به نام خداوند جان و خرد

درس ابزار دقیق

گروه کنترل

مدرس: محمدرضا نیری | تمرین سری سوم | نیمسال دوم ۱۴۰۵-۱۴۰۴

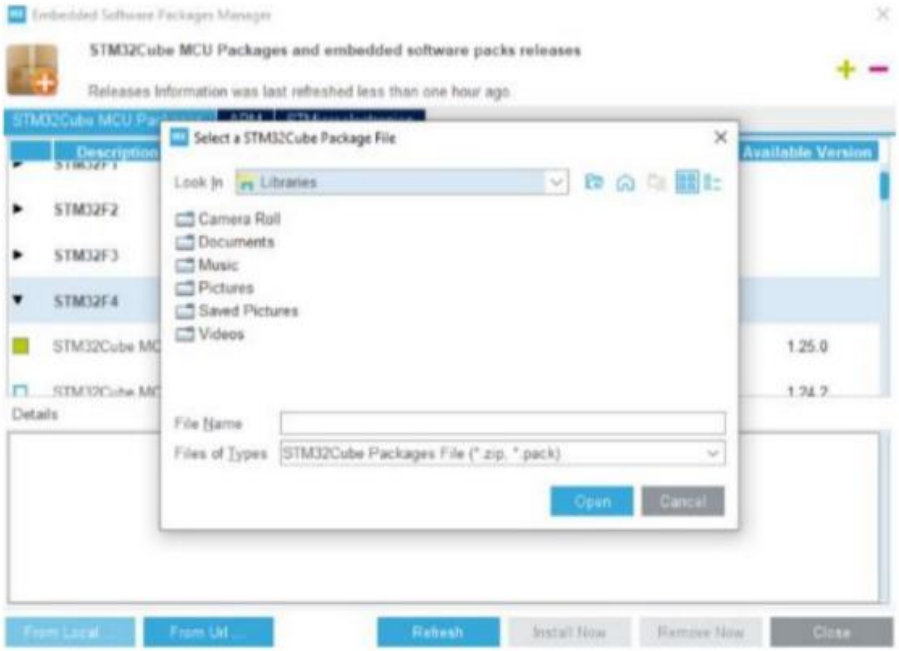
در این تمرین با میکروکنترلر های ARM شرکت ST به نام STM32 آشنا خواهید شد. برای این کار از نرم افزار STM32CubeIDE برای برنامه نویسی میکروکنترلر و از Proteus برای شبیه سازی و دیدن نتایج استفاده خواهیم کرد. الزام است ویدیو های آموزشی موجود در صفحه درس را قبل از انجام تمرین مشاهده کنید

❖ نصب و آماده سازی نرم افزار ها

نرم افزار را از سایت soft98 دانلود و نصب کنید.

سپس پکیج F32STM1 را از سایت sisoog دانلود کنید. برای اینکار در سایت گفته شده فایل "دانلود پکیج F1"را دانلود کنید.(شکل زیر)

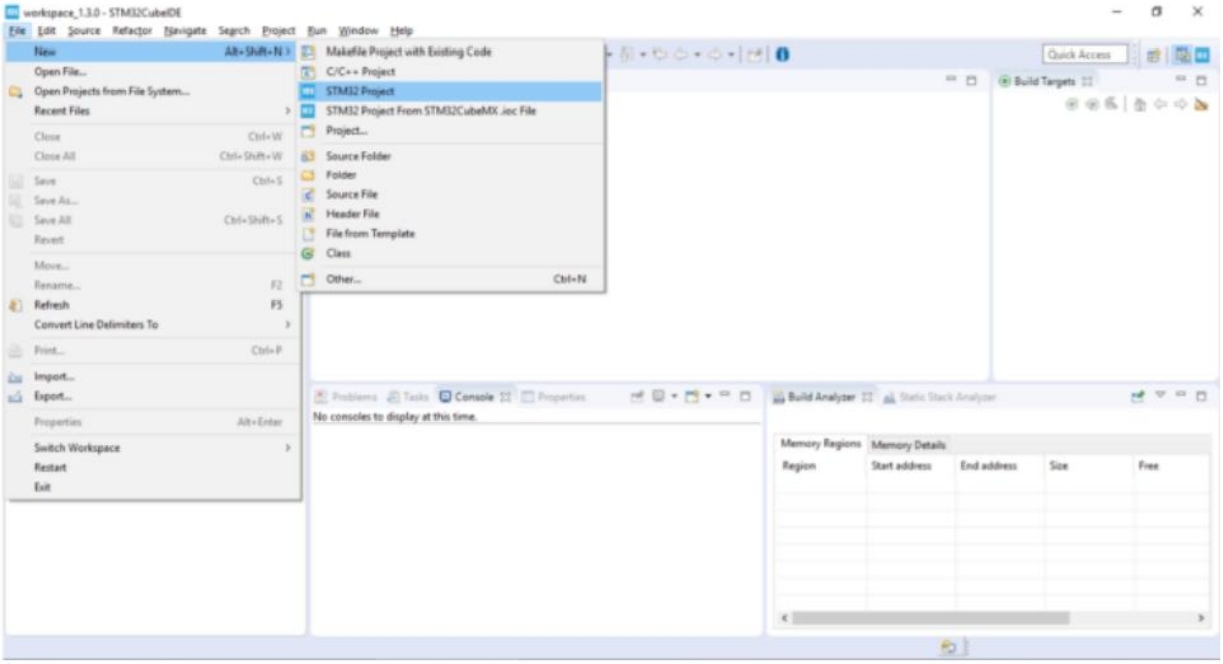
بعد از دانلود پکیج، همانند روش قبلی به قسمت Manage embedded software package برید و از قسمت Local From فایل دانلود شده را انتخاب کنید.



حال نرم افزار آماده ی ساخت پروژه میباشد.

