Meenakshi Shankar
Sr No. 22400
M.Tech EPD

EMBEDDED SYSTEM DESIGN (E3-257)

LAB ASSIGNMENT – 5

Explanation of Code

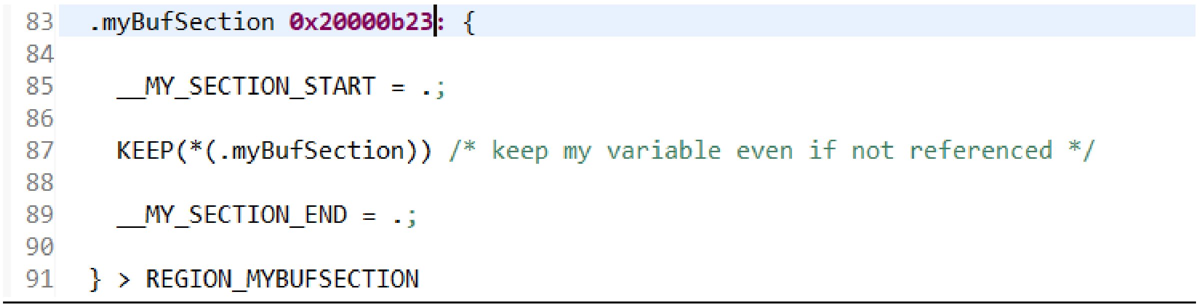
Apart from the functionalities implemented until lab 3 following new additions were made:

1. Creating a new section in Linker script

Following additions were made to the .lds file of the project to customize the memory mapping

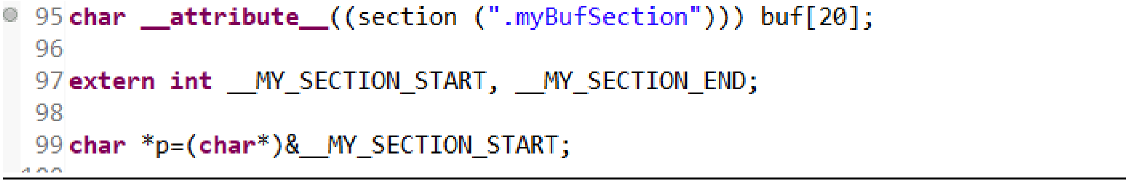
```
REGION_ALIAS("REGION_MYBUFSECTION", SRAM);
```

The new section I am creating is myBufSection within SRAM at a default start of 0x20000b23



```
  .myBufSection 0x20000b23: {

    __MY_SECTION_START = .;

    KEEP(*(.myBufSection)) /* keep my variable even if not referenced */

    __MY_SECTION_END = .;

  } > REGION_MYBUFSECTION
```

A new variable storing string is assigned to the newly created section as follows from the main.c



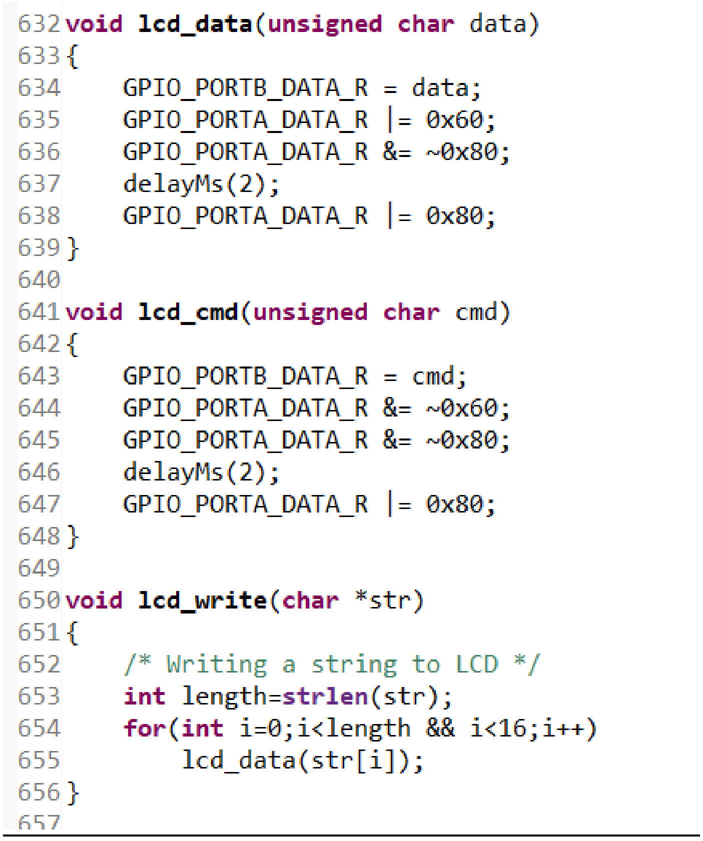
```
char __attribute__((section (".myBufSection"))) buf[20];

extern int __MY_SECTION_START, __MY_SECTION_END;

char *p=(char*)&__MY_SECTION_START;
```

```
__MY_SECTION_START, __MY_SECTION_END;
```

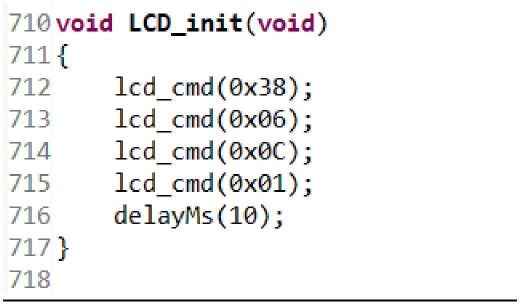
These denote the start and end of the section in SRAM that is assigned.

2. Working and Initializing the LCD.

Following functions are used to print data of implement lcd commands to clear or go next line etc..

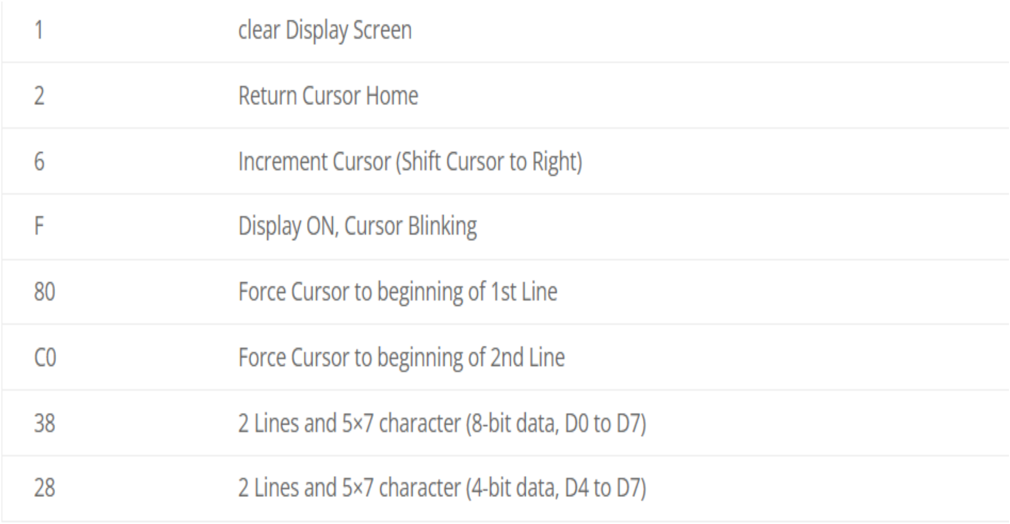


```
void lcd_data(unsigned char data)
{
    GPIO_PORTB_DATA_R = data;
    GPIO_PORTA_DATA_R |= 0x60;
    GPIO_PORTA_DATA_R &= ~0x80;
    delayMs(2);
    GPIO_PORTA_DATA_R |= 0x80;
}

void lcd_cmd(unsigned char cmd)
{
    GPIO_PORTB_DATA_R = cmd;
    GPIO_PORTA_DATA_R &= ~0x60;
    GPIO_PORTA_DATA_R &= ~0x80;
    delayMs(2);
    GPIO_PORTA_DATA_R |= 0x80;
}

void lcd_write(char *str)
{
    /* Writing a string to LCD */
    int length=strlen(str);
    for(int i=0;i<length && i<16;i++)
        lcd_data(str[i]);
}
```



```
void LCD_init(void)
{
    lcd_cmd(0x38);
    lcd_cmd(0x06);
    lcd_cmd(0x0C);
    lcd_cmd(0x01);
    delayMs(10);
}
```

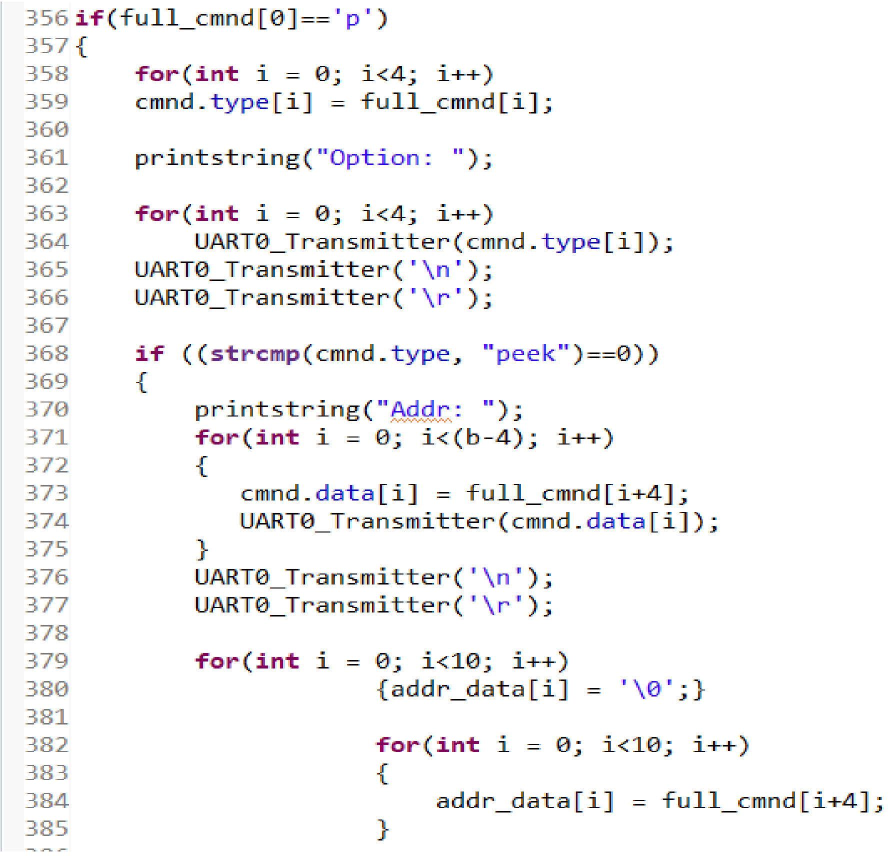
The initializing is guided as following



1	clear Display Screen
2	Return Cursor Home
6	Increment Cursor (Shift Cursor to Right)
F	Display ON, Cursor Blinking
80	Force Cursor to beginning of 1st Line
C0	Force Cursor to beginning of 2nd Line
38	2 Lines and 5×7 character (8-bit data, D0 to D7)
28	2 Lines and 5×7 character (4-bit data, D4 to D7)

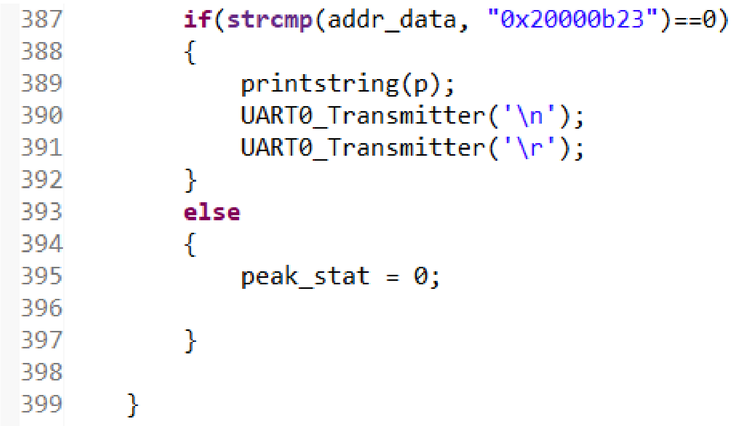
3. Implementing "Peek" functionality

First the uart console command is processed as required to separate the command and the address for the string.



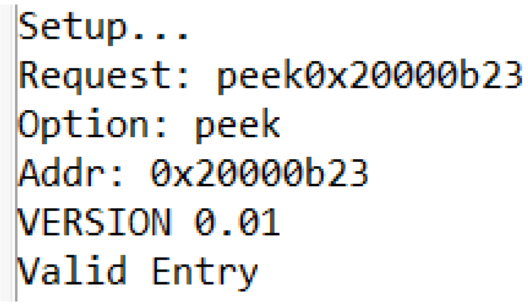
```
if(full_cmnd[0]=='p')
{
    for(int i = 0; i<4; i++)
    cmnd.type[i] = full_cmnd[i];

    printstring("Option: ");

    for(int i = 0; i<4; i++)
        UART0_Transmitter(cmnd.type[i]);
    UART0_Transmitter('\n');
    UART0_Transmitter('\r');

    if ((strcmp(cmnd.type, "peek")==0))
    {
        printstring("Addr: ");
        for(int i = 0; i<(b-4); i++)
        {
            cmnd.data[i] = full_cmnd[i+4];
            UART0_Transmitter(cmnd.data[i]);
        }
        UART0_Transmitter('\n');
        UART0_Transmitter('\r');

        for(int i = 0; i<10; i++)
                    {addr_data[i] = '\0';}

                    for(int i = 0; i<10; i++)
                    {
                        addr_data[i] = full_cmnd[i+4];
                    }
```

Then the comparison to the address were we initialized the string is done.

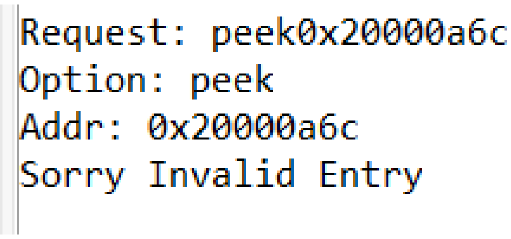


```
        if(strcmp(addr_data, "0x20000b23")==0)
        {
            printstring(p);
            UART0_Transmitter('\n');
            UART0_Transmitter('\r');
        }
        else
        {
            peak_stat = 0;

        }

    }
```

If the address is not valid pointer to string stored it will be flagged using the peak_stat.



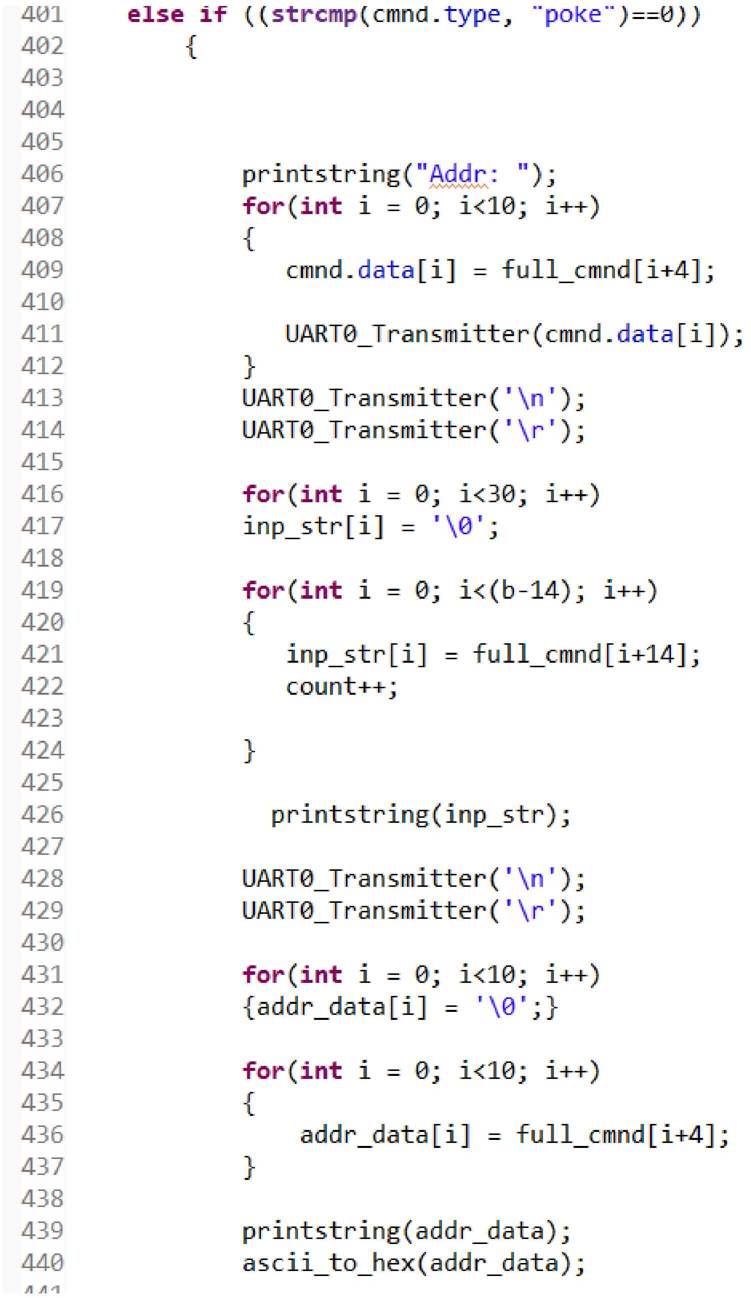
```
Setup...
Request: peek0x20000b23
Option: peek
Addr: 0x20000b23
VERSION 0.01
Valid Entry
```



```
Request: peek0x20000a6c
Option: peek
Addr: 0x20000a6c
Sorry Invalid Entry
```

4. Implementing “Poke” functionality

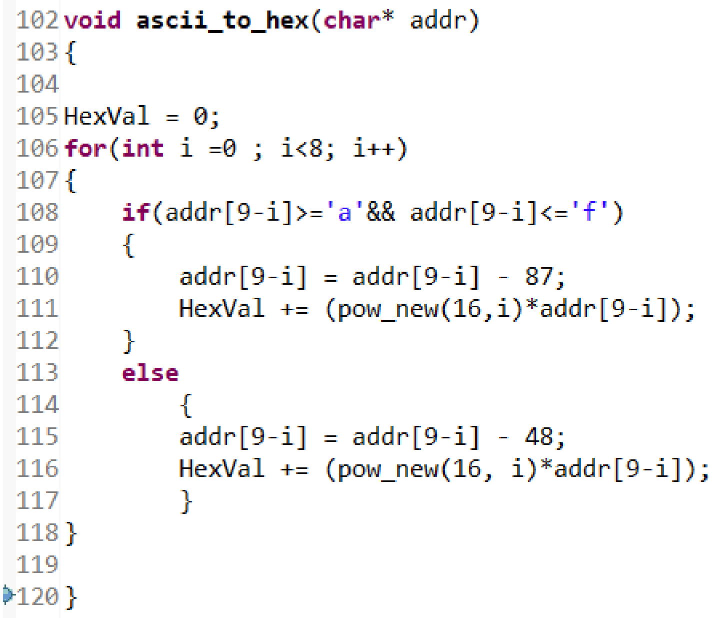
First step was to accept input from uart console and split to command , address from were change is to be reflected and input string to alter into the present region.



```
else if ((strcmp(cmnd.type, "poke")==0))
    {



        printstring("Addr: ");
        for(int i = 0; i<10; i++)
        {
           cmnd.data[i] = full_cmnd[i+4];

           UART0_Transmitter(cmnd.data[i]);
        }
        UART0_Transmitter('\n');
        UART0_Transmitter('\r');

        for(int i = 0; i<30; i++)
        inp_str[i] = '\0';

        for(int i = 0; i<(b-14); i++)
        {
           inp_str[i] = full_cmnd[i+14];
           count++;

        }

          printstring(inp_str);

        UART0_Transmitter('\n');
        UART0_Transmitter('\r');

        for(int i = 0; i<10; i++)
        {addr_data[i] = '\0';}

        for(int i = 0; i<10; i++)
        {
            addr_data[i] = full_cmnd[i+4];
        }

        printstring(addr_data);
        ascii_to_hex(addr_data);
```

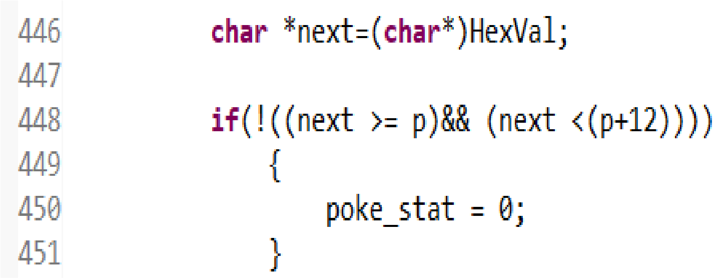
Since we are extracting the address in string format it is necessary to be able to manipulate the string in ‘buf’ at only the required byte positions to convert to hex to point to the SRAM actual locations. For this the following hex conversion from ascii function is used:

Meenakshi Shankar
Sr No. 22400
M.Tech EPD



```
void ascii_to_hex(char* addr)
{

HexVal = 0;
for(int i =0 ; i<8; i++)
{
    if(addr[9-i]>='a'&& addr[9-i]<='f')
    {
        addr[9-i] = addr[9-i] - 87;
        HexVal += (pow_new(16,i)*addr[9-i]);
    }
    else
        {
        addr[9-i] = addr[9-i] - 48;
        HexVal += (pow_new(16, i)*addr[9-i]);
        }
}

}
```

Then based on comparison criteria that if either the hex value is < or > = the hex value + the total allowed 12 bytes of data then it becomes an invalid request, else it is processed and the string is inserted to the place of starting address given.

