la pantalla y también se ajusta el tamaño. Es importante que en las características de PIC se seleccione la imagen de botón apagado y en la característica PIC2 se selecciona la imagen del botón encendido, es decir, viceversa de la pagina cero.

Para poder mandar datos al Arduino nos vamos a la pestaña "Touch Release Event()" en la cual seleccionaremos "Send component ID" y en la sección "User code" se agrega el código "page 0" para que al momento de presionar el botón se cambie a la página 0. En la figura N° 8 podemos visualizar los cambios que se acaban de describir.

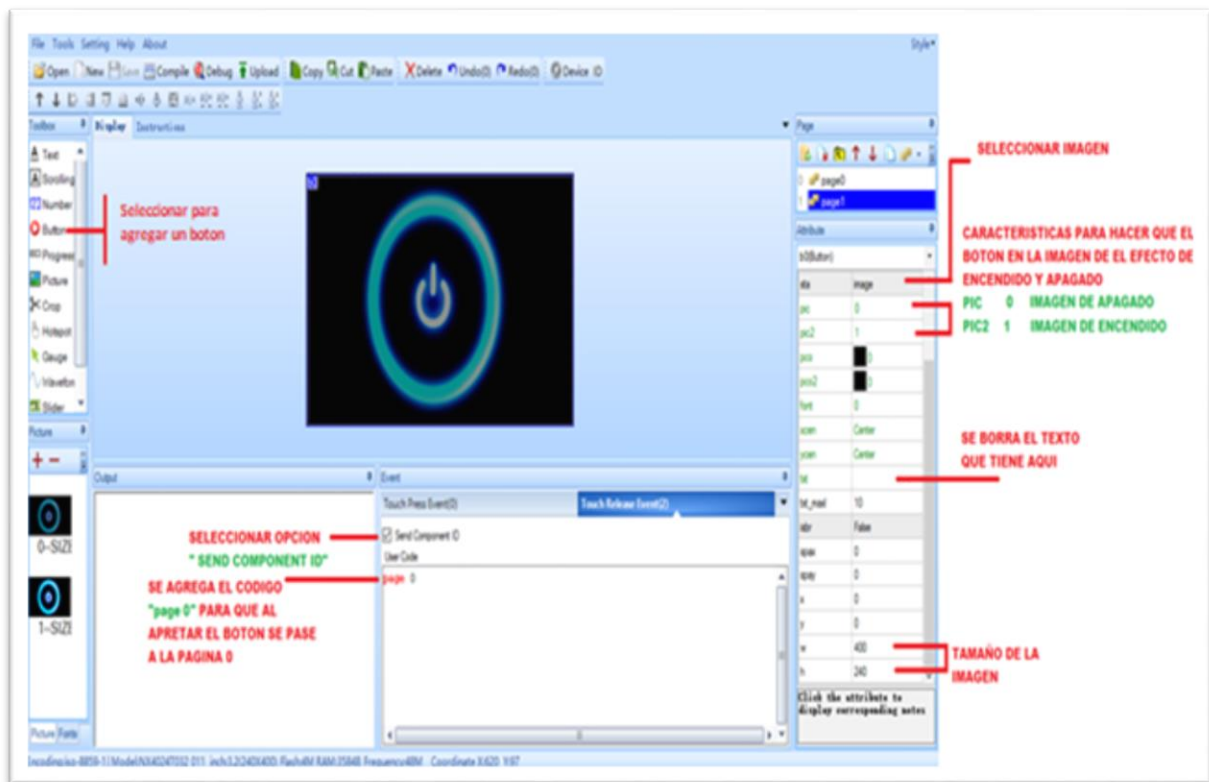


Figura N° 10: Configuración del botón en la Página 1.

Nota: Se requiere agregar una fuente para que el programa pueda funcionar sin errores, en la figura N° 11 se muestra como generar una fuente con el formato “.zi”. Si tienes alguna fuente en el formato indicado solo es necesario agregarla como se muestra en el paso 4 de la figura N° 11.

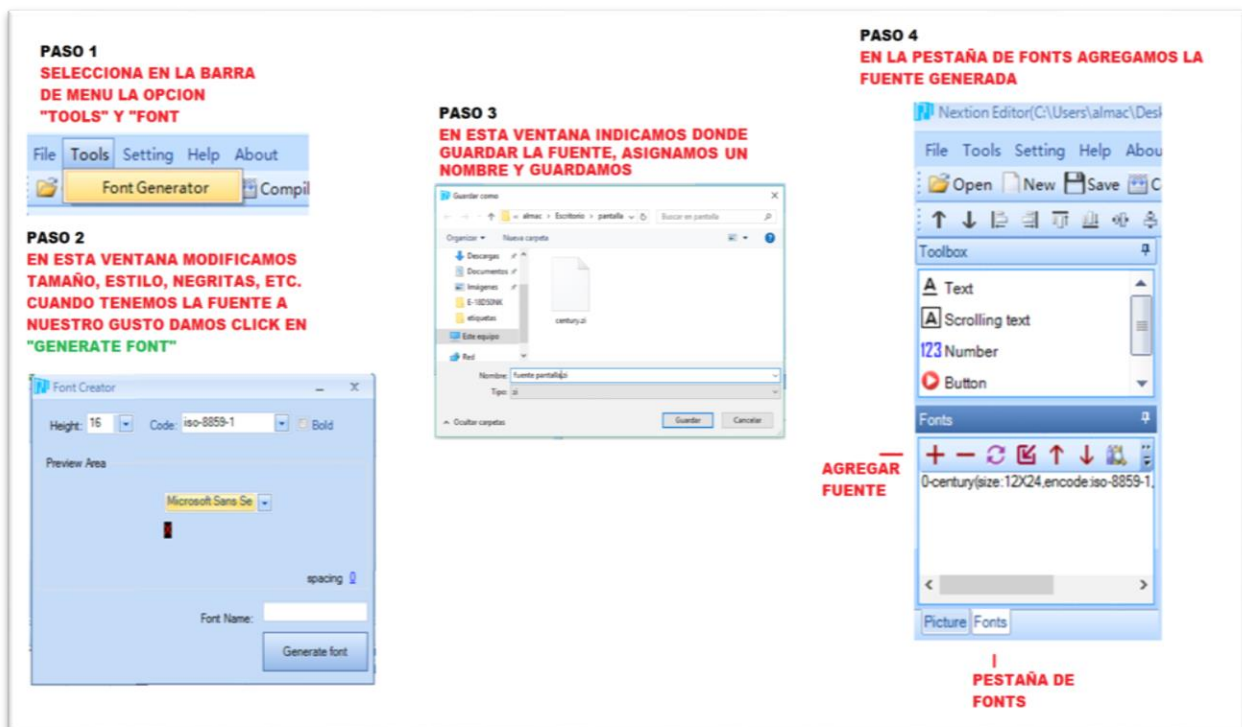


Figura N° 11: Pasos para generar una fuente.

Simulación.

Una vez finalizada la programación de la pantalla se puede corroborar el funcionamiento antes de cargar el programa a la pantalla, solo es necesario compilarlo y luego presionar la opción de “debug” en la figura N° 12 se muestra la simulación ejecutándose.

