



دانشکده مهندسی کامپیوتر

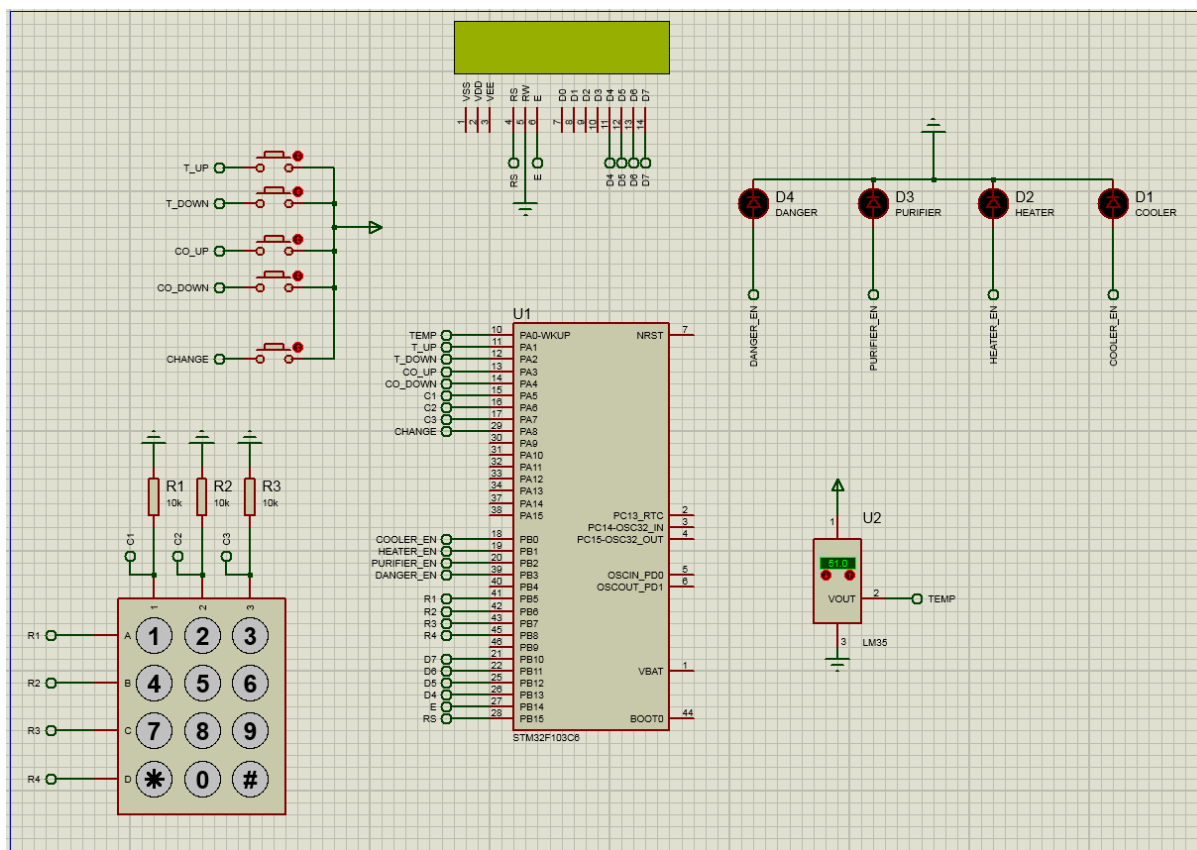
سیستم‌های نهفته و بی‌درنگ

تمرین سری دوم

علی صداقی

۹۷۵۲۱۳۷۸

ابتدا شماتیک و سخت افزار را در نرم افزار Proteus طراحی کردیم.



میکروکنترلر استفاده شده: STM32F103C6

پورت PA0 را برای اتصال سنسور LM35 به همراه ADC در نظر گرفتیم.

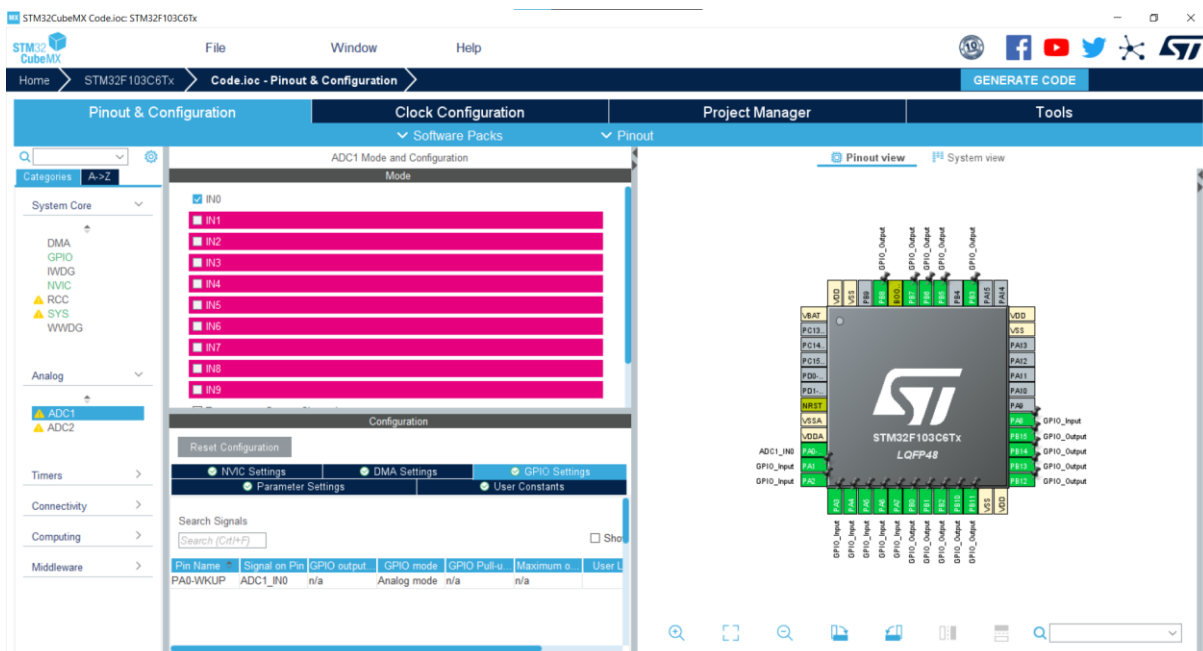
نحوه کارکرد Keypad به صورت Row sweeping (جاروب سطری می باشد).

