AMIT Project (Smart Home)

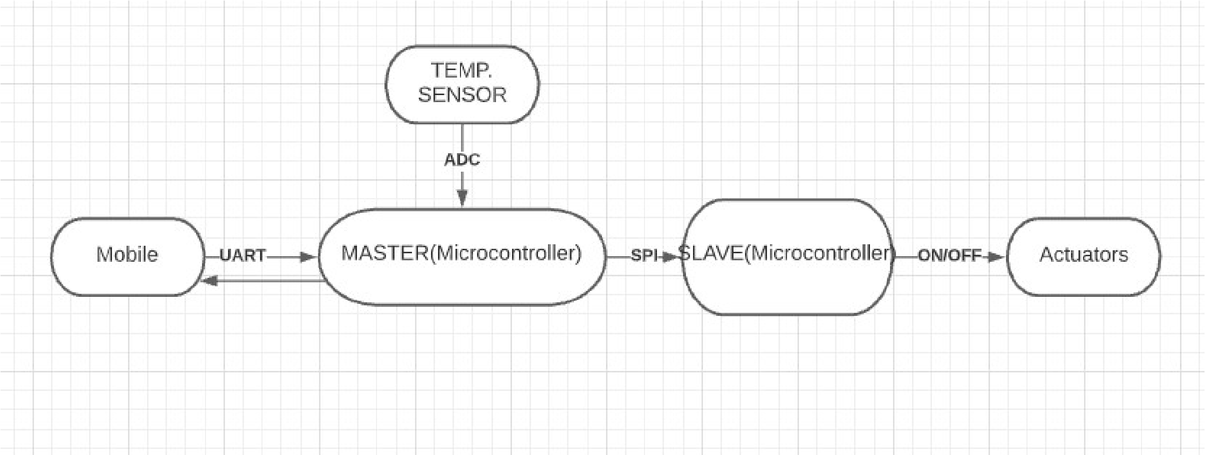
Name: Eslam Adham Attia Saleh

Email: eslam_maghed@hotmail.com

Mobile: 01229926486

Group: N-04

DESCRIPTION:



The system consists of Master and Slave micro-controller which communicate with each other. The Master microcontroller is connected with Bluetooth HC-05 by using UART in order to take orders from mobile from the user. The first controller reads temperature from temp. Sensor using ADC.

The Slave microcontroller is used to control the actuators by getting commands from the master Micro-controller by using SPI communication protocol.

The actuators are Two LEDS and FAN

Two LEDS work when get commands from Mobile

FAN works when temp. sensor read more than 30 degree Celsius.

Command are chars:

a →open RED led

b → open GREEN led

c →Close RED LED

d → Close Green LED

e →Open FAN

f →Close FAN

x→ Close system (Emergency Stop)

s→ Start system

CODE:

Master microcontroller code

```
 /*
          * Final PROJECT.c
          *
          * Created: 1/15/2021 8:24:11 PM
          * Author : HP
          */


         #include <avr/io.h>
         #include "UART.h"
         #include "ADC.h"
         #include "Timer.h"
         #include "SPI.h"
         #include <avr/interrupt.h>
         static int fanflag =0;
         int main(void)
         {
             /* Replace with your application code */
                DDRB |= (1<<4) | (1<<5) | (1<<7);
                DDRB &= ~(1<<6);
                ADC_inti(0,1,7);

                TIMER_Inti();
                SPI_inti(1);
                UART_inti();


                while(1){
                       char a=UART_RECIVE();
                       if(a=='x' & fanflag==0){
                              SPI_SEND(a);
                              fanflag=1;
                       }
                       else if(a=='s' & fanflag==1){
                              SPI_SEND(a);
                              fanflag=0;
                       }
                       else if(fanflag==0){
                              SPI_SEND(a);

                       }
                       }}
```

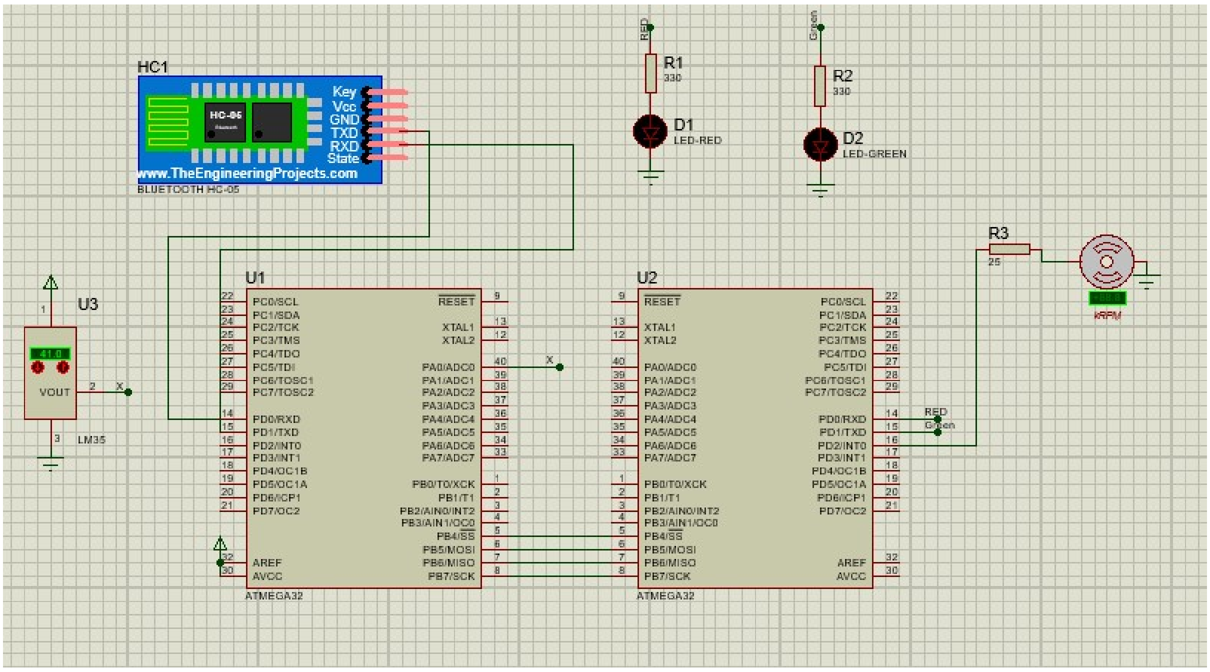
```
ISR(TIMER0_OVF_vect){
                                static int x=0;

                                x++;
                                if(x==152 & fanflag==0){
                                        int a=ADC_READ();

                                        if(a>=30 ){
                                                SPI_SEND('e');

                                        }
                                        else if(a<30){
                                                SPI_SEND('f');

                                        }

                                        UART_SEND_NUMB(a);
                                        x=0;

                                }
                                else if(x==152){
                                        x=0;
                                }



                        }
```

SLAVE microcontroller code:

```
#include
<avr/io.h>
             #include "SPI.h"
             #include "UART.h"


             int main(void)
             {
                    SPI_inti(0);

                    DDRB |= (1<<6);
                    DDRB &= ~(1<<4);
                    DDRB &= ~(1<<5);
                    DDRB &= ~(1<<7);
                    DDRD =0xff;
                 /* Replace with your application code */
                 while (1)
                 {
                           char test = SPI_RECIVE();
                           switch(test){
                                  case 'a':
                                                      PORTD |=(1<<0);
                                                      break;
                                  case 'b':
                                                      PORTD |=(1<<1);
                                                      break;
                                  case 'c':
                                                      PORTD &=~(1<<0);
                                                      break;
                                  case 'd':
                                                      PORTD &=~(1<<1);
                                                      break;

                                  case 'e':
                                                      PORTD |=(1<<2);
                                                      break;
                                  case 'f':
                                                      PORTD &=~(1<<2);


                                                      break;

                                  case 'x':
                                                      PORTD =0;
```

```
                                            break;
               }


    }
}
```

Proteus Simulations:



Simulation video:

https://drive.google.com/drive/folders/1jShxjU9b-rc4Aj3Uv4xlflTYJQwCwtAo?usp=sharing

GitHUB LINK:

https://github.com/eslamasaleh/AMIT_PROJECT