AMIT Graduation project

2021

Name : Omar Mohamed Hamza

Email : Omar.mhamza98@gmail.com

Group : Q10

Introduction:

Technology is the most common language in the world now and the human mindset prefers things to be automated, and the best place People can have this advantage is their homes , number of smart homes in massively increasing where people can control their home appliances and machines by their smart phones and this lead to save time and effort

Abstract:

This presentation will illustrate a simple smart home project where

The user can open and close room lamps

Methodology:

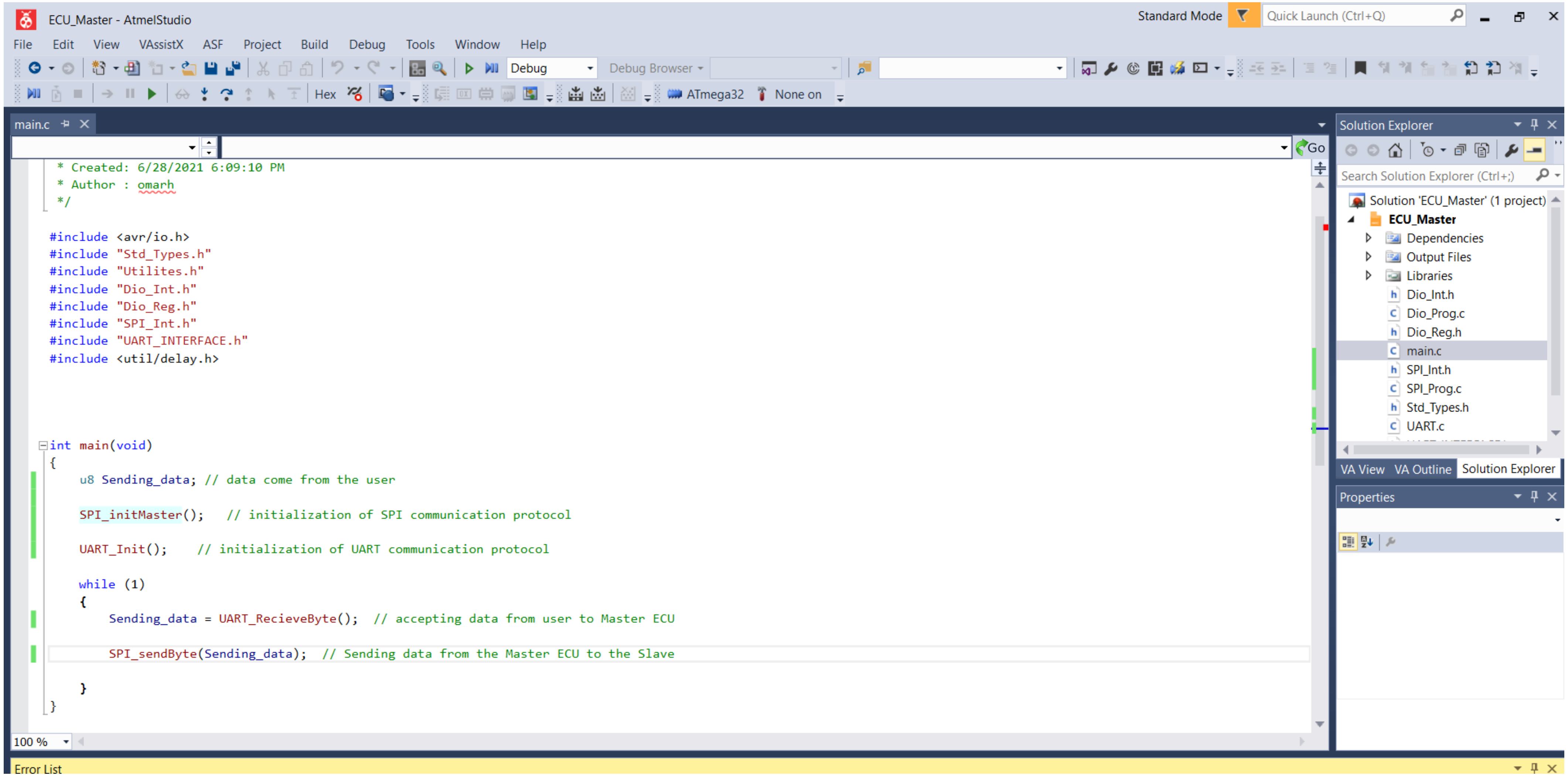
This project consist of two master and slave ECUs communicate with each other by SPI protocol and the master ECU communicate with the Bluetooth module (Virtual Terminal) by UART

in this project we use drivers that helps the code to be optimized and clean

Used drivers : (Std_Types, utilities, DIO, UART, SPI)

