AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 06

Experiment Title: Communication between two Arduino Boards using SPI.

Date of Perform:	27 April 2025	**Date of Submission:**	20 May 2025
Course Title:	Microprocessor and Embedded Systems Lab		
Course Code:	EE4103	**Section:**	G
Semester:	Spring 2024-25	**Degree Program:**	BSc in CSE
Course Teacher:	**Prof. Dr. Engr. Muhibul Haque Bhuyan**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case Study is my/our original work; no part has been copied from any other student's work or any other source except where due acknowledgment is made.
3. No part of this Assignment/Case Study has been written for me/us by any other person except where such collaboration has been authorized. by the concerned teacher and is acknowledged in the assignment.
4. I/we have not previously submitted or am submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived to detect plagiarism.
6. I/we permit a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic, and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

** Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 02

Sl No	Name	ID	PROGRAM	SIGNATURE
1	MAHBUB HASAN	22-47419-2	BSc in CSE	
2	MUJTAHID TABSSUM MOHONA	22-47430-2	BSc in CSE	
3	MD TOYEB ALI	23-51819-2	BSc in CSE	

Faculty use only

FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table of Contents

Marking Rubrics (to be filled by Faculty):

Level Category	Excellent [5]	Proficient [4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]
Title and Objectives	Able to clarify the understanding of the lab, no issues are missing and formatting is good.	Able to clarify the understanding of the lab experiment, no issues are missing but its formatting is not good.	Able to clarify the understanding of the lab experiment, but a few issues are wrong, and its formatting is bad.	Able to clarify the understanding of the lab experiment, but it lacks a few important issues of the experiment without maintaining the format.	Unable to clarify the understanding of the lab experiment.	No Response/ copied from others/ identical submissions with gross errors/image file printed
Codes and Methods	Able to explain the experimental codes and simulation methods using Proteus very well.	Able to explain the experimental codes and simulation methods using Proteus but is not formatted well.	Able to explain the experimental codes but simulation method using Proteus is not explained well.	Presents the experimental codes but didn't explain simulation methods using Proteus clearly.	Presents the experimental codes but didn't explain simulation methods using Proteus.	
Results	Key results and images are there. Figures/Tables have all identifications and refer to them properly in the texts.	Key results and images are there. Figures/Tables have all identifications, such as the axis labels, numbers, and captions with a few minor errors; the texts refer them.	Key results and images are there. Figures/Tables lack a few identifications, such as the axis labels, numbers, and captions; the texts refer them.	Misses several key results and images. Figures/Tables lack identification, such as the axis labels, numbers, and captions; the texts don't refer them.	Major results, such as experimental and simulation results' images are not included. Figures and tables are poorly constructed or not presented.	
Discussion and Conclusion	Proper interpretation of results and summarizes the results to draw a conclusion, discusses its applications in real-life situations to connect with the report's conclusion.	Proper interpretation of results and summarizes the results to draw a conclusion but didn't discuss its applications in real-life situations to connect with the conclusion of the report.	Interpretation of results is presented. However, there is a disconnect between the results and discussion.	Misses the interpretation of key results. There is little connection between the results and discussion.	Very poor interpretation of the results. No connection between results and discussions.	
Question and Answer	Able to produce all questions' answers correctly maintaining the lab report format.	Able to produce all questions' answers but didn't maintain the lab report format.	Able to produce all questions' answers but wrong answers to a few questions.	Able to produce all questions' answers but wrong/missing answers to multiple questions.	Unable to produce all questions' answers and completely wrong answers.	
Comments						**Total Marks (25)**

Objectives:

The objectives of this experiment are to

a. Study the SPI protocol used in Arduino.
b. Write assembly language programming code for SPI communication with Arduinos.
c. Use SPI protocol for communication between two Arduinos.
d. Build a circuit to control the master side LED by the push button at the slave side and vice versa using the SPI Serial communication protocol.
e. Know the working principles of the SPI used in Arduino.

Equipment List:

1. Arduino IDE (2.0.1 or any recent version)
2. Arduino UNO (2)
3. LED (2)
4. Push Button (2)
5. Resistors 10 k, 2.2 k (2 + 2)
6. Breadboard
7. Connecting Wires

Circuit Diagram:

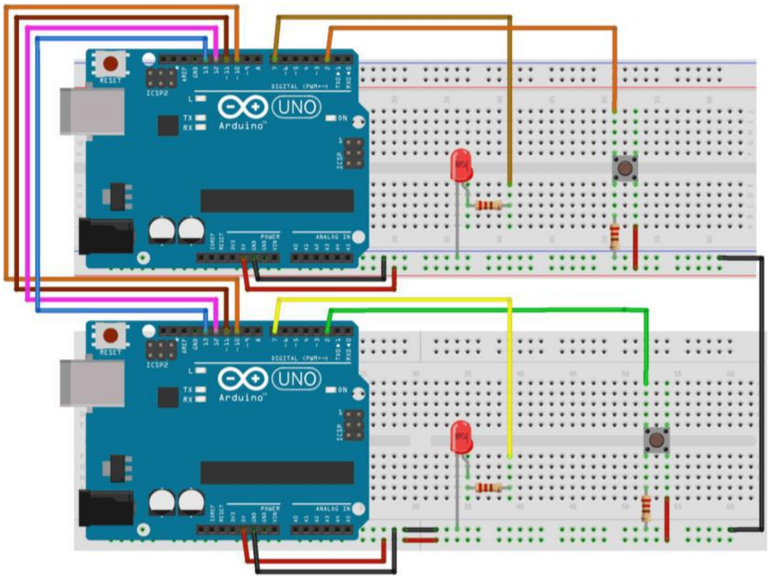


Figure 1: Circuit schematic of the experiment communication between two Arduino Boards using SPI.

Experimental Output Results:

Here are the output results of the experiment on communication between two Arduino Boards using SPI. One of them is Master (controller) circuit and another is Slave (Peripheral) circuit. In this program if the peripheral circuit switch is pressed the controller circuit LED will turn on and if the controller circuit switch is pressed the peripheral circuit LED will turn on.