از چهار LED برای نمایش حالت های زیر استفاده کردیم (از LED چپ به راست):

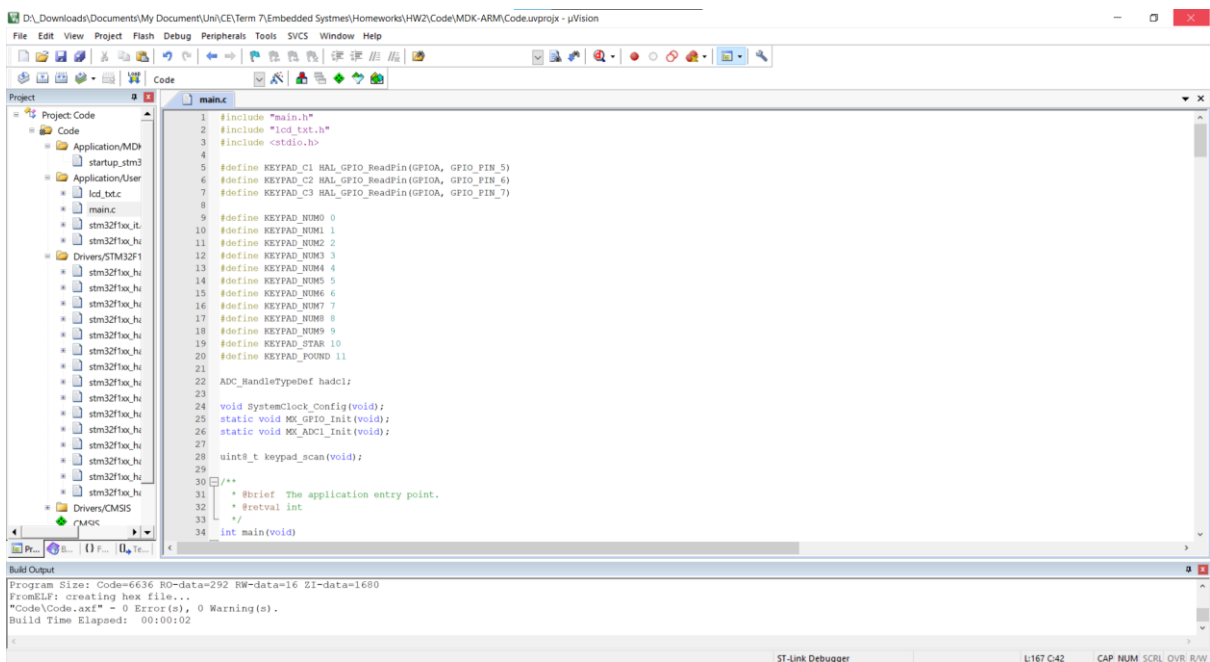
- رنگ آبی: روشن بودن سیستم سرمایشی
- رنگ نارنجی: روشن بودن سیستم گرمایشی
- رنگ سبز: روشن بودن سیستم تهویه
- رنگ قرمز: بیشتر بودن غلظت کربن مونو اکسید از 6ppm

از LCD کاراکتری 16x2 برای نمایش اطلاعات استفاده کردیم.

از نرم افزار STM32CubeMX برای ساختن Boilerplate پروژه و همچنین ست کردن پورت‌ها و ادوات میکروکنترلر استفاده کردیم.



خروجی حاصل شده از این بخش را در نرم افزار Keil uvision MDK-ARM اکسپورت کردیم.



توضیح کد:

ابتدا در دو فایل lcd_txt.h و lcd_txt.c به طراحی Interface مربوط به LCD پرداختم و پورتهای LCD را به پورتهای میکروکنترلر اتصال دادم.

```

1 #ifndef LCDTXT_H
2 #define LCDTXT_H
3
4 #include "stm32f1xx_hal.h"
5
6 /*----- Define LCD Use -----*/
7
8 /*Note: Comment which not use */
9
10 #define LCD16xN //For lcd16x2 or lcd16x4
11 // #define LCD20xN //For lcd20x4
12
13 /*----- Define For Connection -----*/
14
15 #define RS_PORT GPIOB
16 #define RS_PIN GPIO_PIN_15
17
18 #define EN_PORT GPIOB
19 #define EN_PIN GPIO_PIN_14
20
21 #define D7_PORT GPIOB
22 #define D7_PIN GPIO_PIN_10
23
24 #define D6_PORT GPIOB
25 #define D6_PIN GPIO_PIN_11
26
27 #define D5_PORT GPIOB
28 #define D5_PIN GPIO_PIN_12
29
30 #define D4_PORT GPIOB
31 #define D4_PIN GPIO_PIN_13
32
33

```

منطق قسمت بررسی رمز عبور:

```

main.c
63 if (change_mode) {
64     lcd_clear();
65     lcd_puts(0, 0, (int8_t*)"Enter Password:");
66     i = 0;
67
68     key_res = keypad_scan();
69     while (1) {
70         if (key_res == KEYPAD_POUND) {
71             if (entered == password) {
72                 lcd_clear();
73                 lcd_puts(0, 0, (int8_t*)"Entered Succesfully!");
74                 HAL_Delay(2000);
75             }
76             else {
77                 lcd_clear();
78                 lcd_puts(0, 0, (int8_t*)"Wrong Password!");
79                 HAL_Delay(2000);
80                 lcd_clear();
81                 lcd_puts(0, 0, (int8_t*)"TMP:");
82                 lcd_puts(1, 0, (int8_t*)"PPM:");
83                 lcd_puts(0, 8, (int8_t*)"ADC:");
84                 change_mode = 0;
85                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, (GPIO_PinState)0);
86                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, (GPIO_PinState)0);
87                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, (GPIO_PinState)0);
88                 HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, (GPIO_PinState)0);
89                 break;
90             }
91         }
92         else if (key_res != 255) {
93             i++;
94             entered *= 10;
95             entered += key_res;
96             sprintf(string, "%d", key_res);
97             lcd_puts(1, i, (int8_t*)string);
98         }
99         key_res = keypad_scan();
100         HAL_Delay(5);
101     }
}

```

در ابتدای تابع main و بیرون از حلقه بینهایت به Initialize کردن موارد مختلف پرداختم.

```
int32_t ADC_val;
uint8_t tempearture = 25;
uint8_t co = 3;
char string[2];
uint8_t param_temp_high = 25;
uint8_t param_temp_low = 20;
uint8_t param_co = 6;
uint8_t key_res;
uint8_t change_mode = 0;
uint16_t password = 3421;
uint16_t entered = 0;
uint8_t i;

HAL_Init();
SystemClock_Config();
MX_GPIO_Init();
MX_ADC1_Init();

// LCD
lcd_init();
lcd_puts(0, 0, (int8_t*)"TMP:");
lcd_puts(1, 0, (int8_t*)"PPM:");
lcd_puts(0, 8, (int8_t*)"ADC:");
```

سپس مقدار دمای سنسور LM35 را بوسیله ADC خواندم و تبدیل به مقدار قابل استفاده کردم:

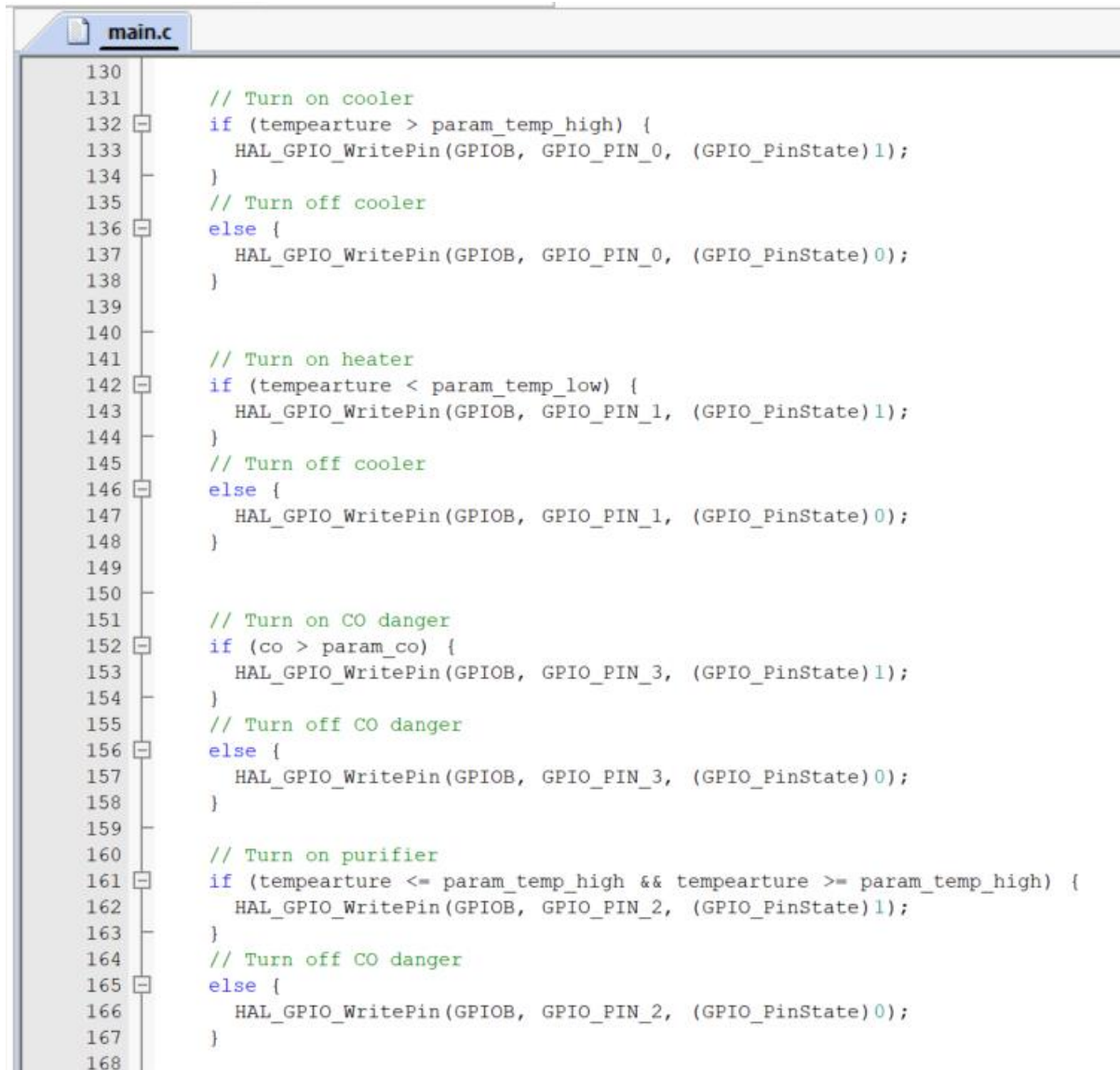
```
125 HAL_ADC_Start(&hadcl);
126 HAL_ADC_PollForConversion(&hadcl, 100);
127 ADC_val = HAL_ADC_GetValue(&hadcl);
128 ADC_val *= 8;
129 HAL_ADC_Stop(&hadcl);
```

نمایش اطلاعات بر روی LCD:

```
169
170 sprintf(string, "%d", tempearture);
171 lcd_puts(0, 4, (int8_t*)string);
172
173 sprintf(string, "%d", co);
174 lcd_puts(1, 4, (int8_t*)string);
175
176 sprintf(string, "%d", ADC_val);
177 lcd_puts(0, 12, (int8_t*)string);
178
179 HAL_Delay(400);
```

در نهایت با توجه به پارامترها و مقادیری که از محیط ورودی گرفتیم رله‌های مربوط به سیستم گرمایشی، سرمایشی، تهویه، اخطار کربن مونو اکسید را تنظیم کردیم:

رد



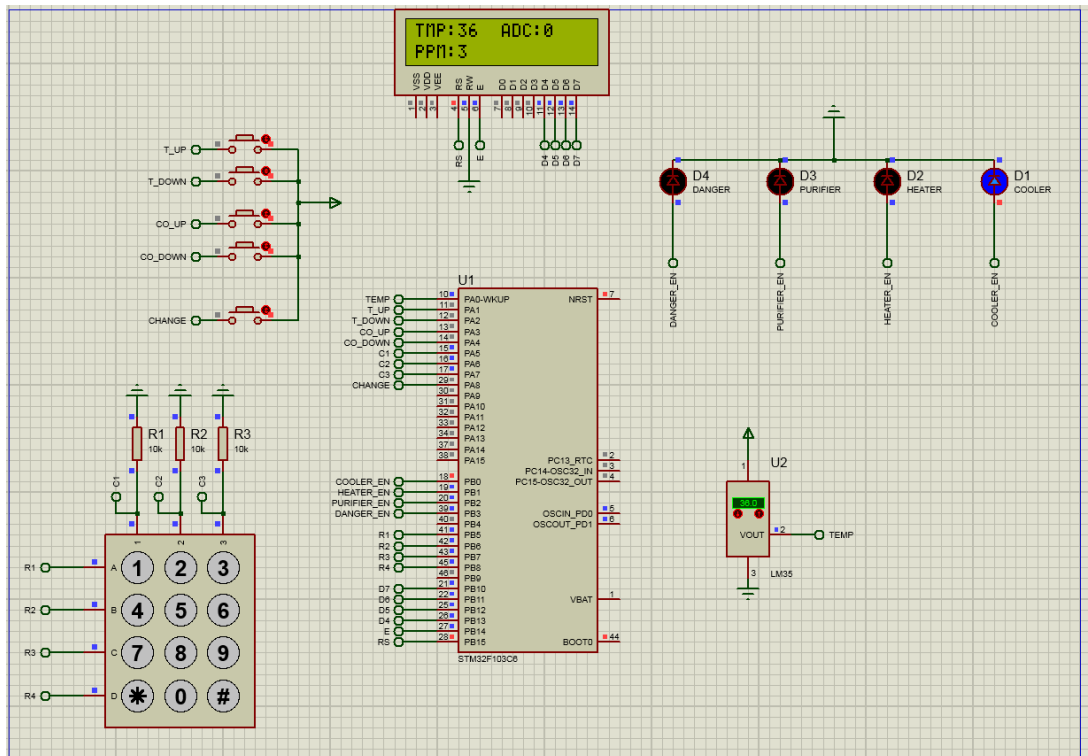
```
130
131 // Turn on cooler
132 if (tempearture > param_temp_high) {
133     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, (GPIO_PinState)1);
134 }
135 // Turn off cooler
136 else {
137     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, (GPIO_PinState)0);
138 }
139
140
141 // Turn on heater
142 if (tempearture < param_temp_low) {
143     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, (GPIO_PinState)1);
144 }
145 // Turn off cooler
146 else {
147     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, (GPIO_PinState)0);
148 }
149
150
151 // Turn on CO danger
152 if (co > param_co) {
153     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, (GPIO_PinState)1);
154 }
155 // Turn off CO danger
156 else {
157     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, (GPIO_PinState)0);
158 }
159
160 // Turn on purifier
161 if (tempearture <= param_temp_high && tempearture >= param_temp_high) {
162     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, (GPIO_PinState)1);
163 }
164 // Turn off CO danger
165 else {
166     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, (GPIO_PinState)0);
167 }
168
```

