Seven Segment Display and 4x4 keypad

ASSIGNMET – 04

BASATI SIVAKRISHNA

22976

DESE EPD

In this assignment we are adding more commands, seven segment display and keypad support. The 1st two keys available in EDUARM board is used to change the state of the machine to pause or resume and start or stop.

The first two digits in SSD display are used to indicate the number of times we changed the led colour, and fourth digit of SSD display is used to indicate the type of the led blinking.

Adding command support via Console:

```
}

if((strcmp(parsed_cmnd,"color")==0))
    command_type = 1;
else if(((strcmp(parsed_cmnd,"blink")==0)))
    command_type = 2;
else if(((strcmp(parsed_cmnd,"pause")==0)))
    command_type = 3;
else if(((strcmp(parsed_cmnd,"start")==0)))
    command_type = 4;
else if(((strcmp(parsed_cmnd2,"resume")==0)))
    command_type = 5;
else if(((strcmp(parsed_cmnd1,"stop")==0)))
    command_type = 6;
else
    command_type = 7;
```

Based on the entered command the command_type variable will be modified. This entire comparison is going iin command_parser() function(explained in previous assignment).

Once the command_type varible gets its value then further validation of command will be done according to the command entered. If the entered command is invalid, then following message will be printed.

```
if(valid == 0){
label: printstring("**********  ENTERED COMMAND IS INVALID  *******\n\r");
printstring("ENTER THE FOLLOWING COMMANDS ONLY\n\r");
printstring("1. blink blink_rate        2. color color_type\n\r");
printstring("3. pause                   4. resume\n\r");
printstring("5. stop                    6. start\n\r");
valid = 0;
```

Using Seven Segment Display:

The four digits connected to four pins of port A and data of 8 bits are connected to Port B. So to display a digit on single display then we need to write 1 into particular A port pin and enter coded hex data into port B. The written data should be toggled very often to use all the digits of SSD. Data to displayed is stored in posX_data , and the following code will be executed continuously in delayMs() loop.

```
//refresh display

    GPIO_PORTA_DATA_R &= ~(0xF0);
    GPIO_PORTB_DATA_R = 0;
    GPIO_PORTA_DATA_R |= (1<<(4));
    GPIO_PORTB_DATA_R = pos0_data;
    for(k = 0; k < 30; k++) {
    }

    GPIO_PORTA_DATA_R &= ~(0xF0);
    GPIO_PORTB_DATA_R = 0;
    GPIO_PORTA_DATA_R |= (1<<(6));
    GPIO_PORTB_DATA_R = pos2_data;
    for(k = 0; k < 30; k++) {
    }
    GPIO_PORTA_DATA_R &= ~(0xF0);
    GPIO_PORTB_DATA_R = 0;
    GPIO_PORTA_DATA_R |= (1<<(7));
    GPIO_PORTB_DATA_R = pos3_data;
    for(k = 0; k < 30; k++) {
    }
//read keypad
```

Interfacing Keypad:

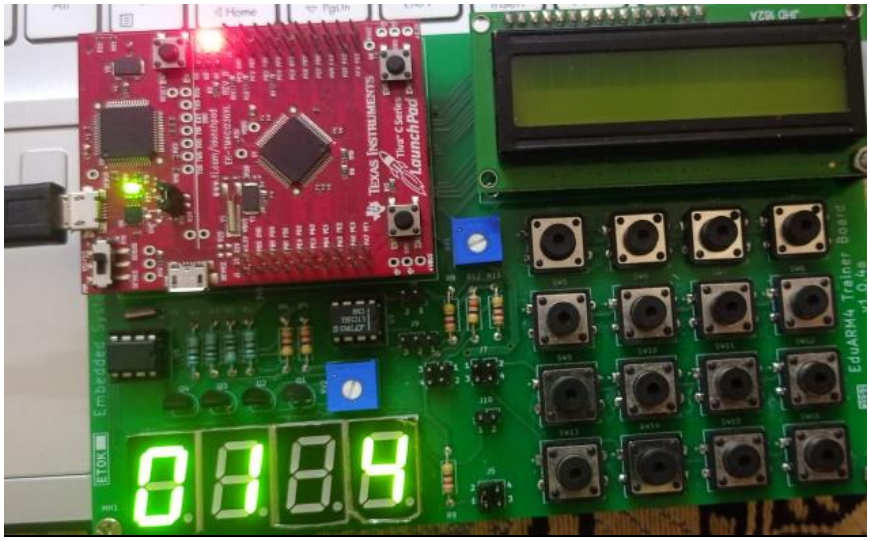
We need to initialize the keypad ports by using keypad_init() function available in keypad.h file and by using iskeypressed() and readkey () functions we can access the when any key is pressed. Based on the keypress the start_stop and resume_pause variables will toggle their values between 1 and 0.

```
//read keypad
    if(isKeyPressed())
    {
        wait(40);

        if(isKeyPressed())
        {
            data = readkey();
            if(data == 2)
            {
                (start_stop == 1)?(start_stop = 0):(start_stop = 1);
            }
            else if(data == 1)
                (resume_pause == 1)?(resume_pause= 0):(resume_pause= 1);
        }
    }
```

Results:

SSD DISPLAY:



COMMANDS:

```
COM4
Initialization done
Entered command is: colorred
||-------------STATUS OF THE MACHINE--------------||
COMMAND GIVEN ----> colorred
LED CODE ---->  0
Entered command is: colorgreenen
||-------------STATUS OF THE MACHINE--------------||
COMMAND GIVEN ----> colorgreen
LED CODE ---->  3
Entered command is: pausee
||-------------STATUS OF THE MACHINE--------------||
COMMAND GIVEN ----> pause
LED CODE ---->  0
Entered command is: resume
||-------------STATUS OF THE MACHINE--------------||
COMMAND GIVEN ----> resume
LED CODE ---->  0
Entered command is: startt
||-------------STATUS OF THE MACHINE--------------||
COMMAND GIVEN ----> start
LED CODE ---->  0
Entered command is: stopp
||-------------STATUS OF THE MACHINE--------------||
COMMAND GIVEN ----> stop
LED CODE ---->  0
Entered command is: starpp
**********  ENTERED COMMAND IS INVALID  *******
ENTER THE FOLLOWING COMMANDS ONLY
1. blink blink_rate        2. color color_type
3. pause                   4. resume
5. stop                    6. start
```