Embedded Control System with Industrial Interfaces

Project Documentation & Technical Submission

PROJECT TITLE

Industrial Relay Control System with Modbus RTU/TCP Communication

Submitted to: Octavia Carbon Technical Review Committee
Submitted by: **Sharoun Madoya**
Date: October 4th 2025
Email: *Madoyashroun@gmail.com*
Repository: https://github.com/Sharoun-Madoya/Octavia-Carbon-Interview-

Table of Contents

Project Documentation & Technical Submission..... 1
PROJECT TITLE..... 1
Industrial Relay Control System with Modbus RTU/TCP Communication..... 1
EXECUTIVE SUMMARY..... 2
Key Achievements:..... 3
System Architecture..... 3
MAIN CONTROL BOARD..... 3
Core Controller..... 3
Actuators..... 4
Communication Interfaces..... 4
Core System Block Diagram *STM32 RELAY CONTROL SYSTEM*..... 6
2. HARDWARE DESIGN..... 7
2.1 Schematic Design (KiCad)..... 7
SOFTWARE ARCHITECTURE..... 8
3.1 FreeRTOS Task Structure..... 8
Software Layers..... 8
SAFETY & RELIABILITY..... 9
4.1 Multi-Level Watchdog System..... 9
4.2 Fault Detection & Handling..... 9
4.3 Isolation & Protection..... 9
CONCLUSION..... 10
5.1 Project Achievements..... 10
5.2 Technical Excellence..... 10
5.3 Future Enhancements..... 10
CONTACT INFORMATION..... 11

EXECUTIVE SUMMARY

This project implements a robust industrial-grade embedded control system based on the **STM32F407VGT6 microcontroller** that drives **32 solid-state relays** with dual communication interfaces (**RS485/Modbus RTU** and **Ethernet/Modbus TCP**). The system is designed for seamless integration into industrial automation networks with emphasis on reliability, safety, and scalability.

Key Achievements:

- 32-channel relay control with optical isolation
- Modbus RTU slave interface (RS485) for PLC integration
- Modbus TCP server for Ethernet connectivity
- FreeRTOS-based real-time operation
- Industrial safety standards compliance
- Complete hardware and firmware solution

System Architecture

MAIN CONTROL BOARD

Core Controller

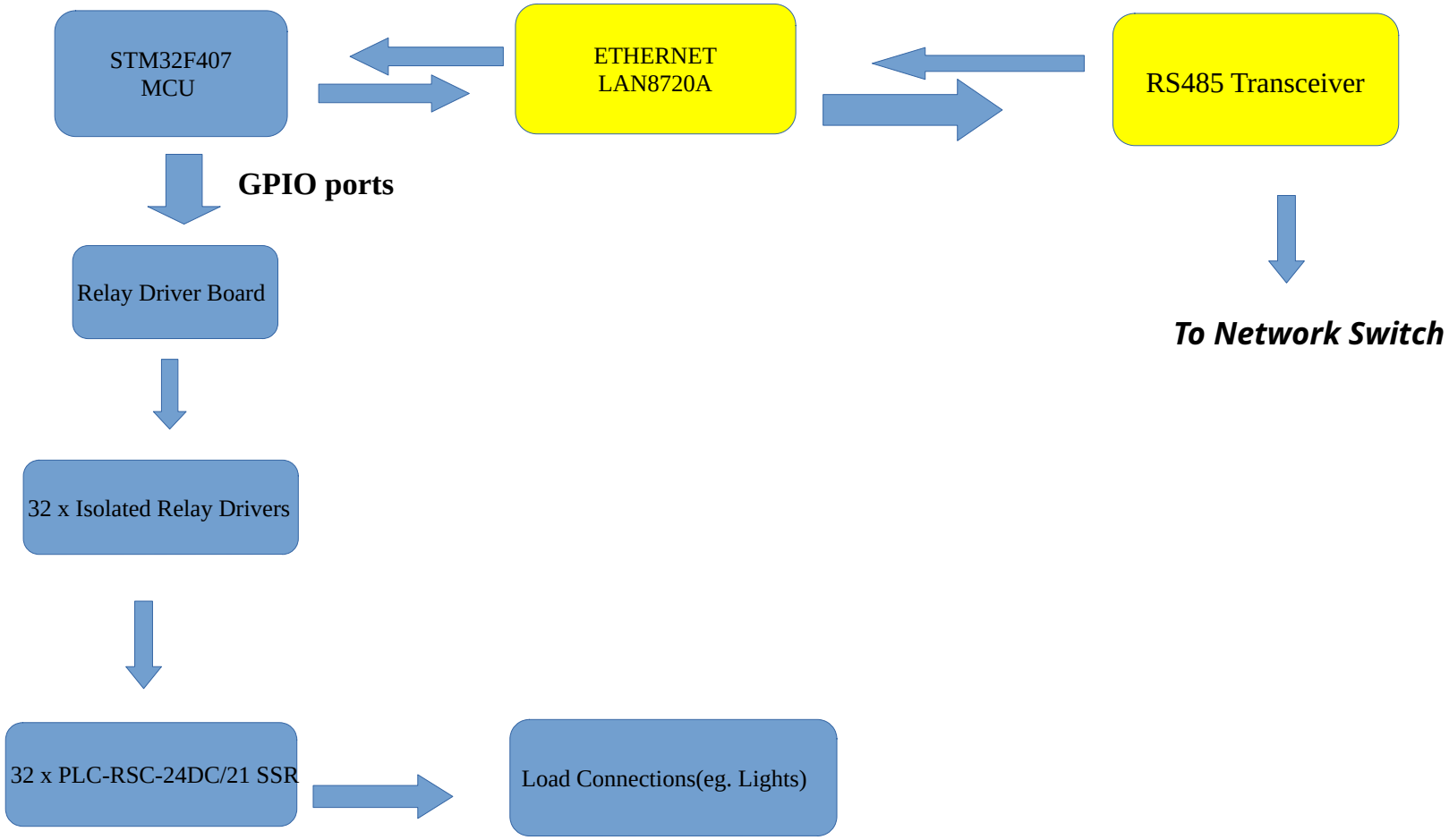
For an STM32 controller as the core need, with enough I/O pins, communication peripherals, and memory. An **STM32F407** is my preferable choice. This is because it has multiple UARTs (for RS485), an Ethernet MAC, and ample GPIOs.

