Basics ESP32/Arduino:

ESP32 Smart Home Prototyping: Wireless communication, LEDs and OLED display

Work Order: Flashing LEDs with ESP32 microcontroller

Goal: The goal of this project is to make two LEDs blink using an ESP32 microcontroller. This work order serves as a guide for completing this simple project.

Materials:

- ESP32 microcontroller (in your mailbox in the Secretariat-H)
- 2 LEDs (different colors for better distinction)
- Resistors (if needed for LED protection)
- Breadboard and connection cable
- Computer with Arduino IDE installed

Steps:

1. Hardware setup:
   - o Connect the ESP32 microcontroller to the breadboard.
   - o Connect the anodes of the LEDs (longer leg) each to a GPIO pin of the ESP32 (e.g. GPIO2 and GPIO3).
   - o Connect the cathodes of the LEDs (shorter leg) to the GND (Ground) pin of the ESP32.
   - o If necessary, use resistors between the GPIO pins and the Anodes of the LEDs to protect the LEDs.
2. Set up Arduino IDE:
   - o Download and install the Arduino IDE from the official website if not already exists.
   - o Add ESP32 board support to Arduino IDE (see instructions on ESP32 website).
3. Programming:
   - o Open the Arduino IDE and create a new sketch (File -> New Sketch).
4. Documentation:
   - o Please wire the structure using suitable software document.
   - o Add a comment to each line of code, briefly describing the function explained.
5. Questions (for self-control):
   - o How are the LEDs connected to the ESP32 and what purpose do they serve Resistance in this context?
   - o Why is the selection of GPIO pins important when connecting the LEDs to the ESP32?
   - o What steps are required to use the Arduino IDE for programming the ESP32 prepare?
   - o What is the basic code snippet to make the LEDs blink in the desired pattern let?
   - o What information should be recorded in the log to ensure work progress to be able to understand later?
   - o How can you ensure that the wiring is correct to avoid possible malfunctions to avoid?

Step 2:

Extended work order: OLED display with ESP32 and LEDs

Goal: The goal of this extended project is to add an OLED display to the existing setup and control it using a text or icon while the LEDs continue to flash.

Additional materials:

- OLED display (e.g., SSD1306 based display)
- Connection cable for the display

Extension of steps:

1. Hardware Setup:
   - o Connect the OLED display to the ESP32.
   - o Note the connection details of the display and use the appropriate ones GPIO pins (e.g., SDA, SCL).
   - o Make sure the display's I2C address is configured correctly.
2. Software setup:
   - o Download the required OLED display library in the Arduino IDE and install it.
   - o Customize the code to include OLED display functionality.
3. Programming:
   - o Expand the code to include the OLED instructions in the setup section
4. Documentation.
5. Questions (for self-control):
   - o How is the OLED display connected to the ESP32 and why is the selection of GPIO Pins important?
   - o Which libraries need to be installed in the Arduino IDE to use the OLED display steer?
   - o Why is it necessary to update the code of the first ESP32 to the OLED display to integrate?
   - o How can you ensure that the I2C address of the OLED display is configured correctly?
   - o Why should the code for the OLED display be placed in the setup area?
   - o What possible difficulties could arise when integrating the OLED display, and how can they be solved?

3rd step:

Extended work order: Communication between two ESP32 via proprietary Wireless technology with a shared OLED display

Goal: The goal of this expanded project is to integrate a second ESP32 into the existing setup. This second ESP32 is supposed to use proprietary radio technology (e.g. ESP-NOW) to send data to the first ESP32, which displays them on the shared OLED display.

Additional materials:

- Second ESP32
- Connection cable for the second ESP32

Extension of steps:

1. Hardware update:
   - o Connect the second ESP32 to its own set of LEDs (e.g., GPIO4 and GPIO5).
   - o Make sure that the wiring of the second ESP32 is correct.
2. Software setup:
   - o Download and install the ESP-NOW library in the Arduino IDE.
3. Programming for the first ESP32 (receiver):
   - o Update the code of the first ESP32 to use ESP-NOW and receive data from the second ESP32 and display it on the OLED display.

1. Programming for the first ESP32 (receiver):
   - o Update the code of the first ESP32 to use ESP-NOW and receive data from the second ESP32 and display it on the OLED display.

2. Programming for the second ESP32 (transmitter):
   - o Create a separate sketch for the second ESP32 that sends data to the first ESP32.
3. Documentation.
4. Questions (for self-control):
   - o What additional materials are needed for the third step of the project?
   - o How is the second ESP32 connected to the existing setup and what role do the antennas play in this context?
   - o Why is the selection of GPIO pins important for the second ESP32, especially for the LEDs?
   - o Which special radio technology is used for communication between the two ESP32 Modules used and why?
   - o Which library is installed in the Arduino IDE to enable the ESP-NOW functionality support?

- o How are the MAC addresses of the two ESP32 modules used in the code and why are they important?
- o What function was added in the code of the first ESP32 to get data from the second ESP32 to recieve?
- o Why should the function for receiving data be organized in its own function be?
- o What information could be recorded in the protocol for the third step in order to to look back on the work progress later?