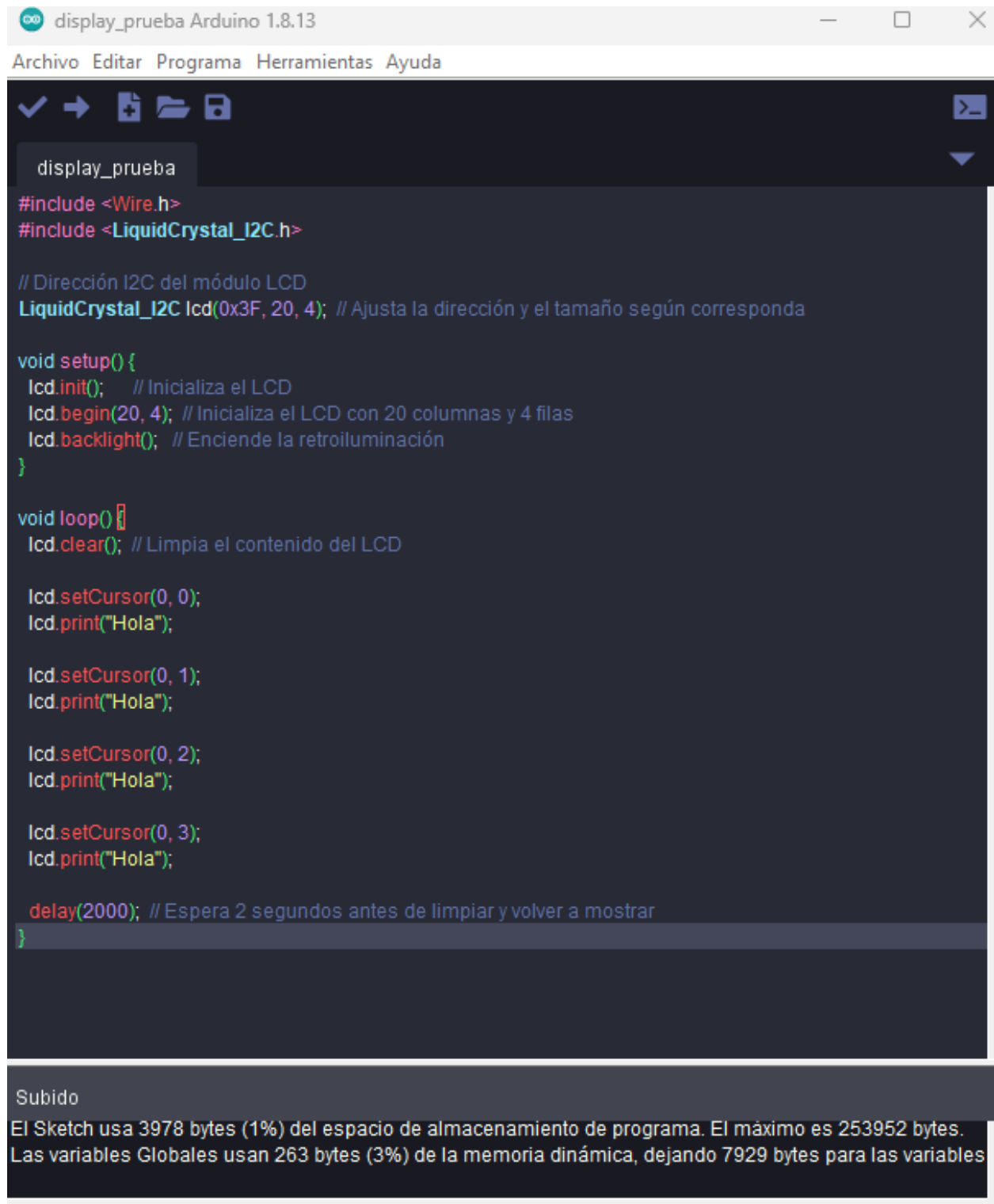


TUTORIAL PROBLEMAS DISPLAY 20x4 – 16x2

Antes que nada debemos recordar que los display pueden tener distintos tipos de direcciones, por ejemplo los displays azules tiene generalmente una dirección 0X3F, mientras que los displays verdes tienen generalmente una dirección 0x27.

En la imagen de abajo vemos el caso de un código que imprime en las 4 líneas del display “hola”, y que es un display de color azul.



```
display_prueba Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

display_prueba
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Dirección I2C del módulo LCD
LiquidCrystal_I2C lcd(0x3F, 20, 4); // Ajusta la dirección y el tamaño según corresponda

void setup() {
  lcd.init(); // Inicializa el LCD
  lcd.begin(20, 4); // Inicializa el LCD con 20 columnas y 4 filas
  lcd.backlight(); // Enciende la retroiluminación
}

void loop() {
  lcd.clear(); // Limpia el contenido del LCD

  lcd.setCursor(0, 0);
  lcd.print("Hola");

  lcd.setCursor(0, 1);
  lcd.print("Hola");

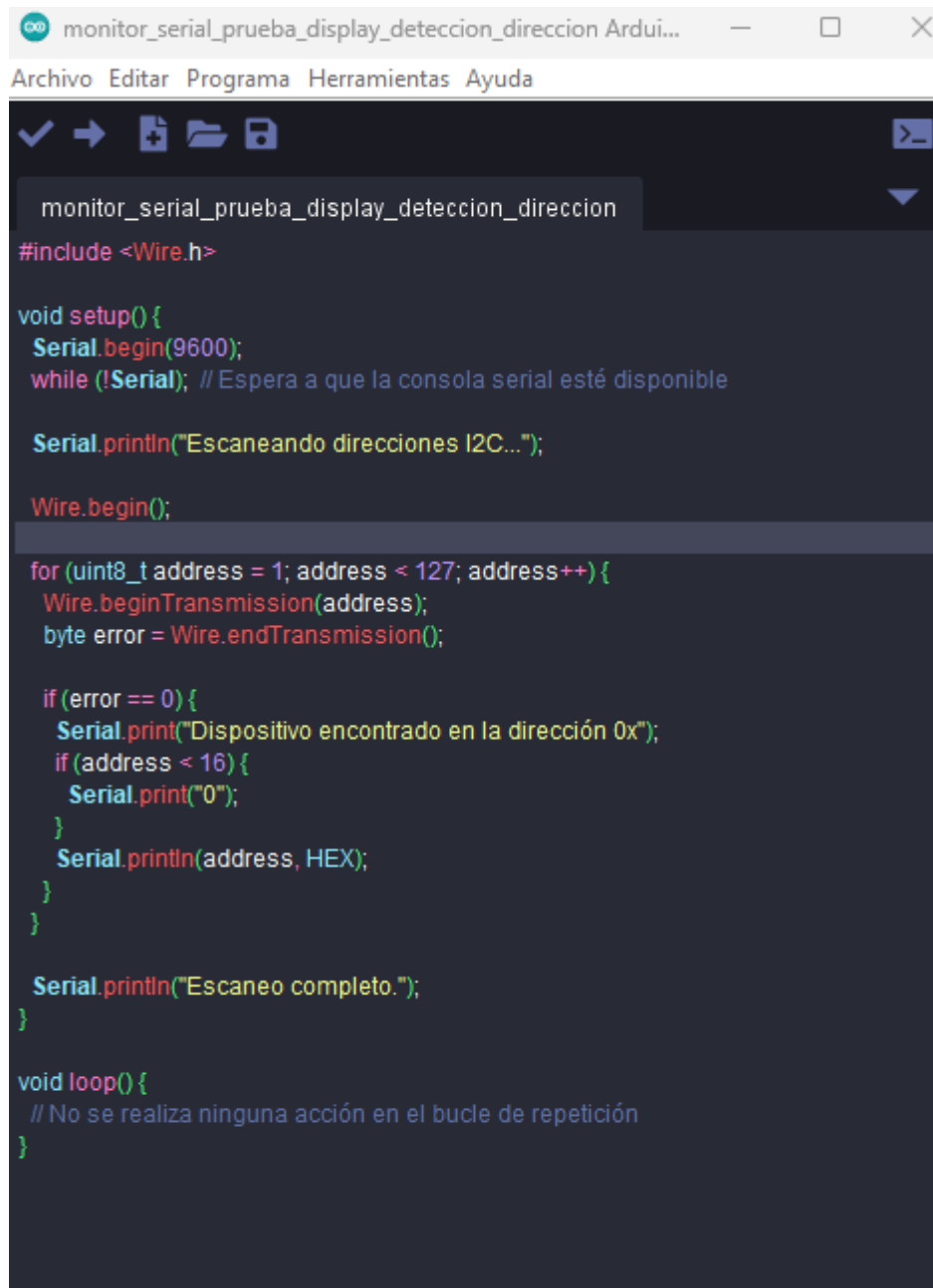
  lcd.setCursor(0, 2);
  lcd.print("Hola");

  lcd.setCursor(0, 3);
  lcd.print("Hola");

  delay(2000); // Espera 2 segundos antes de limpiar y volver a mostrar
}

Subido
El Sketch usa 3978 bytes (1%) del espacio de almacenamiento de programa. El máximo es 253952 bytes.
Las variables Globales usan 263 bytes (3%) de la memoria dinámica, dejando 7929 bytes para las variables
```