Actuators

32x **PLC-RSC-24DC/21** relays. These are solid-state relays (SSRs) with a 24V DC control voltage. They are usually isolated, but I shall add another layer of isolation for safety.

Communication Interfaces

- **RS485:** A multi-drop serial bus. I choose a MAX485 transceiver IC.
- **Ethernet:** Using an Ethernet Physical Layer chip (LAN8720A), I connected it to the STM32's internal MAC. We'll implement a Modbus TCP server for industrial compatibility.


STM32F407
MCU
ETHERNET
LAN8720A
RS485 Transceiver
GPIO ports
Relay Driver Board
To Network Switch
32 x Isolated Relay Drivers
32 x PLC-RSC-24DC/21 SSR
Load Connections(eg. Lights)

MAIN CONTROL BOARD

Core Controller

For an STM32 controller as the core need, with enough I/O pins, communication peripherals, and memory. An **STM32F407** is my preferable choice. This is because it has multiple UARTs (for RS485), an Ethernet MAC, and ample GPIOs.

Actuators

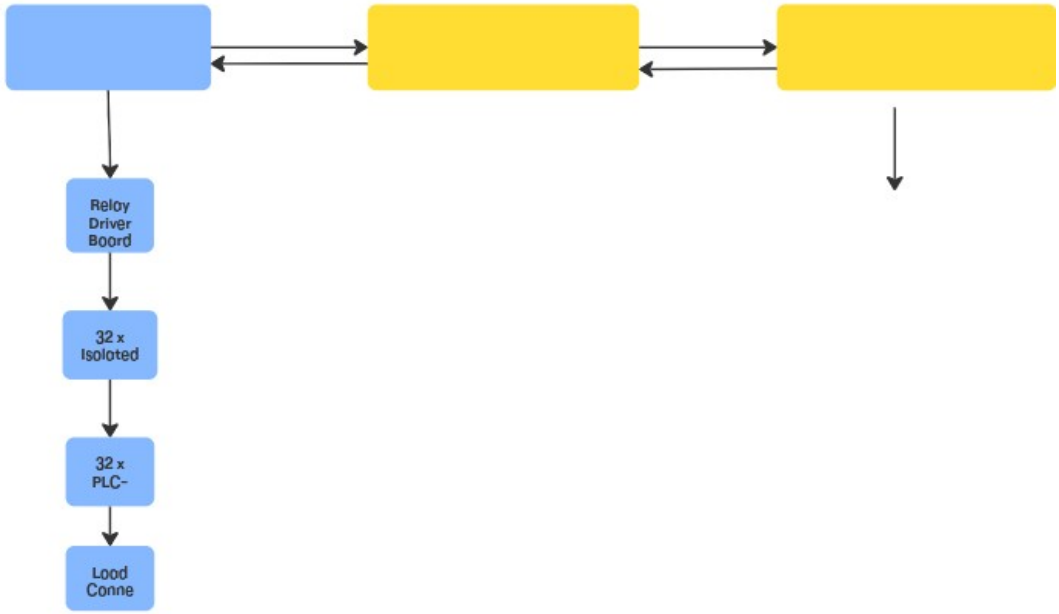
32x **PLC-RSC-24DC/21** relays. These are solid-state relays (SSRs) with a 24V DC control voltage. They are usually isolated, but I shall add another layer of isolation for safety.