Master code

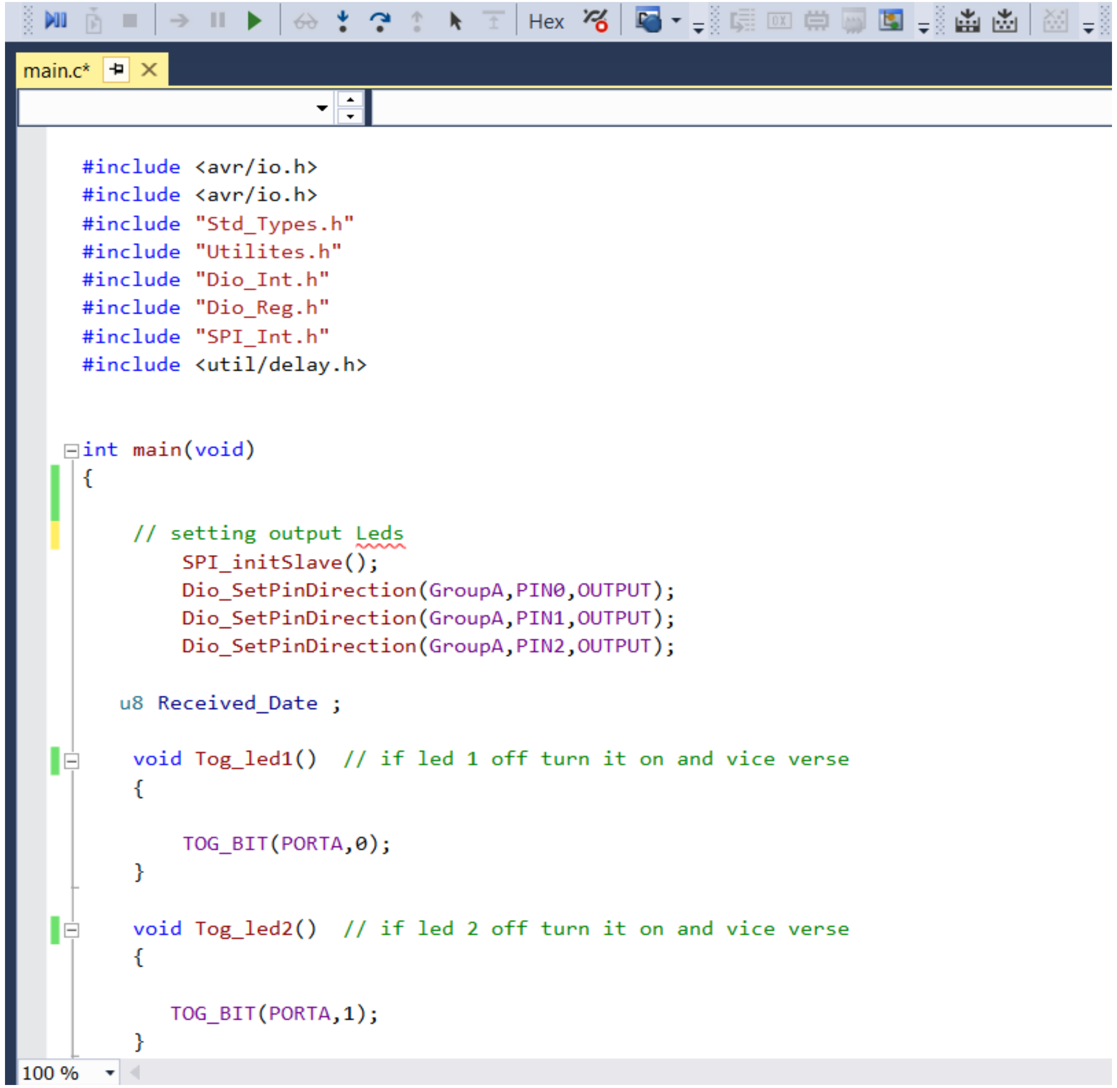
ECU Master code



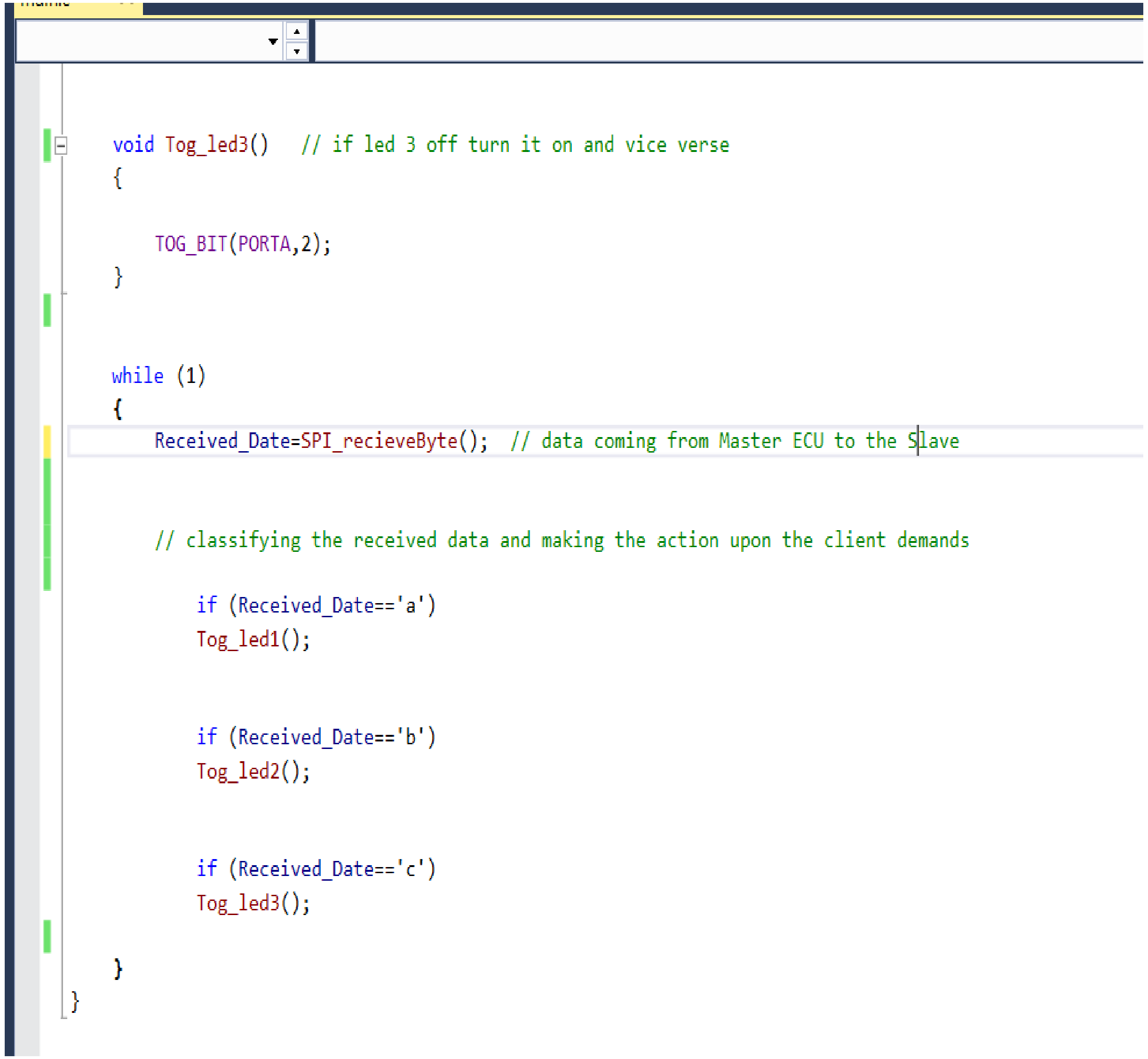
```c
 * Created: 6/28/2021 6:09:10 PM
 * Author : omarh
 */

#include <avr/io.h>
#include "Std_Types.h"
#include "Utilites.h"
#include "Dio_Int.h"
#include "Dio_Reg.h"
#include "SPI_Int.h"
#include "UART_INTERFACE.h"
#include <util/delay.h>



int main(void)
{
    u8 Sending_data; // data come from the user

    SPI_initMaster();   // initialization of SPI communication protocol

    UART_Init();    // initialization of UART communication protocol

    while (1)
    {
        Sending_data = UART_RecieveByte();  // accepting data from user to Master ECU

        SPI_sendByte(Sending_data);  // Sending data from the Master ECU to the Slave

    }
}
```

