

Display Custom Text on a 20x4 LCD with ESP32

Objective:

To interface a **20x4 LCD** with the **ESP32 microcontroller** using the **LiquidCrystal library** and display a **personal message** across all four rows.

Required Components:

- ESP32 development board
- 20x4 LCD (HD44780 compatible)
- 10K potentiometer (for contrast control)
- Breadboard & jumper wires
- Optional: 220Ω resistor for LCD backlight

Pin Configuration (LCD to ESP32):

The LCD uses 6 digital pins to communicate in 4-bit mode.		
LCD Pin	Function	Connected to ESP32 GPIO
RS	Register Select	GPIO 21
E	Enable	GPIO 22
D4	Data line 4	GPIO 18
D5	Data line 5	GPIO 19
D6	Data line 6	GPIO 23
D7	Data line 7	GPIO 5
Note: VSS → GND, VDD → 5V, RW → GND, VO → Potentiometer center pin Backlight pins (A & K) can be connected to 5V and GND via a resistor.		

How the LCD Works (HD44780):

- The LCD has **20 columns and 4 rows**.
- You use commands to set the **cursor** and send characters.
- In 4-bit mode, data is sent in **nibbles** (4-bit chunks) using D4–D7.
- **RS pin:** Selects whether you're sending a command (RS=0) or data (RS=1).
- **E pin:** Acts as a **latch** – data is latched on the **rising edge** of this signal.

Code Explanation

```
#include <Arduino.h>

#include <LiquidCrystal.h>

// Pin mapping: RS, E, D4, D5, D6, D7

LiquidCrystal lcd(21, 22, 18, 19, 23, 5);

void setup() {

    lcd.begin(20, 4);          // 20 columns, 4 rows

    lcd.setCursor(0, 0);      // column 0, row

    lcd.print("Hello, Guys!");

    lcd.setCursor(0, 1);      // column 0, row 1

    lcd.print("I'm Gurucharan.G.R");

    lcd.setCursor(0, 2);      // column 0, row 1

    lcd.print("Let's connect and");

    lcd.setCursor(0, 3);      // column 0, row 1

    lcd.print("        grow....");

}

void loop() {

    // You can update text here

}
```

(Line by Line)

1. Include Required Libraries

```
#include <Arduino.h>
#include <LiquidCrystal.h>
```

- **Arduino.h**: Core Arduino functionality (e.g., `pinMode`, `delay`).
- **LiquidCrystal.h**: Provides functions for controlling HD44780 LCDs.

2. Initialize LCD Object

```
LiquidCrystal lcd(21, 22, 18, 19, 23, 5);
```

- Creates an LCD object using **6 ESP32 GPIO pins**.
- The order is:
 - RS = GPIO 21
 - E = GPIO 22
 - D4 = GPIO 18
 - D5 = GPIO 19
 - D6 = GPIO 23
 - D7 = GPIO 5
- This sets up communication in **4-bit mode**, saving GPIOs.

3. Setup Function – LCD Initialization

```
void setup() {
```

- This function runs **once** when the ESP32 powers on or resets.

```
    lcd.begin(20, 4);
```

- Initializes the display with **20 columns and 4 rows**.
- Internally sends configuration commands to the LCD controller.

```
    lcd.setCursor(0, 0);    // column 0, row 0
    lcd.print("Hello, Guys!");
```

- Moves cursor to **column 0, row 0** (top-left corner).
- Prints "Hello, Guys!" starting from that position.

```
    lcd.setCursor(0, 1);    // column 0, row 1
    lcd.print("I'm Gurucharan.G.R");
```

- Moves cursor to **row 1** (second row) and prints your name.

```
    lcd.setCursor(0, 2);    // column 0, row 2
    lcd.print("Let's connect and");
```

- Prints the text "Let's connect and" on **row 2**.

```
lcd.setCursor(0, 3);    // column 0, row 1  
lcd.print("    grow....");
```

- Prints "grow...." on **row 3** (bottom row).
- The **leading spaces** indent the text for visual styling.

4. Loop Function

```
void loop() {  
    // You can update text here  
}
```

- The `loop()` is empty here.
- In future, you could add code to **scroll, blink, or change the display text**.

Summary of Operation:

1. ESP32 powers on.
2. LCD is initialized in 4-bit mode.
3. Text is printed on each of the 4 rows.
4. LCD remains static unless updated in the loop.

Tips & Troubleshooting:

- If the screen is **blank**, adjust the **potentiometer** connected to the **VO** pin.
- Ensure that **RS/E/D4–D7** match the pin numbers in `LiquidCrystal()`.
- Always connect **RW pin to GND** (write mode only).
- Make sure the LCD is powered with **5V**, not 3.3V.

Visual Layout Example:

Row 0: Hello, Guys!

Row 1: I'm Gurucharan.G.R

Row 2: Let's connect and

Row 3: grow....

