Indian Institute of Technology Bombay

Microprocessor Lab
EE 337

Lab 9B Part I
April 8, 2025

Mridul Choudhary
23B3933

Contents

1 Board to Board communication using SPI

1.1 Aim of the experiment

The aim of Part 1 in the EE337 Lab 9B project is to establish board-to-board communication using the SPI (Serial Peripheral Interface) protocol. A Master and a Slave microcontroller board are connected through four SPI lines: CLK, MISO, MOSI, and CS. The Master board sends two prime numbers to the Slave. The Slave computes their sum and checks whether it is a prime number. The result, along with the sum, is displayed on the LCDs attached to both boards. The Master receives a signal from the Slave and shows either "PRIME" or "NOT PRIME" on its LCD.

1.2 Design

1. **Hardware Connections:**
    - Connect both the boards using the SPI lines: CLK, MISO, MOSI, and CS.
    - Attach an LCD to both boards for output display.
    - Connect necessary power and ground pins between boards.
2. **Initialization:**
    - Initialize the SPI interface on both Master and Slave boards using **spi_master_init()** and **spi_slave_init()** respectively.
    - Initialize the LCD on both boards using **lcd_init()**.
3. **Execution Flow:**
    - The Master sends two predefined prime numbers ($p1 = 2$, $p2 = 11$) to the Slave using the SPI transmit function **spi_trx()**.
    - The Slave receives the two numbers, adds them, and checks if the result is also a prime using the **is_prime()** function.
    - The result of the prime check is sent back to the Master.
    - The Slave displays the two numbers and their sum on its LCD.
    - The Master displays either "**PRIME**" or "**NOT PRIME**" based on the result.
    - Displays remain indefinitely.

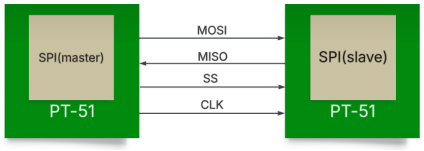

Figure 1: Master - Slave Configuration

1.3 Function Explanations

- **main() in SPI_master.c**
  The main function in the Master board initializes SPI and LCD. It then sends two hard-coded prime numbers to the Slave via SPI. After short delays to allow data transmission, it awaits the response from the Slave. If the response is 1, it displays "PRIME"; if 2, it shows "NOT PRIME"; otherwise, it outputs "ERROR 404". This function demonstrates SPI transmission and LCD feedback effectively.

```
void main(void) {
    unsigned int p1 = 2;
    unsigned int p2 = 11;
    unsigned int result = 0;
    char adc_ip_data_ascii[6]={0,0,0,0,0,'\0'};

    spi_master_init();
    lcd_init();
    lcd_write_string("Master Initializing");

    SS = 0;
    spi_trx(p1);
    SS = 1;

    msdelay(500);

    SS = 0;
    spi_trx(p2);
    SS = 1;

    msdelay(500);

    result = spi_trx(0);
    lcd_cmd(0x01);

    if(result == 1) {
        lcd_write_string("PRIME");
    }
    else if(result == 2) {
        lcd_write_string("NOT PRIME");
    }
    else {
        lcd_write_string("ERROR 404");
    }

    while (1);
}
```

- **main() in SPI_slave.c**
  In the Slave board, the main function initializes the SPI and LCD. It then waits to receive two numbers via SPI from the Master. Upon receiving, it displays the numbers and their sum on its LCD. It uses the helper function is_prime() to determine if the sum is a prime number and sends the result (1 or 2) back to the Master via SPI.

```
void main(void) {
    unsigned int num1, num2, sum, result;
    char adc_ip_data_ascii[6]={0,0,0,0,0,'\0'};

    spi_slave_init();

    lcd_init();
    lcd_write_string("Slave Initializing");

    lcd_cmd(0x01);

    num1 = spi_trx(0x00);
    int_to_string(num1, adc_ip_data_ascii);
    lcd_write_string(adc_ip_data_ascii);

    num2 = spi_trx(0x00);
    lcd_write_char('+');
    int_to_string(num2, adc_ip_data_ascii);
    lcd_write_string(adc_ip_data_ascii);

    sum = num1 + num2;

    result = is_prime(sum);

    lcd_cmd(0xC0);
    lcd_write_string("Sum is ");
    int_to_string(sum, adc_ip_data_ascii);
    lcd_write_string(adc_ip_data_ascii);

    spi_trx(result);

    while (1);
}
```

- **is_prime(unsigned int n) in SPI_slave.c**
  This utility function checks whether a given number is prime. It returns 2 if the number is not prime and 1 if it is. The logic is based on trial division up to the square root of n, which is efficient enough for small inputs and suitable for microcontroller environments.

```
unsigned int is_prime(unsigned int n) {
    unsigned int i;
    if (n < 2) return 2;
    for (i = 2; i * i <= n; ++i) {
    if (n % i == 0) return 2;
    }
    return 1;
}
```

1.4 Output Screenshots

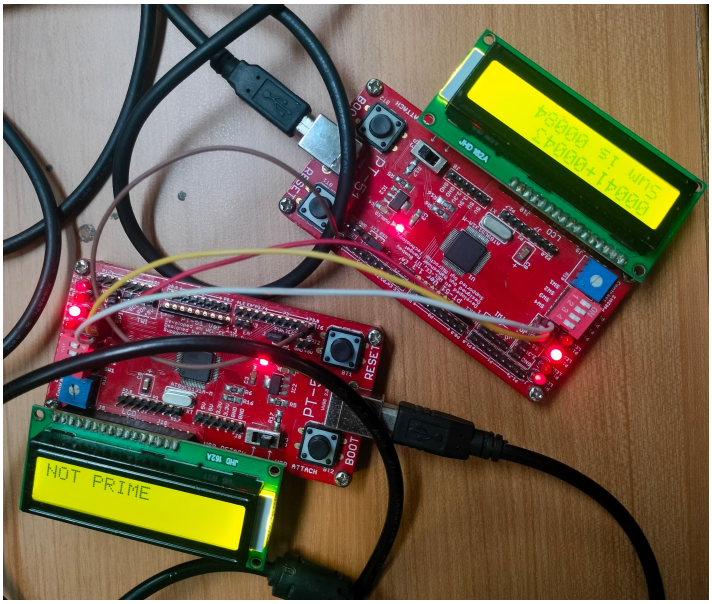

Figure 2: Not Prime Example

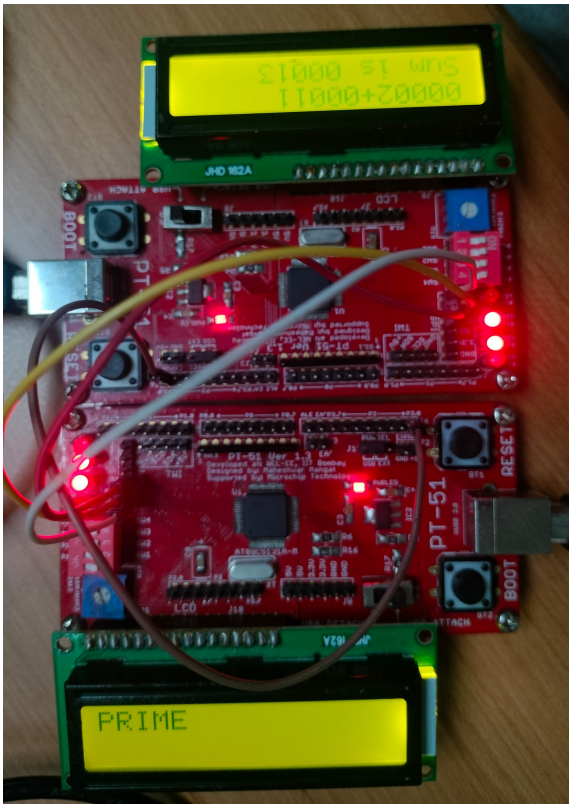

Figure 3: Prime Example

1.5 Conclusion and Inference

This experiment successfully demonstrates SPI-based communication between two microcontroller boards. The Master sends data, the Slave performs processing, and the result is transmitted back, with real-time display on LCDs. The use of SPI provides a practical experience in synchronous communication, and the exercise reinforces concepts in embedded systems, such as peripheral interfacing, basic algorithm implementation, and microcontroller programming.