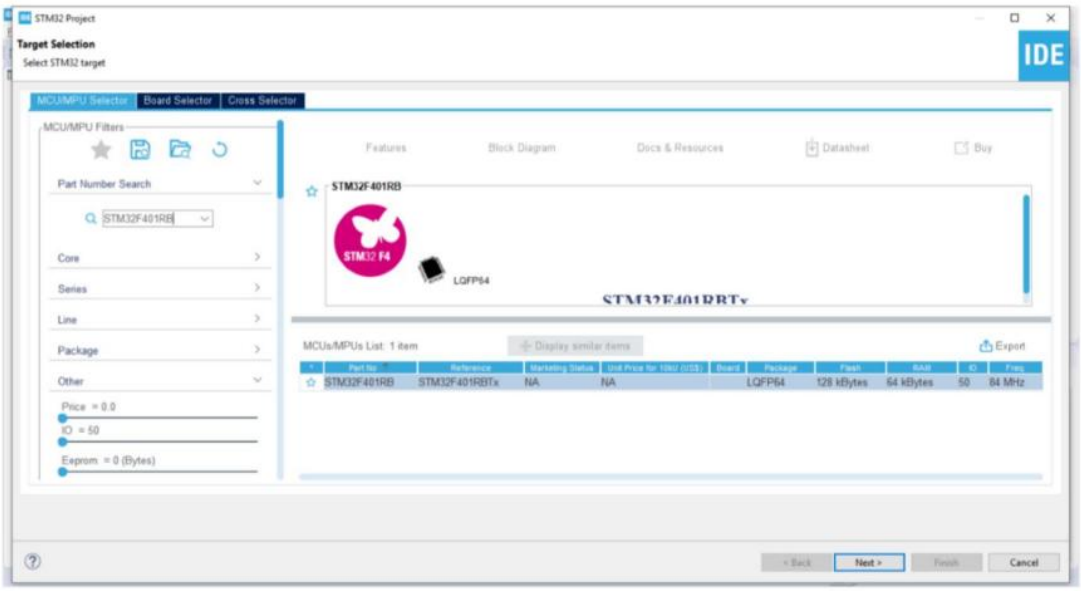
❖ **ساخت پروژه**

از مسیر شکل زیر یک پروژه جدید بسازید. (جهت افزایش سرعت نرم افزار، پیشنهاد میشود که در این قسمت اینترنت خود را خاموش کنید)

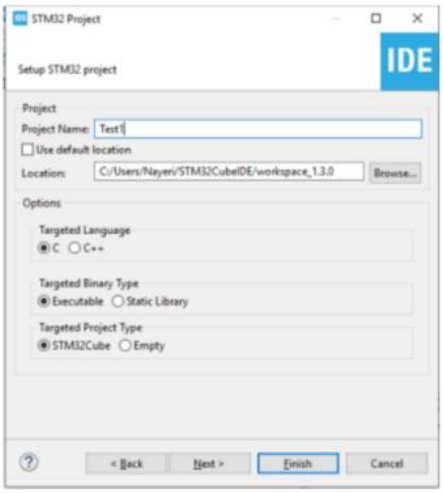


سپس صفحه ای به صورت شکل زیر باز خواهد شد. در قسمت Part Number Search میکروکنترلر مطابق جدول زیر را تایپ کرده و در پنجره وسط آن را انتخاب کنید و بر روی Next کلیک کنید.

میکروکنترلر	باقیمانده شماره دانشجویی بر ۳
STM32F103C6	۰
STM32F103R6	۱
STM32F103T6	۲

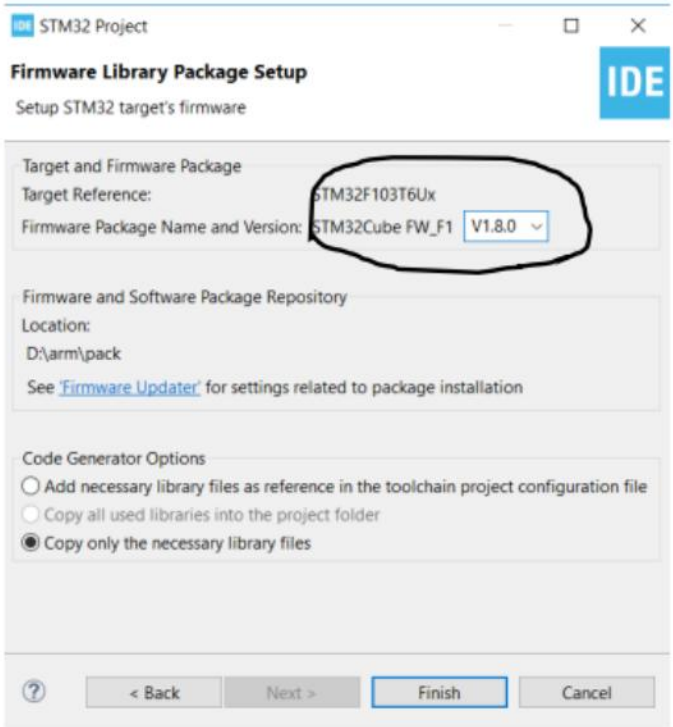


سپس پنجره ای به صورت شکل زیر باز خواهد شد که در آن میبایست یک نام برای پروژه انتخاب کرده و سپس مسیری را که پروژه در آن ذخیره می شود انتخاب کنید



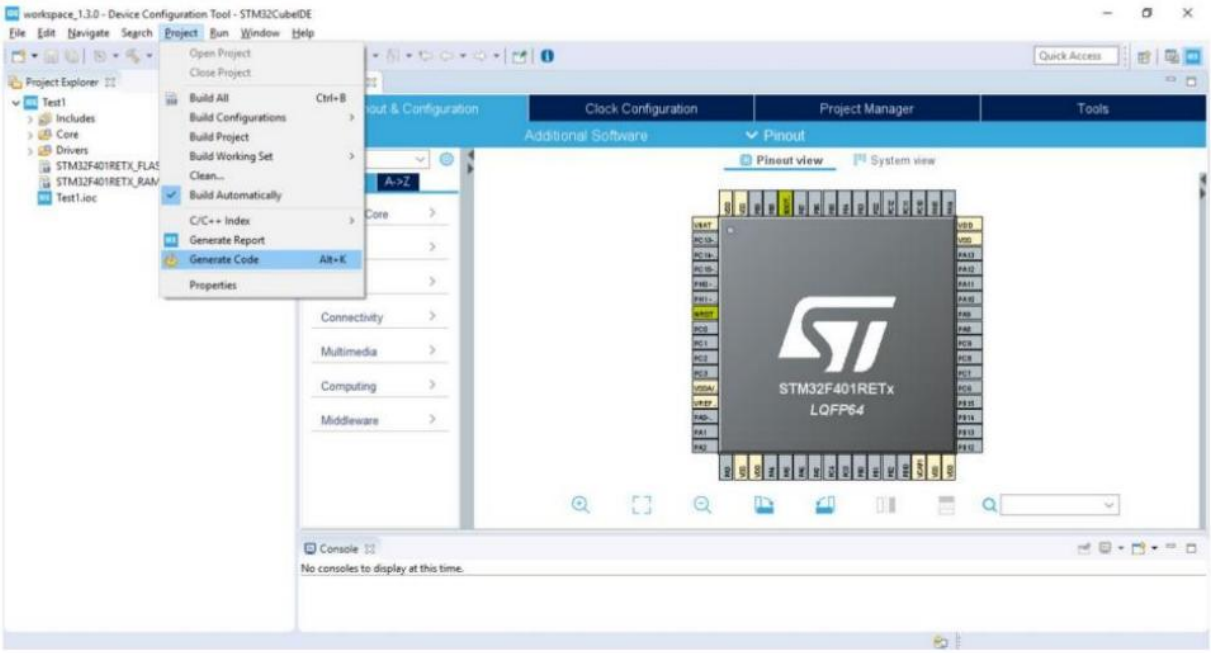
توجه : پس از انتخاب نام و آدرس پروژه بر روی Next کلیک کنید. در قسمت علامت زده ی شکل زیر دقت کنید که باید دقیقا همان ورژن پکیجی انتخاب شود که بر روی نرم افزار دانلود شده است. که اگر از روش دوم در

نصب نرم افزار استفاده کرده باشید باید V 1.8.0را انتخاب کنید در غیر این صورت پروژه با موفقیت ساخته نمیگردد.

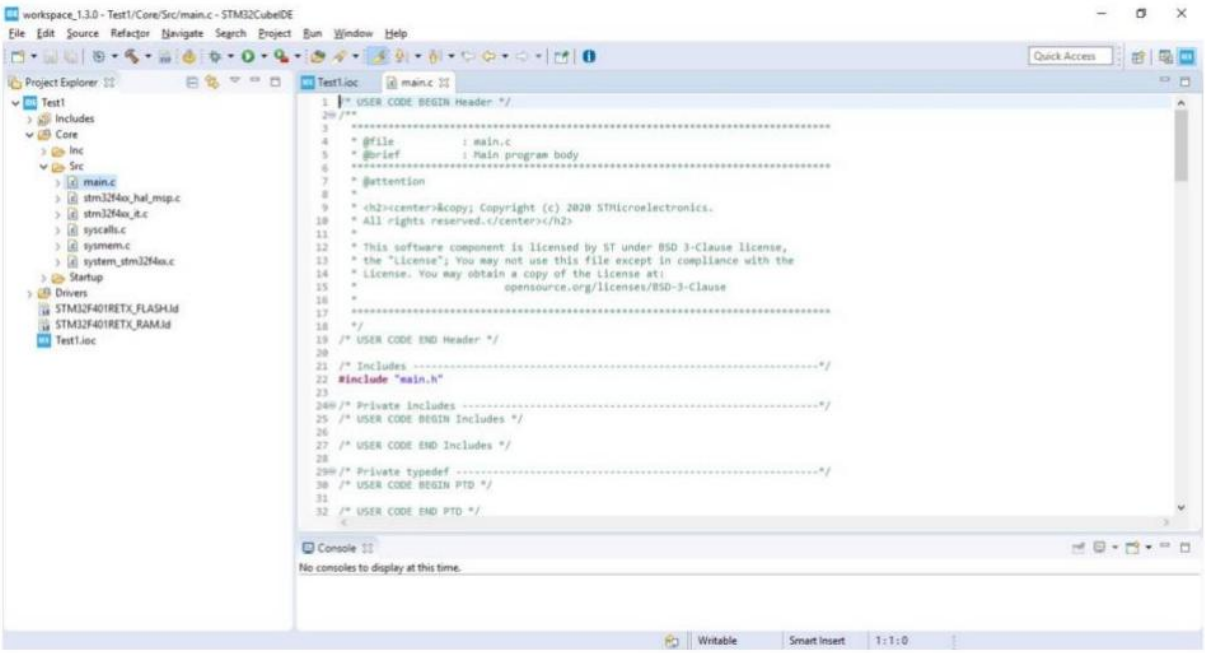


سپس بر روی Finish کلیک کنید. در ادامه محیطی مثل نرم افزار CubeMX ظاهر خواهد شد که به راحتی میتوانید تنظیمات مربوط به بخش های مختلف میکروکنترلر را روی آن انجام دهید.

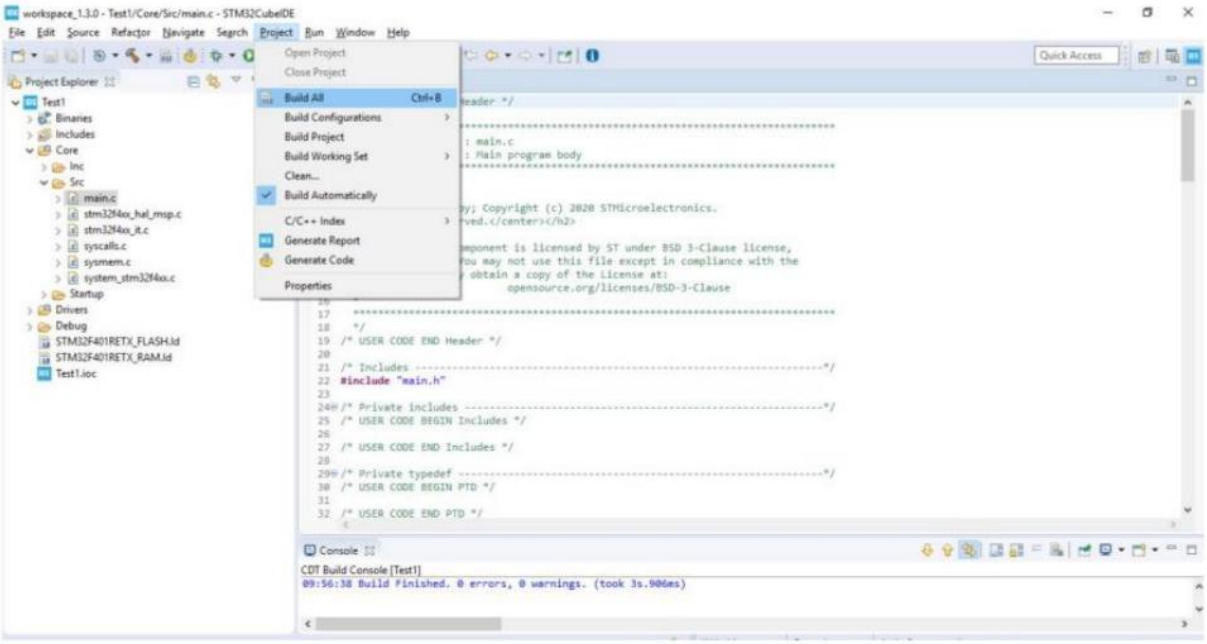
پس از انجام تغییرات مورد نظر، همانند شکل زیر بر روی Generate Code کلیک کنید تا فایل های مربوط به برنامه نویسی میکروکنترلر تولید شود.



سپس مطابق شکل زیر از سمت چپ فایل main.c را انتخاب کنید. صفحه ای مطابق شکل زیر باز خواهد شد که بدنه اصلی برنامه ست و میتوانید بر روی آن کد های خود را بنویسید.

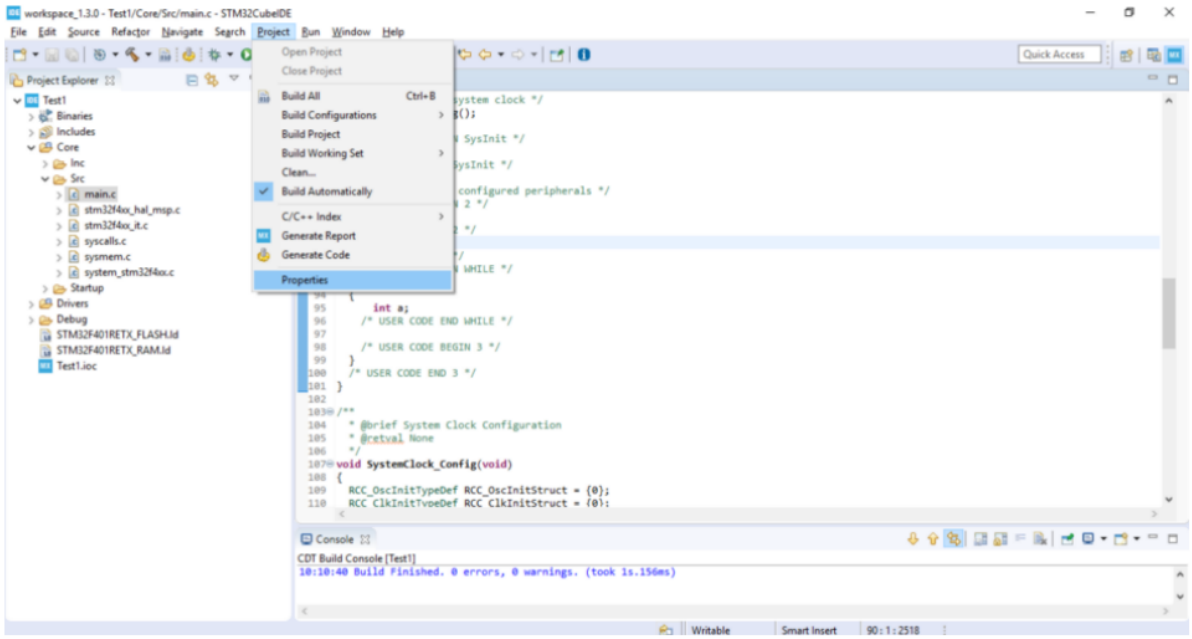


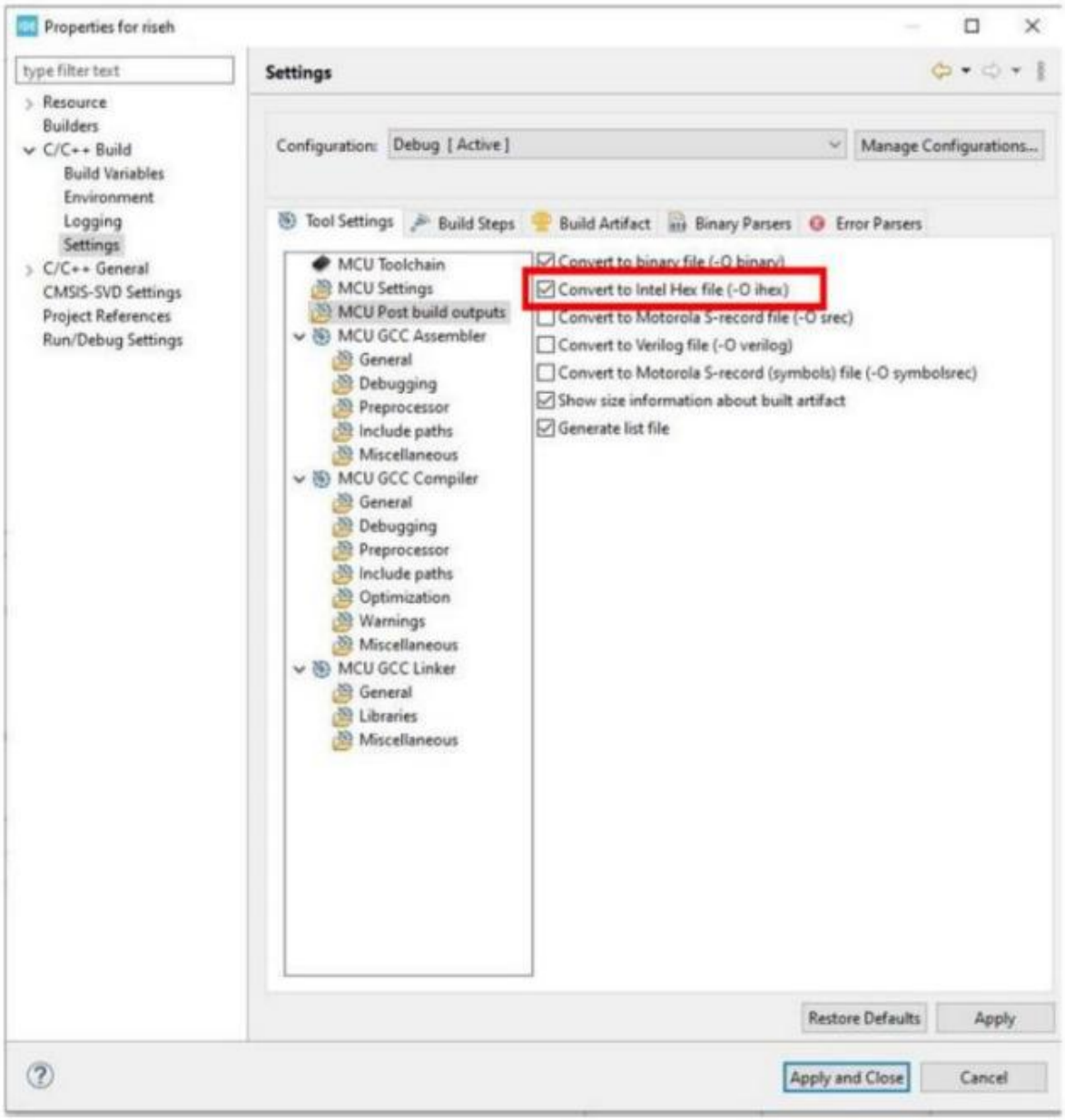
بعد از نوشتن کد های مدنظرتان، همانند شکل زیر باید کد های مد نظر را Build all کنید تا از خطا ها و ارور های احتمالی که در باکس console مشاهده میشود مطلع گردید.