SLAVE CODE

ECU Slave code

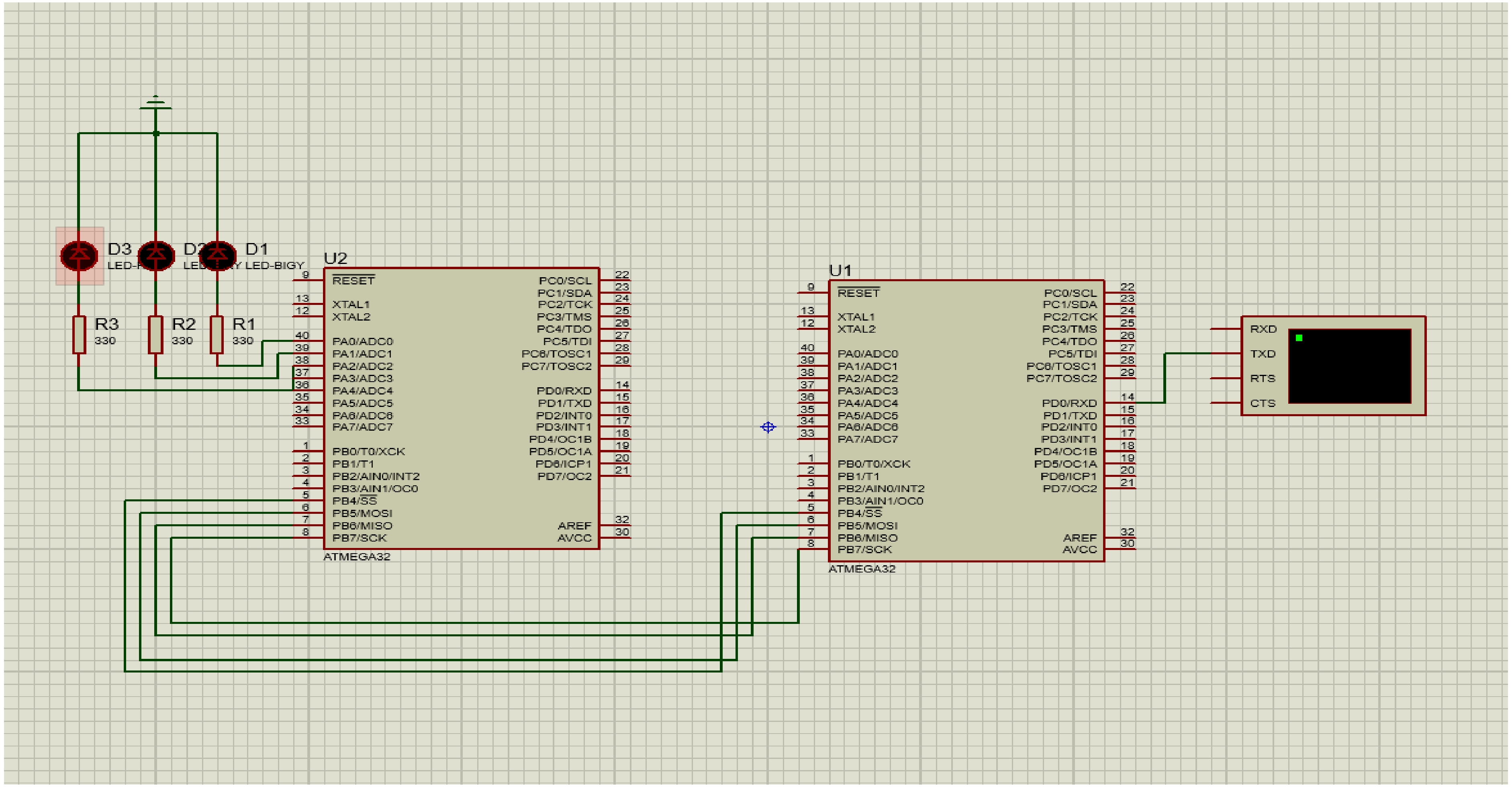


```c
#include <avr/io.h>
#include <avr/io.h>
#include "Std_Types.h"
#include "Utilites.h"
#include "Dio_Int.h"
#include "Dio_Reg.h"
#include "SPI_Int.h"
#include <util/delay.h>


int main(void)
{

    // setting output Leds
        SPI_initSlave();
        Dio_SetPinDirection(GroupA,PIN0,OUTPUT);
        Dio_SetPinDirection(GroupA,PIN1,OUTPUT);
        Dio_SetPinDirection(GroupA,PIN2,OUTPUT);

   u8 Received_Date ;

    void Tog_led1()  // if led 1 off turn it on and vice verse
    {

        TOG_BIT(PORTA,0);
    }

    void Tog_led2()  // if led 2 off turn it on and vice verse
    {

       TOG_BIT(PORTA,1);
    }
```