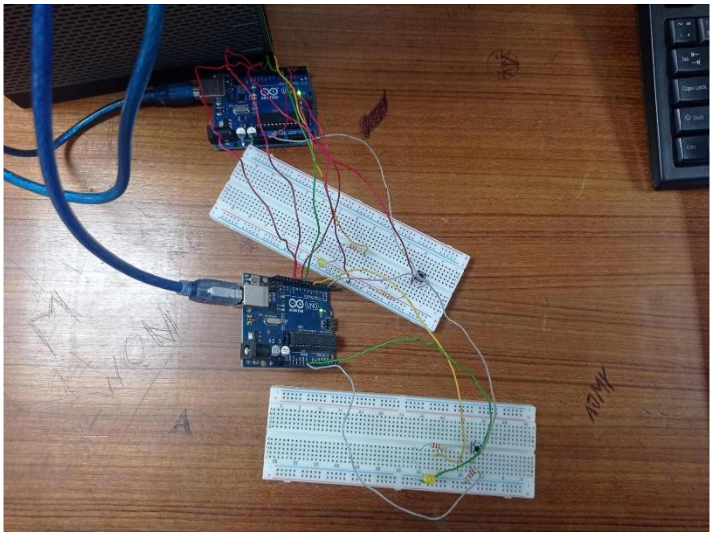


Figure 2: Both master and slave circuit LEDs are in off state

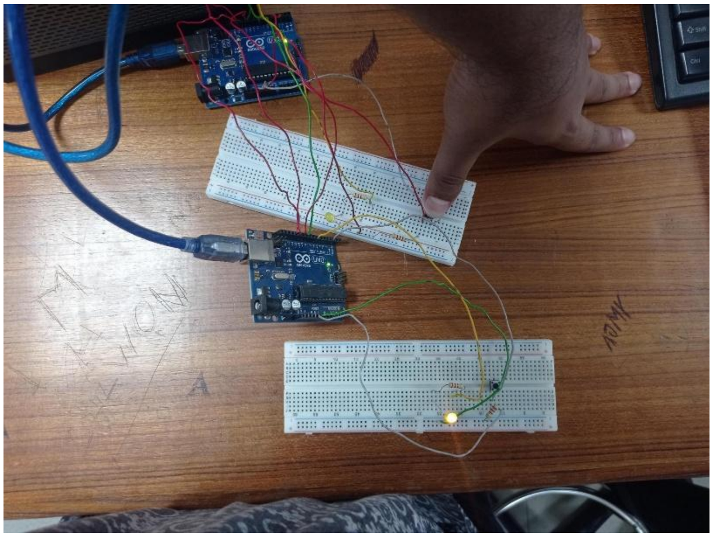


Figure 3: Pressing slave button to turn on master circuit LED

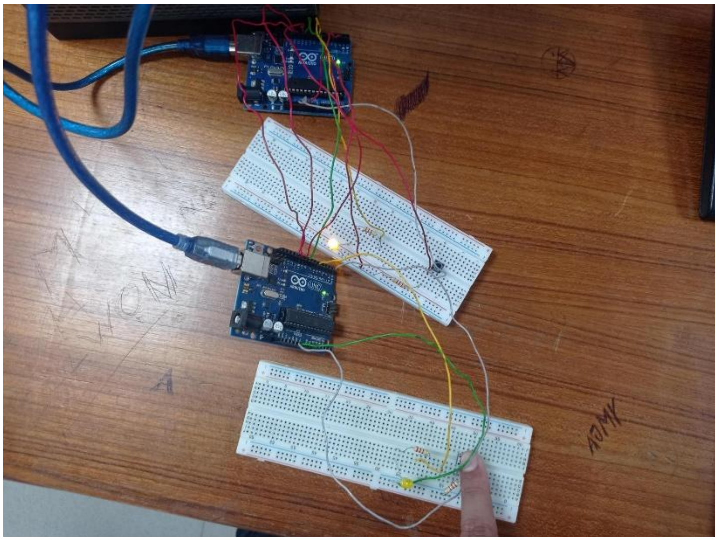
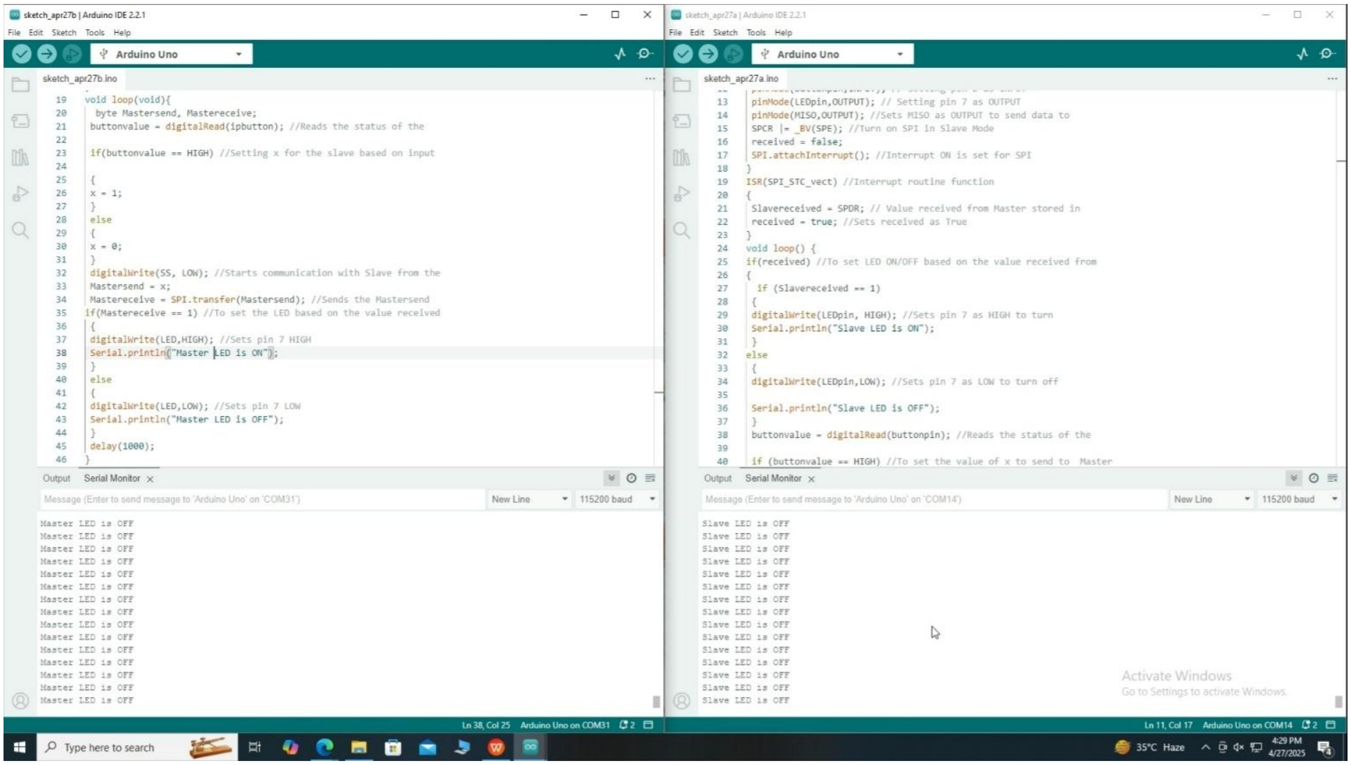