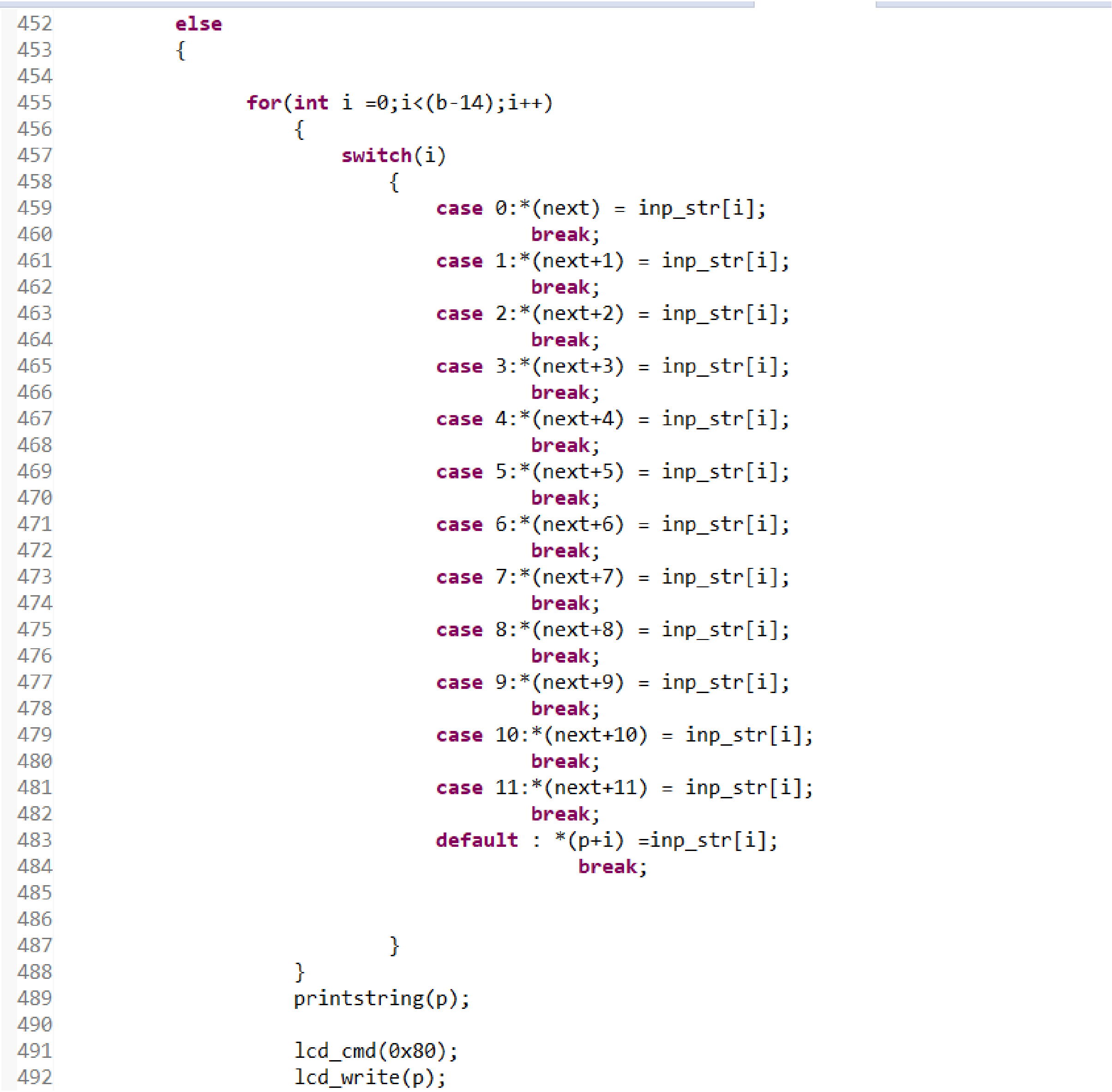


```
        char *next=(char*)HexVal;

        if(!((next >= p)&& (next <(p+12))))
            {
                poke_stat = 0;
            }
```

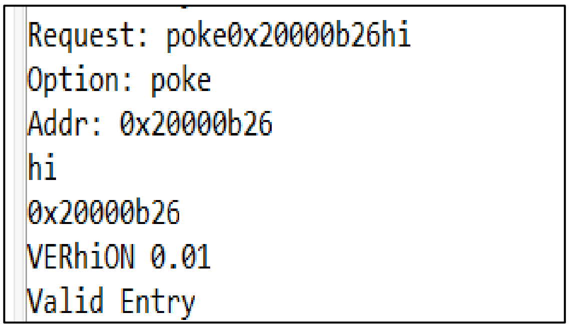
The pointer of char type is assigned to do the updation byte by byte once above criteria is satisfied.

The updation at the memory location is further reflected onto the LCD.

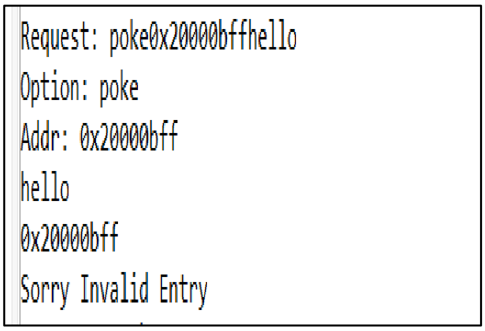
Meenakshi Shankar
Sr No. 22400
M.Tech EPD



```
        else
        {

              for(int i =0;i<(b-14);i++)
                  {
                      switch(i)
                          {
                              case 0:*(next) = inp_str[i];
                                      break;
                              case 1:*(next+1) = inp_str[i];
                                      break;
                              case 2:*(next+2) = inp_str[i];
                                      break;
                              case 3:*(next+3) = inp_str[i];
                                      break;
                              case 4:*(next+4) = inp_str[i];
                                      break;
                              case 5:*(next+5) = inp_str[i];
                                      break;
                              case 6:*(next+6) = inp_str[i];
                                      break;
                              case 7:*(next+7) = inp_str[i];
                                      break;
                              case 8:*(next+8) = inp_str[i];
                                      break;
                              case 9:*(next+9) = inp_str[i];
                                      break;
                              case 10:*(next+10) = inp_str[i];
                                      break;
                              case 11:*(next+11) = inp_str[i];
                                      break;
                              default : *(p+i) =inp_str[i];
                                          break;


                          }
                  }
                  printstring(p);

                  lcd_cmd(0x80);
                  lcd_write(p);
```



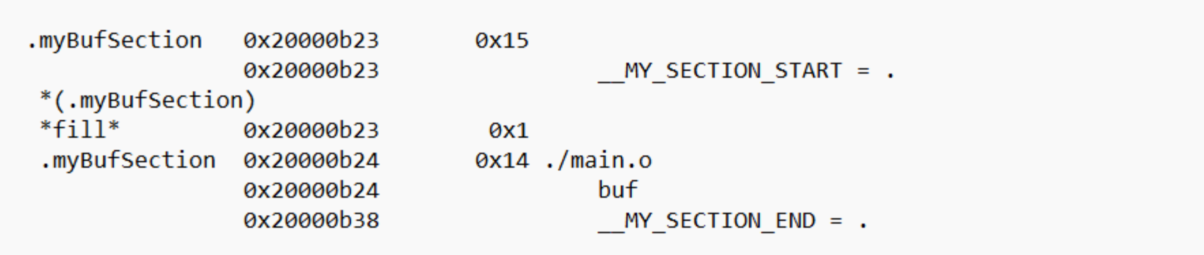
```
Request: poke0x20000b26hi
Option: poke
Addr: 0x20000b26
hi
0x20000b26
VERhiON 0.01
Valid Entry
```



```
Request: poke0x20000bffhello
Option: poke
Addr: 0x20000bff
hello
0x20000bff
Sorry Invalid Entry
```

A valid request is made and updation has occurred at b26 whereas the string starts at b23. For some address beyond the dedicated section myBufSection the command is invalid.

The memory allocation is reflected and verified from the .map file of the project



```
.myBufSection   0x20000b23       0x15
                0x20000b23                __MY_SECTION_START = .
 *(.myBufSection)
 *fill*         0x20000b23        0x1
 .myBufSection  0x20000b24       0x14 ./main.o
                0x20000b24                buf
                0x20000b38                __MY_SECTION_END = .
```