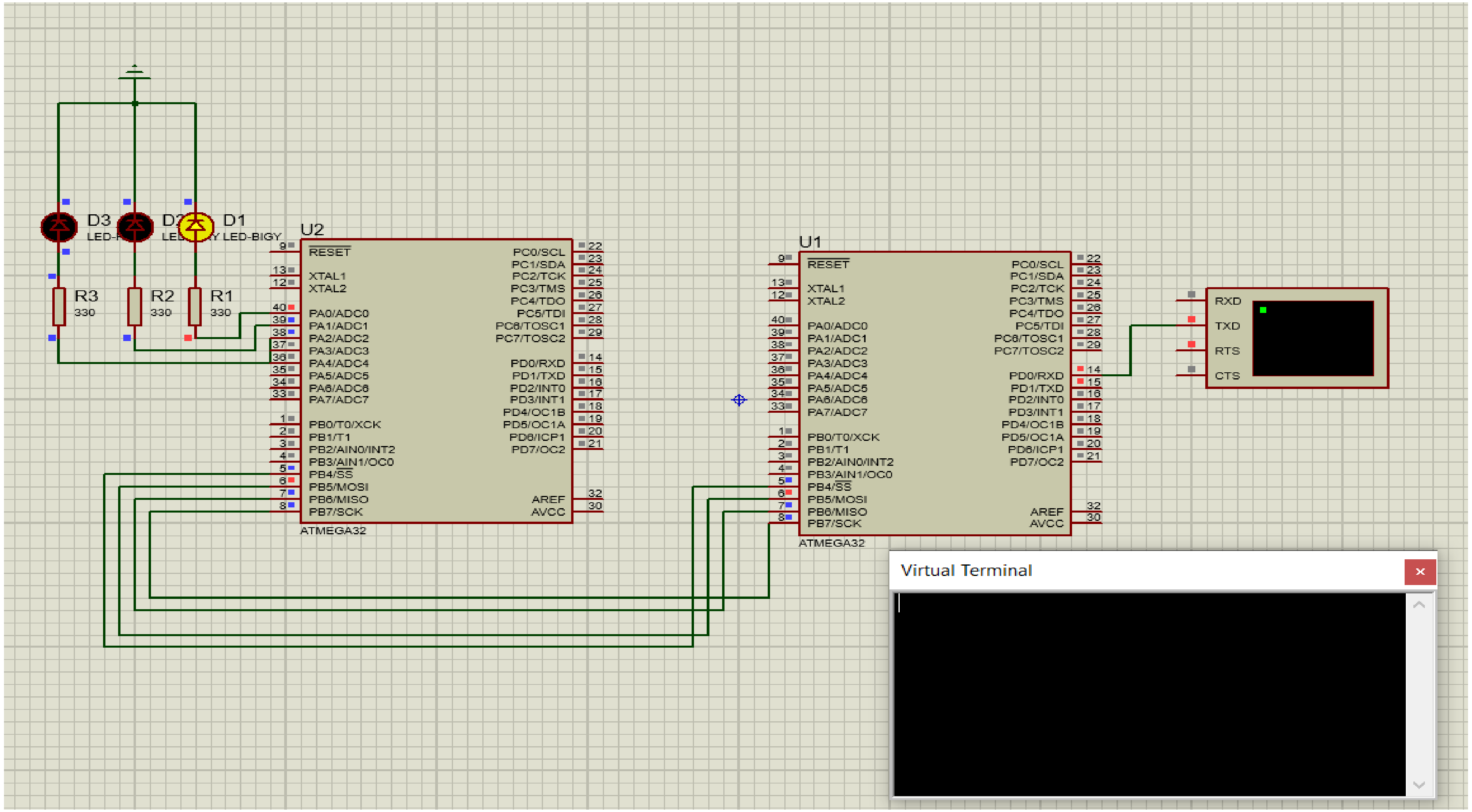


```c
    void Tog_led3()   // if led 3 off turn it on and vice verse
    {

        TOG_BIT(PORTA,2);
    }


    while (1)
    {
        Received_Date=SPI_recieveByte();  // data coming from Master ECU to the Slave


        // classifying the received data and making the action upon the client demands

            if (Received_Date=='a')
            Tog_led1();


            if (Received_Date=='b')
            Tog_led2();


            if (Received_Date=='c')
            Tog_led3();

    }
}
```