Figure 4: Pressing master button to turn on slave circuit LED


sketch_apr27b | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_apr27b.ino
Output
Serial Monitor
Ln 38, Col 25 Arduino Uno on COM31
sketch_apr27a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_apr27a.ino
Output
Serial Monitor
Activate Windows
Go to Settings to activate Windows.
Ln 11, Col 17 Arduino Uno on COM14
Type here to search
35°C Haze
4:29 PM
4/27/2025

```
void loop(void){
  byte Mastersend, Mastereceive;
 buttonvalue = digitalRead(ipbutton); //Reads the status of the

 if(buttonvalue == HIGH) //Setting x for the slave based on input

 {
 x = 1;
 }
 else
 {
 x = 0;
 }
 digitalWrite(SS, LOW); //Starts communication with Slave from the
 Mastersend = x;
 Mastereceive = SPI.transfer(Mastersend); //Sends the Mastersend
if(Mastereceive == 1) //To set the LED based on the value received
 {
 digitalWrite(LED,HIGH); //Sets pin 7 HIGH
 Serial.println("Master LED is ON");
 }
 else
 {
 digitalWrite(LED,LOW); //Sets pin 7 LOW
 Serial.println("Master LED is OFF");
 }
 delay(1000);
}
```

Message (Enter to send message to 'Arduino Uno' on 'COM31') New Line 115200 baud

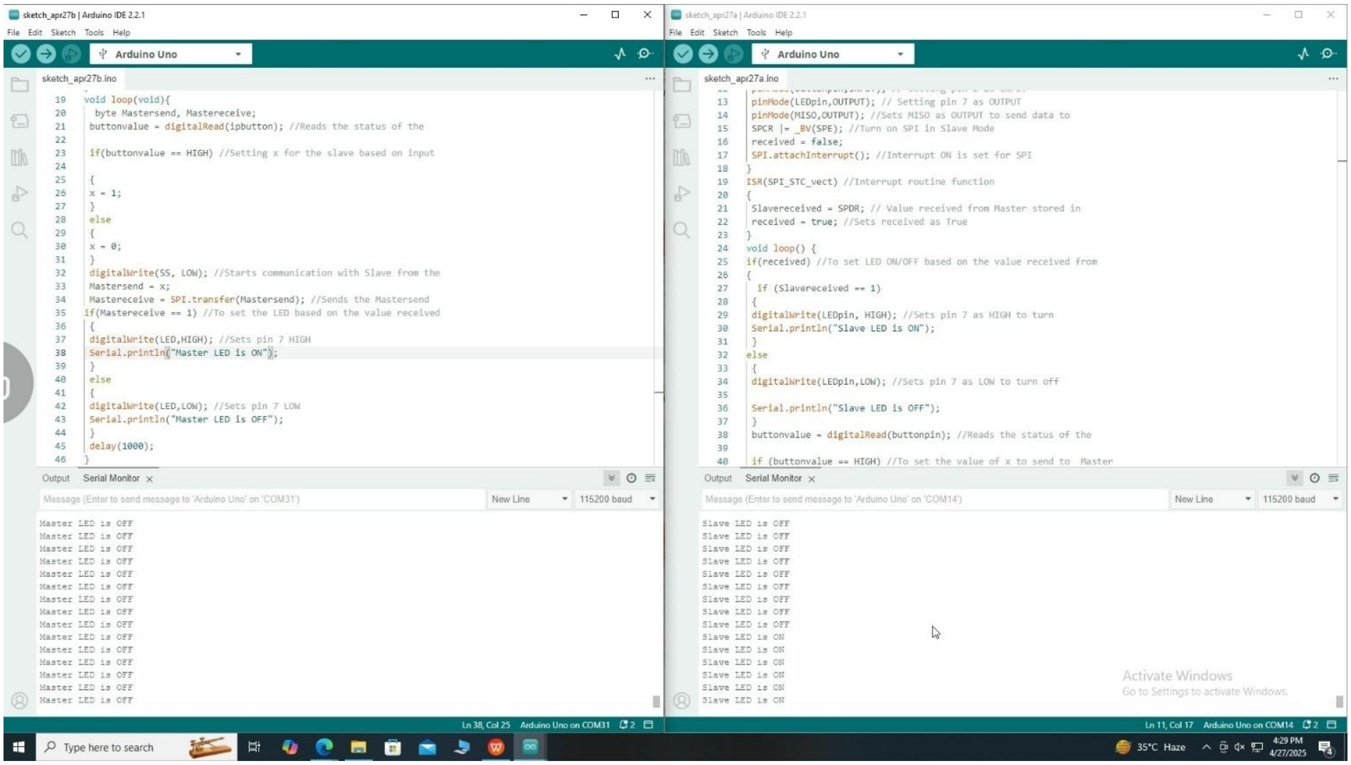
```
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
```

```
 pinMode(LEDpin,OUTPUT); // Setting pin 7 as OUTPUT
 pinMode(MISO,OUTPUT); //Sets MISO as OUTPUT to send data to
 SPCR |= _BV(SPE); //Turn on SPI in Slave Mode
 received = false;
 SPI.attachInterrupt(); //Interrupt ON is set for SPI
}
ISR(SPI_STC_vect) //Interrupt routine function
{
 Slavereceived = SPDR; // Value received from Master stored in
 received = true; //Sets received as True
}
void loop() {
if(received) //To set LED ON/OFF based on the value received from
{
  if (Slavereceived == 1)
 {
 digitalWrite(LEDpin, HIGH); //Sets pin 7 as HIGH to turn
 Serial.println("Slave LED is ON");
 }
else
 {
 digitalWrite(LEDpin,LOW); //Sets pin 7 as LOW to turn off

 Serial.println("Slave LED is OFF");
 }
 buttonvalue = digitalRead(buttonpin); //Reads the status of the

 if (buttonvalue == HIGH) //To set the value of x to send to  Master
```

Message (Enter to send message to 'Arduino Uno' on 'COM14') New Line 115200 baud

```
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
```

Figure 5: Both switches were not pressed


sketch_apr27b | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_apr27b.ino
Output
Serial Monitor
Ln 38, Col 25 Arduino Uno on COM31
sketch_apr27a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_apr27a.ino
Output
Serial Monitor
Activate Windows
Go to Settings to activate Windows.
Ln 11, Col 17 Arduino Uno on COM14
Type here to search
35°C Haze
4:29 PM
4/27/2025

```
void loop(void){
  byte Mastersend, Mastereceive;
 buttonvalue = digitalRead(ipbutton); //Reads the status of the

 if(buttonvalue == HIGH) //Setting x for the slave based on input

 {
 x = 1;
 }
 else
 {
 x = 0;
 }
 digitalWrite(SS, LOW); //Starts communication with Slave from the
 Mastersend = x;
 Mastereceive = SPI.transfer(Mastersend); //Sends the Mastersend
if(Mastereceive == 1) //To set the LED based on the value received
 {
 digitalWrite(LED,HIGH); //Sets pin 7 HIGH
 Serial.println("Master LED is ON");
 }
 else
 {
 digitalWrite(LED,LOW); //Sets pin 7 LOW
 Serial.println("Master LED is OFF");
 }
 delay(1000);
}
```

