PV284 Colloquium Project

Benjamín Lapoš

February 2026

1 Project brief description

The project is an example of a room thermometer and a main boiler control unit. The system consists of 2 nodes: Node A - the main boiler control unit with touch display, Node B - a room thermometer with OLED display and control buttons. Communication between nodes is performed using ESP-NOW, native direct communication between ESP family boards.

Wiring:
Node A:

- T_IRQ - GPIO 36
- T_OUT - GPIO 39
- T_DIN - GPIO 32
- T_CS - GPIO 33
- T_CLK - GPIO 25
- SDO - GPIO 12
- LED - GPIO 21
- SCK - GPIO 14
- SDI - GPIO 13
- D/C - GPIO 2
- RESET - EN/RESET
- CS - GPIO 15
- GND - GND
- VCC - 5V (VIN)

Node B:

- GND - 3V3
- VCC - 5V
- SCL - GPIO 9 (OLED and ATH10)
- SDA - GPIO 14 (OLED and ATH10)
- Btn up - GPIO 18 (second btn pin to GND)
- Btn down - GPIO 19 (second btn pin to GND)

2 Project solution

The main hurdle to overcome was implementation of the project using my own devices.
Node A:

Initially, I wanted to use a beautiful chinese all-in-one display Guition JC3248W535EN, but due to the lack of proper documentation from the manufacturer I was not able to draw anything on the display, despite turning the display on and having working touch feedback. Thus, I went with ESP32 and a reliable old ILI9341 LCD TFT SPI touch display.
Node B:

This node was more straightforward. It uses an ESP32C6, I2C SSD1306 OLED display, 2 push buttons and I2C AHT10 thermometer.

Logic of the system is folowing: Node B measures the temperature and humidity in a room, displays them on the OLED display, and allows the user to use the 2 buttons to set the desired temperature. Buttons use interrupts and time delay debounce for more reliable interaction. The current and desired temperatures are sent to node A over ESP-NOW on each change.
Node B shows the current temperature received from node A and the desired temperature, that is synced both ways with node A, on the display. It allows the user to change the desired temperature on the touch display with graphic buttons. Whether the boiler is on or off, depending on the temperatures, is indicated by a small color circle and text in the corner. In real life use it would be feasible to add relay controlled by a gpio pin (not implemented here).

3 Conclusion

My biggest lesson learned was to buy chinese boards only from known sellers with enough documentation. Moreover, the experience of setting up my own connection of the modules provided me with a deepened understanding of electronics and the communication protocols used. This project acts as a proof of concept for multi-room temperature control, with temperature sensor in each room that I want to implement in near future.

4 References

ESP32 ILI9341 tutorial.
Guition display I was not able to make work and library I tried to use for it.
Parts used:
Node A:

- ILI9341 display equivalent
- ESP32 dev kit (chinese clone)

Node B:

- ESP32C6 official devkit
- ATH10 temperature sensor
- chinese OLED display
- Common small push buttons, breadboards and jumper wires I had spare

Libraries used:

- TFT eSPI fast SPI display library

- XPT2046 Touchscreen library
- ESP-NOW library included in the official Arduino ESP32 package
- Adafruit SSD1306 OLED library
- Adafruit GFX core graphics library
- Adafruit AHTX0 library for AHT10 and AHT20 sensors

5 Appendixes

5.1 Software for both nodes

Software for both nodes is provided here on Github. Folder ‘main_control_board‘ contains the code for node A and ‘esp32c6_temp_display‘ contains the code for node B.