

گزارش کار جلسه هفتم ۴ آذر

آزمایشگاه سیستمهای ریزپردازنده و مدارهای واسطه گروه ۳

آنوشا شریعتی ۹۹۲۳۰۴۱

مهشاد اکبری سریزدی ۹۹۲۳۰۹۳

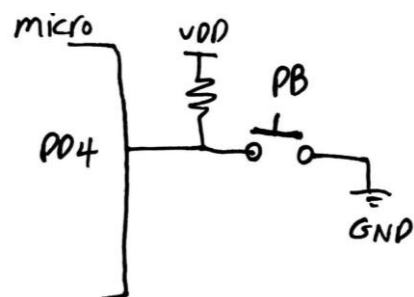
امیرحسین منصوری

آزمایش سری ۶ :

بخش پنجم : برنامه ای بنویسید که با زدن یک باتن خارجی، LED به صورت toggle رفتار کند.

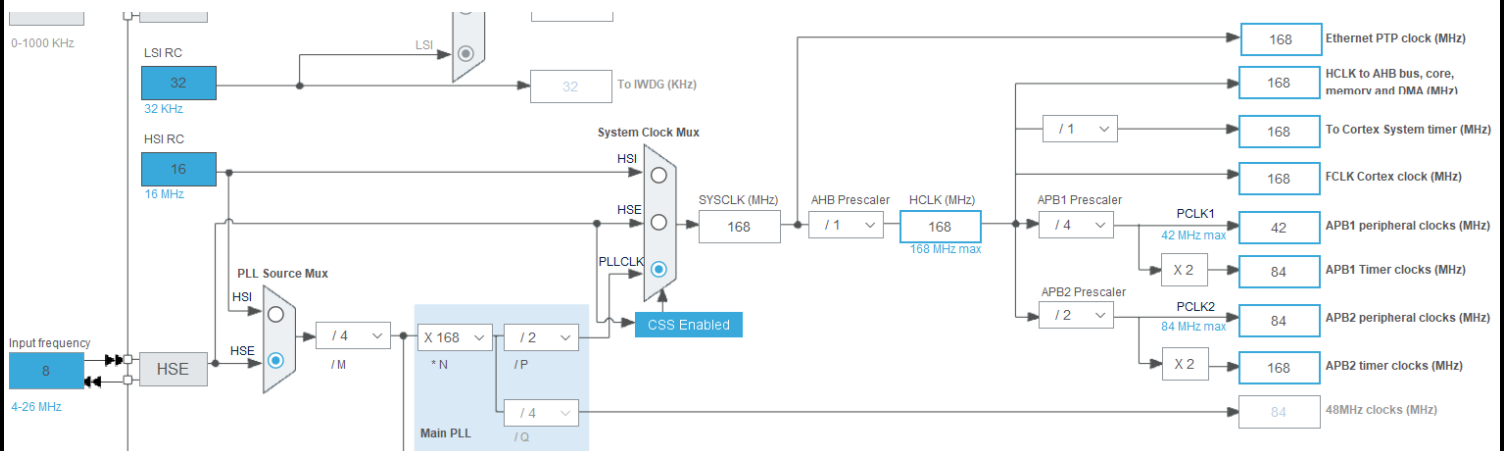
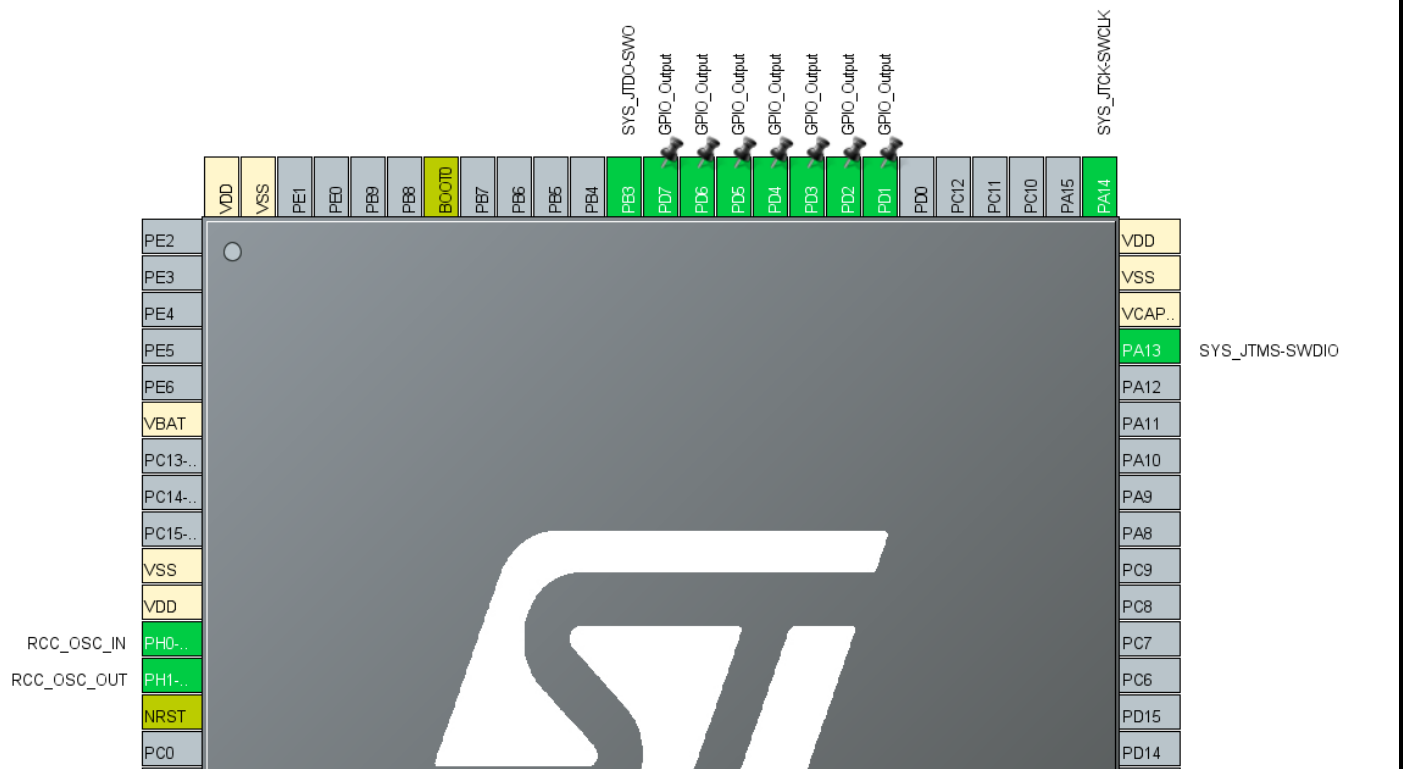
برای پیاده کردن کد قسمت قبل روی یک مدار خارجی مدار را به صورت زیر میبندیم . در مدار زیر اگر پوش باتن فشار داده نشود پایه میکرو پول آپ میشود و اگر فشرده شود زمین میشود. پس کد قسمت قبل بدون تغییر میماند و فقط پین D۴ را از بیرون استفاده میکنیم.

```
93  /* USER CODE BEGIN WHILE */
94  while (1)
95  {
96      /* USER CODE END WHILE */
97
98      /* USER CODE BEGIN 3 */
99
100     if (! HAL_GPIO_ReadPin(GPIOD , GPIO_PIN_4)){
101         HAL_Delay(100);
102         HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_12);
103         HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_13);
104         HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_14);
105         HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_15);
106         while(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_4));
107         HAL_Delay(100);
108     }
109 }
110 }
111 /* USER CODE END 3 */
112 }
```



آزمایش سری ۷:

برای انجام دادن این آزمایش ابتدا کتابخانه های مربوط به LCD را به پروژه اضافه کرده تا بتوانیم از توابع مورد نیاز استفاده کنیم. همچنین تنظیمات مربوط به کلاک پروژه را مانند آزمایش های قبل انجام میدهیم و پایه هایی که میخواهیم به LCD متصل کنیم را به صورت خروجی تعریف میکنیم.



بخش سوم :

درون فایل LCD_۱۶*۲.h توابع مورد نیاز برای کار کردن با LCD آمده است و برای آشنایی با نحوه کارکرد توابع میتوان به فایل LCD_۱۶*۲.c مراجعه کرد. برای مثال از تابع LCD_Set_Cursor برای تغییر مکان مکان نما استفاده میشود که دو مقدار c و r را به صورت ورودی میگیرد. طبق تابع تعریف شده زیر، کاراکتر r مقدار ۱ یا ۲ را میگیرد که نشان دهنده ستون است و c هم مکان نما را در یک خط جا به جا میکند.

```
57 void LCD_Set_Cursor(unsigned char r, unsigned char c)|
58 {
59     unsigned char Temp,Low4,High4;
60     if(r == 1)
61     {
62         Temp = 0x80 + c - 1; //0x80 is used to move the cursor
63         High4 = Temp >> 4;
64         Low4 = Temp & 0x0F;
65         LCD_CMD(High4);
66         LCD_CMD(Low4);
67     }
68     if(r == 2)
69     {
70         Temp = 0xC0 + c - 1;
71         High4 = Temp >> 4;
72         Low4 = Temp & 0x0F;
73         LCD_CMD(High4);
74         LCD_CMD(Low4);
75     }
76     DWT_Delay_ms(2);
77 }
```

بخش پنجم :

با مراجعه به فایل LCD_۱۶*۲_cfg.c پورت و پین هایی که میخواهیم برای LCD استفاده کنیم را تغییر میدهیم.

```
12
13 #include "LCD16x2.h"
14
15 const LCD16x2_CfgType LCD16x2_CfgParam =
16 {
17     GPIOD,
18     GPIO_PIN_4,
19     GPIO_PIN_5,
20     GPIO_PIN_6,
21     GPIO_PIN_7,
22     GPIO_PIN_1,
23     GPIO_PIN_2,
24     20
25 };
26
```

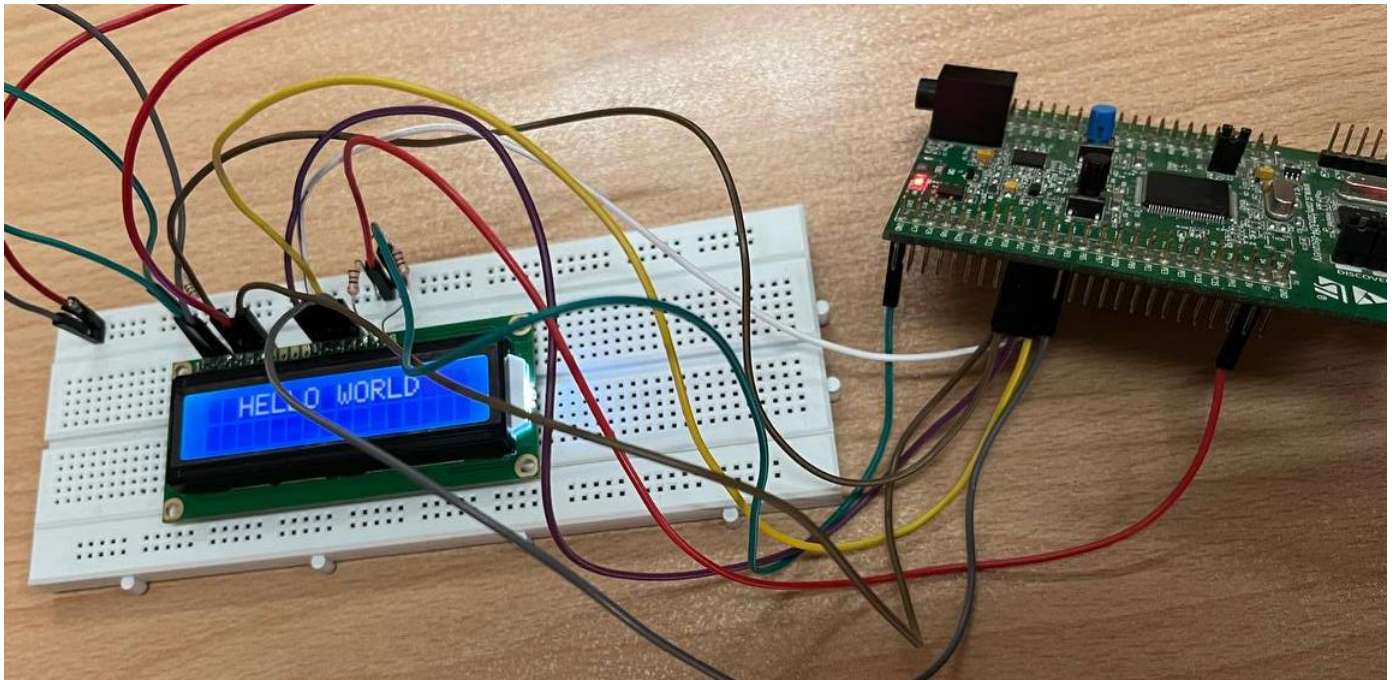
بخش هفتم و هشتم :

قبل از تابع وایل اصلی با استفاده از توابع زیر صفحه LCD را پاک کرده و مکان نما را مشخص میکنیم. و چون میخواهیم پیام HELLO WORLD یک بار چاپ شود دستور LCD_Write_String را بیرون از وایل مینویسیم.

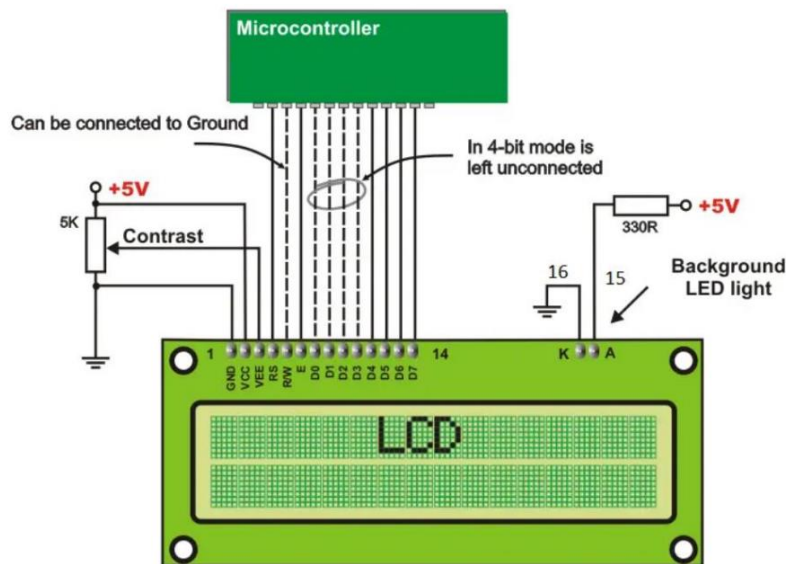
```

65 int main(void)
66 {
67     /* USER CODE BEGIN 1 */
68
69     /* USER CODE END 1 */
70
71     /* MCU Configuration-----*/
72
73     /* Reset of all peripherals, Initializes the Flash interface and the Systick.
74     HAL_Init();
75
76     /* USER CODE BEGIN Init */
77
78     /* USER CODE END Init */
79
80     /* Configure the system clock */
81     SystemClock_Config();
82
83     /* USER CODE BEGIN SysInit */
84
85     /* USER CODE END SysInit */
86
87     /* Initialize all configured peripherals */
88     MX_GPIO_Init();
89     /* USER CODE BEGIN 2 */
90     LCD_Init();
91     LCD_Clear();
92     LCD_Set_Cursor(1, 1);
93     LCD_Write_String(" HELLO WORLD ");
94     /* USER CODE END 2 */
95

```



همچنین دقت شود که برای روشن کردن LCD و تنظیم کنتراست و روشنایی پس زمینه از اطلاعات داده شده در دستور کار برای اتصال پایه های LCD استفاده میکنیم.



بخش نهم :

در این بخش می‌خواهیم در داخل حلقه وایل به کمک تابع دیلی ساعتی طراحی کنیم که هر ثانیه مقدار آن تغییر کند. برای زدن این برنامه تنظیمات ابتدایی زیر را در بیرون از حلقه وایل انجام می‌دهیم و متغیرهای second و min و hr را از نوع اینتیجر و رشته str را تعریف می‌کنیم.

```

85
86 /* USER CODE END SysInit */
87
88 /* Initialize all configured peripherals */
89 MX_GPIO_Init();
90 /* USER CODE BEGIN 2 */
91 LCD_Init();
92 LCD_Clear();
93 LCD_Set_Cursor(1, 1);
94
95 int second = 0 ;
96 int min = 0 ;
97 int hr = 0 ;
98
99
100 char str[16] ;
101 /* USER CODE END 2 */

```


سپس درون حلقه وایل الگوریتمی با سه ایف تو در تو به صورت زیر پیاده سازی میکنیم. در نهایت با استفاده از مقادیر اینتیجر های تعریف شده که در هر حلقه تغییر پیدا میکنند رشته str را ساخته و روی LCD چاپ میکنیم.

```

103  /* Infinite loop */
104  /* USER CODE BEGIN WHILE */
105  while (1)
106  {
107      /* USER CODE END WHILE */
108
109      /* USER CODE BEGIN 3 */
110
111      HAL_Delay(999);
112      LCD_Clear();
113      HAL_Delay(100);
114      if (second != 60){
115          second++;
116      }
117      else {
118          second = 0 ;
119      if (min != 60 ){
120          min++;}
121      else{
122          min =0 ;
123          if (hr != 24){
124              hr++;}
125          else {
126              hr = 0 ;
127          }
128      }
129      }
130      sprintf(str, "%d : %d: %d ", hr , min , second);
131      LCD_Write_String(str);
132
133  }
```

