# **ESP32-Modbus**

ESP32 Modbus Communication via RS485 Module

## **Overview**

This project provides an implementation of Modbus communication using an ESP32 microcontroller and an RS485 module. Modbus is a widely used communication protocol in the industrial automation field, allowing for communication between various devices such as PLCs, HMIs, and sensors.

The ESP32 microcontroller, equipped with built-in Wi-Fi and Bluetooth capabilities, provides a reliable and versatile platform for implementing Modbus communication. By connecting an RS485 module to the ESP32, we can establish communication with Modbus devices using the RS485 serial communication standard.

## **Features**

* Modbus master functionality: The ESP32 acts as the master device and can send requests to Modbus slave devices for data acquisition or control purposes.
* RS485 communication: The RS485 module enables long-distance communication and supports multi-node communication on the same bus.
* Support for various Modbus function codes: The implementation supports common Modbus function codes such as Read Holding Registers, Write Single Register, and Write Multiple Registers.
* Automatic baud rate detection: The system automatically detects the baud rate of the connected Modbus devices, ensuring seamless communication.
* Temperature monitoring: The project includes temperature monitoring using a thermocouple sensor, allowing for automatic control of the system based on temperature thresholds.
* Auto/Manual mode switching: The system can operate in either Auto or Manual mode, allowing for manual control of the motors and sensors if needed.

## **Hardware Requirements**

To replicate this project, you will need the following hardware components:

* ESP32 development board
* RS485 module
* Thermocouple sensor
* HMI or Modbus slave devices for testing (e.g., PLCs, sensors)

## **Software Requirements**

The project is developed using the Arduino framework for ESP32. You will need the following software tools:

* Arduino IDE: To write, compile, and upload the code to the ESP32.
* ESP32 board support package for Arduino: To enable ESP32 development within the Arduino IDE.
* ModbusMaster library: To handle Modbus communication.

## **Setup Instructions**

* Install the Arduino IDE on your computer.
* Install the ESP32 board support package for Arduino.
* Connect the RS485 module and thermocouple sensor to the ESP32 according to the provided pin configuration in the code.
* Open the Arduino IDE and create a new sketch.
* Install the ModbusMaster library by navigating to Sketch > Include Library > Manage Libraries and searching for "ModbusMaster".
* Copy and paste the provided code into the Arduino IDE.
* Adjust the configuration variables in the code to match your hardware setup and Modbus device addresses.
* Compile and upload the code to the ESP32 board.
* Open the Serial Monitor in the Arduino IDE to view the system logs and debug messages.
* Verify that the ESP32 successfully establishes communication with the Modbus devices and performs the desired operations.

## **Usage**

The system operates in two modes: Auto and Manual.

* Auto mode: In this mode, the ESP32 continuously monitors the temperature and controls the motors and sensors based on predefined thresholds. Make sure the temperature limits and other configuration variables are set according to your requirements.
* Manual mode: In this mode, you can manually control the motors and sensors using the buttons or switches connected to the ESP32. This mode is useful for testing or manual intervention.

Use the Serial Monitor in the Arduino IDE to monitor the system logs and debug messages. Make sure to set the baud rate in the Serial Monitor to match the one configured in the code.

## **Contributing**

Contributions to this project are welcome. If you find any bugs or have suggestions for improvements, please open an issue or submit a pull request.

## **License**

This project is licensed under the [MIT License](https://chat.openai.com/LICENSE).

## **Acknowledgments**

* [ModbusMaster](https://github.com/4-20ma/ModbusMaster) library for Modbus communication.
* The Arduino community for providing the ESP32 board support package and development resources.

## **Disclaimer**

This project is provided as-is without any warranty. Use it at your own risk. Ensure proper safety precautions when working with electrical and industrial automation systems.