**EE084IU**

**Micro-processing Systems**

**LAB 6:**

**SPI AND I2C INTERFACE**

**Student:**

**Student ID:**

**Class:**

**Date:**

I. LAB OBJECTIVES

After completion of this Lab, Students will:

* Know how to use SPI serial interface
* Know how to use I2C serial interface
* Develop multi-tasks program for embedded system
* Design the embedded system hardware and software for specific application

I. PROCEDURE

Student Carry out all the Lab tasks and get Lab Instructor Signature check mark for each task.

I.1 TASK1: SPI inter-system connection

I.1.1 Problem Statement: Design two embedded systems using ATMEGA32 (working at 8Mhz) communicate together by SPI serial interface. One system work as the Master SPI, One system work as the Slave SPI. The slave system has a LCD to display received data and has 8 LEDs. The Master has LCD to display the data, and 8 buttons, when button0 is pressed the Master system will send character ‘0’ to slave system, and button1 is pressed the system will send character ‘1’ and so on … button7 is pressed the system will send character ‘7’. When the slave system receiving new character from the SPI interface the system will display this character to LCD and toggle the relevant LED ( for example the character received is ‘0’ LED0 will be toggled). Write C program to control the system with above requirements.

I.1.2 Circuit Design:

Student Design the application circuit in Proteus

I.1.3 Theory:

Student review related theory knowledge about SPI and describe in the report, with following details:

- SPI Control Registers

- SPI initialization explanation for the task1 requirements

I.1.4 Algorithm

Master Algorithm

1. Develop a function to initialize SPI Master

2. Develop a function to send a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to LCDs

+ Setting up PORTA as input connect to 8 Buttons

+ Initialize LCD (2x16) which connect to PORTC working in 4 bit mode

+ Enable SPI receiver interrupt

+ While(1) loop :

- Check each of 8 buttons is pressed of not.

If any button is pressed, send the relevant character to SPI.

Slave Algorithm

1. Develop a function to initialize SPI Slave

2. Develop a function to receive a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to LCDs

+ Setting up PORTA as input connect to 8 Leds

+ Initialize LCD (2x16) which connect to PORTC working in 4 bit mode

+ Enable SPI receiver polling

+ While(1) loop :

- Check SPI receive a character .

If any character is received, control toggle relevant led.

I.1.5 Code

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Master Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Slave Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

I.1.6 Result

Check the designed circuit.

Check the button pressed in one system and check the led toggled and the data display in other system.

|  |  |  |
| --- | --- | --- |
| For Office use only | Signature of Lab Instructor | Remarks |
| Circuit Design in Proteus |  |  |
| Code Running in Proteus correctly |  |  |

I.2 TASK2: I2C inter-system connection

I.2.1 Problem Statement: Design two embedded systems using ATMEGA32 (working at 8Mhz) communicate together by I2C serial interface. One system work as the Master , One system work as the Slave I2C. The slave system has a LCD to display received data and has 8 LEDs. The Master has LCD to display the data, and 8 buttons, when button0 is pressed the Master system will send character ‘0’ to slave system, and button1 is pressed the system will send character ‘1’ and so on … button7 is pressed the system will send character ‘7’. When the slave system receiving new character from the I2C interface the system will display this character to LCD and toggle the relevant LED ( for example the character received is ‘0’ LED0 will be toggled). Write C program to control the system with above requirements.

I.2.2 Circuit Design:

Student Design the application circuit in Proteus

I.2.3 Theory:

Student review related theory knowledge about I2C and describe in the report, with following details:

- I2C Control Registers

- I2C initialization explanation for the task1 requirements

I.2.4 Algorithm

1. Develop a function to initialize I2C Master and SLAVE.

2. Develop a function to send a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to LCDs

+ Setting up PORTD as output connect to 8 LEDs

+ Setting up PORTB as input connect to 8 Buttons

+ Initialize LCD (2x16) which connect to PORTD working in 4 bit mode

+ Enable I2C receiver interrupt

+ While(1) loop :

- Check each of 8 buttons is pressed of not.

If any button is pressed, send the relevant character to I2C.

I.2.5 Code

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Master Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Slave Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

I.2.6 Result

Check the designed circuit.

Check the button pressed in one system and check the led toggled and the data display in other system.

|  |  |  |
| --- | --- | --- |
| For Office use only | Signature of Lab Instructor | Remarks |
| Circuit Design in Proteus |  |  |
| Code Running in Proteus correctly |  |  |

I.3 TASK3: SPI inter-system connection- Assembly Programming

I.3.1 Problem Statement: Design two embedded systems using ATMEGA32 (working at 8Mhz) communicate together by SPI serial interface. One system work as the Master SPI, One system work as the Slave SPI. The slave system has a 7-segment led to display received data and has 8 LEDs. The Master has a 7-segment to display the data, and 8 buttons, when button0 is pressed the Master system will send character ‘0’ to slave system, and button1 is pressed the system will send character ‘1’ and so on … button7 is pressed the system will send character ‘7’. When the slave system receiving new character from the SPI interface the system will display this character to a 7-segment and toggle the relevant LED ( for example the character received is ‘0’ LED0 will be toggled). Write Assembly program to control the system with above requirements.

