**Interfacing an LCD display with an ESP8266 without using I2C**

**Components Needed:**

1. ESP8266 (like NodeMCU or Wemos D1 Mini)
2. LCD 16x2 (HD44780 compatible)
3. Breadboard and jumper wires
4. Resistor (220Ω for backlight)
5. Potentiometer (10kΩ for contrast adjustment)

**Pin Connections:**

For a 16x2 LCD in 4-bit mode, you will connect it as follows:

**LCD Pins:**

* **VSS**: Ground
* **VDD**: 5V (or 3.3V for ESP8266, but 5V is preferable for LCD)
* **V0**: Contrast (Connect to the middle pin of the potentiometer)
* **RS**: Digital Pin (e.g., D2)
* **RW**: Ground (to set the LCD in write mode)
* **EN**: Digital Pin (e.g., D1)
* **D4**: Digital Pin (e.g., D5)
* **D5**: Digital Pin (e.g., D6)
* **D6**: Digital Pin (e.g., D7)
* **D7**: Digital Pin (e.g., D8)

**Wiring Diagram:**

**LCD Pin | ESP8266 Pin**

**-------------|--------------**

**VSS | GND**

**VDD | 5V (or 3.3V)**

**V0 | Potentiometer**

**RS | D2**

**RW | GND**

**EN | D1**

**D4 | D5**

**D5 | D6**

**D6 | D7**

**D7 | D8**

### Code Example:

Below is an example code using the LiquidCrystal library to control the LCD:

**#include <LiquidCrystal.h>**

**// Initialize the library with the numbers of the interface pins**

**LiquidCrystal lcd(D2, D1, D5, D6, D7, D8);**

**void setup() {**

**// Set up the LCD's number of columns and rows**

**lcd.begin(16, 2);**

**// Print a message to the LCD**

**lcd.print("Hello, ESP8266!");**

**}**

**void loop() {**

**// You can update the display here**

**}**

**Notes:**

* Make sure to adjust the contrast using the potentiometer connected to V0.
* If you want to write more complex functions, you may need to create custom functions for sending commands and data in 4-bit mode.

**Troubleshooting:**

* Ensure all connections are secure.
* Verify the power supply to the LCD.
* Adjust the contrast if the display is not visible.