I2C EEPROM Communication with STM32

1. Introduction

This document describes the implementation of I2C communication between an STM32 microcontroller and an EEPROM using direct register manipulation. The project demonstrates reading and writing operations on an EEPROM chip via the I2C1 peripheral of the STM32.

2. Requirements

2.1 Hardware Requirements

- STM32-based microcontroller
- EEPROM with I2C interface (e.g., AT24Cxx)
- Serial debugging interface (optional)
- Power supply (3.3V or 5V as required by the EEPROM)
- Connecting wires

2.2 Software Requirements

- STM32CubeIDE (optional)
- Keil uVision or any ARM-compatible compiler
- Embedded C programming knowledge

3. System Design and Implementation

3.1 I2C Peripheral Configuration

The I2C1 peripheral is configured by enabling the corresponding GPIOB pins (PB6 and PB7), configuring them for alternate function, open-drain mode, and pull-up resistors.

3.2 Register Definitions

Relevant memory-mapped registers are defined using their respective addresses. These include RCC registers for enabling GPIOB and I2C1, GPIOB registers for pin configuration, and I2C1 registers for communication.

3.3 I2C Initialization

The `I2C1_Init` function performs the following:

- Enables GPIOB and I2C1 clock
- Configures PB6 and PB7 as alternate function open-drain
- Resets and initializes I2C1 peripheral
- Configures the clock speed and rise time
- Enables the I2C1 peripheral

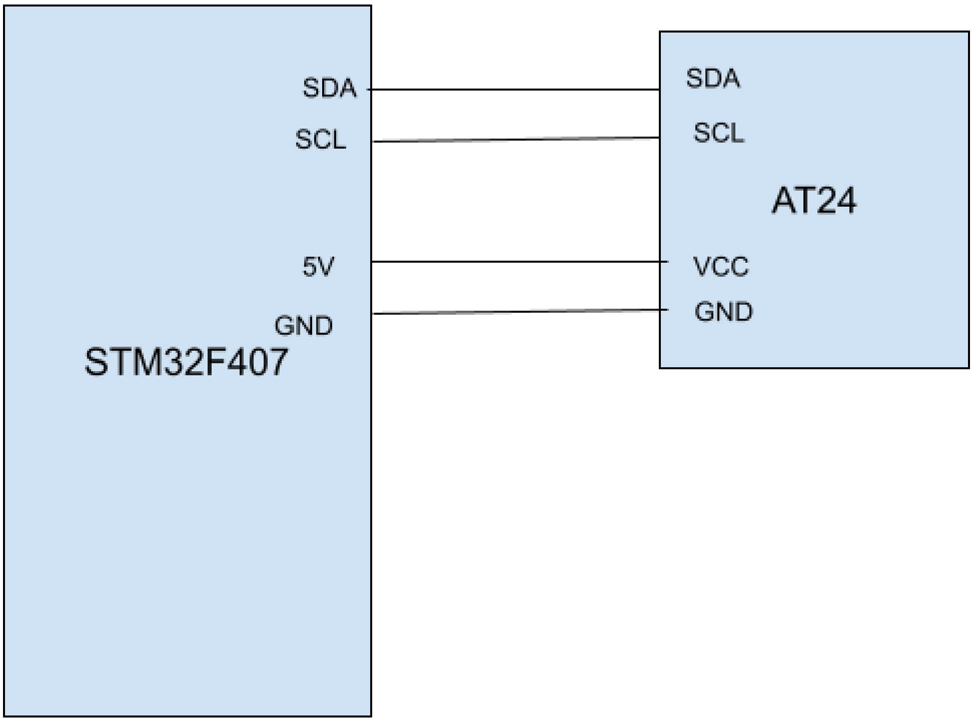
3.4 I2C Communication Functions

- `I2C1_Start(uint8_t address, uint8_t direction)`: Generates a start condition and sends the address.
- `I2C1_WriteByte(uint8_t data)`: Sends a byte over I2C.
- `I2C1_ReadByte(uint8_t ack)`: Reads a byte from the I2C bus.
- `I2C1_Stop()`: Generates a stop condition.

3.5 EEPROM Read/Write Functions

- `EEPROM_Write(uint8_t mem_addr, char data)`: Writes a byte to a specified memory address in EEPROM.
- `EEPROM_Read(uint8_t mem_addr)`: Reads a byte from a specified memory address in EEPROM.

4. Block Diagram



5. Implementation and Execution

The `main` function initializes I2C1 and executes the following steps in an infinite loop:

1. Writes a string "Nishadh" to EEPROM starting at memory address 0x10.
2. Introduces a delay for EEPROM write operation.
3. Reads back the stored data into a buffer.
4. Compares the written and read data.
5. Introduces a delay before repeating the process.

5. Conclusion

This project successfully demonstrates EEPROM communication using I2C on an STM32 microcontroller by implementing low-level register access. The developed functions provide a foundation for future enhancements, such as block read/write operations and integrating error-handling mechanisms.

This documentation serves as a reference for developers working on similar I2C-based EEPROM applications.