Una buena forma de probar en qué dirección está conectado el display es hacer un chequeo de los pines del arduino. El siguiente código chequea los pines y muestra en el monitor serie del arduino en que Address se encuentra conectada



```
monitor_serial_prueba_display_deteccion_direccion Ardui...
Archivo Editar Programa Herramientas Ayuda

monitor_serial_prueba_display_deteccion_direccion
#include <Wire.h>

void setup() {
  Serial.begin(9600);
  while (!Serial); // Espera a que la consola serial esté disponible

  Serial.println("Escaneando direcciones I2C...");

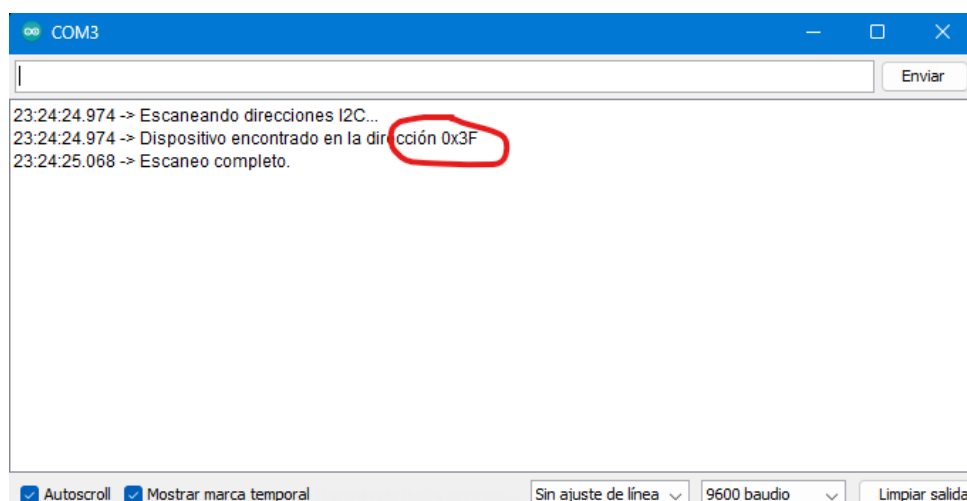
  Wire.begin();

  for (uint8_t address = 1; address < 127; address++) {
    Wire.beginTransmission(address);
    byte error = Wire.endTransmission();

    if (error == 0) {
      Serial.print("Dispositivo encontrado en la dirección 0x");
      if (address < 16) {
        Serial.print("0");
      }
      Serial.println(address, HEX);
    }
  }

  Serial.println("Escaneo completo.");
}

void loop() {
  // No se realiza ninguna acción en el bucle de repetición
}
```



```
COM3
|
| Enviar
23:24:24.974 -> Escaneando direcciones I2C...
23:24:24.974 -> Dispositivo encontrado en la dirección 0x3F
23:24:25.068 -> Escaneo completo.

Autoscroll Mostrar marca temporal Sin ajuste de línea 9600 baudio Limpiar salida
```

Problemas con el display:

- 1) Algunos de los problemas se pueden deber a la conexión de los cables. Por ejemplo si los cables estuvieran bien conectados SCL y SDA en los correspondientes pines y el cheque no muestra ninguna dirección, como en la imagen de arriba, pues entonces puede que algún cable este haciendo falso contacto
- 2) Chequear que el brillo de la pantalla este regulado adecuadamente.
- 3) muy posiblemente se encuentre con el problema de que, estando todo conectado, habiendo configurado la dirección correctamente, y el display está encendido y aun así solo muestra las líneas vacías como la imagen de abajo.



En este caso el problema está relacionado con el software seguramente, falta inicializar el display, lo que sucede es que muchos display ya implícitamente se inicializan, pero en otros no, así que la siguiente línea de código `"lcd.init()"` debería ir antes del `begin`.

El código se vería así:



display_prueba

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Dirección I2C del módulo LCD
LiquidCrystal_I2C lcd(0x3F, 20, 4); // Ajusta la dirección y el tamaño según corresponda

void setup() {
  lcd.init(); // Inicializa el LCD
  lcd.begin(20, 4); // Inicializa el LCD con 20 columnas y 4 filas
  lcd.backlight(); // Enciende la retroiluminación
}

void loop() {
  lcd.clear(); // Limpia el contenido del LCD

  lcd.setCursor(0, 0);
  lcd.print("Hola");

  lcd.setCursor(0, 1);
  lcd.print("Hola");

  lcd.setCursor(0, 2);
  lcd.print("Hola");

  lcd.setCursor(0, 3);
  lcd.print("Hola");

  delay(2000); // Espera 2 segundos antes de limpiar y volver a mostrar
}
```

Subido

El Sketch usa 3978 bytes (1%) del espacio de almacenamiento de programa. El máximo es 253952 bytes.
Las variables Globales usan 263 bytes (3%) de la memoria dinámica, dejando 7929 bytes para las variables