DWIN & ESP32 MODBUS Interfacing

Saikishen P V

This project interfaces an ESP32 with a DWIN DMG80480C070_15WTR HMI that is pre-flashed with a Modbus Slave OS (Slave ID = 1) to read the runtime value of a UI toggle implemented at VP address 5500 and print it to the serial monitor for verification and debugging.
The communication is implemented over Modbus RTU on an RS485 physical layer using a MAX485 RS485-to-TTL module, with the ESP32 acting as the Modbus master.

From	Pin	To	Pin
ESP32	GPIO17 (TX2)	MAX485 module	DI
ESP32	GPIO16 (RX2)	MAX485 module	RO
ESP32	GPIO4	MAX485 module	DE + RE (tied together)
ESP32	GND	MAX485 module	GND
ESP32	3V3	MAX485 module	VCC
MAX485 module	A	DWIN HMI	485A
MAX485 module	B	DWIN HMI	485B / B

START

→ **Setup phase**

- Initialize USB debug serial (`Serial.begin(115200)`).
- Configure RS485 direction pin:
  - Set `MAX485_DE_RE` as OUTPUT.
  - Drive it **LOW** → MAX485 in **receive** mode.
- Configure status LED pin `LED_PIN` as OUTPUT.
- Initialize Modbus UART (`Serial1.begin(115200, 8N1, RX_PIN, TX_PIN)`).
- Configure Modbus master object:
  - `node.begin(1, Serial1)` → target slave ID = 1.
  - Register callbacks:
    - `preTransmission()` → sets DE/RE HIGH before sending (TX mode).
    - `postTransmission()` → sets DE/RE LOW after sending (RX mode).
- Print "Modbus Master Started".

→ **Main loop (repeats forever)**

1. **Send Modbus request**
   - Call `node.readHoldingRegisters(0x5500, 1)`
   - Internally:
     - `preTransmission()` runs → DE/RE = HIGH (TX enable).
     - Library sends Modbus RTU frame (Function 0x03, start addr 0x5500, length 1).
     - `postTransmission()` runs → DE/RE = LOW (RX enable).
     - Library waits for slave response (or timeout).
2. **Check result**
   - **If SUCCESS (`result == ku8MBSuccess`)**
     - Read returned register:
       - `data = node.getResponseBuffer(0)`
     - Print:
       - `"VP 5500 Value: " + data`
     - **Decision: `data == 1` ?**
       - YES → LED_PIN = HIGH
       - NO → LED_PIN = LOW
   - **Else (FAIL)**

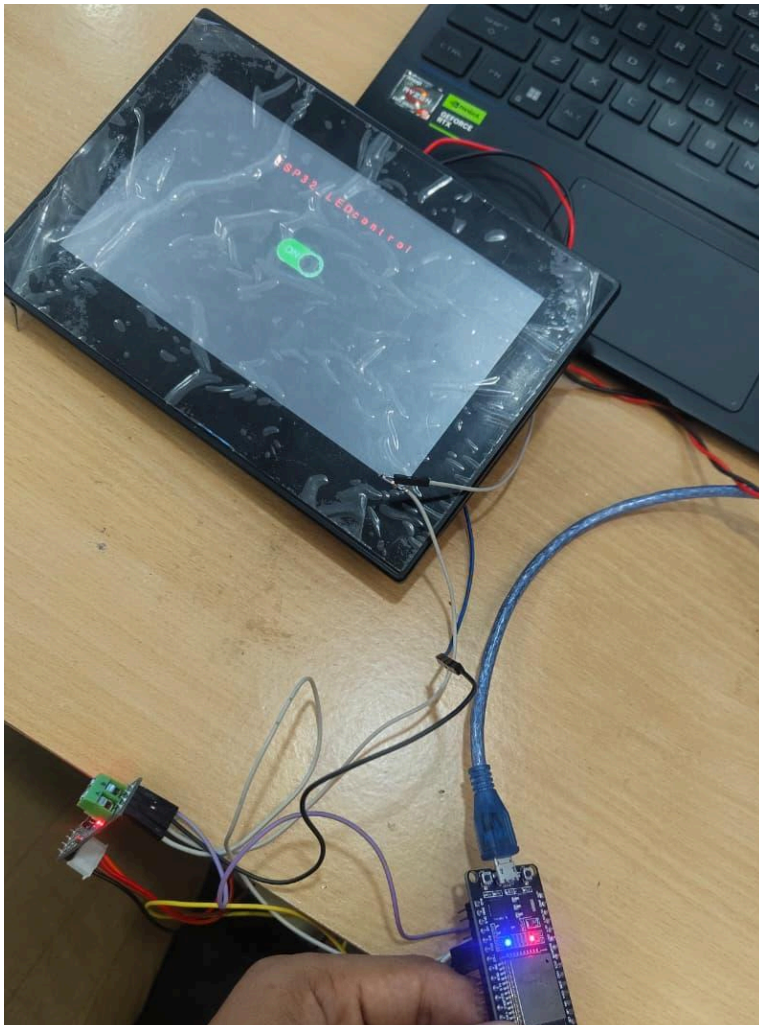
        - Print:
            - `"Read Failed. Error: " + result`
3. **Delay**
    - Wait 500 ms

→ Go back to **Main loop**

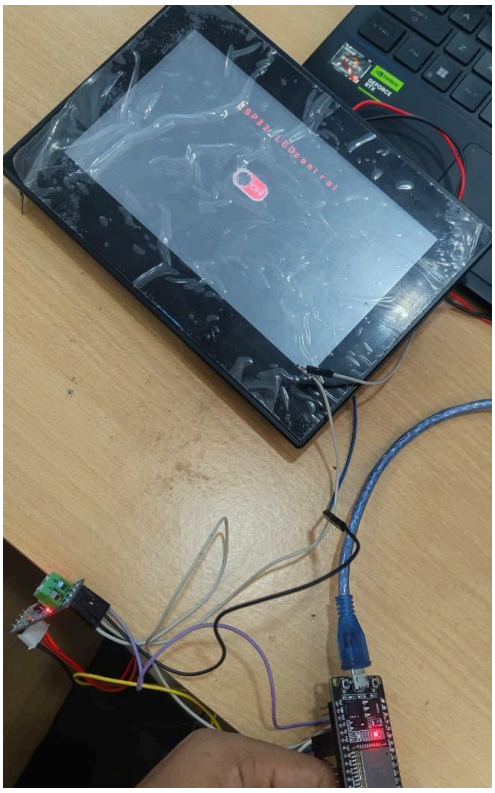
Results:

```
VP 5500 Value: 1
VP 5500 Value: 1
VP 5500 Value: 0
VP 5500 Value: 0
```

ESP32 serial monitor displaying the current value at VP 5500 (switch)



The LED turns on when the switch is toggled ON



The LED Turns OFF when the switch is toggled OFF