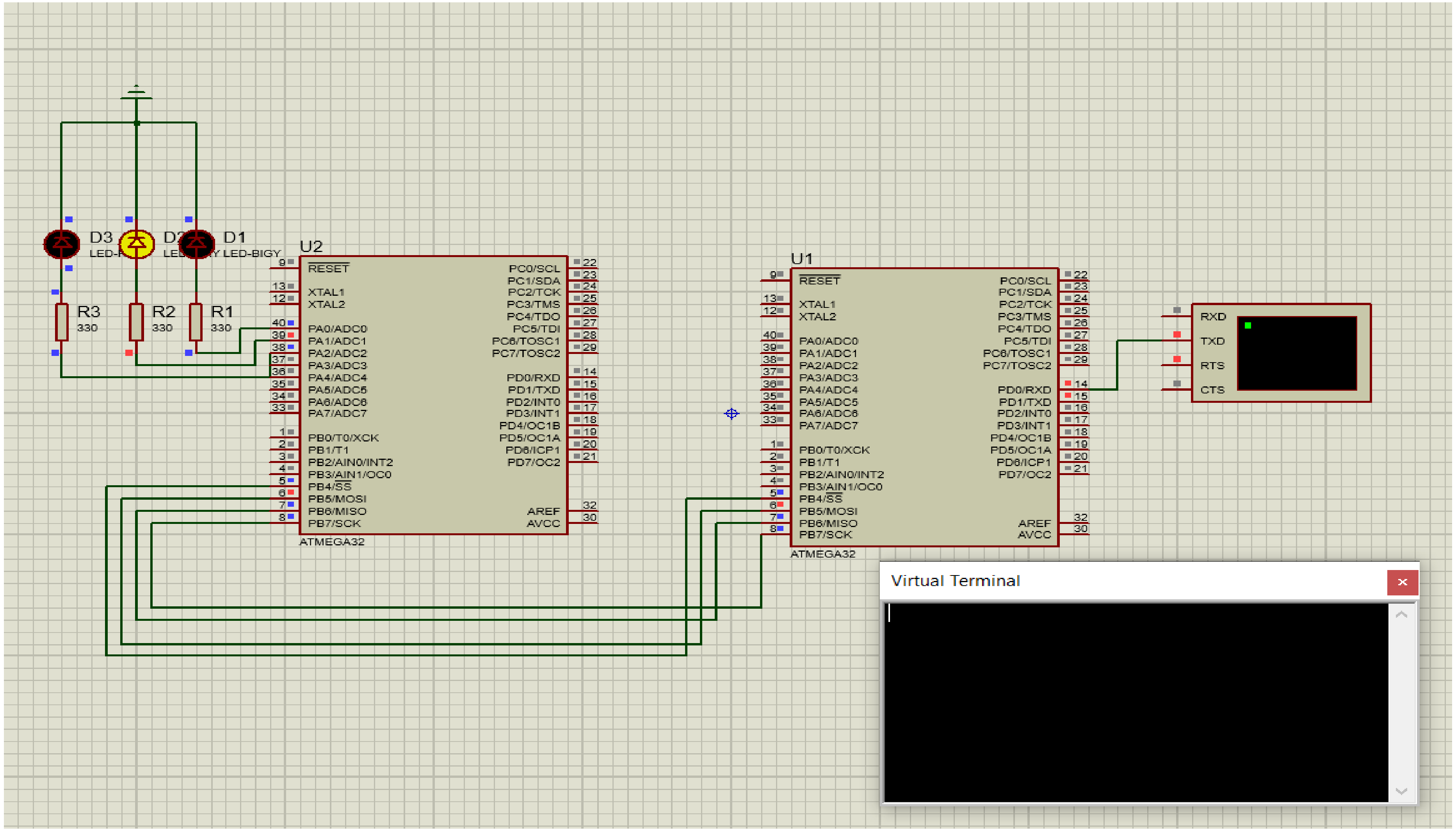
Simulation on proteus



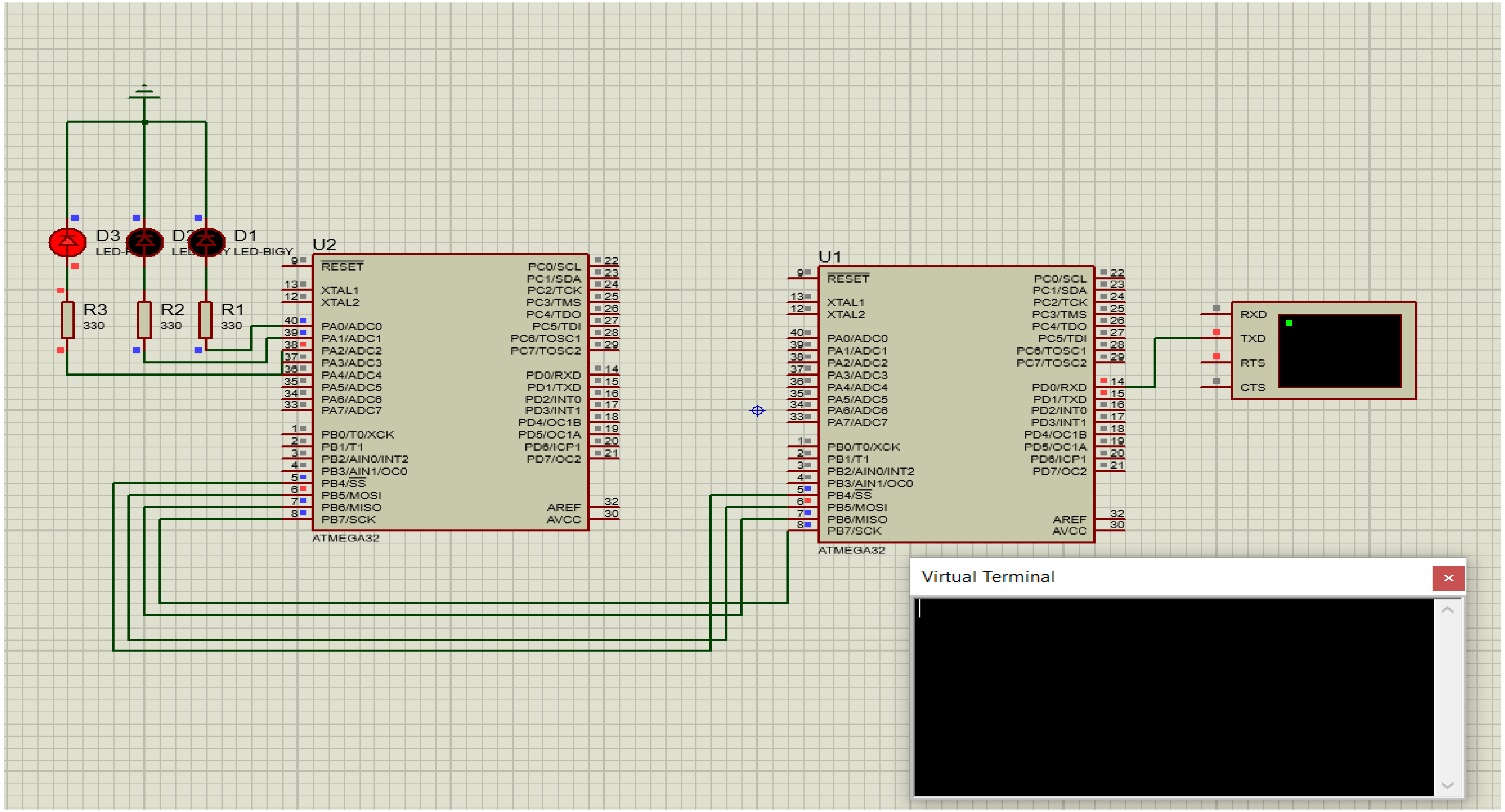
If user send ‘a’ led 1 turn on and off



If user send ‘b’ led 2 turn on and off



If user send ‘c’ led 3 turn on and off


D3
D2
D1
LED-BIGY
R3
330
R2
330
R1
330
U2
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA32
U1
ATMEGA32
RXD
TXD
RTS
CTS
Virtual Terminal
×