برای اسکن کردن Keypad از روش جاروب روی سطر استفاده کردم. به این ترتیب که در هر مرحله یک سطر روشن میشد و منتظر کلیک روی دکمه بودیم تا ستون آن سطر نیز روشن شود:

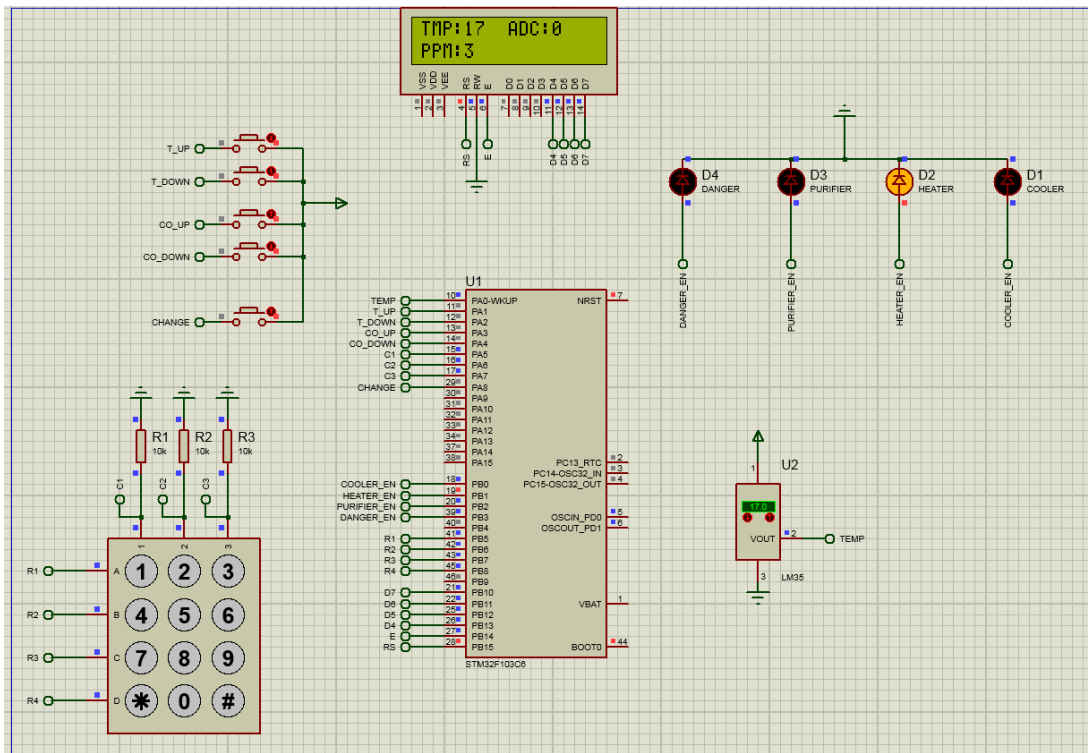
```
184 uint8_t keypad_scan() {
185     uint8_t result = 255;
186     uint16_t delay = 100;
187     //////////////////////////////////// ROW1 ////////////////////////////////////
188     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, (GPIO_PinState)1);
189     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, (GPIO_PinState)0);
190     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, (GPIO_PinState)0);
191     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, (GPIO_PinState)0);
192
193     HAL_Delay(delay);
194     if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_5))
195         result = KEYPAD_NUM1;
196     else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6))
197         result = KEYPAD_NUM2;
198     else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7))
199         result = KEYPAD_NUM3;
200
201     //////////////////////////////////// ROW2 ////////////////////////////////////
202     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, (GPIO_PinState)0);
203     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, (GPIO_PinState)1);
204     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, (GPIO_PinState)0);
205     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, (GPIO_PinState)0);
206
207     HAL_Delay(delay);
208     if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_5))
209         result = KEYPAD_NUM4;
210     else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6))
211         result = KEYPAD_NUM5;
212     else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7))
213         result = KEYPAD_NUM6;
214
215     //////////////////////////////////// ROW3 ////////////////////////////////////
```

نتایج و خروجی پروژه:

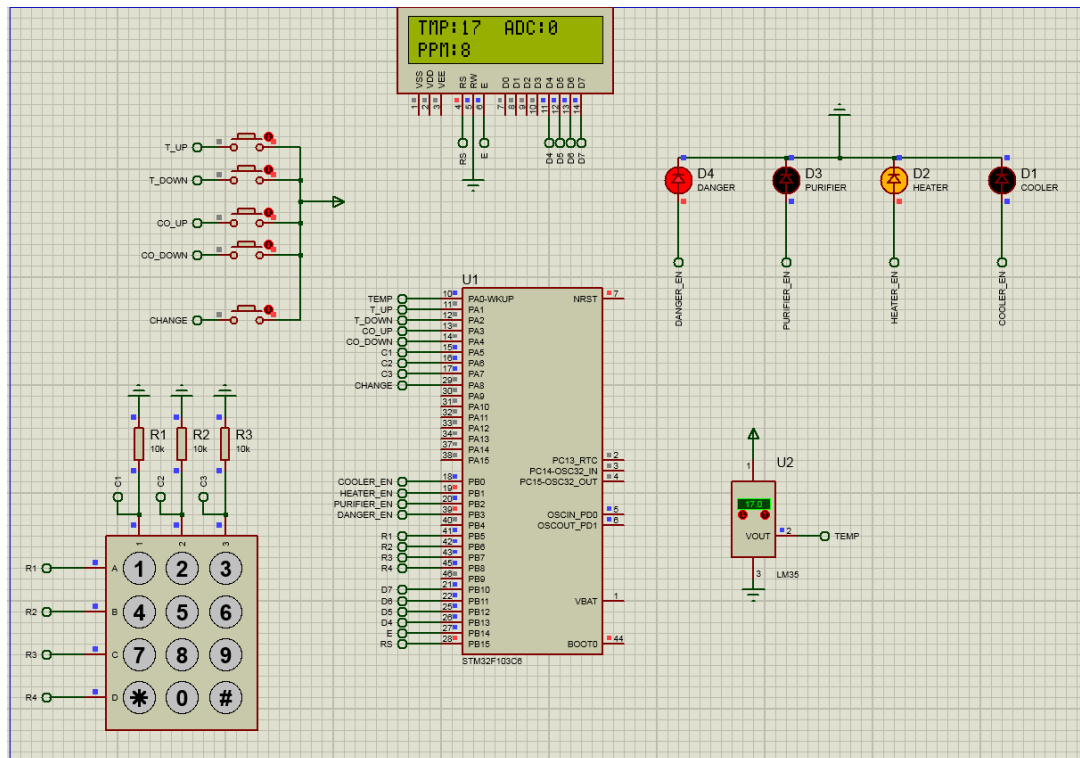
۱- روشن بودن کولر وقتی دما از حد تعیین شده بالاتر می‌رود:



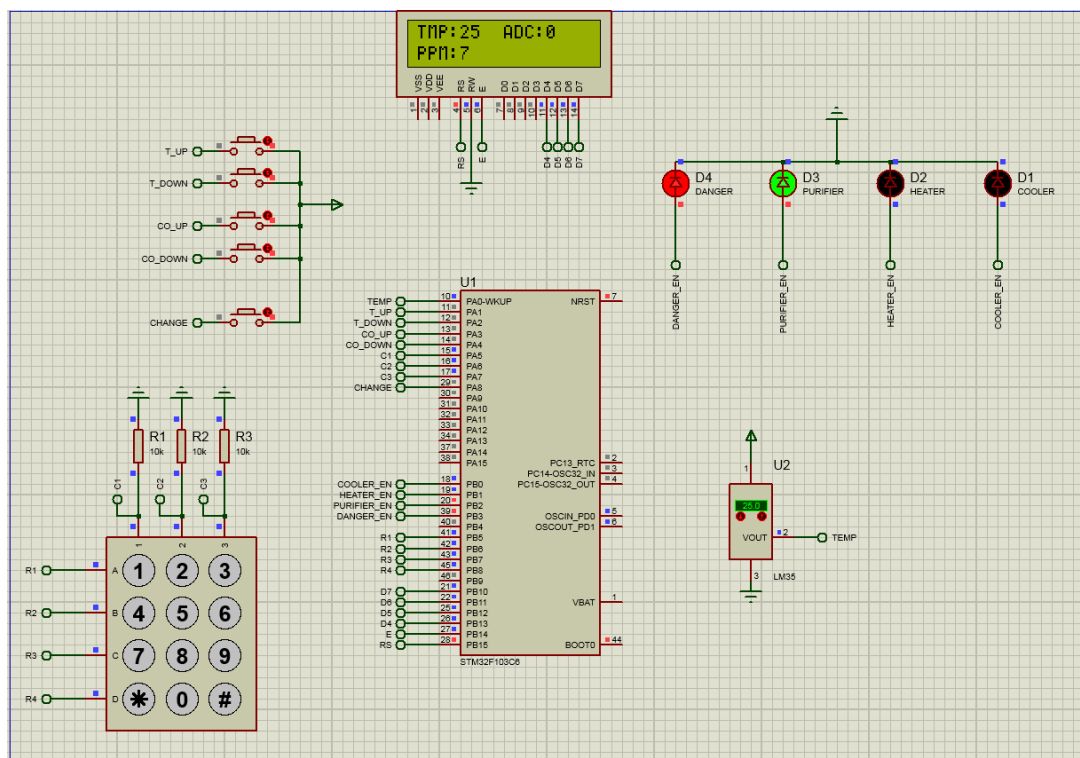
۲- روشن شدن سیستم گرمایشی هنگامی که دما از حد تعیین شده پایین تر می‌رود:



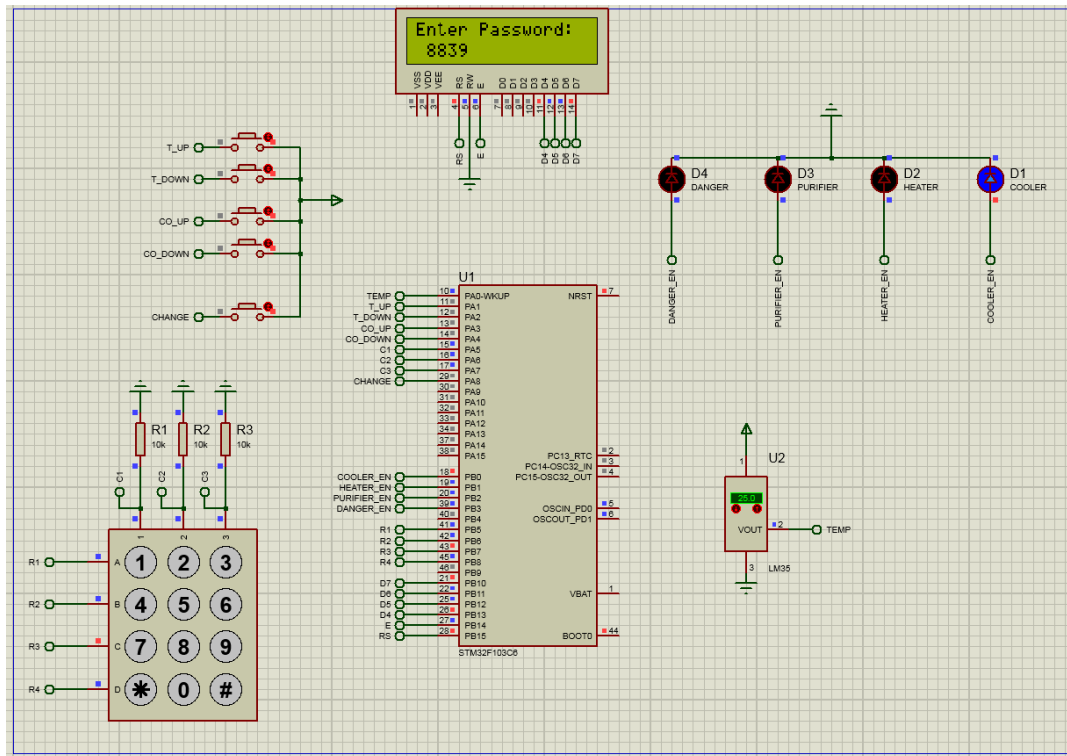
۳- روشن شدن LED قرمز هنگامی که میزان کربن مونو اکسید بیشتر از 6ppm است.



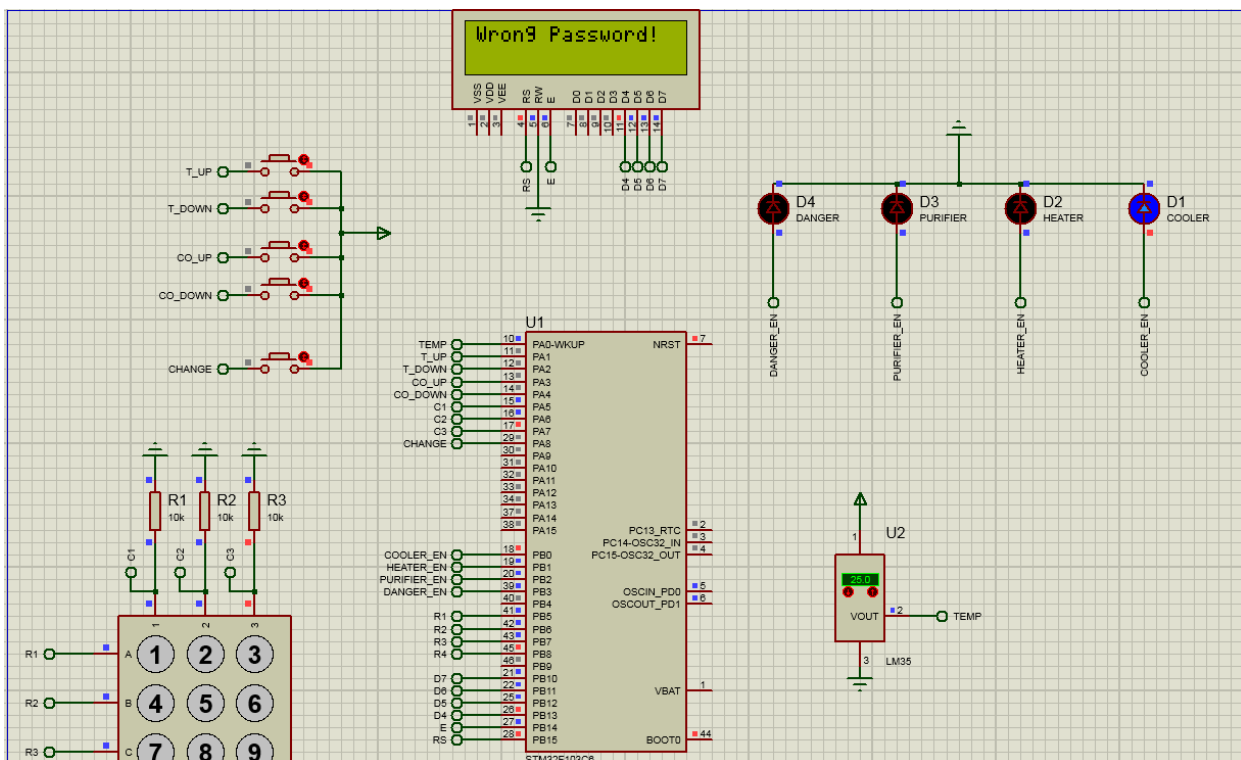
۴- روشن شدن تهویه هوا:



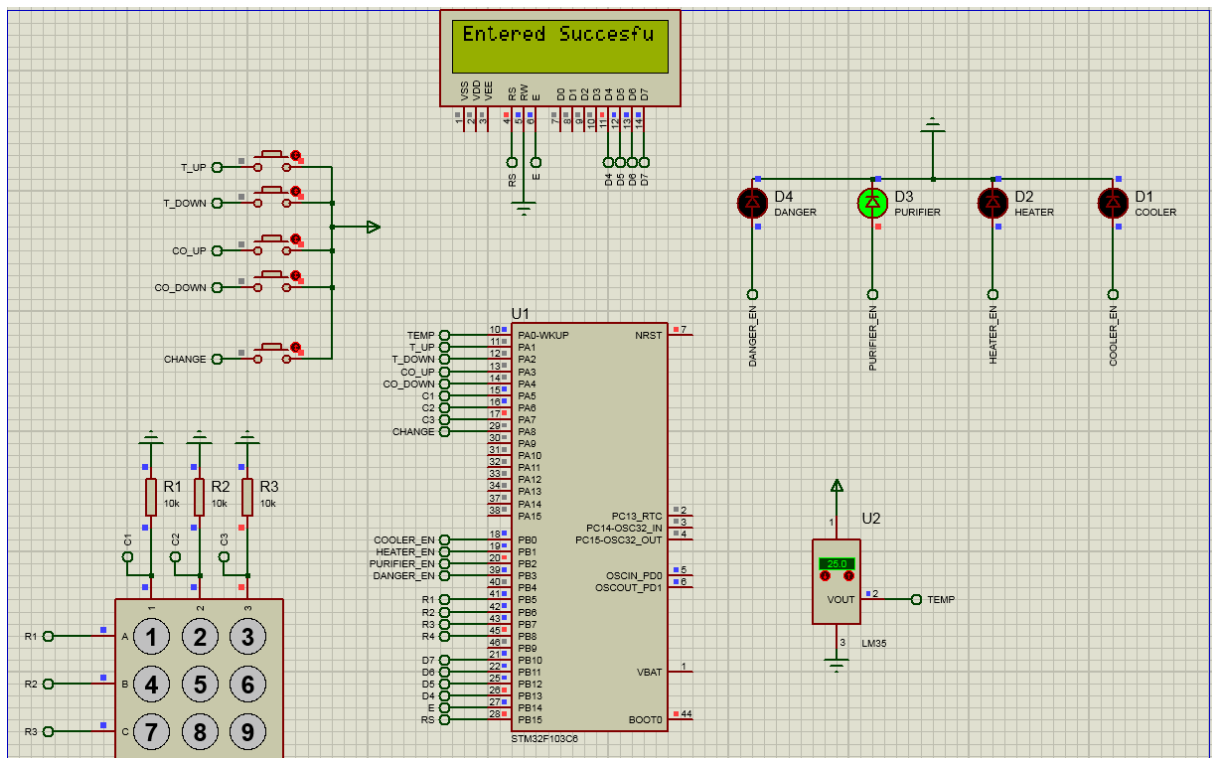
۵- وارد کردن پسورد برای انجام تغییرات



۶- اشتباه بودن پسورد (در این صورت پیام Wrong به مدت دو ثانیه نمایش داده می شود سپس از صفحه رمز خارج می شویم).



۷- در صورت صحیح بودن پسورد ابتدا پیام Successfully برای ۲ ثانیه نمایش داده می شود سپس پارامترها تغییر می یابند.



چالش‌ها:

۱- میکروکنترلر STM32 دارای بیشینه فرکانس 72MHz است و دستیابی به این فرکانس در شبیه‌سازی بسیار اذیت‌کننده هست. عملکرد میکروکنترلر در هنگام خواندن اطلاعات ADC، نوشتن روی LCD، خواندن دکمه فشار داده شده در Keypad و ... مورد آسیب قرار می‌گیرد و در بسیاری از اوقات اخطارهای زیر در فرکانس بالا دریافت می‌شود که باعث می‌شود شبیه‌ساز نتواند همه چیز را طبق برنامه انجام دهد و بخشی از اعمال miss می‌شوند:

Simulation is not running in real time due to excessive CPU load.

[HD44780] Controller received data whilst busy. [LCD1]

اگر فرکانس را پایین بیاوریم باعث می‌شود زمان برخی از Task‌ها بیشتر از کلاک طول بکشد و عملاً آن Task صورت نگیرد.

۲- منابع در خصوص STM32 بسیار کم بود و اکثراً ناقص یا با مشکل همراه بود.

۳- تمرین خیلی بیش از حد انتظار وقت گرفت. ۶ روز کامل (روزی ۶-۷ ساعت) زمان برد تا این تمرین انجام شود.