Meenakshi Shankar
Sr No. 22400
M.Tech EPD

EMBEDDED SYSTEM DESIGN (E3-257)

LAB ASSIGNMENT – 6 and 7

→**Lab 6:** *Implementing Stop watch Timer*

1. Setting up the SSD for Stop-Watch.

We use the NVIC SysTick Handler to manage the timer setting in the Seven segment display

```
128     NVIC_ST_RELOAD_R = 1000000-1;
129     NVIC_ST_CTRL_R |= 7 ; // enable counter, interrupt and select system bus clock
130     NVIC_ST_CURRENT_R  = 0;
131
```

```
746 void SysTick_Handler(void)
747 {
748
749    if((pause_stat == 1))
750    {
751
752        GPIO_PORTF_DATA_R  ^= 0x04;
753    }
754
755    else
756    {
757        count = count+1;
758
759        GPIO_PORTF_DATA_R  ^= 0x08;  //toggle PF3 pin
760    }
761
762 }
763
```

Here the Handler toggles in default and the pause state conditions are checked for when the flag is raised.

Since the 0 pattern in the 4th SSD is interfering with the LCD cmnd mode operation its not printing for seconds with 3rd digit in 0 value and it starts showing when count becomes 100 ms onwards

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

```
158       if(count4 == 0)
159
160       { for(int i =0;i<60;i++){}
161
162        GPIO_PORTA_DATA_R &= ~(0xF0);
163        GPIO_PORTA_DATA_R |= 0x10;
164        GPIO_PORTB_DATA_R = 0;
165        GPIO_PORTB_DATA_R = digitPattern[count];
166
167        for(int i =0;i<60;i++){}
168        GPIO_PORTA_DATA_R &= ~(0xF0);
169        GPIO_PORTA_DATA_R |= 0x20;
170        GPIO_PORTB_DATA_R = 0;
171        GPIO_PORTB_DATA_R = digitPattern[count2];
172
173        for(int i =0;i<60;i++){}
174        GPIO_PORTA_DATA_R &= ~(0xF0);
175        GPIO_PORTA_DATA_R |= 0x40;
176        GPIO_PORTB_DATA_R = 0;
177        GPIO_PORTB_DATA_R = digitPattern[count3];
178       }
179
```

```
180         else
181         {
182             for(int i =0;i<60;i++){}
183
184              GPIO_PORTA_DATA_R &= ~(0xF0);
185              GPIO_PORTA_DATA_R |= 0x10;
186              GPIO_PORTB_DATA_R = 0;
187              GPIO_PORTB_DATA_R = digitPattern[count];
188
189              for(int i =0;i<60;i++){}
190              GPIO_PORTA_DATA_R &= ~(0xF0);
191              GPIO_PORTA_DATA_R |= 0x20;
192              GPIO_PORTB_DATA_R = 0;
193              GPIO_PORTB_DATA_R = digitPattern[count2];
194
195              for(int i =0;i<60;i++){}
196              GPIO_PORTA_DATA_R &= ~(0xF0);
197              GPIO_PORTA_DATA_R |= 0x40;
198              GPIO_PORTB_DATA_R = 0;
199              GPIO_PORTB_DATA_R = digitPattern[count3];
200
201
202              for(int i =0;i<60;i++){}
203              GPIO_PORTA_DATA_R &= ~(0xF0);
204              GPIO_PORTA_DATA_R |= 0x80;
205              GPIO_PORTB_DATA_R = 0;
206              GPIO_PORTB_DATA_R = digitPattern[count4];
207
208
209         }
```

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

2.Push Buttons are set as follows and works based on:

SW1: Will be used to start and stop.

SW2: Will be used to pause and resume.

```
761
762  if ((GPIO_PORTF_MIS_R & 0x10) == 0x10) /* check if interrupt causes by PF4/SW1*/
763    {
764        GPIO_PORTF_ICR_R  = 0x10; /* clear the interrupt flag */
765
766     if((strcmp(cmnd.data, "stop")==0)||(stat_key_1 == 1))
767     {
768
769                                          stat_key_1 = 0;
770
771                                          printstring("Stop\n\r");
772                                          GPIO_PORTF_DATA_R = 0x08;
773                                          count = 0;
774                                              count2 = 0;
775                                              count3 = 0;
776                                              count4 = 0;
777
778                                          NVIC_ST_CTRL_R = 0;
779                                          flag = 1;
780                                          stop_stat = 1;
781
782
783                                          lcd_cmd(0x01);
784                                          lcd_cmd(0x02);
785                                          lcd_cmd(0x80);
786                                          lcd_write("Timer  ");
787                                          lcd_cmd(0xC0);
788                                          lcd_write("Ready   ");
789
790                                          printstring("Valid Entry\n\r");
791                                          check= 1;
792
793
794     }
795     else if((strcmp(cmnd.data, "start")==0)||(stat_key_1 == 0))
796     {
```

```
821  if ((GPIO_PORTF_MIS_R & 0x01) == 0x01) /* check if interrupt causes by PF4/SW2*/
822      {
823          GPIO_PORTF_ICR_R  = 0x01; /* clear the interrupt flag */
824
825       if((strcmp(cmnd.data, "pause")==0)||(stat_key_2 == 0))
826       {
827
828                                           stat_key_2 = 1;
829
830                                           printstring("Pause \n\r");
831
832                                           GPIO_PORTF_DATA_R &= 0x00;
833                                           GPIO_PORTF_DATA_R |= 0x04;
834
835                                           NVIC_ST_CTRL_R |= 7;
836                                           flag = 1;
837                                           pause_stat = 1;
838
839
840                                           lcd_cmd(0x01);
841                                           lcd_cmd(0x02);
842                                           lcd_cmd(0x80);
843                                           lcd_write("Timer   ");
844                                           lcd_cmd(0xC0);
845                                           lcd_write("Paused   ");
846
847                                           printstring("Valid Entry\n\r");
848                                           check= 1;
849
850
851       }
852       else if((strcmp(cmnd.data, "resume")==0)||(stat_key_2 == 1))
853       {
854                                          stat_key_2 = 0;
855                                          printstring("Resume\n\r");
856                                          GPIO_PORTF_DATA_R &= 0x00;
857                                          GPIO_PORTF_DATA_R |= 0x08;
858
```

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

Both SW1 and SW2 are configured to falling edge interrupts and the same is checked in the port F handler as above.

→**Lab 7:** *Tic-Tac-Toe*

1. Logic:

Detecting a Row Strike.

```
426             {
427                 for (int r_val = 0; r_val < 3; r_val++) {
428                         if (arr[r_val][0] == arr[r_val][1]
429                             && arr[r_val][1] == arr[r_val][2]) {
430                             if (arr[r_val][0] == 'x')
431                                 {printstring("User x Won\n\r");
432                                  x_flag = 1;
433                                 }
434                             else if (arr[r_val][0] == 'o' )
435                                 {printstring("User o Won \n\r");
436                                  o_flag = 1;
437                                 }
438                         }
439                 }
```

Detecting a Column Strike.

```
440                 for (int c_val = 0; c_val < 3; c_val++)
441                         {
442                             if ((arr[0][c_val] == arr[1][c_val]) && (arr[1][c_val] == arr[2][c_val]))
443                             {
444                                 if (arr[0][c_val] == 'x')
445                                     {printstring("User x Won\n\r");
446                                      x_flag = 1;
447                                     }
448
449                                 else if (arr[0][c_val] == 'o')
450                                     {printstring("User o Won\n\r");}
451                             }
452                         }
453
```

Detecting Diagonal Strike.

```
465                 if (arr[0][2] == arr[1][1] && arr[1][1] == arr[2][0]) {
466                     if (arr[0][2] == 'x')
467                         {printstring("User x Won\n\r");
468                          x_flag = 1;
469                         }
470                     else if (arr[0][2] == 'o')
471                         {printstring("User o Won\n\r");
472                          o_flag = 1;
473                         }
474                 }
```

Detecting a Draw Game.

```
477                 if((x_flag == 0)&&(o_flag == 0)&&(inp_count>9))
478                 {
479                     printstring("Game Draw!!\n\r");
480                     x_flag = 1;
481                     o_flag = 1;
482                 }
483
484
```

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

Clearing a Game after result:

```
490             if((x_flag == 1)||(o_flag == 1))
491             {
492                 for(int i =0;i<3;i++)
493                 {
494                     for(int j = 0;j <3;j++)
495                     {
496                         arr[i][j] = ' ';
497                     }
498                 }
499 
500                 inp_count = 0;
501 
502 
503             }
504 
505             x_flag = 0;
506             o_flag = 0;
507 
508         }
509 
```

2. Mapping and Recognizing Key Presses.

➔ Implementation of col recognition using interrupts:

```
887 void GPIOC_Handler(void)
888 {
889     if ((GPIO_PORTC_MIS_R & 0x10) == 0x10)
890                 {
891                     GPIO_PORTC_ICR_R = 0x10;
892                     key_col = 1;
893 
894                 }
895     if ((GPIO_PORTC_MIS_R & 0x20) == 0x20)
896                 {
897                     GPIO_PORTC_ICR_R = 0x20;
898                     key_col = 1;
899 
900                 }
901     if ((GPIO_PORTC_MIS_R & 0x40) == 0x40)
902                 {
903                     GPIO_PORTC_ICR_R = 0x40;
904                     key_col = 1;
905 
906                 }
907     if ((GPIO_PORTC_MIS_R & 0x80) == 0x80)
908                 {
909                     GPIO_PORTC_ICR_R = 0x80;
910                     key_col = 1;
911 
912                 }
913 }
914 
```

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

➔ Recognition of the row pressed:

```
322         if(key_col)
323          {
324              int row = 0, col = 0, num = 0;
325              for(row = 0; row < 3; row ++)
326              {
327
328                  GPIO_PORTE_DATA_R = ( 0x0F & ~(1 << row) );
329
330                  num = GPIO_PORTC_DATA_R & 0xF0 ;
331
332                  if( num == 0xE0)
333                  {
334                      col = 0;
335                      key_val = (row*3 + col);
336
337                      r = row;
338                      c = key_val%3;
339
340
341                  }
342                  else if( num == 0xD0)
343                  {
344                      col = 1;
345                      key_val = (row*3 + col);
346
347                      r = row;
348                      c = key_val%3;
349
350
351                  }
352                  else if( num == 0xB0)
353                  {
354                      col = 2;
355                      key_val = (row*3 + col);
356
357                      r = row;
358                      c = key_val%3;
359
360
361
362                  }
363
```

Recognizing and storing the x or o intake to the particular position in arr[][] as mapped from the keypad button position(set based on the winner of previous game. On setup it is X)

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

```
373         if(isKeyPressed())
374             {
375 
376 
377 
378                 if((user == 0)&&(arr[r][c]!='o'))
379                 {
380                     if(x_flag == 1)
381                     {
382                         arr[r][c] = 'o';
383                         user = 0;
384                     }
385 
386                     else
387                     {
388                         arr[r][c] = 'x';
389                         user  = 1;
390                     }
391 
392 
393 
394 
395                 }
396 
397                 else if((user == 1)&&(arr[r][c]!='x'))
398                 {
399                     if(o_flag == 1)
400                     {
401                         arr[r][c] = 'x';
402                         user = 1;
403                     }
404 
405                     else
406                     {
407                         user = 0;
408                         arr[r][c] = 'o';
409 
410                     }
411 
412                 }
413 
414                 else
```

Pressing of already pressed position in the 3x3 matrix is not allowed and is realised as “Invalid Move”

```
397             else if((user == 1)&&(arr[r][c]!='x'))
398             {
399                 if(o_flag == 1)
400                 {
401                     arr[r][c] = 'x';
402                     user = 1;
403                 }
404 
405                 else
406                 {
407                     user = 0;
408                     arr[r][c] = 'o';
409 
410                 }
411 
412             }
413 
414             else
415             {
416                 printstring("Invalid Move!!");
417             }
418 
419             inp_count = inp_count + 1;
```

Meenakshi Shankar
Sr No. 22400
M.Tech EPD

3. UART console commands are also accepted. The commands are parsed as below in the separate_funct()

```
void separate_funct(void)
{
    int a = 0, b = 0;
    for(int i = 0; i<6; i++)
    {
        cmnd.data[i] = '\0';
        cmnd.type[i] = '\0';
    }


while((cmnd_val[a] != '\r'))
{
        if(!(((cmnd_val[a]>='A')&&(cmnd_val[a]<='Z')) || ((cmnd_val[a]>='a')&&(cmnd_val[a]<='z')) || ((cmnd_val[a]>='0')&&(cmnd_val[a]<='9'))))
            {
            a++;
            continue;
            }
        else
        {
            if((cmnd_val[a]>='A') && (cmnd_val[a]<='Z'))
                cmnd_val[a] = cmnd_val[a] + 32;
            full_cmnd[b] = cmnd_val[a];
            a++;
            b++;
        }
}
printstring("Request: ");
for(int i = 0; i<b; i++)
    UART0_Transmitter(full_cmnd[i]);

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

And different operation conditions are mapped to the received state as below:

```
if ((strcmp(cmnd.data, "stop")==0))
{   // GPIO_init();

    LCD_init();
    count = 0;
    count2 = 0;
    count3 = 0;
    count4 = 0;




    NVIC_ST_CTRL_R = 0;
    flag = 1;
    stop_stat = 1;
    GPIO_PORTF_DATA_R = 0x08;

    lcd_cmd(0x01);
    lcd_cmd(0x02);
    lcd_cmd(0x80);
    lcd_write("Timer   ");
    lcd_cmd(0xC0);
    lcd_write("Ready   ");

    printstring("Stop");

    printstring("\n\r");


    printstring("Valid Entry\n\r");
    check= 1;



}

```