

Smart Home Project

❖ Table of contents:

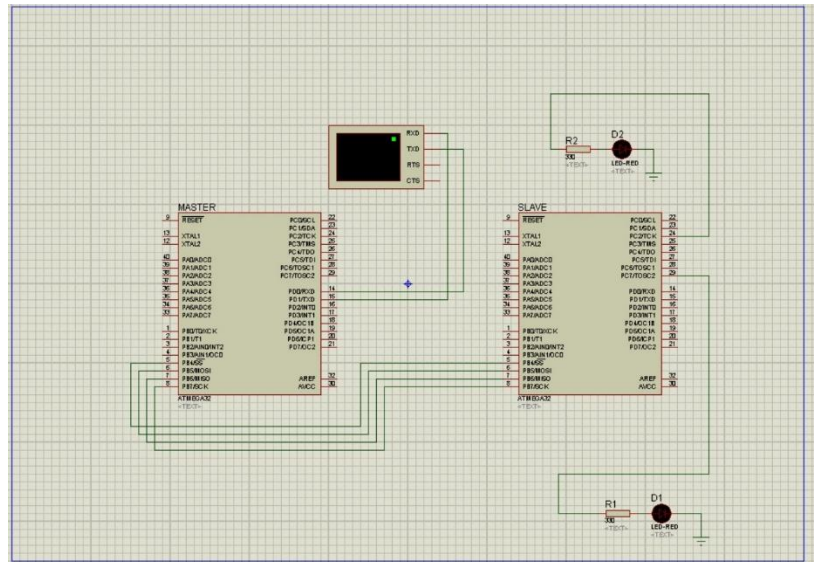
1. Circuit Explanation.
2. Code Explanation.
 - a) Master Code.
 - b) Slave Code.
3. Flowchart of the project.

1. Circuit Explanation:

As shown in the opposite figure:

There are 2 microcontrollers:
Master and Slave.

- Master MC is connected to a virtual terminal to simulate the Bluetooth module through TX and RX pins.
- Slave MC is connected to 2 LEDs to simulate home appliances.
- Also, the master and slave are connected to each other through MISO, MOSI, SS, SCK pins to communicate with each other through SPI Protocol.



2. Code Explanation:

a) Master Code:

As shown in the figure: the code is built using layered architecture concept and has a module for each driver.

First, the initialization of Master SPI and UART and set the directions of pins in portC as output. creating a string to be

sent to the user "Please Enter a Button", and send it to Bluetooth module using `MCAL_UART_sendByteBusyWait` function, and finally create two variables to send and receive them through UART and SPI.

In the program loop:

Master receives the char from Bluetooth using UART receive function.

then, we check on the char if it's '0' then we turn the LEDs connected to the output pins off by set their values to LOW as a kind of testing that master MC receive '0' and we send the char to the slave MC using `SPI_Master_DataBusyWait` function.

The same steps are done if the char is '1'.

```
1 #include "STD_TYPES.h"
2 #include "BIT_MATH.h"
3 #include "AVR_DIO_REG.h"
4 #include "DIO_interface.h"
5 #include "Compiler.h"
6 #include "AVR_UART_REG.h"
7 #include "UART_interface.h"
8 #include "AVR_SPI_REG.h"
9 #include "SPI_interface.h"
10
11 int main(){
12     str_uartConfig_t obj = {9600, UART_8_BIT_MODE, UART_POLLING};
13     MCAL_SPI_MasterInit(SPI_MASTER_FREQUENCY_64);
14     MCAL_UART_init(&obj);
15     MCAL_DIO_uSetPinDir(DIO_PORTC, DIO_PIN_2, DIO_OUTPUT);
16     MCAL_DIO_uSetPinDir(DIO_PORTC, DIO_PIN_7, DIO_OUTPUT);
17     uint8_t arr[] = "Please Enter a Button:";
18     uint8_t i = 0;
19     while (arr[i] != '\0'){
20         MCAL_UART_sendByteBusyWait(arr[i]);
21         i++;
22     }
23     uint8_t chr;
24     uint8_t recieve;
25     while(1)
26     {
27         MCAL_UART_recieveByteBusyWait(&chr);
28         if (chr == '0')
29         {
30             MCAL_DIO_uSetPinValue(DIO_PORTC, DIO_PIN_2, DIO_LOW); //to check that the program went through this condition
31             MCAL_DIO_uSetPinValue(DIO_PORTC, DIO_PIN_7, DIO_LOW);
32             MCAL_SPI_Master_DataBusyWait(chr, &recieve);
33         }
34         else if (chr == '1')
35         {
36             MCAL_DIO_uSetPinValue(DIO_PORTC, DIO_PIN_2, DIO_HIGH); //to check that the program went through this condition
37             MCAL_DIO_uSetPinValue(DIO_PORTC, DIO_PIN_7, DIO_HIGH);
38             MCAL_SPI_Master_DataBusyWait(chr, &recieve);
39         }
40     }
41 }
42
```

b) Slave Code:

As shown in the figure:
First initialization of SPI of Slave MC and also the LEDs.

Declare two variables to send and receive through SPI with Master MC.

In the loop:

Slave MC receive the data from Master MC

through SPI_Slave_DataBusyWait function and check on the data:

If the received char is '0' then will turn LEDs off.

else if received char '1' then will turn LEDs on.

```
1 #include "STD_TYPES.h"
2 #include "BIT_MATH.h"
3 #include "AVR_DIO_REG.h"
4 #include "DIO_interface.h"
5 #include "Compiler.h"
6 #include "AVR_UART_REG.h"
7 #include "UART_interface.h"
8 #include "AVR_SPI_REG.h"
9 #include "SPI_interface.h"
10 #include "LED_interface.h"
11
12 int main(){
13     MCAL_SPI_SlaveInit();
14     HAL_LEDInit(DIO_PORTC,DIO_PIN_2);
15     HAL_LEDInit(DIO_PORTC,DIO_PIN_7);
16     uint8_t recieve_char;
17     uint8_t send = NULL;
18     while(1){
19         MCAL_SPI_Slave_DataBusyWait(send,&recieve_char);
20         if(recieve_char == '0'){
21             HAL_LEDSetValue(DIO_PORTC,DIO_PIN_2,DIO_LOW);
22             HAL_LEDSetValue(DIO_PORTC,DIO_PIN_7,DIO_LOW);
23         }
24         else if (recieve_char == '1'){
25             HAL_LEDSetValue(DIO_PORTC,DIO_PIN_2,DIO_HIGH);
26             HAL_LEDSetValue(DIO_PORTC,DIO_PIN_7,DIO_HIGH);
27         }
28     }
29 }
30
31
```

3. Flowchart of the project:

