|  |
| --- |
| ICBRAMR  **School of Electronics Engineering (SENSE)**  **B. Tech – Electronics & Computer Engineering**  **BECE403E – EMBEDDED SYSTEM DESIGN**  **LAB RECORD**  **(L15+L16)**  **Submitted By**  **21BLC1228 – Mayukh Ray**    **Submitted To**  **Dr. Manoj Kumar Rajagopal**  **DATE: 29/02/2024**  **Slot:** L15+L16  **Date: 29/2/2024**  **LAB – 08: Working with SPI**  **AIM:**  Implement and verify the logic on the STM32 Nucleo-64 board using Keil Studio Cloud IDE.  **Software Required:** ARM Keil Studio (Mbed Online Compiler)  **Hardware Required:** Micro USB cable, NUCLEO64-STM32L152 Board, LEDs, Jumper Wires (M-F and M-M), Breadboard  **Procedure:**   1. Go to ARM Keil Studio (<https://studio.keil.arm.com>) and log in 2. Select File → New → Mbed Project 3. Click the Example project drop-down list and select “mbed2-example-blinky” 4. In Project name field, provide the name of the new project and click Add project 5. Double click on the “main.cpp” file from the newly created project folder 6. Modify the code in the editor window as per the logic of your application 7. Check for any errors in the program under the “Problems” tab of the panels window 8. If no errors, connect the Nucleo Board to the computer using Micro USB Cable 9. Click Play icon (Run project) to upload and start the code execution on the board.   **PROGRAM:**  **Lab Task 1:**  Write a program to implement a SPI communication between two Nucleo boards. Configure one of the Nucleo as master and other as slave. Establish a SPI communication between master and slave display each key press on the master’s Teraterm to the slave Teraterm terminal.  **Code:**  **Program (Master)**  #include “mbed.h”  SPI spi(PB\_15,PB\_14,PB\_13);  DigitalOut cs(PB\_12);  Serial pc(USBTX,USBRX);  int main(){  char send\_val;  pc.printf(“Press any key to start…\n”);  while(1){  send\_val=pc.getc();  pc.printf(“%c”,send\_val);  cs=0;  spi.write(send\_val);  cs=1;  wait(0.01);  }  }  **Program (Slave)**  #include “mbed.h”  SPISlave spi(PB\_15,PB\_14,PB\_13,PB\_12);  Serial pc(USBTX,USBRX);  Char recd\_val;  int main(){  pc.printf(“Received word is…\n”);  while(1){  if(spi.receive()){  recd\_val=spi.read();  pc.printf(“%c”,recd\_val);  }  }  }  **Output:**        **Output Verification:**    **Lab Task 2:**  Write a program to implement a SPI communication between two Nucleo boards. Configure one of the Nucleo as master and other as slave. Both Nucleo are attached with a LED & a push button separately. Master LED can be controlled by using slave Nucleo’s push button and slave Nucleo’s LED can be controlled by master Nucleo’s push button using SPI communication protocol.  **Code:**  **Program (Master)**  #include “mbed.h”  SPI ser\_port(PB\_15,PB\_14,PB\_13);  Digitalout led(PC\_8);  DigitalIn switch\_ip(PC\_4);  DigitalOut cs(PB\_12);  char switch\_word;  char recd\_val;  int main()  {  while(1)  {  switch\_word=0xa0;  if(switch\_ip==1)  switch\_word=switch\_word|0x01;  cs=0;  recd\_val=ser\_port.write(switch\_word);  cs=1;  wait(0.01);  led=0;  recd\_val=recd\_val&0x01;  if(recd\_val==1)  led=1;  }  }  **Program (Slave)**  #include “mbed.h”  SPISlave ser\_port(PB\_15,PB\_14,PB\_13,PB\_12);  DigitalOut led(PC\_8);  DigitalIn switch\_ip(PC\_4);  char recd\_val;  int main()  {  while(1)  {  switch\_word=0xa0;  if(switch\_ip==1)  switch\_word=switch\_word|0x01;  if(ser\_port.receive())  {  recd\_val=ser\_port.read();  ser\_port.reply(switch\_word);  }  led=0;  recd\_val=recd\_val&0x01;  if(recd\_val==1)  led=1;  }  }  **Output:**      **Output Verification:**      **INFERENCE:**   1. We need to identify the pin connected to the LED. 2. Write the program to toggle the state of the LED between ON and OFF in a loop. 3. Compile and upload the code to the microcontroller.   **RESULT:**  **Lab-1:**  Hence, we were able to implement a SPI communication between two Nucleo boards. Configured one of the Nucleo as master and other as slave. Establish a SPI communication between master and slave display each key press on the master’s Teraterm to the slave Teraterm terminal.  **Lab-2:**  Hence, we were able to implement a SPI communication between two Nucleo boards. Configured one of the Nucleo as master and other as slave. Both Nucleo are attached with a LED & a push button separately. Master LED can be controlled by using slave Nucleo’s push button and slave Nucleo’s LED can be controlled by master Nucleo’s push button using SPI communication protocol. |