Message (Enter to send message to 'Arduino Uno' on 'COM31') New Line 115200 baud

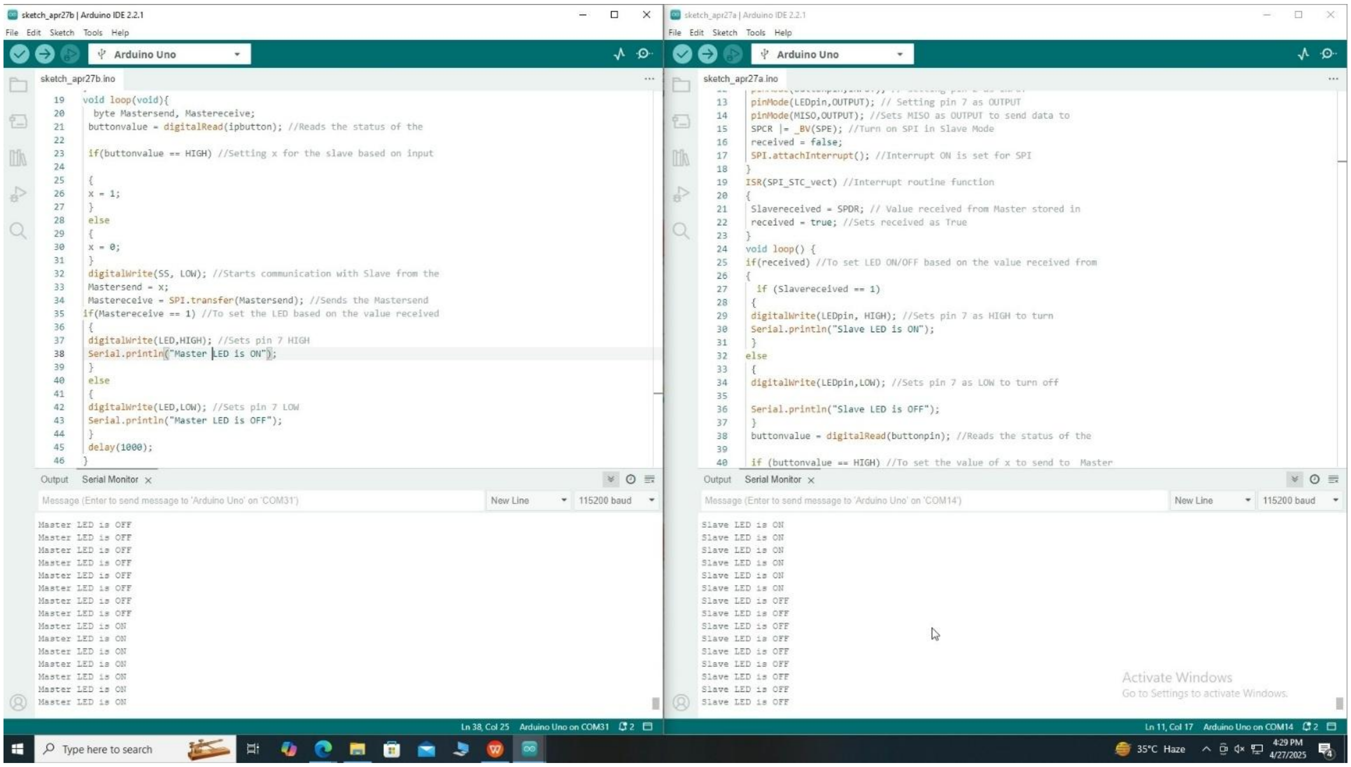
```
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
```

```
 pinMode(LEDpin,OUTPUT); // Setting pin 7 as OUTPUT
 pinMode(MISO,OUTPUT); //Sets MISO as OUTPUT to send data to
 SPCR |= _BV(SPE); //Turn on SPI in Slave Mode
 received = false;
 SPI.attachInterrupt(); //Interrupt ON is set for SPI
}
ISR(SPI_STC_vect) //Interrupt routine function
{
 Slavereceived = SPDR; // Value received from Master stored in
 received = true; //Sets received as True
}
void loop() {
if(received) //To set LED ON/OFF based on the value received from
{
  if (Slavereceived == 1)
 {
 digitalWrite(LEDpin, HIGH); //Sets pin 7 as HIGH to turn
 Serial.println("Slave LED is ON");
 }
else
 {
 digitalWrite(LEDpin,LOW); //Sets pin 7 as LOW to turn off

 Serial.println("Slave LED is OFF");
 }
 buttonvalue = digitalRead(buttonpin); //Reads the status of the

 if (buttonvalue == HIGH) //To set the value of x to send to  Master
```

Message (Enter to send message to 'Arduino Uno' on 'COM14') New Line 115200 baud

```
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is ON
```

Figure 6: Pressing Master Circuit Button



sketch_apr27b | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_apr27b.ino

```
void loop(void){
  byte Mastersend, Mastereceive;
 buttonvalue = digitalRead(ipbutton); //Reads the status of the

 if(buttonvalue == HIGH) //Setting x for the slave based on input

 {
 x = 1;
 }
 else
 {
 x = 0;
 }
 digitalWrite(SS, LOW); //Starts communication with Slave from the
 Mastersend = x;
 Mastereceive = SPI.transfer(Mastersend); //Sends the Mastersend
if(Mastereceive == 1) //To set the LED based on the value received
 {
 digitalWrite(LED,HIGH); //Sets pin 7 HIGH
 Serial.println("Master LED is ON");
 }
 else
 {
 digitalWrite(LED,LOW); //Sets pin 7 LOW
 Serial.println("Master LED is OFF");
 }
 delay(1000);
}
```

```
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is OFF
Master LED is ON
Master LED is ON
Master LED is ON
Master LED is ON
Master LED is ON
Master LED is ON
Master LED is ON
```

Ln 38, Col 25 Arduino Uno on COM31
sketch_apr27a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_apr27a.ino

```
 pinMode(LEDpin,OUTPUT); // Setting pin 7 as OUTPUT
 pinMode(MISO,OUTPUT); //Sets MISO as OUTPUT to send data to
 SPCR |= _BV(SPE); //Turn on SPI in Slave Mode
 received = false;
 SPI.attachInterrupt(); //Interrupt ON is set for SPI
}
ISR(SPI_STC_vect) //Interrupt routine function
{
 Slavereceived = SPDR; // Value received from Master stored in
 received = true; //Sets received as True
}
void loop() {
if(received) //To set LED ON/OFF based on the value received from
{
  if (Slavereceived == 1)
 {
 digitalWrite(LEDpin, HIGH); //Sets pin 7 as HIGH to turn
 Serial.println("Slave LED is ON");
 }
else
 {
 digitalWrite(LEDpin,LOW); //Sets pin 7 as LOW to turn off

 Serial.println("Slave LED is OFF");
 }
 buttonvalue = digitalRead(buttonpin); //Reads the status of the

 if (buttonvalue == HIGH) //To set the value of x to send to  Master
```

```
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is ON
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
Slave LED is OFF
```

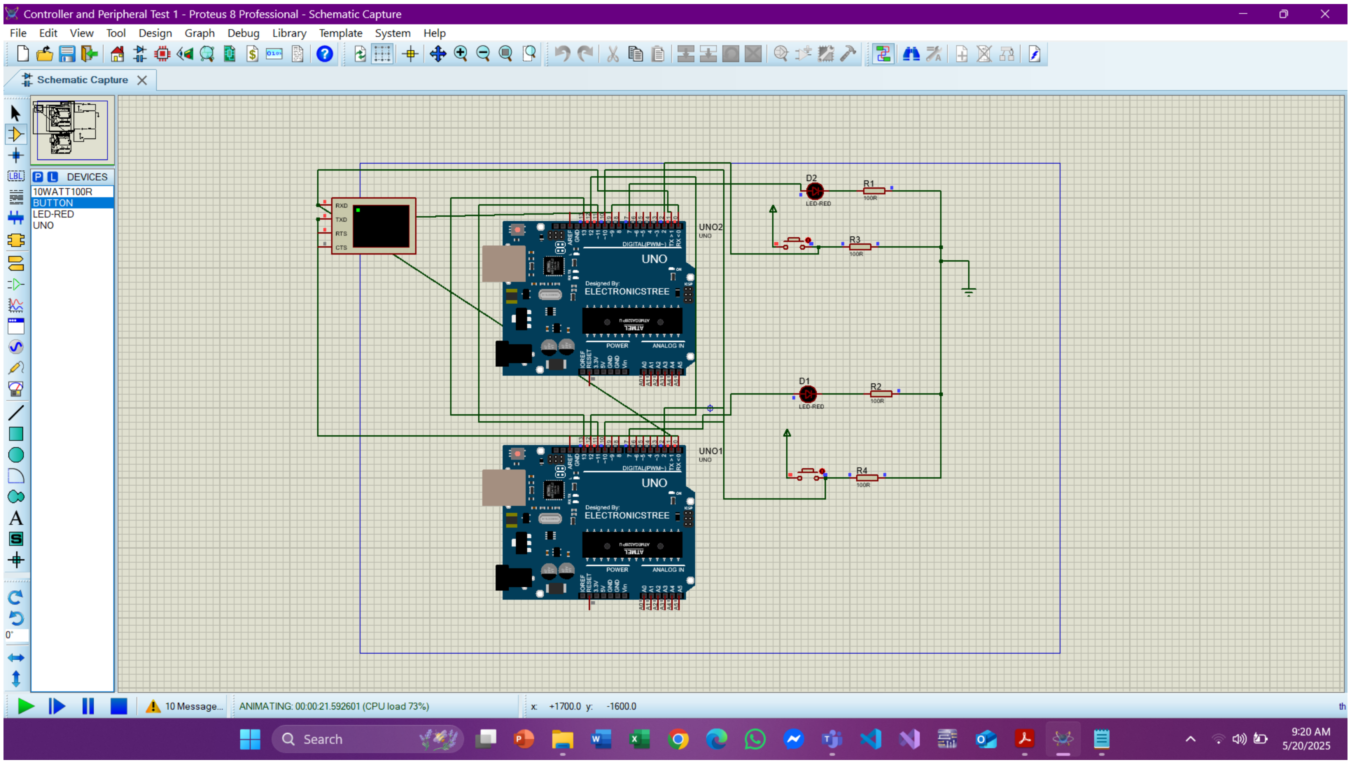
Activate Windows
Go to Settings to activate Windows.
Ln 11, Col 17 Arduino Uno on COM14
Type here to search
35°C Haze
4:29 PM
4/27/2025

Figure 7: Pressing Slave Circuit Button

Using the serial monitor the operation can be seen when the Master led turns on and when the Slave led turns on. If the master circuit button is pressed the Slavereceived variable will be high, and it will print Slave LED is ON and if the slave circuit button is pressed the Mastereceived variable will be high, and it will print Master LED is ON. Otherwise, both LEDs were turned off by default.

Simulation Output Results:

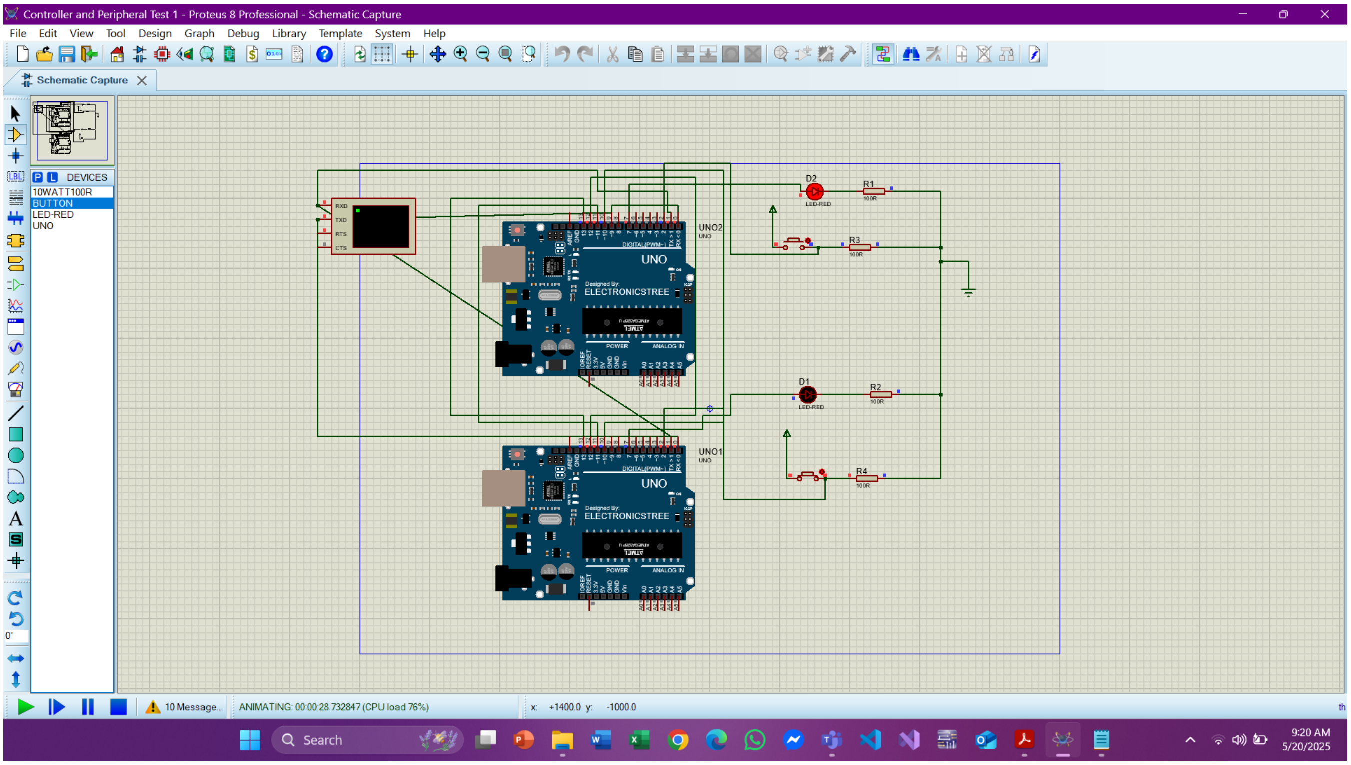
The simulation was done using proteus software. Two Arduino were used one as master(controller) and another one is as slave(peripheral) for SPI communication between two Arduino. Arduinos SS, MISO, MOSI and SCK pin are connected for that communication. Pin 7 is configured as an output pin which is connected to LEDs a pin 2 is configured as input pin which is connected to switch on both circuits. The codes were compiled using Arduino IDE and generates the .hex file for 2 different codes master and slave. The .hex files were uploaded to the respective arduino then the simulation was run successfully.



ANIMATING: 00:00:21.592601 (CPU load 73%)

x: +1700.0 y: -1600.0

Figure 8: Both Circuit LEDs are in off state



ANIMATING: 00:00:28.732847 (CPU load 76%)

x: +1400.0 y: -1000.0

Figure 9: Master Circuit switch pressed and slave circuit LED on.

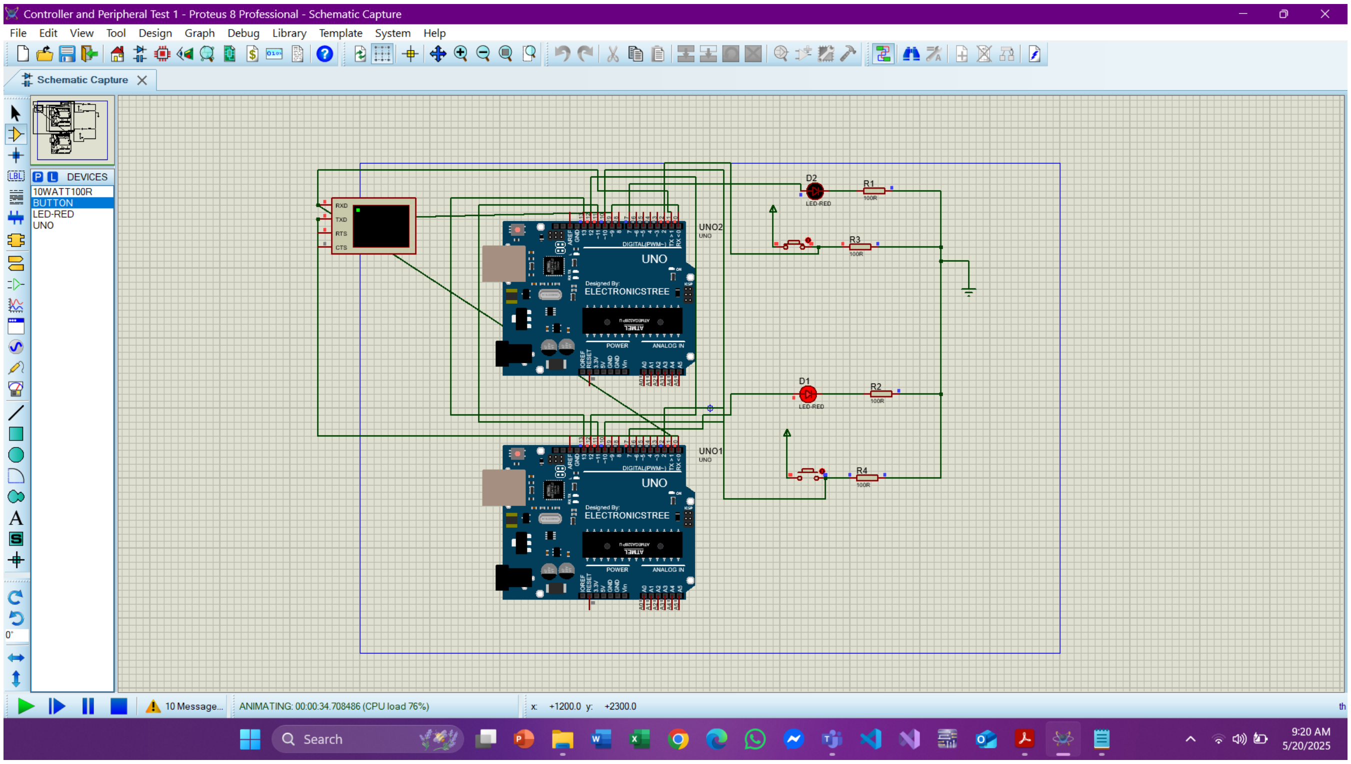


Figure 10: Slave circuit switch pressed and master circuit LED on.

]

Answers to the Questions in the Lab Manual:

1.

Master/Controller Arduino Code:

```
#include<SPI.h> //Library for SPI
#define LED 7
#define ipbutton 2
int buttonvalue;
int x;
void setup (void){
Serial.begin(115200); //Starts Serial Communication at Baud Rate 115200
pinMode(ipbutton,INPUT); //Sets pin 2 as input
pinMode(LED,OUTPUT); //Sets pin 7 as Output
SPI.begin(); //Begins the SPI communication
SPI.setClockDivider(SPI_CLOCK_DIV8); //Sets clock for SPI communication at
// 8 (16/8 = 2 MHz)
digitalWrite(SS,HIGH); //Setting SS to HIGH do disconnect master from slave
}
void loop(void){
byte Mastersend, Mastereceive;
buttonvalue = digitalRead(ipbutton); //Reads the status of the pin 2
if(buttonvalue == HIGH) //Setting x for the slave based on input at pin 2
{
x = 1;
}
else
{
x = 0;
}
digitalWrite(SS, LOW); //Starts communication with Slave from the Master
Mastersend = x;
Mastereceive = SPI.transfer(Mastersend); //Sends the Mastersend value to
//the slave and also receives value from the slave
if(Mastereceive == 1) //To set the LED based on the value received from slave
{
digitalWrite(LED,HIGH); //Sets pin 7 HIGH
Serial.println("Master LED is ON");
}
else
{
digitalWrite(LED,LOW); //Sets pin 7 LOW
Serial.println("Master LED is OFF");
}
delay(1000);
}
```

Slave/Peripherals Arduino Code:

```
#include<SPI.h>
#define LEDpin 7
#define buttonpin 2
volatile boolean received;
volatile byte Slavereceived, Slavesend;
int buttonvalue;
int x;
void setup(){
Serial.begin(115200);
pinMode(buttonpin,INPUT); // Setting pin 2 as INPUT
pinMode(LEDpin,OUTPUT); // Setting pin 7 as OUTPUT
pinMode(MISO,OUTPUT); //Sets MISO as OUTPUT to send data to Master In
SPCR |= _BV(SPE); //Turn on SPI in Slave Mode
received = false;
SPI.attachInterrupt(); //Interrupt ON is set for SPI communication
}
ISR(SPI_STC_vect) //Interrupt routine function
{
Slavereceived = SPDR; // Value received from Master stored in Slavereceived
received = true; //Sets received as True
}
```

```
void loop() {
if(received) //To set LED ON/OFF based on the value received from Master
{
if (Slavereceived == 1)
{
digitalWrite(LEDpin, HIGH); //Sets pin 7 as HIGH to turn on LED
Serial.println("Slave LED is ON");
}
else
{
digitalWrite(LEDpin,LOW); //Sets pin 7 as LOW to turn off LED
Serial.println("Slave LED is OFF");
}
buttonvalue = digitalRead(buttonpin); //Reads the status of the pin 2
if (buttonvalue == HIGH) //To set the value of x to send to Master
{
x = 1;
}
else
{
x=0;
}
Slavesend = x;
SPDR = Slavesend; //Sends the x value to the Master via SPDR
delay(1000);
}
}
```

3.

My ID is 23-51819-2 compared with XY-PQABC-Z, B = 1 and C = 9. That means pin 1 will be configured as input pin and pin 9 will be configured as output pin. After applying these changes the program and output will be,

```
// Program for master device
#include<SPI.h> //Library for SPI
#define LED 9
#define ipbutton 1
int buttonvalue;
int x;
void setup (void){
Serial.begin(115200); //Starts Serial Communication at Baud Rate 115200
pinMode(ipbutton,INPUT); //Sets pin 2 as input
pinMode(LED,OUTPUT); //Sets pin 7 as Output
SPI.begin(); //Begins the SPI communication
SPI.setClockDivider(SPI_CLOCK_DIV8); //Sets clock for SPI communication at
// 8 (16/8 = 2 MHz)
digitalWrite(SS,HIGH); //Setting SS to HIGH do disconnect master from slave
}
void loop(void){
byte Mastersend, Mastereceive;
buttonvalue = digitalRead(ipbutton); //Reads the status of the pin 2
if(buttonvalue == HIGH) //Setting x for the slave based on input at pin 2
{
x = 1;
}
else
{
x = 0;
}
digitalWrite(SS, LOW); //Starts communication with Slave from the Master
Mastersend = x;
Mastereceive = SPI.transfer(Mastersend); //Sends the Mastersend value to
//the slave and also receives value from the
if(Mastereceive == 1) //To set the LED based on the value received from slave
{
digitalWrite(LED,HIGH); //Sets pin 7 HIGH
Serial.println("Master LED is ON");
}
else
{
digitalWrite(LED,LOW); //Sets pin 7 LOW
Serial.println("Master LED is OFF");
}
delay(1000);
}
```

Program for Slave device:

```
#include<SPI.h>
#define LEDpin 9
#define buttonpin 1
volatile boolean received;
volatile byte Slavereceived, Slavesend;
int buttonvalue;
int x;
void setup(){
Serial.begin(115200);
pinMode(buttonpin,INPUT); // Setting pin 2 as INPUT
pinMode(LEDpin,OUTPUT); // Setting pin 7 as OUTPUT
pinMode(MISO,OUTPUT); //Sets MISO as OUTPUT to send data to Master In
SPCR |= _BV(SPE); //Turn on SPI in Slave Mode
received = false;
SPI.attachInterrupt(); //Interrupt ON is set for SPI communication
}
```

```
ISR(SPI_STC_vect) //Interrupt routine function
{
Slavereceived = SPDR; // Value received from Master stored in Slavereceived
received = true; //Sets received as True
}
void loop() {
if(received)
{ //To set LED ON/OFF based on the value received from Master
if (Slavereceived == 1)
{
digitalWrite(LEDpin, HIGH); //Sets pin 7 as HIGH to turn on LED
Serial.println("Slave LED is ON");
}
else
{
digitalWrite(LEDpin,LOW); //Sets pin 7 as LOW to turn off LED
Serial.println("Slave LED is OFF");
}
buttonvalue = digitalRead(buttonpin); //Reads the status of the pin 2
if (buttonvalue == HIGH) //To set the value of x to send to Master
{
x = 1;
}
else
{
x=0;
}
Slavesend = x;
SPDR = Slavesend; //Sends the x value to the Master via SPDR
delay(1000);
}
}
```

The Simulation result is,

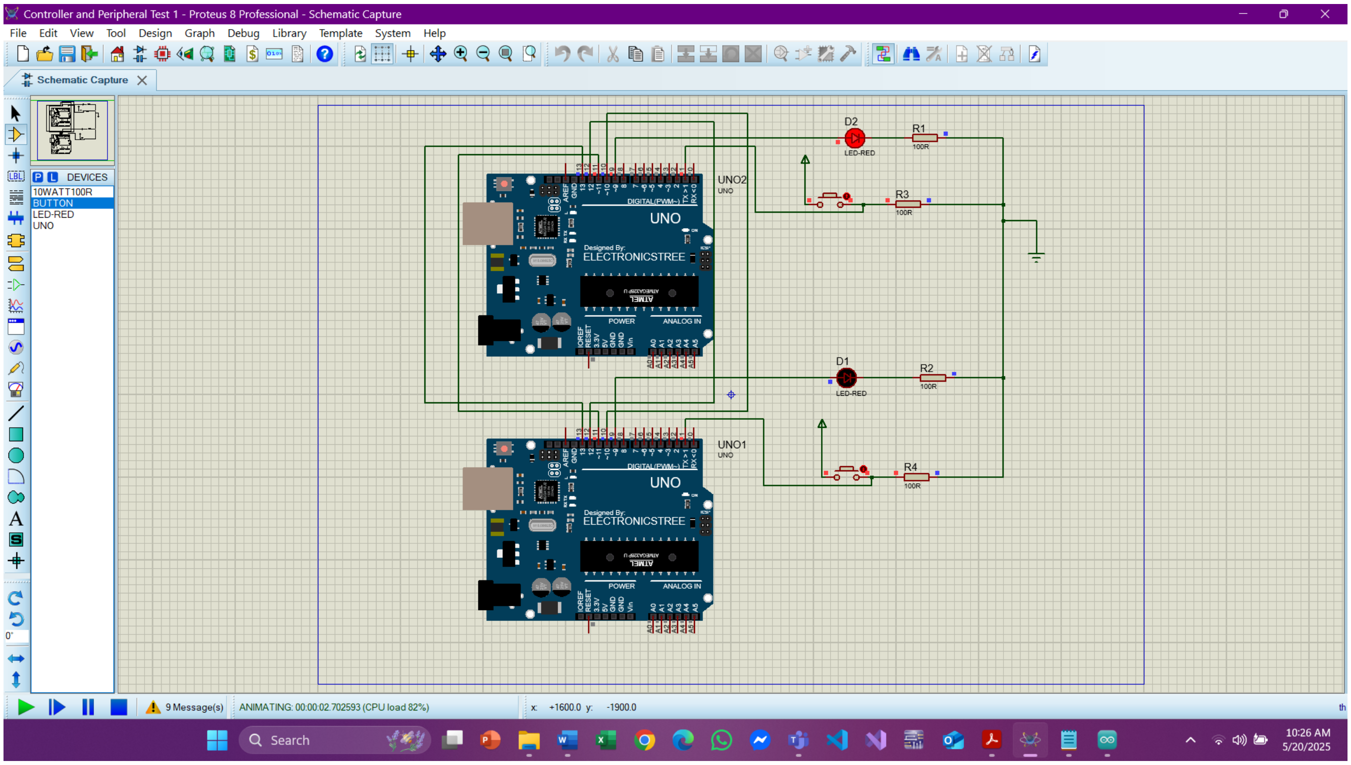

Controller and Peripheral Test 1 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
10WATT100R
BUTTON
LED-RED
UNO
D2
LED-RED
R1
100R
R3
100R
UNO2
UNO
D1
LED-RED
R2
100R
UNO1
UNO
R4
100R
9 Message(s)
ANIMATING: 00:00:02.702593 (CPU load 82%)
x: +1600.0 y: -1900.0
Search
10:26 AM
5/20/2025

Figure 11: Without pressing and switch the master device LED blink with 2 second delay

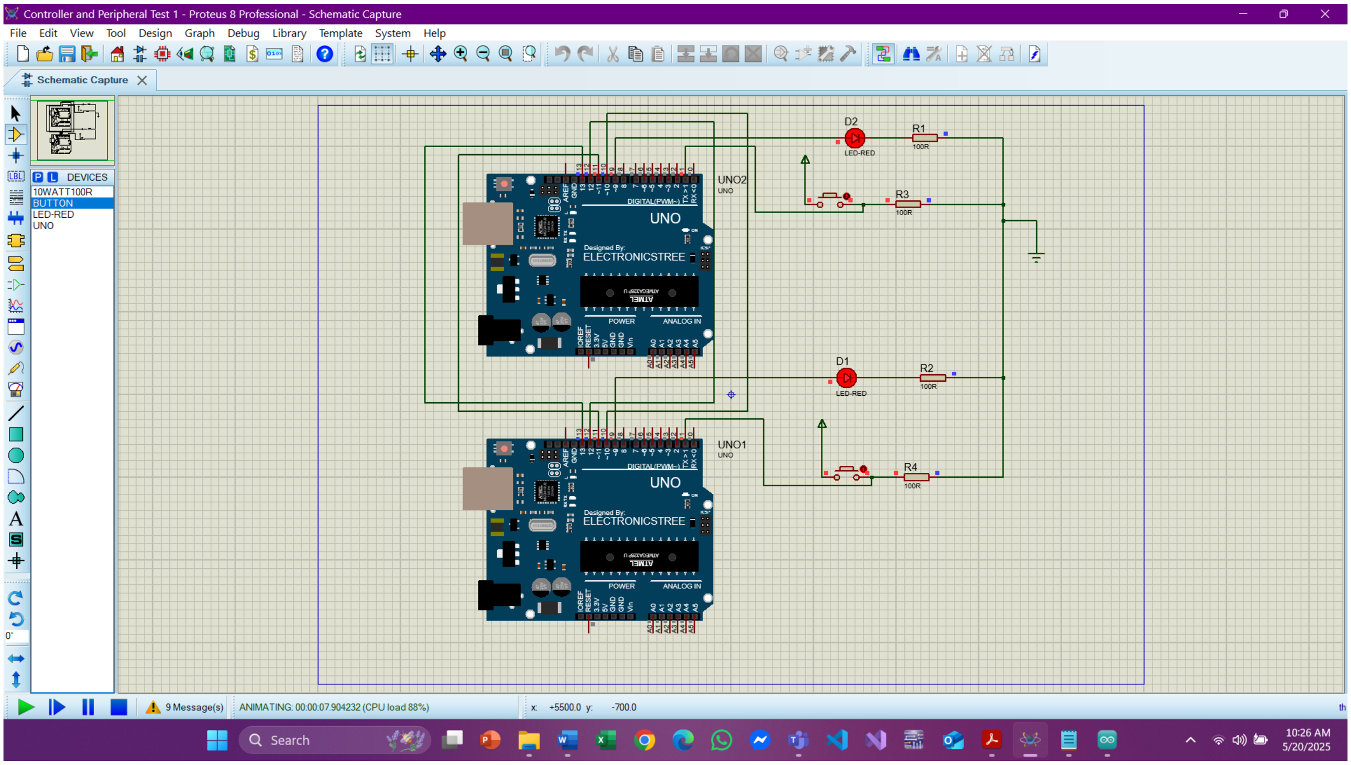

Controller and Peripheral Test 1 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
10WATT100R
BUTTON
LED-RED
UNO
D2
LED-RED
R1
100R
R3
100R
UNO2
UNO
D1
LED-RED
R2
100R
UNO1
UNO
R4
100R
9 Message(s)
ANIMATING: 00:00:07.904232 (CPU load 88%)
x: +5500.0 y: -700.0
Search
10:26 AM
5/20/2025

Figure 12: After pressing the Master device switch the Master Device LED keeps in ON state.

In the simulation there is no effect of slave switch. The slave device LED is always turned on.

Discussion:

In this lab experiment, SPI communication was implemented between two Arduino boards, where one was configured as the master and the other as the slave. The objective was to control the LED on the master board using a push button on the slave board, and vice versa. The SPI pins (MOSI, MISO, SCK, and SS) were properly connected, and code was written to transmit button states and control LEDs accordingly. It was observed that pressing the slave's button caused the master's LED to toggle, while the master's button was used to control the slave's LED in return. Through this experiment, a fundamental understanding of SPI communication was gained, including the master-driven data transfer and the importance of synchronization. Real-life applications such as home automation, modular robotics, and industrial control systems were identified, where similar microcontroller communication techniques are often used. Overall, valuable practical experience was acquired in the field of embedded system communication.

References:

[1] https://www.arduino.cc/.

[2] ATMega328 manual

[3] https://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-timers

[4] http://maxembedded.com/2011/06/avr-timers-timer0/