I.3.2 Circuit Design:

Student Design the application circuit in Proteus

I.3.3 Theory:

Student review related theory knowledge about SPI and describe in the report, with following details:

- SPI Control Registers

- SPI initialization explanation for the task1 requirements

I.3.4 Algorithm

Master Algorithm

1. Develop a function to initialize SPI Master

2. Develop a function to send a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to 7-segment leds

+ Setting up PORTA as input connect to 8 Buttons

+ Enable SPI receiver interrupt

+ While(1) loop :

- Check each of 8 buttons is pressed of not.

If any button is pressed, send the relevant character to SPI.

Slave Algorithm

1. Develop a function to initialize SPI Slave

2. Develop a function to receive a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to 7-segment leds

+ Setting up PORTA as input connect to 8 Leds

+ Enable SPI receiver polling

+ While(1) loop :

- Check SPI receive a character .

If any character is received, control toggle relevant led.

I.3.5 Code

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Master Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Slave Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

I.3.6 Result

Check the designed circuit.

Check the button pressed in one system and check the led toggled and the data display in other system.

|  |  |  |
| --- | --- | --- |
| For Office use only | Signature of Lab Instructor | Remarks |
| Circuit Design in Proteus |  |  |
| Code Running in Proteus correctly |  |  |

I.4 TASK4: I2C inter-system connection- Assembly Programming

I.4.1 Problem Statement: Design two embedded systems using ATMEGA32 (working at 8Mhz) communicate together by I2C serial interface. One system work as the Master I2C, One system work as the Slave I2C. The slave system has a 7-segment led to display received data and has 8 LEDs. The I2C has a 7-segment to display the data, and 8 buttons, when button0 is pressed the Master system will send character ‘0’ to slave system, and button1 is pressed the system will send character ‘1’ and so on … button7 is pressed the system will send character ‘7’. When the slave system receiving new character from the I2C interface the system will display this character to a 7-segment and toggle the relevant LED ( for example the character received is ‘0’ LED0 will be toggled). Write Assembly program to control the system with above requirements.

I.4.2 Circuit Design:

Student Design the application circuit in Proteus

I.4.3 Theory:

Student review related theory knowledge about SPI and describe in the report, with following details:

- SPI Control Registers

- SPI initialization explanation for the task1 requirements

I.4.4 Algorithm

Master Algorithm

1. Develop a function to initialize I2C Master

2. Develop a function to send a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to 7-segment leds

+ Setting up PORTA as input connect to 8 Buttons

+ Enable I2C receiver interrupt

+ While(1) loop :

- Check each of 8 buttons is pressed of not.

If any button is pressed, send the relevant character to I2C.

Slave Algorithm

1. Develop a function to initialize I2C Slave

2. Develop a function to receive a character

3. Develop an interrupt function to receive a character, when receiving a new character this function will toggle the relevant led.

4. Write main program

+ Setting up PORTC as output connect to 7-segment leds

+ Setting up PORTA as input connect to 8 Leds

+ Enable SPI receiver polling

+ While(1) loop :

- Check I2C receive a character .

If any character is received, control toggle relevant led.

I.4.5 Code

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Master Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

|  |  |
| --- | --- |
| Write general comments about the program here  /\*----------------------------------------------------------------  This program is a multi-tasks program in Slave Side.  Author: xxxxxxxxx Date: xx.xx.20xx  ---------------------------------------------------------------------\*/ | |
| Instruction | Comments |
|  |  |

I.4.6 Result

Check the designed circuit.

Check the button pressed in one system and check the led toggled and the data display in other system.

|  |  |  |
| --- | --- | --- |
| For Office use only | Signature of Lab Instructor | Remarks |
| Circuit Design in Proteus |  |  |
| Code Running in Proteus correctly |  |  |

II. REPPORT

After finish all tasks student analyze the laboratory results and submit your report in blackboard. The report for each task should include

* Theory ( Related theory)
* Designed Circuit diagram
* Algorithm
* Code with clear comments

The format of the report should follow the Lab report guideline of School of Electrical Engineering

**GRADING GUIDELINE FOR LAB REPORT**

Name of Student: ID:

Subject: Lab Number:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Content** |  | **Score** | **Comment** |
| 1 | **Format (max 9%)** | |  |  |
| * Font type | Yes No |  |
| * Font size | Yes No |  |
| * Lab title | Yes No |  |
| * Page number | Yes No |  |
| * Table of contents | Yes No |  |
| * Header/Footer | Yes No |  |
| * List of figures (if exists) | Yes No |  |
| * List of tables (if exists) | Yes No |  |
| * Lab report structure | Yes No |  |
| 2 | **English Grammar and Spelling (max 6%)** | |  |  |
| * Grammar | Yes No |  |
| * Spelling | Yes No |  |
| 3 | **Data and Result Analysis (max 85%)**   * Running result check * Report contents * Code with clear comments | |  |  |
| **Total Score** | |  | |  |

Signature:

Date: