اسم الطالب : أيمن محمد نبيل محمد

سكشن : 2

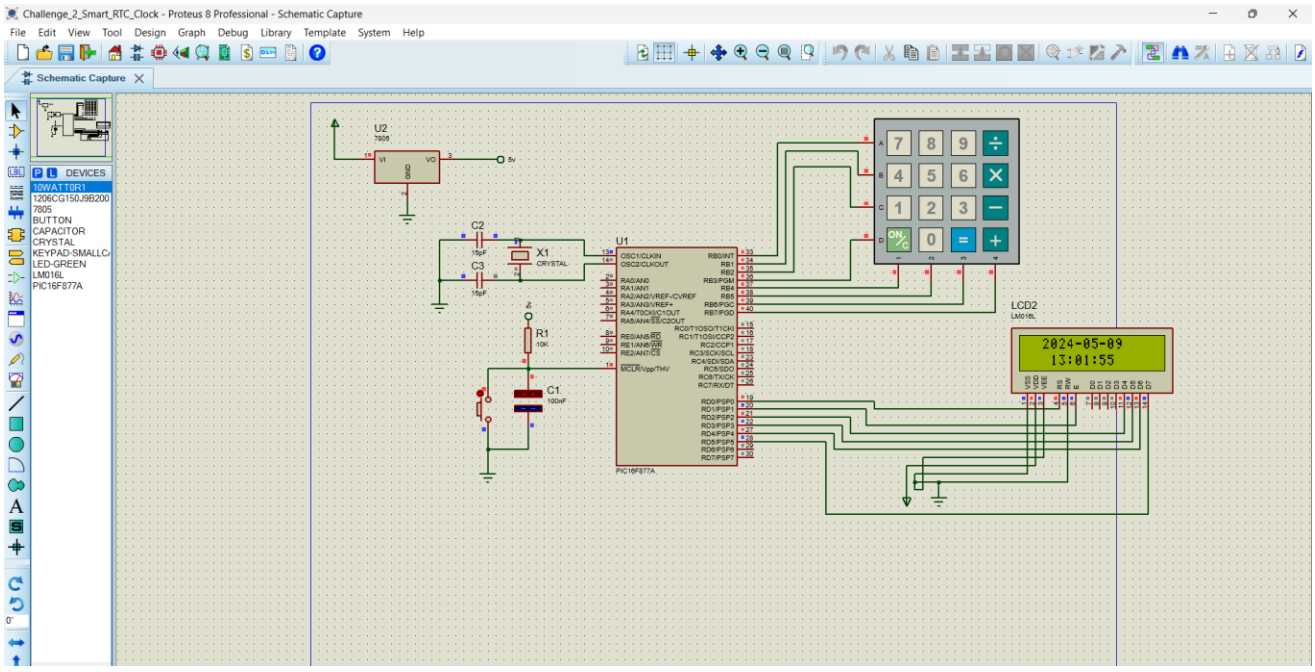
قسم هندسة الحاسبات ونظم التحكم

Challenge 2 – Task

Smart RTC Clock

.....

Screenshot of the circuit



Code:

```
 /*
 * File:   main.c
 * Author: engay
 *
 * Created on May 8, 2024, 4:19 PM
 */



#include <xc.h>
#include "./lib/common.h"
#include "lib/keypad/keypad.h"
#include "lib/lcd/lcd.h"
#include "lib/timer/timer.h"



#define Leap_Year_Days 31622400 //(366*24*60*60)
#define Year_Days 31536000 //(365*24*60*60)
#define IS_LEAP_YEAR(year) ( (year%4==0) && ( (year%100!=0) ||
(year%400==0) ) )



static unsigned long long epoch=0; // Global Variable.


void My_Init_Code()
{
    timer0_init(_TIMER_PRESCALER_2);
    lcd_init();
    keypad_init();
```

```
}

char Get_Num_str(char* stnum)
{
    lcd_set_cursor(2,7);
    char key=1,i=0;
    for(i=0;i<10;i++)
    {
        key='\0';
        while(key=='\0')
        {
            key=keypad_get_key();
        }
        while(key==keypad_get_key());

        if(key=='=')
        {
            stnum[i]='\0';
            break;
        }
        else
        {
            stnum[i]=key;
            lcd_chr_cp(key);
        }
    }
    keypad_init();
    return i-1;
}
```

```
unsigned long long power(char num,char repeat)
{
    unsigned long long result=0;
    result = num - 48 ;
    while(repeat!=0)
    {
        result*=10;
        repeat--;
    }
    return result;
}

unsigned long long Conv_Num(char *snum,char end)
{
    unsigned long long num=0;
    while(*snum!=0)
    {
        num+= power(*snum,end);
        end--;
        snum++;
    }
    return num;
}

void Show_4(int four)
{
    char Show[5]={5,0,0,0,0};
    char i=3;
    while(i)
    {
```

```
        Show[i] = (four%10)+48;
        four/=10;
        i--;
    }
    Show[0]=four+48;
    lcd_out_cp(Show);
}

void Show_2(int two)
{
    char Show[3]={0,0,0};
    char i=1;
    while(i)
    {
        Show[i] = (two%10)+48;
        two/=10;
        i--;
    }
    Show[0]=two+48;
    lcd_out_cp(Show);
}

    int year =1970;
    char month=1,day=1,hour=0,minute=0;

void Print_On_Screen()
{
    // First Line
    lcd_set_cursor(1,3);
    Show_4(year);
```

```
    lcd_chr_cp('-');
    Show_2(month);
    lcd_chr_cp('-');
    Show_2(day);
    // Second Line
    lcd_set_cursor(2,4);
    Show_2(hour);
    lcd_chr_cp(':');
    Show_2(minute);
    lcd_chr_cp(':');
    Show_2((char)epoch);
}

extern unsigned long timer0_time_passed_ms;
extern unsigned short overflow_cnt;

void __interrupt() my_isr()
{
    GIE=0;
    if(TMR0IF==1 && TMR0IE==1)
    {
        overflow_cnt++;
        if(overflow_cnt==4)
        {
            timer0_time_passed_ms++;
            overflow_cnt = 0;
        }
        TMR0IF=0;
    }
    TMR0= 6+40;
```

```
    GIE=1;
}

void my_delay(unsigned long mytime)
{
    unsigned long start_time=0,end_time=0;
    end_time = timer_get_time_since_init();
    while(end_time-start_time<=mytime)
    {
        end_time = timer_get_time_since_init();
    }
    timer0_time_passed_ms=0;
}

void main(void) {
    char months[12]={31,28,31,30,31,30,31,31,30,31,30,31};


    My_Init_Code();

    lcd_cmd(_LCD_CLEAR);


    char myname[]="Ayman Mohamed";
    lcd_out(1,6,"Welcome");
    lcd_out(2,3,myname);
    my_delay(2000); // 2 Seconds

    lcd_cmd(_LCD_CLEAR);
```

```
    char snum[11];
    lcd_out(1,1,"Enter Epoch Unix");
    lcd_out(2,1,"Time :");
    char end=0;
    end = Get_Num_str(snum);
    lcd_cmd(_LCD_CLEAR);
    epoch = Conv_Num(snum,end);









    while( (IS_LEAP_YEAR(year) && epoch>=Leap_Year_Days) ||
(epoch>=Year_Days && !IS_LEAP_YEAR(year)))
    {
        if(IS_LEAP_YEAR(year))
        {
            epoch-=Leap_Year_Days;
        }
        else
        {
            epoch-=Year_Days;
        }
        year++;
    }
    // Year is Done.


    if(epoch>=86400) // 24*60*60
    {
```

```
        int num_of_days=epoch/(86400);
        int desired = 0 ;
        int i=0;
        for( i=0; i<11 && desired + months[i] <num_of_days &&
num_of_days!=0 ;i++)
        {
            if(i==1 && IS_LEAP_YEAR(year) )
            {
                desired+=(months[i]+1);
                continue;
            }
            desired+=months[i];
        }
        month = i+1;
        epoch-=(desired*86400);
    }
    // Month is Done.



    if(epoch>=86400) // 24*60*60
    {
        day = epoch / 86400; // I added one because we started from 1
Jan 1970.
        day++; // I added one because we started from 1 Jan 1970.
        epoch= (epoch % 86400) ;
    } // Day is Done.




    if(epoch>=3600) // 60*60
```

```
    {
        hour = epoch/(3600);
        epoch= (epoch%3600);
    }
    // Hour is Done.

    if(epoch>=60) // 60
    {
        minute = epoch/(60);
        epoch -= (minute*60);
    }
    // Minutes is Done.
    // Seconds is Done. I will use epoch as a variable for seconds.
(Ayman Mohamed)



    while(1)
    {
        Print_On_Screen();
        my_delay(1000);
        if(epoch==59)
        {
            epoch=0;
            if(minute==59)
            {
                minute=0;
                if(hour==23)
                {
                    hour =0;
                    // Check Day
```

```
                    if(IS_LEAP_YEAR(year) && month == 2 && day==29)
                    {
                        day = 0;
                        month++;
                    }
                    else if(IS_LEAP_YEAR(year) && month == 2 && day
==28)
                    {
                        day=29;
                    }
                    else if(IS_LEAP_YEAR(year) && month == 2)
                    {
                        day++;
                    }
                    else
                    {
                        if(day < months[month-1] )
                        {
                            day++;
                        }
                        else
                        {
                            day=0;
                            if(month == 12)
                            {
                                month = 0 ;
                                year++;
                            }
                            else
                            {
```

```
                                month++;
                            }
                        }
                    }
                }
                else
                {
                    hour++;
                }
            }
            else
            {
                minute++;
            }
        }
        else
        {
            epoch++;

        }
    }
    return;
}
```