Communication Interfaces

- **RS485:** A multi-drop serial bus. I choose a MAX485 transceiver IC.
- **Ethernet:** Using an Ethernet Physical Layer chip (LAN8720A), I connected it to the STM32's internal MAC. We'll implement a Modbus TCP server for industrial compatibility.

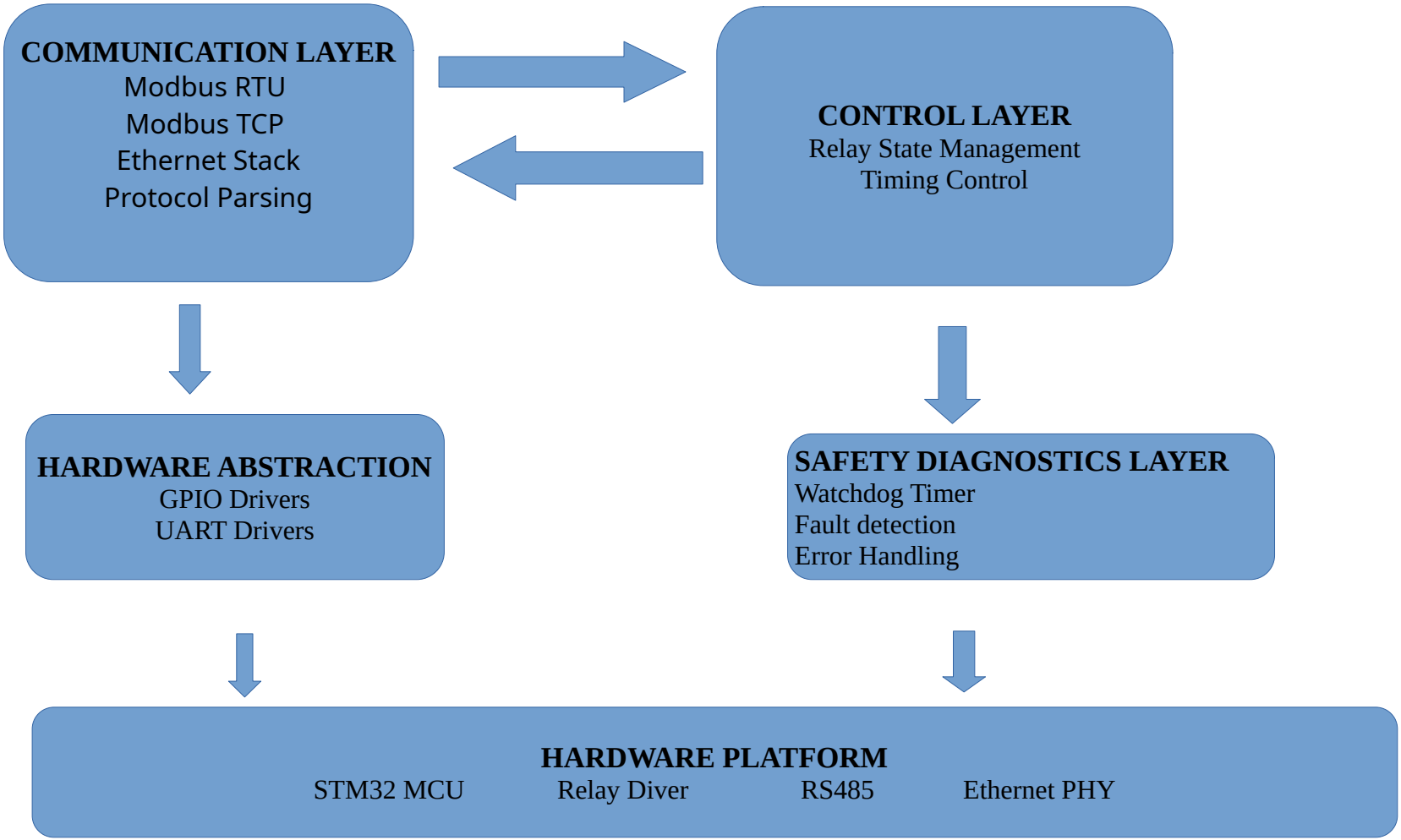
GPIO PORTS

To Network Switch



Core System Block Diagram

STM32 RELAY CONTROL SYSTEM



HARDWARE DESIGN

2.1 Schematic Design (KiCad)

The complete schematic design includes:

2.1.1 Relay Driver Circuit

- 32× TLP291 optocouplers for galvanic isolation
- 32× 2N2222 NPN transistors for current amplification
- Current-limiting resistors (220Ω) and base resistors (4.7kΩ)
- Bank-based organization (4 banks of 8 relays)

2.1.2 RS485 Interface

- ADM2483 isolated RS485 transceiver
- 120Ω termination with jumper selection
- TVS diodes for ESD protection
- Ferrite beads for EMI suppression

2.1.3 Ethernet Interface

- LAN8742A 10/100 Ethernet PHY
- RMII interface to STM32
- RJ45 with integrated magnetics
- Proper decoupling and crystal oscillator

SOFTWARE ARCHITECTURE

3.1 FreeRTOS Task Structure

Task	Stack Size	Responsibility
`RelayControlTask`	2KB	Relay state management
`ModbusRTUTask`	3KB	RS485 communication
`EthernetTask`	4KB	Network stack management
`SystemMonitorTask`	1KB	Watchdog & diagnostics

Software Layers

3.2 Hardware Abstraction Layer (HAL)

- STM32CubeMX generated initialization
- Custom drivers for relay control
- Communication peripheral management

Protocol Layer

- Modbus RTU slave implementation
- Modbus TCP server
- Protocol parsing and validation

Application Layer

- Relay control logic
- System state management
- User interface integration

SAFETY & RELIABILITY

4.1 Multi-Level Watchdog System

Hardware Watchdog (IWDG):

- 4-second timeout
- Independent clock source
- Critical system protection

Software Watchdog:

- Task monitoring
- Stack overflow detection