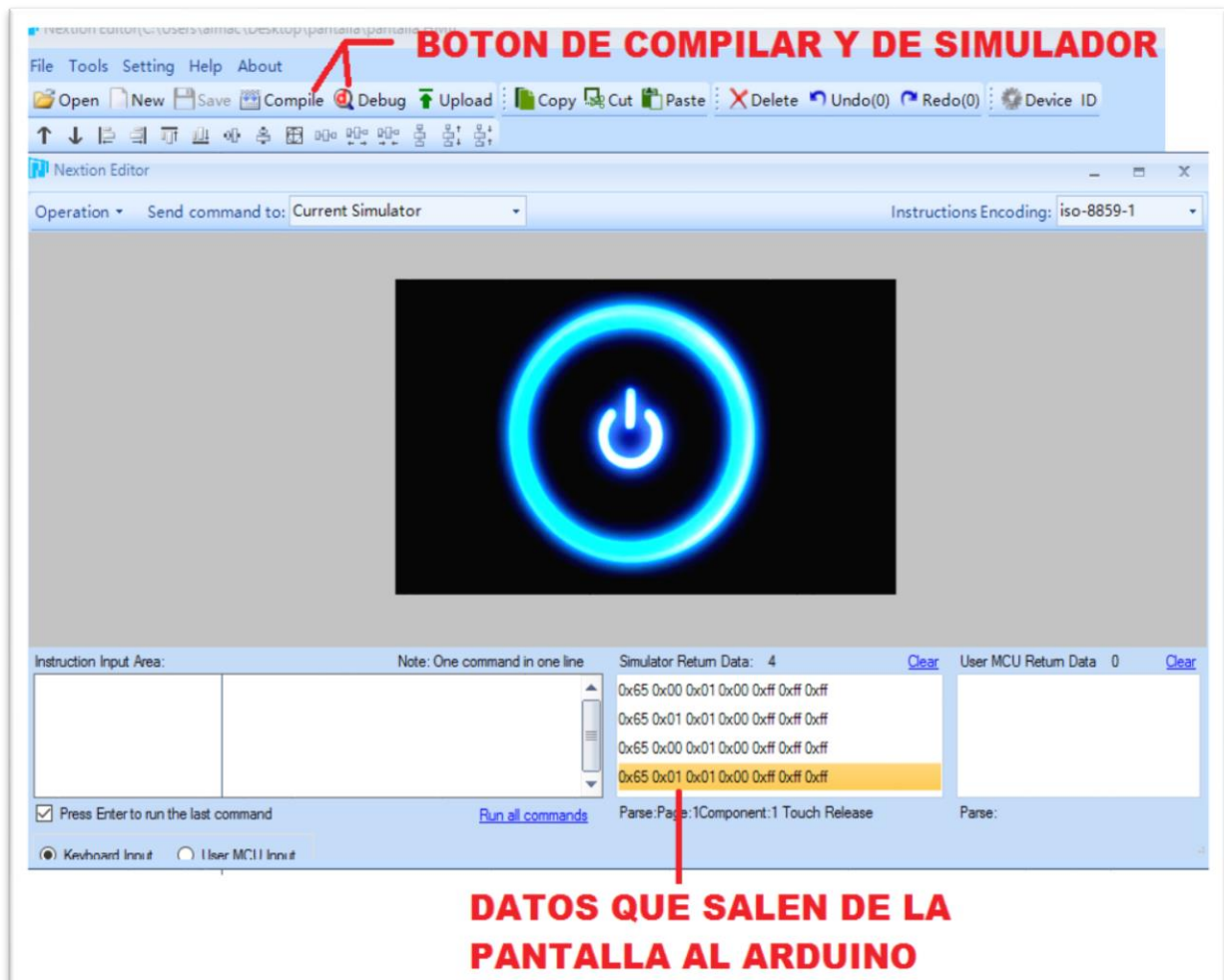


Figura N° 12: Ventana de simulación.

Imágenes de Funcionamiento.

En la figura N° 13 se puede ver la página inicial donde el botón esta encendido y por lo tanto el led esta encendido.

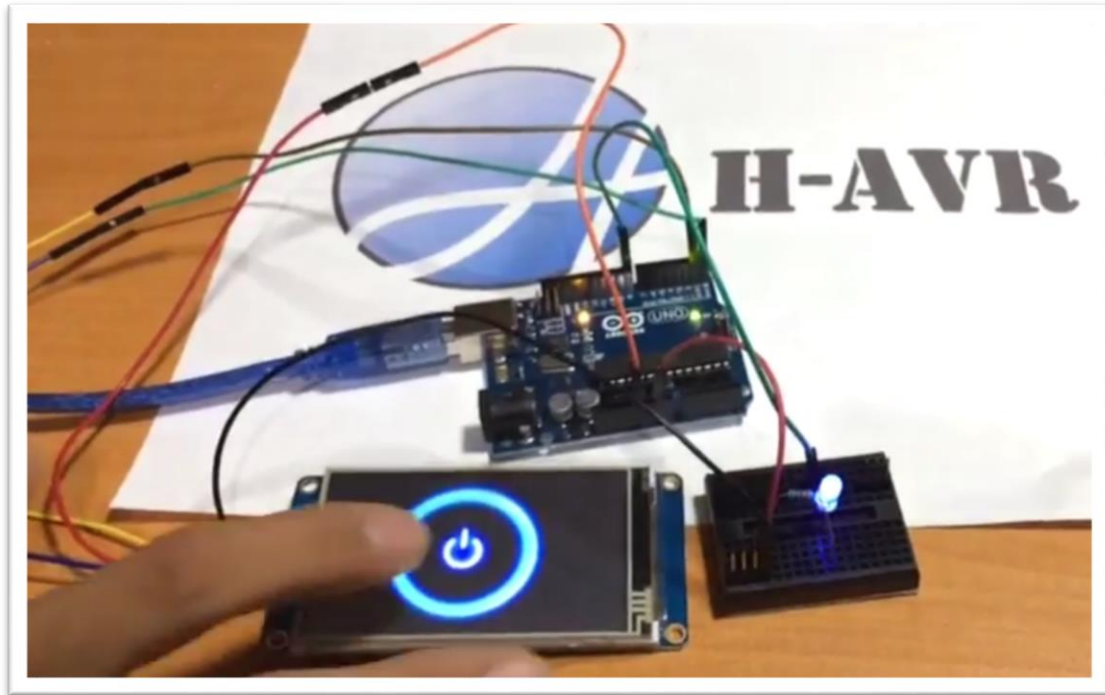


Figura N° 13: Pantalla Nextion 3.2" en la pantalla inicial programada.

