



به نام خدا
دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر



طراحی بر اساس ریزپردازنده

نیمسال دوم (01-02)

پروژه نهایی

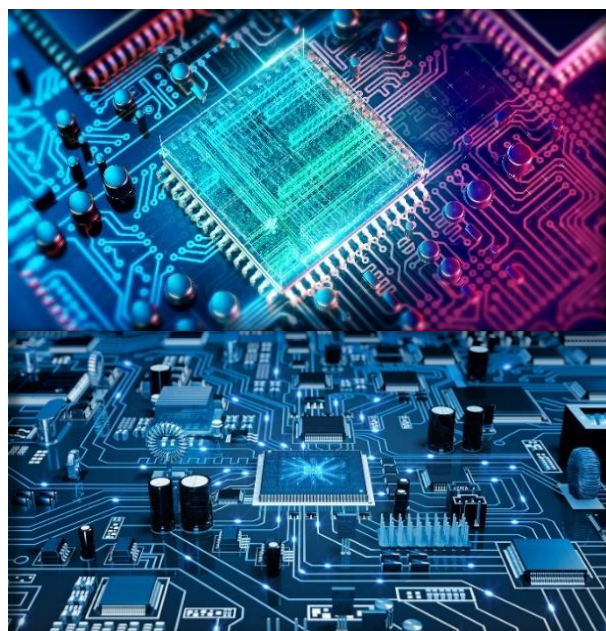
استاد درس: دکتر مهدی سیفی پور

محمدجواد شمسایی

810 198 419

محمد مهدی عبدالحسینی

810 198 434



Microprocessor Based Design

فهرست مطالب

1	مقدمه: طرح پروژه.....
1	طراحی مدار.....
1	ATmega32.....
2	Keypad.....
4	LCD.....
5	7-Segment.....
5	7448.....
6	سنسور دما.....
6	سنسور نور.....
7	سنسور فاصله.....
7	LEDs.....
7	LED های نمایش سنسور.....
7	LED های بحرانی.....
7	LED های انیمیشن.....
8	برنامه نویسی.....
8	تابع read_adc().....
8	تابع keyboard().....
9	تعریف متغیرها.....
10	تعیین ورودی و خروجی و مقدار اولیه.....
11	دریافت مقادیر سنسورها و keypad.....
11	وارد شدن عدد 7، 8 و 9 و مدیریت 7-Segment.....
12	وارد شدن عدد 0 و 1:.....
13	وارد شدن عدد 2.....
14	وارد شدن عدد 3.....
15	وارد شدن عدد 4.....
16	وارد شدن عدد 5.....
17	وارد شدن عدد 6.....
18	شرط doAnimate.....
19	شرط doLEDanimate.....
19	LED های بحرانی.....
20	نتایج نهایی.....

مقدمه: طرح پروژه

در این پروژه تبدیل سیگنال آنالوگ ورودی سنسورهای دما، نور و فاصله به سیگنال دیجیتال در خروجی کار شده است.

در ورودی همانطور که گفته شد از سنسورهای فوق و یک keypad جهت کنترل نمایشگرهای خروجی استفاده شده است.

در خروجی نیز از یک LCD برای نمایش داده سنسورها و یک 7-Segment جهت نمایش عدد وارد شده در keypad استفاده شده است. که این نمایشگر با واسطه آی سی 7448 مقداردهی میشود. همچنین LED هایی در خروجی استفاده شده است که کاربرد مفصل آنها در بخش های بعدی ارائه میشود.

طراحی مدار

در این بخش به بررسی و توضیح قطعات استفاده شده در مدار و چرایی استفاده آنها میپردازیم:

ATmega32

از میکروکنترلر ATmega32 که از جمله میکروکنترلرهای AVR شرکت Atmel میباشد جهت عملیات پردازش سیگنال با سرعت بالا در داخل سیستم استفاده میکنیم. در واقع هسته مرکزی مدار ارائه شده یا مغز مدار این تراشه است که تصویر آن را مشاهده میکنید.

در این مدار از پایه های PA0 تا PA2 برای ورودی سنسورها و از پایه های PD0 تا PD2 برای ورودی keypad استفاده کرده ایم. از سایر پایه های این تراشه نیز به شرح زیر برای خروجی استفاده شده است:

پایه PA3 تا PA6 برای نمایش 7-segment با استفاده از 7448

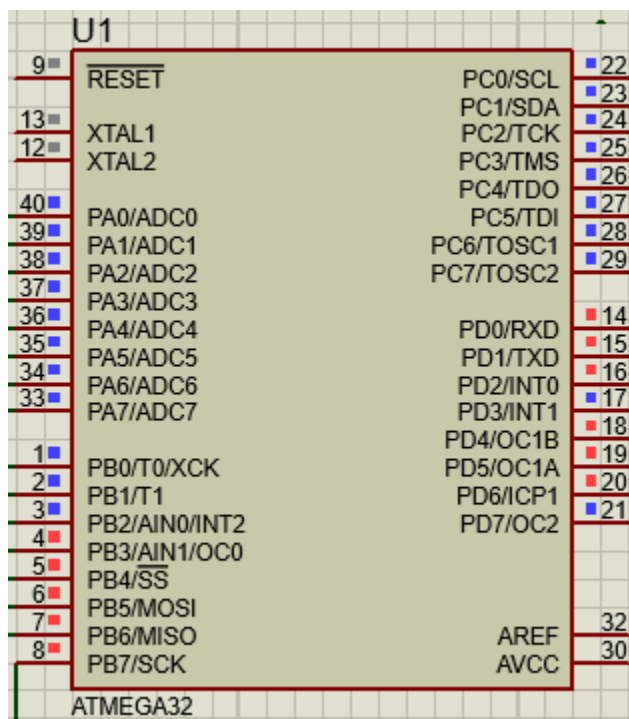
پایه PB0 تا PB2 برای LED های نمایش سنسور

پایه PB3 تا PB7 برای LED های انیمیشن

پایه PC0 تا PC2 و PC4 تا PC7 برای صفحه نمایش LCD

پایه PD3 تا PD6 برای keypad

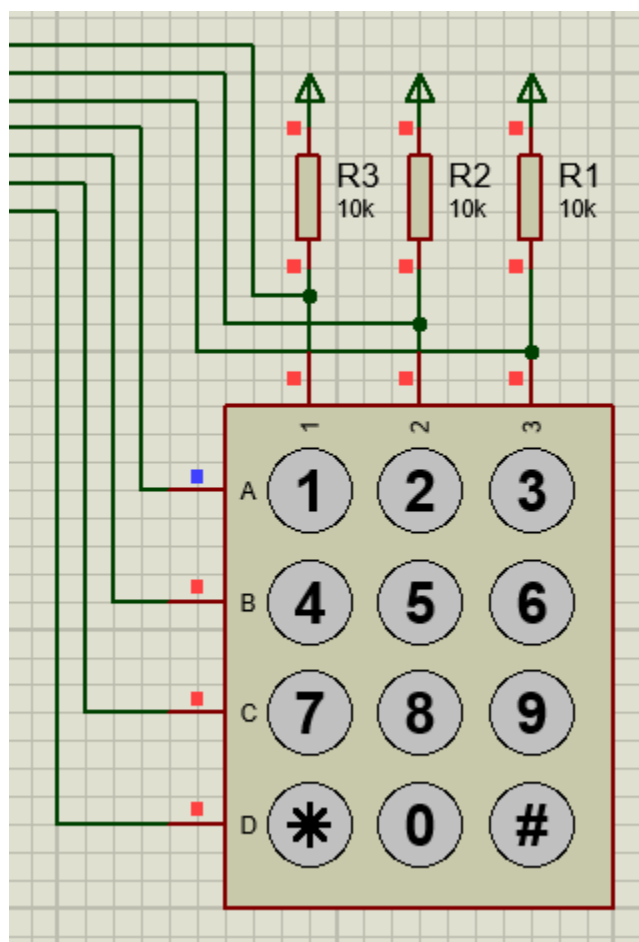
پایه PC3 و PA7 و PD7 برای LED های بحرانی



Keypad

از این صفحه کلید برای کنترل LCD و LED های انیمیشن و نمایش سنسور استفاده شده است.

ورودی‌های A تا D که توسط atmega32 فرمان میگیرند با فاصله هر 2ms مقدارشان از 0 به 1 و برعکس تغییر میکند. و خروجی های 1,2,3 که به atmega32 وارد میشوند در حالتی که کاربر عددی وارد نکرده به Vcc متصل اند. وقتی کاربر عدد را وارد میکند در یک لحظه 0 خوانده میشود و با استفاده از این منطق میتوان به عدد وارد شده پی برد که کد آن در بخش های بعد توضیح داده شده است.



نحوه کار صفحه کلید به شرح زیر است:

عدد 0: ریست شدن LCD و نمایش عبارت "Use Keypad" روی آن

عدد 1: نمایش مقدار دمای احساس شده توسط سنسور دما بر حسب سانتی گراد و روشن شدن LED نمایش سنسور دما.

عدد 2: نمایش مقدار دمای احساس شده توسط سنسور دما بر حسب فارنهایت و روشن شدن LED نمایش سنسور دما.

عدد 3: نمایش مقدار دمای احساس شده توسط سنسور دما بر حسب کلوین و روشن شدن LED نمایش سنسور دما.

عدد 4: نمایش شدت نور احساس شده توسط سنسور نور بر حسب uW/cm^2 و روشن شدن LED نمایش سنسور نور.

عدد 5: نمایش فاصله اندازه‌گیری شده توسط سنسور فاصله بر حسب cm و روشن شدن LED نمایش سنسور فاصله.

عدد 6: نمایش فاصله اندازه‌گیری شده توسط سنسور فاصله بر حسب inch و روشن شدن LED نمایش سنسور فاصله.

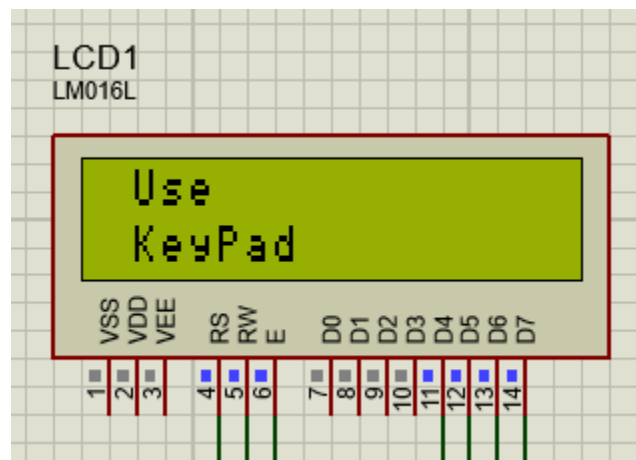
عدد 7: روشن شدن LED های انیمیشن.

عدد 8: متحرک شدن عبارت نوشته شده روی LCD.

عدد 9: ثابت شدن عبارت نوشته شده روی LCD و خاموش شدن LED های انیمیشن.

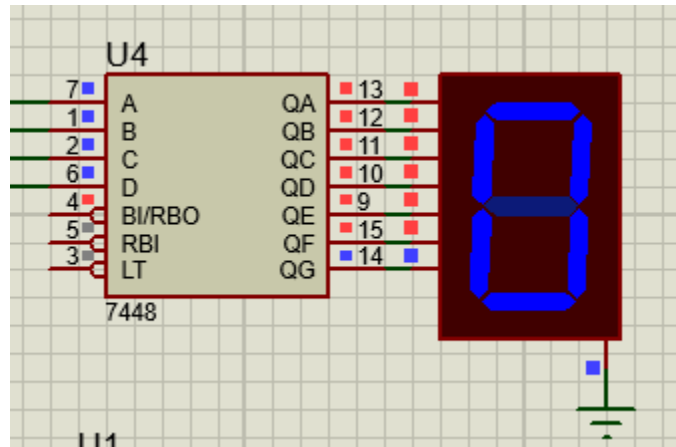
LCD

با استفاده از این نمایشگر مقادیر اندازه‌گیری شده توسط سنسورها را نمایش می‌دهیم.



7-Segment

از این ابزار برای نشان دادن عدد وارد شده با keypad توسط کاربر استفاده کرده ایم. این 7-Segment دارای 8 ورودی میباشد که به پایه های خروجی 7448 که توضیحات آن در پایین آورده شده متصل شده است

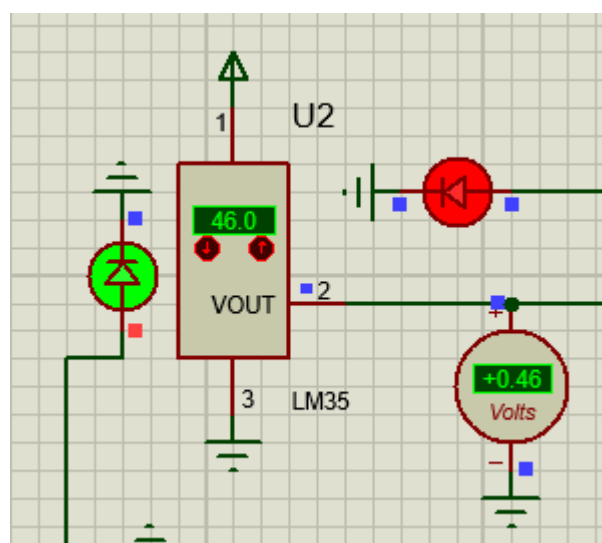
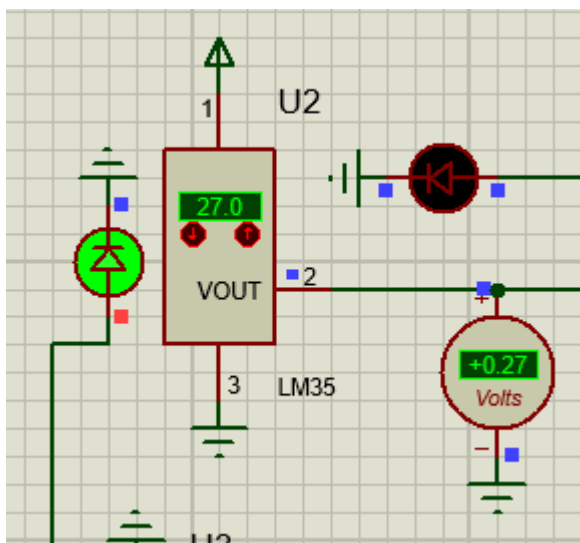


7448

برای اینکه کار با 7-Segment راحت تر شود از این IC استفاده کرده ایم که 4 ورودی را از atmega32 میگیرد که یک عدد باینری 4 بیتی در مبنای 2 را تشکیل میدهد که میتواند از اعداد 1 تا 15 باشد. کاربر هر عددی را در keypad وارد کند به ورودی این IC منتقل میشود و وظیفه این ابزار روشن کردن LED های مربوطه عدد وارد شده در 7-Segment است.

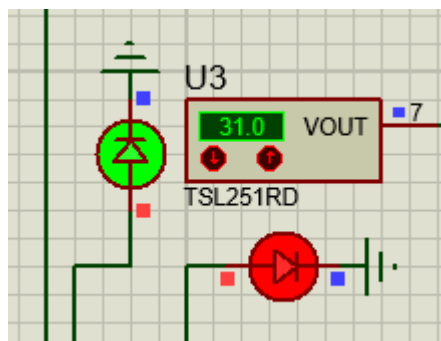
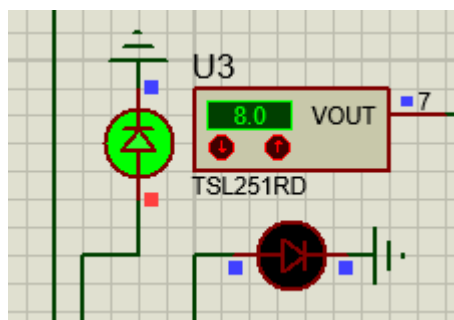
سنسور دما

این سنسور دما را اندازه میگیرد و در برنامه ای که نوشته ایم اگر این مقدار بیشتر از 45 درجه سانتی گراد شد LED بحرانی این سنسور شروع به چشمک زدن میکند.



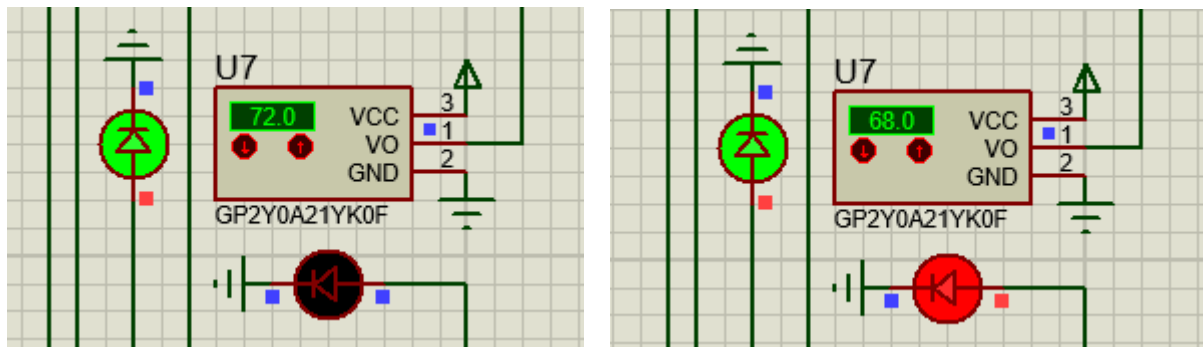
سنسور نور

این سنسور نور را اندازه میگیرد و در برنامه ای که نوشته ایم اگر این مقدار بیشتر از 30 uW/cm^2 شد LED بحرانی این سنسور شروع به چشمک زدن میکند.



سنسور فاصله

این سنسور فاصله را اندازه میگیرد و در برنامه ای که نوشته ایم اگر این مقدار کمتر از 70 cm شد LED بحرانی این سنسور شروع به چشمک زدن میکند.



LEDs

LED های نمایش سنسور

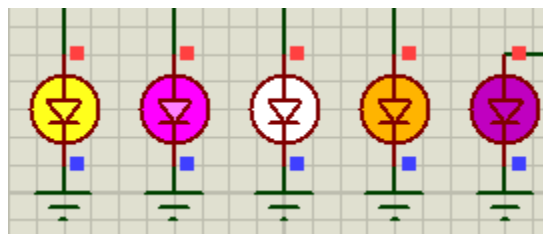
داده های هر سنسوری روی LCD نمایش داده شود سنسور مربوطه آن روشن میشود

LED های بحرانی

اگر عدد دریافت شده توسط سنسورها در شرایط بحرانی تعریف شده بود شروع به چشمک زدن میکنند

LED های انیمیشن

با کلید 7 در keypad به صورت چشمک زن روشن شده و با کلید 9 خاموش میشوند. بیشتر جهت زیبایی استفاده شده اند.



برنامه نویسی

در این بخش به بررسی توابع و قطعه کدهای استفاده شده میپردازیم:

تابع read_adc()

این تابع در ورودی عدد 0 تا 7 را میتواند دریافت کند که نشان میدهد از کدام پایه PORTA در atmega32 استفاده میکنیم. و داده آنالوگ این سنسورها را تبدیل به دیجیتال کرده و بر میگرداند.

```

33 //===== Read the AD conversion result
34 unsigned char read_adc(unsigned char adc_input)
35 {
36     ADMUX=adc_input | ADC_VREF_TYPE;
37     // Delay needed for the stabilization of the ADC input voltage
38     delay_us(10);
39     // Start the AD conversion
40     ADCSRA|=(1<<ADSC);
41     // Wait for the AD conversion to complete
42     while ((ADCSRA & (1<<ADIF))==0);
43     ADCSRA|=(1<<ADIF);
44     return ADCW;
45 }

```

تابع keyboard()

این تابع بسته به عددی که کاربر وارد میکند یک عدد را به عنوان خروجی بر میگرداند همانطور که پیشتر گفته شد PORT های ورودی به keypad با فاصله 2ms مقدارشان از 0 به 1 و برعکس تغییر میکند در این حین هر کدام از پین های خروجی از keypad و ورودی به atmega32 اگر مقدار 0 گرفت عدد مورد نظر وارد شده است.

```
47 //===== keyboard handler:
48 int keyboard(){
49     int key=20;
50     //---- ROW1 ----
51     PORTD.3=0;
52     delay_ms(2);
53     if(PIND.0==0) key=1;
54     if(PIND.1==0) key=2;
55     if(PIND.2==0) key=3;
56     PORTD.3=1;
57     //---- ROW2 ----
58     PORTD.4=0;
59     delay_ms(2);
60     if(PIND.0==0) key=4;
61     if(PIND.1==0) key=5;
62     if(PIND.2==0) key=6;
63     PORTD.4=1;
64     //---- ROW3 ----
65     PORTD.5=0;
66     delay_ms(2);
67     if(PIND.0==0) key=7;
68     if(PIND.1==0) key=8;
69     if(PIND.2==0) key=9;
70     PORTD.5=1;
71     //---- ROW4 ----
72     PORTD.6=0;
73     delay_ms(2);
74     if(PIND.1==0) key=0;
75     PORTD.6=1;
76
77     return key;
78 }
```

تعریف متغیرها.

در عکس زیر متغیرهای مورد نیاز تعریف شده اند که استفاده هر کدام در بخش های بعدی توضیح داده میشود.

```

80 #define LCD_WIDTH 16 // LCD has 16 columns
81
82 void main(void){
83     // Declare your local variables here
84     unsigned char temp;
85     unsigned char light;
86     unsigned char distance;
87
88     char s[15];
89     int key = 20;
90     int flag = 10;
91     int backupFlag = 10;
92
93     int doAnimate = 0;
94     int doLEDanimate = 0;
95     int x = 0; // Initial x position
96     int y = 0; // Initial y position
97     int direction = 1; // Initial direction (1: right, -1: left)

```

تعیین ورودی و خروجی و مقدار اولیه

در این بخش از کد باید تعیین کنیم که کدام پایه ها به عنوان ورودی و کدام به عنوان خروجی استفاده شوند. و همچنین خروجی ها را مقدار اولیه دهیم. (بخشی ابتدایی توسط کدویزارد generate شده است)

```

217 PORTC = 0x00;
218 DDRC = 0xF7;
219 ADMUX = ADC_VREF_TYPE & 0xFF;
220 ADCSRA = 0x83;
221
222 DDRD = 0xF8;
223 PORTD = 0xF8;
224 DDRB = 0xFF;
225 PORTB = 0xF8;
226 DDRA.3 = 1;
227 DDRA.4 = 1;
228 DDRA.5 = 1;
229 DDRA.6 = 1;
230
231 DDRA.7 = 1;
232 DDRD.7 = 1;
233 DDRC.3 = 1;

```

دریافت مقادیر سنسورها و keypad

در این خطوط از کد مقادیر سنسور های دما نور و فاصله را در متغیرهای distance, light, temp ریخته ایم. همچنین عدد وارد شده توسط کاربر در keypad را در متغیر key ذخیره میکنیم.

```
235     while (1){  
236  
237         // analog signals input  
238         temp = read_adc(0);  
239         light = read_adc(1);  
240         distance = read_adc(2);  
241  
242         key=keyboard(); // keypad input
```

وارد شدن عدد 7، 8 و 9 و مدیریت 7-Segment

عکس زیر شامل دو بخش کد است بخش اول جهت مدیریت متحرک شدن یا ثابت شدن نوشته روی LCD و بخش دوم جهت نمایش عدد وارد شده روی 7-Segment.

```

244 //===== handle keypad input:
245 if (key != 20){
246     if (key == 7 || key == 8 || key == 9) {backupFlag = flag;}
247
248     flag = key;
249
250     lcd_gotoxy(0, 0);
251     lcd_puts("                ");
252     lcd_gotoxy(0, 1);
253     lcd_puts("                ");
254
255     PORTB = 0x00;
256     PORTA.3 = 0;
257     PORTA.4 = 0;
258     PORTA.5 = 0;
259     PORTA.6 = 0;
260
261     // update 7-segments
262     if (key == 1) { PORTA.6 = 1;}
263     else if (key == 2) {PORTA.5 = 1;}
264     else if (key == 3) {PORTA.5 = 1; PORTA.6 = 1;}
265     else if (key == 4) {PORTA.4 = 1;}
266     else if (key == 5) {PORTA.4 = 1; PORTA.6 = 1;}
267     else if (key == 6) {PORTA.4 = 1; PORTA.5 = 1;}
268     else if (key == 7) {PORTA.4 = 1; PORTA.5 = 1; PORTA.6 = 1;}
269     else if (key == 8) {PORTA.3 = 1;}
270     else if (key == 9) {PORTA.3 = 1; PORTA.6 = 1;}
271 }

```

وارد شدن عدد 0 و 1:

در کدهای زیر مشاهده میکنید اگر عدد 0 وارد شود عبارت "Use KeyPad" روی LCD چاپ شده و اگر عدد 1 وارد شود در صورتی که دما از 50 درجه سانتی گراد کمتر باشد روی LCD نمایش داده میشود و اگر بیشتر باشد عبارت ">50" نمایش داده میشود همچنین LED نمایش سنسور دما نیز روشن میشود

```

273 //===== handle LCD:
274 if (flag == 10 || flag == 0){
275     sprintf(s, " Use ");
276     lcd_gotoxy(x, 0);
277     lcd_puts(s);
278     sprintf(s, " KeyPad ");
279     lcd_gotoxy(x, 1);
280     lcd_puts(s);
281 } //=====
282 else if(flag == 1){
283     if( temp/4 > 49){
284         sprintf(s, " Temp(C) ");
285         lcd_gotoxy(x, 0);
286         lcd_puts(s);
287         sprintf(s, " >50C ");
288         lcd_gotoxy(x, 1);
289         lcd_puts(s);
290     }
291     else{
292         sprintf(s, " Temp(C) ");
293         lcd_gotoxy(x, 0);
294         lcd_puts(s);
295         sprintf(s, " %dC ", temp/4);
296         lcd_gotoxy(x, 1);
297         lcd_puts(s);
298     }
299     PORTB.2 = 1;
300 } //=====

```

وارد شدن عدد 2

در صورتی که دما از 50 درجه سانتی گراد کمتر باشد آن دما بر حسب فارنهایت روی LCD نمایش داده میشود و اگر بیشتر باشد عبارت " >122F" نمایش داده میشود همچنین LED نمایش سنسور دما نیز روشن میشود

```
301     else if(flag == 2){
302         if( temp/4 > 49){
303             sprintf(s, " Temp(F) ");
304             lcd_gotoxy(x, 0);
305             lcd_puts(s);
306             sprintf(s, " >122F ");
307             lcd_gotoxy(x, 1);
308             lcd_puts(s);
309         }
310         else{
311             sprintf(s, " Temp(F) ");
312             lcd_gotoxy(x, 0);
313             lcd_puts(s);
314             sprintf(s, " %dF ", ((temp/4)*(9))/5 + 32);
315             lcd_gotoxy(x, 1);
316             lcd_puts(s);
317         }
318         PORTB.2 = 1;
319     } //=====
```

وارد شدن عدد 3

در صورتی که دما از 50 درجه سانتی گراد کمتر باشد آن دما بر حسب کلوین روی LCD نمایش داده میشود و اگر بیشتر باشد عبارت " >323K" نمایش داده میشود همچنین LED نمایش سنسور دما نیز روشن میشود


```
320     else if(flag == 3){
321         if( temp/4 > 49){
322             sprintf(s, " Temp(K) ");
323             lcd_gotoxy(x, 0);
324             lcd_puts(s);
325             sprintf(s, " >323K ");
326             lcd_gotoxy(x, 1);
327             lcd_puts(s);
328         }
329         else{
330             sprintf(s, " Temp(K) ");
331             lcd_gotoxy(x, 0);
332             lcd_puts(s);
333             sprintf(s, " %dK ", (temp/4) + 273);
334             lcd_gotoxy(x, 1);
335             lcd_puts(s);
336         }
337         PORTB.2=1;
338     } //=====
```

وارد شدن عدد 4

در صورتی که شدت نور از 33uW/cm^2 کمتر باشد روی LCD نمایش داده میشود و اگر بیشتر باشد عبارت $>33\text{uW/cm}^2$ نمایش داده میشود همچنین LED نمایش سنسور نور نیز روشن میشود.

```
339     else if(flag == 4){
340         if( light/6 > 32){
341             sprintf(s, " Light    ");
342             lcd_gotoxy(x, 0);
343             lcd_puts(s);
344             sprintf(s, " >33uW/cm2 ");
345             lcd_gotoxy(x, 1);
346             lcd_puts(s);
347         }
348         else{
349             sprintf(s, " Light    ");
350             lcd_gotoxy(x, 0);
351             lcd_puts(s);
352             sprintf(s, " %duW/cm2 ", light/6);
353             lcd_gotoxy(x, 1);
354             lcd_puts(s);
355         }
356         PORTB.1 = 1;
357     } //-----
```

وارد شدن عدد 5

در صورتی که فاصله اندازه گیری شده توسط سنسور فاصله از 65cm بیشتر باشد روی LCD نمایش داده میشود و اگر کمتر باشد عبارت "<65cm" نمایش داده میشود همچنین LED نمایش سنسور فاصله نیز روشن میشود.

```
358     else if(flag == 5){
359         if ( 161 - distance/2 < 65){
360             sprintf(s, " Dist.      ");
361             lcd_gotoxy(x, 0);
362             lcd_puts(s);
363             sprintf(s, " <65cm      ");
364             lcd_gotoxy(x, 1);
365             lcd_puts(s);
366         }
367         else{
368             sprintf(s," Dist.      ");
369             lcd_gotoxy(x, 0);
370             lcd_puts(s);
371             sprintf(s, " %dcm        ", 161 - distance/2);
372             lcd_gotoxy(x, 1);
373             lcd_puts(s);
374         }
375         PORTB.0 = 1;
376     } //=====
```

وارد شدن عدد 6

در صورتی که فاصله اندازه گیری شده توسط سنسور فاصله از 65 cm بیشتر باشد روی LCD بر حسب اینچ نمایش داده میشود و اگر کمتر باشد عبارت "165inch" نمایش داده میشود همچنین LED نمایش سنسور فاصله نیز روشن میشود.

```

377  ✓   else if(flag == 6){
378  ✓       if ( 161 - distance/2 < 65){
379           sprintf(s, " Dist.   ");
380           lcd_gotoxy(x, 0);
381           lcd_puts(s);
382           sprintf(s, " <165inch ");
383           lcd_gotoxy(x, 1);
384           lcd_puts(s);
385       }
386       else{
387           sprintf(s, " Dist.   ");
388           lcd_gotoxy(x, 0);
389           lcd_puts(s);
390           sprintf(s, " %dinch   ", ((161 - distance/2)*254)/100 );
391           lcd_gotoxy(x, 1);
392           lcd_puts(s);
393       }
394       PORTB.0 = 1;
395   } //=====
396  ✓   else if(flag == 7){
397       doLEDanimate=1;
398       flag = backupFlag;
399   } //=====
400  ✓   else if (key == 8){
401       doAnimate = 1;
402       flag = backupFlag;
403   } //=====
404  ✓   else if (key == 9){
405       doAnimate = 0;
406       doLEDanimate = 0;
407       flag = backupFlag;
408   }

```

شرط doAnimate

در کد زیر شرط مربوط به متحرک شدن نوشته روی LCD آورده شده است. که بالاتر توضیح دادیم در صورتی که کاربر عدد 7 را وارد کند فعال میشود

```

410 //===== handle LCD animation
411 if (doAnimate == 1){
412     lcd_gotoxy(x, y);
413     lcd_puts(" ");
414     x += direction; // Update x position
415     // Check collision with screen edges
416     if (x >= LCD_WIDTH-7){
417         x = LCD_WIDTH-7 - 1; // Adjust x to prevent overflow
418         direction = -1; // Change direction to left
419     }
420     else if (x < 0){
421         x = 0; // Adjust x to prevent underflow
422         direction = 1; // Change direction to right
423     }
424
425     if (y == 0) y = 1;
426     else y = 0;
427 }

```

شرط doLEDAnimate

در این شرط نیز که در صورت وارد شدن عدد 8 توسط کاربر فعال میشود LED های انیمیشن به صورت تصادفی روشن و خاموش میشوند

```

429 //===== handle LEDs animation
430 if (doLEDanimate==1){
431     PORTB.3 = rand() % 2;
432     PORTB.4 = rand() % 2;
433     PORTB.5 = rand() % 2;
434     PORTB.6 = rand() % 2;
435     PORTB.7 = rand() % 2;
436 }

```

LED های بحرانی

در کد زیر نیز شرایط لازم برای چشمک زدن LED های بحرانی آورده شده است

```

438 //===== handle critical situation:
439 if(161 - distance/2 < 70) {PORTD.7 = ~PORTD.7;}
440 else {PORTD.7 = 0;}
441
442 if(temp/4 > 45) {PORTA.7 = ~PORTA.7;}
443 else {PORTA.7 = 0;}
444
445 if(light/6 > 30) {PORTC.3 = ~PORTC.3;}
446 else {PORTC.3 = 0;}

```

نتایج نهایی

در بخش آخر تصاویری از شبیه‌سازی مدار را در شرایطی که کاربر اعداد مختلفی وارد میکند آورده‌ایم:

