

گزارش کار آزمایش ۵ قسمت الف

امین احسانی مهر / شماره دانشجویی: ۹۹۲۴۳۰۰۹

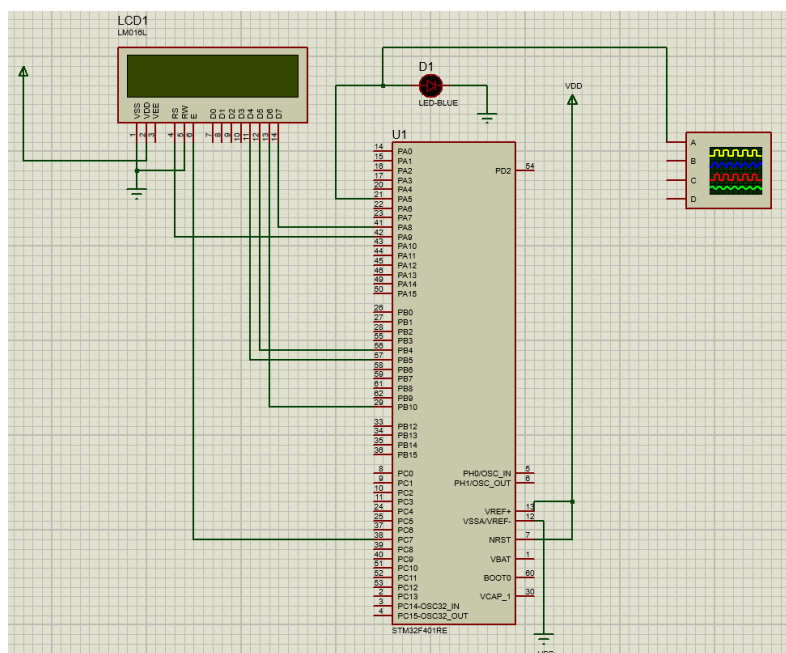
نگار هنرور صدیقیان / شماره دانشجویی: ۹۹۲۴۳۰۷۶

سوالات تحلیلی:

۱. همانطور که می‌دانیم، شیلد LCD دارای ۸ بیت دیتا برای منتقل کردن داده و یک پین RW برای خواندن/نوشتن، یک پین RS برای انتخاب command/data register، یک پین E (enable) یک پین VEE برای افزایش contrast صفحه نمایش است. تمامی این مقادیر باید به LCD داده شود. پس در نتیجه پورت‌های nucleo باید پورت‌های خروجی باشد.

۲. برای راه‌اندازی LCD، ابتدا تابع `SystemCoreClockUpdate` را صدا می‌کنیم که فرکانس میکروکنترلر را ست کند تا در ادامه از آن استفاده کند. سپس تابع `(HAL_Init)` صدا می‌شود تا بتوانیم در ادامه از توابع کتابخانه HAL استفاده کنیم. کلاک‌های هر پورت را ست می‌کنیم. سپس مشخص می‌کنیم که کدام پورت مدنظر ما قرار است به عنوان خروجی از آن استفاده شود. در آخر با فراخوانی تابع `(LCD_Init)` که باعث راه‌اندازی شیلد LCD می‌شود و با تابع `(LCD_Puts)`، مقدار مورد نظر خود را روی LCD نمایش می‌دهیم.

گزارش کار:



شکل ۱ - شماتیک مدار و نحوه اتصال میکروکنترلر به LCD و oscilloscope

ابتدا از شماتیک شروع می‌کنیم. قرار است از ۴ بیت دیتای LCD استفاده کنیم. پس در نتیجه، از D4 تا D7 را به میکروکنترلر متصل می‌کنیم و همینطور مقدار E و RS و RW و VEE را به آن وصل می‌کنیم. همچنین VDD و VSS را به power و ground متصل می‌کنیم. همچنین از یک بیت پورت A برای ایجاد سیگنال موج مربعی استفاده می‌کنیم. این سیگنال را به oscilloscope متصل می‌کنیم تا سیگنال، خروجی، را به صورت گرافیکی مشاهده کنیم.

پین‌های LCD به صورت زیر تعریف می‌شوند:

| Pin | Symbol | I/O | Description |
|-----|--------|-----|--|
| 1 | VSS | -- | Ground |
| 2 | VCC | -- | +5V power supply |
| 3 | VEE | -- | Power supply to control contrast |
| 4 | RS | I | RS = 0 to select command register, RS = 1 to select data register |
| 5 | R/W | I | R/W = 0 for write, R/W = 1 for read |
| 6 | E | I | Enable |
| 7 | DB0 | I/O | The 8-bit data bus |
| 8 | DB1 | I/O | The 8-bit data bus |
| 9 | DB2 | I/O | The 8-bit data bus |
| 10 | DB3 | I/O | The 8-bit data bus |
| 11 | DB4 | I/O | The 4/8-bit data bus |
| 12 | DB5 | I/O | The 4/8-bit data bus |
| 13 | DB6 | I/O | The 4/8-bit data bus |
| 14 | DB7 | I/O | The 4/8-bit data bus |

شکل ۲ - پین‌های شیلد LCD و کارکرد آن‌ها

```

1 #include "stm32f4xx_hal.h"
2 #include "stm32f4xx.h"
3 #include "LCD16x2Lib/LCD.h"
4
5 int main(void)
6 {
7     SystemCoreClockUpdate();
8     HAL_Init();
9
10    __HAL_RCC_GPIOA_CLK_ENABLE();
11    __HAL_RCC_GPIOB_CLK_ENABLE();
12    __HAL_RCC_GPIOC_CLK_ENABLE();
13
14    GPIO_InitTypeDef PinConfig;
15    PinConfig.Pin = GPIO_PIN_5;
16    PinConfig.Mode = GPIO_MODE_OUTPUT_PP;
17    PinConfig.Pull = GPIO_NOPULL;
18    PinConfig.Speed = GPIO_SPEED_FREQ_HIGH;
19    HAL_GPIO_Init(GPIOA, &PinConfig);
20
21
22    LCD_Init();
23    LCD_Puts(1, 0, "Hi there, Amin!");
24    HAL_Delay(5000);

```

حال به سراغ کد می‌رویم. درون برنامه پس از include کردن کتابخانه‌های مورد نیاز و مواردی که در سوال دوم بخش تحلیلی ذکر شد، با دادن رشته مورد نظر به تابع LCD_PUTS() و مشخص کردن محل شروع cursor، آن رشته را روی LCD نمایش می‌دهیم. این مقدار تا ۵ ثانیه باید روی صفحه باشد، پس با تابع HAL_DELAY() یک دلیلی ۵۰۰ میلی ثانیه‌ای قرار می‌دهیم.

شکل ۳ - فراخوانی SystemCoreClockUpdate و

HAL_Init() و مشخص کردن کلاک پروت خروجی بودن پین A

و در آخر، نمایش خوش‌آمدگویی خواسته شده

```

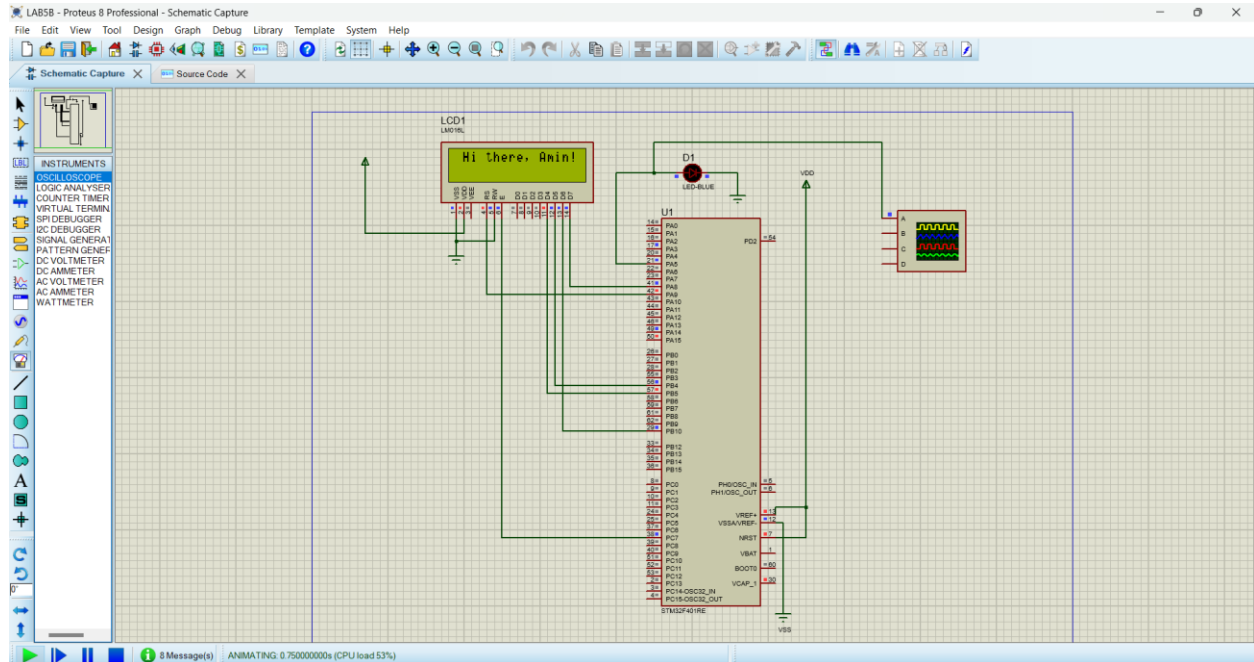
29 while (1)
30 {
31     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
32     HAL_Delay(500);
33     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
34     HAL_Delay(500);
35     counter += 1;
36     if(counter == 1){
37         if(sec_counter == 500){
38             sec_counter = 0;
39         }
40         LCD_Init();
41         sec_counter += 1;
42         sprintf(str, 12, "%d", sec_counter);
43         LCD_Puts(7, 0, str);
44         counter = 0;
45     }
46 }
47

```

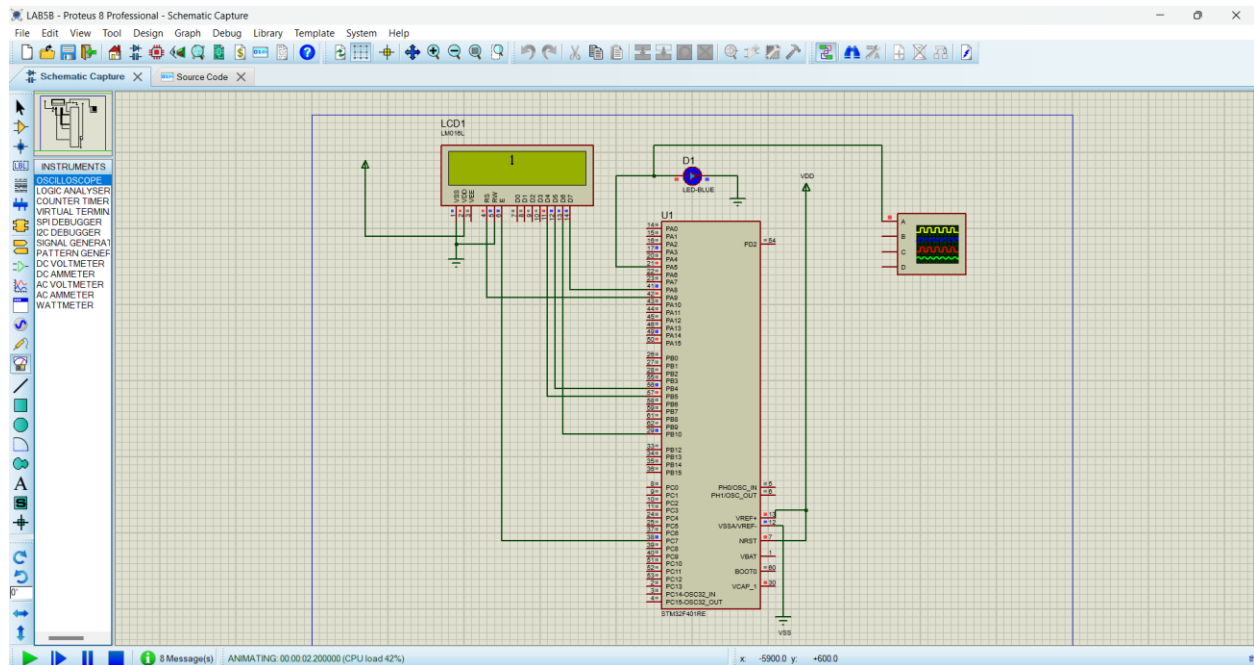
شکل ۴ - نمایش دادن تعداد پالس زده شده در ثانیه / ۵۰۰

حال در بخش بعد، باید یک سیگنال موج مربع با فرکانس 500 هرتز ایجاد کنیم و به ازای هر ۵۰۰ پالس به وجود آمده، تعداد پالس در ثانیه که روی LCD نمایش داده می‌شود را آپدیت کنیم. پس به ازای هر ۵۰۰ پالس به وجود آمده، یک واحد روی LCD اضافه می‌کنیم. برای اینکار، یک متغیر counter داریم که وظیفه آن بررسی آن است که ۵۰۰ پالس انجام شده است یا خیر. اگر بود، مقدار آن را ۱ می‌کنیم تا وارد حلقه شرط بشویم تا عدد مدنظر را روی LCD نمایش بدهیم. این مقدار که با متغیر sec_counter مقاداردهی شده است، اگر از حدی (که در اینجا برابر ۵۰۰ قرار دادیم) بیشتر شد، این مقدار را reset می‌کنیم.

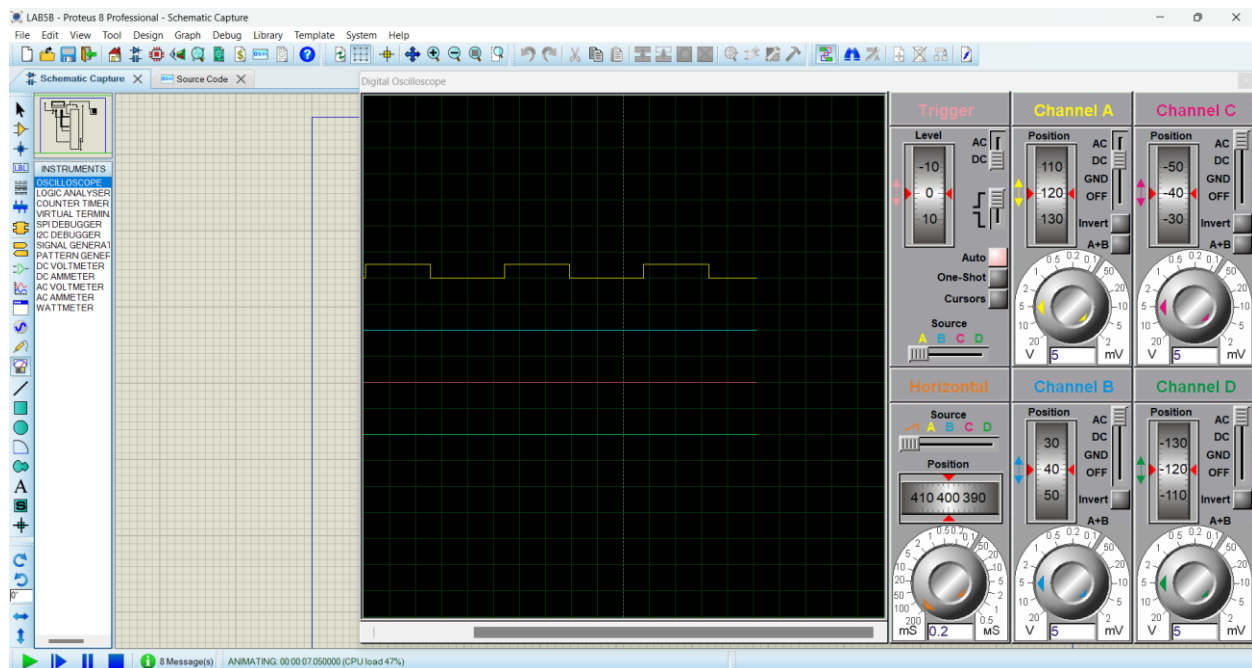
مدار بسته شده را در proteus شبیه‌سازی می‌کنیم. خروجی به دست آمده به صورت زیر خواهد شد:



شکل ۵ - نمایش خوش‌آمدگویی



شکل ۶ - تعداد پالس‌های انجام شده در واحد زمان



شکل ۷- پالس مربعی تولید شده، نمایش داده شده بر روی oscilloscope