En la figura N° 14 se puede ver la segunda página donde el botón se encuentra apagado y el led también se encuentra apagado.

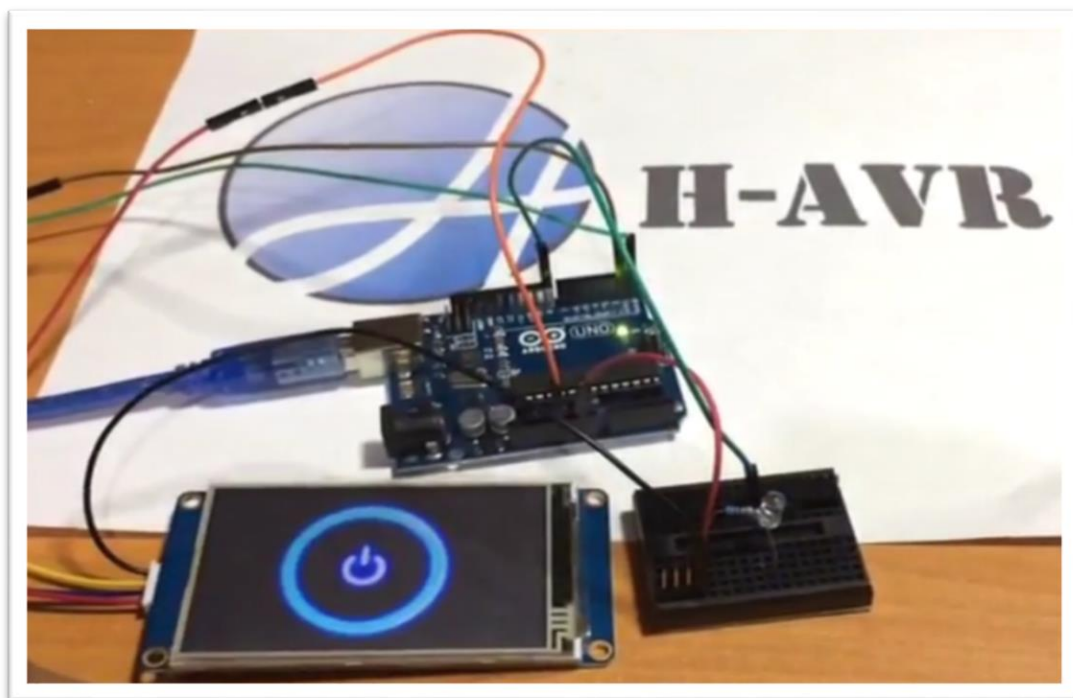


Figura N° 14: Pantalla Nextion 3.2" en la 2da pantalla programada.

Conclusiones.

Esta pantalla táctil demuestra la facilidad al programar teniendo una interfaz amigable para el usuario, ya que es compatible con tarjetas de programación como lo es la plataforma de Arduino o Raspberry ya que las conexiones a realizar son mínimas.

También se puede observar que los comandos de cadena de caracteres reducen la dificultad de depuración, así como el método de control orientado a objetos puede soportar los comandos no solo para dibujar líneas o puntos, sino que también soporta la mayoría de los atributos de los componentes y materiales. Con esto el usuario puede controlar los objetos en una demostración a través de comandos de puerto serie por MCU externo.

Nota: Verificar la hoja de especificaciones para más información.

Contacto.

- <http://www.h-avr.mx/>

Video del Funcionamiento.

- <https://www.youtube.com/watch?v=QQAH-rlMyEs>
- <https://www.youtube.com/watch?v=SyLrP9ZxpxY>

Descarga de la Plataforma Nextion.

- <https://nextion.itead.cc/resources/download/nextion-editor/>

Hoja De Especificaciones.

- <https://www.itead.cc/wiki/NX4024T032>

Donde Comprar:



**mercado
libre**