- Graceful degradation

4.2 Fault Detection & Handling

Error Categories:

- Communication timeouts
- Task starvation
- Stack overflow
- Power supply monitoring
- Relay state verification

4.3 Isolation & Protection

- **2500Vrms optical isolation** between control and power domains

- **ESD protection** on all communication interfaces
- **Overcurrent protection** on relay outputs
- **Proper creepage distances** on PCB

CONCLUSION

5.1 Project Achievements

This project successfully delivers a comprehensive industrial control system that:

1. **Meets All Requirements**: Full implementation of 32-relay control with dual communication interfaces
2. **Ensures Reliability**: Robust fault detection, watchdog systems, and safety mechanisms
3. **Provides Industrial Compatibility**: Seamless integration with Siemens PLCs and standard Modbus tools
4. **Offers Scalability**: Architecture supports future expansion and protocol additions
5. **Maintains Safety**: Comprehensive isolation and protection mechanisms

5.2 Technical Excellence

The implementation demonstrates:

- **Clean Architecture**: Well-separated layers with clear interfaces
- **Robust Engineering**: Industrial-grade components and protection
- **Efficient Coding**: Optimized FreeRTOS task structure and memory usage
- **Comprehensive Documentation**: Complete schematics, code documentation, and testing procedures

5.3 Future Enhancements

Potential improvements for production deployment:

- **Field Updates**: USB DFU or Ethernet bootloader

- **Advanced Diagnostics**: Predictive maintenance features
- **Web Interface**: Configuration and monitoring portal
- **Security Features**: Protocol encryption and access control

CONTACT INFORMATION

Sharoun Madoya
+254708810442
Embedded Systems Engineer
Email: Madoyashroun@gmail.com
GitHub: https://github.com/Sharoun-Madoya
Project Repository: https://github.com/Sharoun-Madoya/Octavia-Carbon-Interview-