❖ **ساخت فایل hex.* جهت استفاده در پروتئوس**

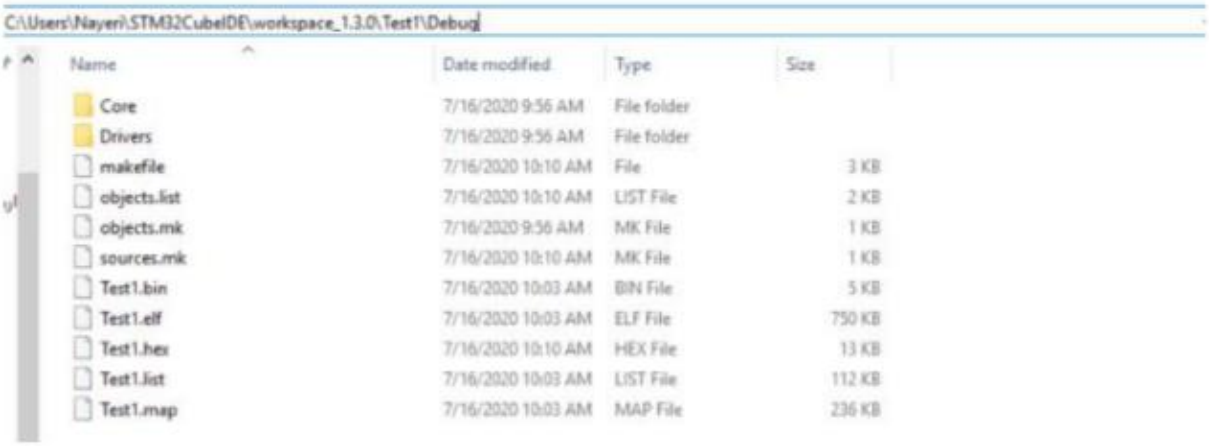
مطابق شکل های زیر عمل کنید:





پس از انتخاب گزینه داخل باکس عکس بالا، Apply and Close را بزنید و برنامه را دوباره Build All کنید.

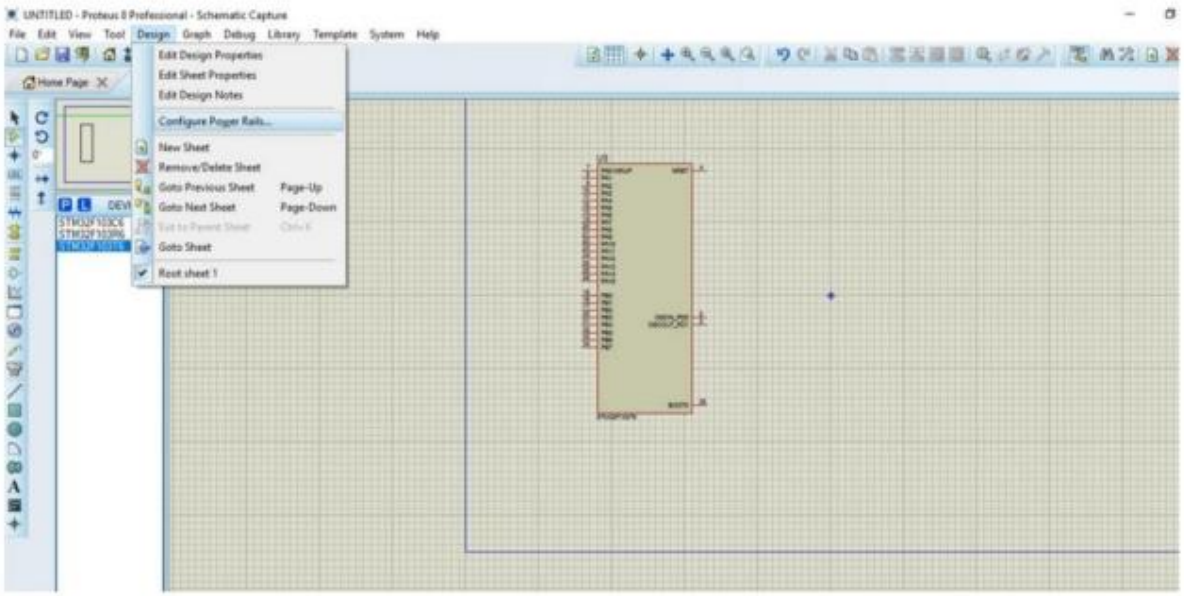
برای دسترسی به فایل * .hex کافیست که به آدرس زیر در محل پروژه خود بروید.(در پوشهDebug)

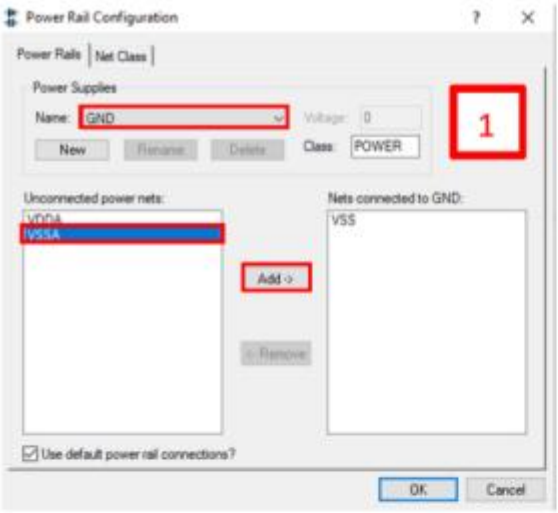


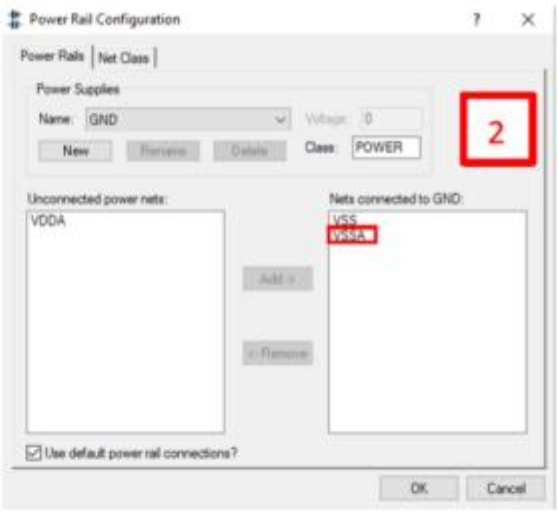
❖ **راه اندازی نرم افزار پروتئوس**

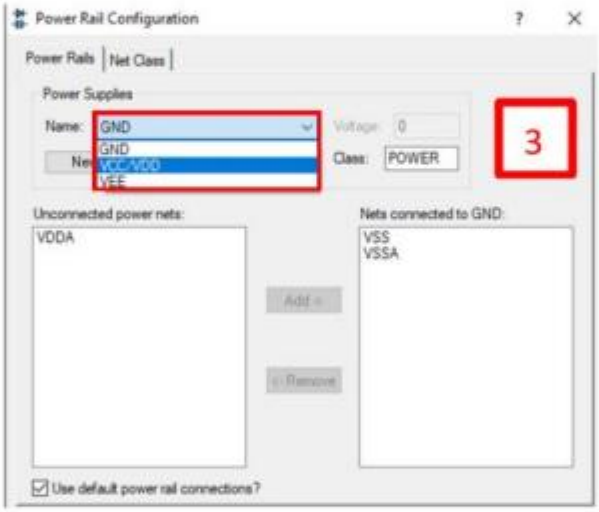
توجه: باتوجه به اینکه این نرم افزار کرک شده است و بسته به کامپیوتر شما و نوع کرک، ممکن است پس از مدتی که نرم افزار باز است، خود به خود بسته شود بنابراین پس از هر تغییر در محیط شبیه سازی بلافاصله پروژه را Save کنید تا تغییرات از دست نروند.

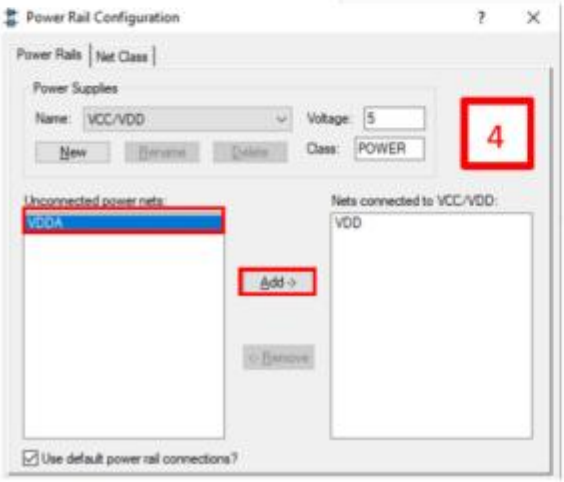
برای شبیه سازی نیاز است که پایه های VDDA و VSSA را به ترتیب به تغذیه و زمین میکروکنترلر متصل کنید. برای این منظور مطابق شکل های زیر عمل کنید.

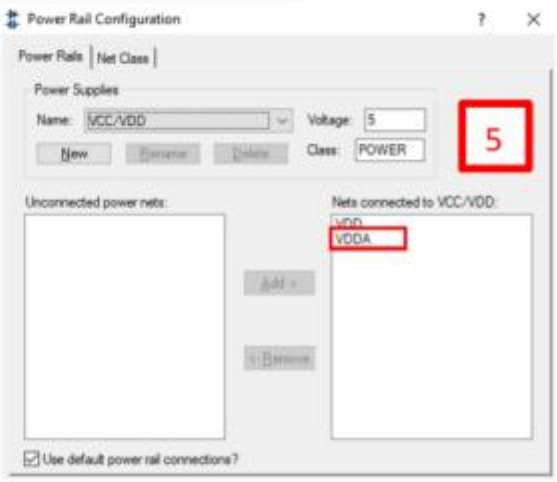



Power Rail Configuration
Power Rails
Net Class
Power Supplies
Name: GND
Voltage: 0
New
Rename
Delete
Class: POWER
1
Unconnected power nets:
VDDA
VSSA
Add ->
<- Remove
Nets connected to GND:
VSS
Use default power rail connections?
OK
Cancel


Power Rail Configuration
Power Rails
Net Class
Power Supplies
Name: GND
Voltage: 0
New
Rename
Delete
Class: POWER
2
Unconnected power nets:
VDDA
Add ->
<- Remove
Nets connected to GND:
VSS
VSSA
Use default power rail connections?
OK
Cancel


Power Rail Configuration
Power Rails
Net Class
Power Supplies
Name: GND
GND
VCC/VDD
VEE
Voltage: 0
Class: POWER
3
Unconnected power nets:
VDDA
Add ->
<- Remove
Nets connected to GND:
VSS
VSSA
Use default power rail connections?


Power Rail Configuration
Power Rails
Net Class
Power Supplies
Name: VCC/VDD
Voltage: 5
New
Rename
Delete
Class: POWER
4
Unconnected power nets:
VDDA
Add ->
<- Remove
Nets connected to VCC/VDD:
VDD
Use default power rail connections?


Power Rail Configuration
Power Rails
Net Class
Power Supplies
Name: VCC/VDD
Voltage: 5
New
Rename
Delete
Class: POWER
5
Unconnected power nets:
Add ->
<- Remove
Nets connected to VCC/VDD:
VDD
VDDA
Use default power rail connections?

ابتدا مقدار کلاک اصلی میکرو کنترلر را به صورت زیر تنظیم کنید.

$HCLK(MHz)$	باقیمانده شماره دانشجویی بر ۹
10	۰
15	۱
20	۲
26	۳
30	۴
36	۵
40	۶
44	۷
52	۸
60	۹

سوال 1)

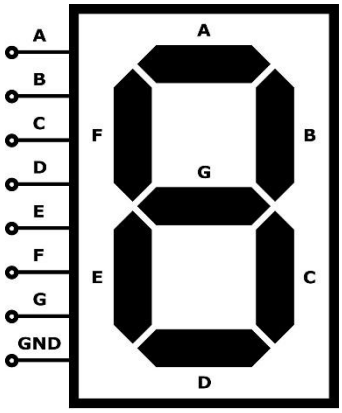
در این سؤال می‌خواهیم یک سیستم ساده برای شبیه‌سازی حرکت یک آسانسور طراحی کنیم. این سیستم دارای یک نمایشگر سون‌سگمنت 7SEG-COM-CATHODE و سه Push Button است. در حالت اولیه، عدد 0 باید روی نمایشگر نشان داده شود.

الف) چند پایه مناسب از میکروکنترلر را انتخاب کنید و پایه‌های نمایشگر سون‌سگمنت را به عنوان خروجی دیجیتال و سه Push Button را به عنوان ورودی دیجیتال راه‌اندازی نمایید، به‌طوری‌که یکی از کلیدها به صورت ورودی عادی و دو کلید دیگر به صورت وقفه خارجی (External Interrupt) تنظیم شوند. مدار مورد نظر را در نرم‌افزار Proteus پیاده‌سازی کنید.

ب) برنامه‌ای بنویسید که در حالت اولیه، عدد 0 روی نمایشگر سون‌سگمنت نمایش داده شود.

ج) برنامه را به گونه‌ای کامل کنید که با فشردن کلید اول، سیستم از حالت اولیه خارج شده و اعداد 0 تا 9 با فاصله زمانی مشخص به ترتیب روی نمایشگر ظاهر شوند و این روند به صورت پیوسته تکرار گردد.

د) یکی از کلیدهای وقفه را به گونه‌ای تنظیم کنید که با فشردن آن در هر لحظه، روند بخش قبل متوقف شده و عدد 1 روی نمایشگر ظاهر شود. همچنین کلید وقفه دیگر را به گونه‌ای تنظیم کنید که با فشردن آن در هر لحظه، روند بخش قبل متوقف شده و عدد 9 روی نمایشگر ظاهر شود. در این حالت، پس از وقوع هر یک از این دو وقفه، سیستم نباید به روند قبلی بازگردد.



7-SEGMENT DISPLAY

برنامه نوشته‌شده را در نرم‌افزار Proteus شبیه‌سازی کنید .

برنامه های میکروکنترلر(main.c, *.hex , *.ioc) را به همراه برنامه پروتئوس(.pdsprj) در پوشه Q1 ذخیره کنید.

سوال 2)

در این سوال قصد داریم سرعت و جهت چرخش یک موتور DC را با استفاده از سیگنال PWM کنترل کنیم.

الف) در مورد واحد TIM تحقیق کنید و چگونگی اضافه کردن آن به stm32 را توضیح دهید. درباره پارامتر های ARR(Auto-Reload Register) و PSC (Prescaler) و رابطه ریاضی آن ها تحقیق کنید. سپس با توجه به این پارامتر ها تایمر را جوری تنظیم کنید که یک سیگنال PWM با فرکانس $1k\ HZ$ تولید کند.

ب) از انجایی که پایه های میکروکنترلر جریان کافی برای راه اندازی مستقیم موتور را ندارند، باید از یک درایور موتور (مانند L298) استفاده کنیم و با استفاده از خروجی های این درایور موتور را به حرکت در بیاوریم. این درایور شامل دو پایه IN1 وIN2 جهت کنترل جهت هستند که با اعمال ۰ و ۱ منطقی میتوان جهت موتور را تغییر داد.

سیستم را به نحوی برنامه نویسی کنید که:

- **مقدار ولتاژ یک پتانسیومتر سرعت موتور را از ۰٪ تا ۱۰۰٪ کنترل کند.**
- **با فشردن Push Button جهت چرخش موتور برعکس شود.**

ج) در محیط Proteus، با استفاده از میکرو کنترلر، یک درایور موتور L298 و یک موتور DC اتصالات را برقرار کنید.

برای عملکرد صحیح شبیه سازی به نکات زیر توجه کنید:

- **پایه ENA در درایور: این پایه فعال ساز درایور است و میزان خروجی را تعیین میکند. برای کنترل سرعت باید سیگنال PWM مستقیم به آن وارد شود.**
- **پایهSEN_A برای اندازه گیری و حسگر جریان موتور است. دقت کنید برای روشن شدن موتور در شبیه سازی این پایه باید حتما زمین باشد.**

برنامه های میکروکنترلر(main.c, *.hex , *.ioc) را به همراه برنامه پروتئوس(.pdsprj) در پوشه Q2 ذخیره کنید.

سوال 3)

طراحی سیستم شمارش معکوس 15 ثانیه با استفاده از وقفه تایمر(interrupt تایمر) درSTM32 :

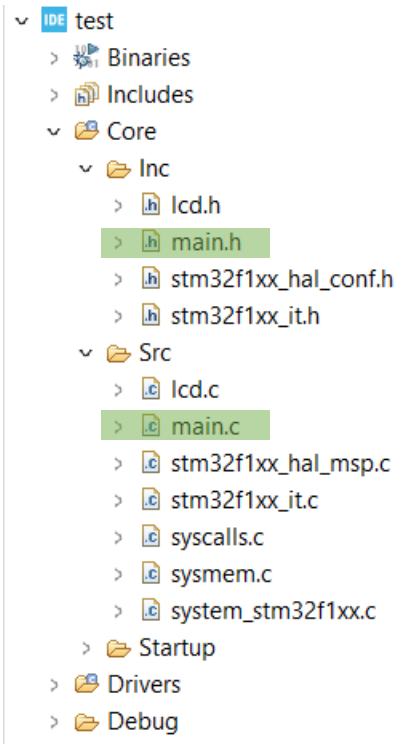
هدف این است که سیستمی بسازیم که با فعال شدن یک ورودی لاجیک (Logicstate)، یک شمارش معکوس 15 ثانیه‌ای را شروع کند. در حین این شمارش، یک LED هر ثانیه چشمک بزند و یک بازر(Buzzer) نیز هر ثانیه بوق بزند. در پایان شمارش که به صفر رسید، LED به طور ممتد روشن شود و بازر نیز به طور ممتد بوق بزند. همچنین در هر لحظه‌ای که ورودی لاجیک برابر صفر شود، سیستم باید ریست شده، LED و بازر خاموش شوند و شمارش متوقف

گردد. در نهایت نیز مقدار شمارش باید روی LCD نمایش داده شود. نکته مهم این است که برای زمان‌بندی دقیق، حتماً باید از وقفه تایمر استفاده شود.

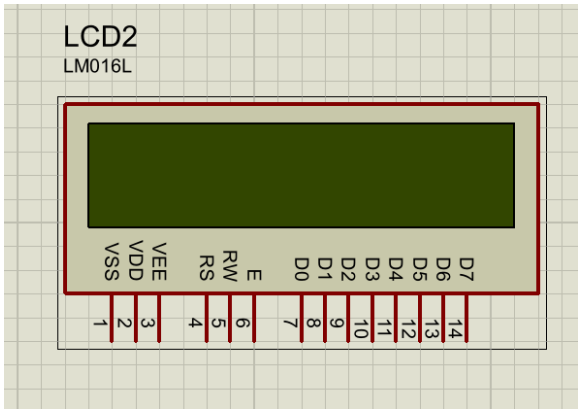
برنامه های میکروکنترلر(main.c, *.hex , *.ioc) را به همراه برنامه پروتئوس(.pdsprj) در پوشه Q3 ذخیره کنید.

آموزش استفاده از LCD :

برای استفاده از LCD باید ابتدا کتابخانه LCD را به پوشه ای که فایل های h. و c. پروژه در آن قرار دارند اضافه کرد. به همین دلیل یک کتابخانه با نام lcd در اختیار قرار داده شده است که فایل lcd.h باید در پوشه inc که در پوشه core قرار دارد اضافه شود. فایل lcd.c باید در پوشه src که در پوشه core قرار دارد اضافه شود.



LCD ای که ما از آن استفاده می کنیم به صورت زیر است:



این LCD ۸ پایه برای اتصال به میکرو دارد که داده ها را به صورت سریال از این ۸ پایه از میکرو دریافت می کند یا ارسال می کند.

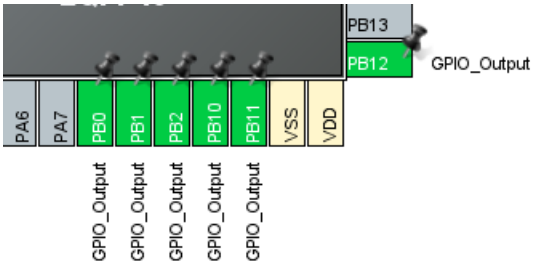
همچنین می توان ۴ پایه D4 تا D7 را به میکرو متصل کرد و داده ها را ارسال کرد.

پایه های E و RS نیز به میکرو متصل می شوند و پایه RW را چون قرار است داده ها را به LCD ارسال کنیم زمین می کنیم.

حال در برنامه stm32cubeIDE برای راه اندازی LCD به صورت زیر عمل می کنیم:

ابتدا در قسمت pinout تراشه ۴ خروجی برای ارسال داده به LCD انتخاب می کنیم.(برای استفاده از سیم کمتر از روش دوم که استفاده از ۴ پایه D4 تا D7 است استفاده می کنیم.)

حال در قسمت کد برنامه بعد از include کردن کتابخانه LCD باید پورت ها و پین هایی که در قسمت pinout مشخص کردیم را برای lcd مشخص کنیم:



```
/* USER CODE BEGIN PV */
Lcd_PortType port[] = {GPIOB, GPIOB, GPIOB, GPIOB};
Lcd_PinType pin[] = {GPIO_PIN_0, GPIO_PIN_1, GPIO_PIN_2, GPIO_PIN_10};
Lcd_PortType rs_port = GPIOB;
Lcd_PinType rs_pin = GPIO_PIN_11;
Lcd_PortType e_port = GPIOB;
Lcd_PinType e_pin = GPIO_PIN_12;

Lcd_HandleTypeDef lcd;

```

هم چنین یک نمونه (instance) از این کتابخانه lcd باید گرفته شود که در آخرین خط عکس بالا مشاهده می شود.

حال با استفاده از دستور زیر نمونه را تعریف کرده و همچنین قبل از شروع حلقه while حتما باید lcd را initialize کنیم.

```
/* USER CODE BEGIN 2 */
lcd = Lcd_create(port, pin, rs_port, rs_pin, e_port, e_pin, LCD_4_BIT_MODE);  // Configure LCD
Lcd_init(&lcd);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
```

حال می توان از lcd استفاده کرد.

به عنوان مثال اگر بخواهیم کلمه Hello world را در خط اول و عدد ۵ را در خط دوم lcd چاپ کنیم به صورت زیر عمل می کنیم :

```
void Hello_World()
{
    Lcd_cursor(&lcd, 0, 0);
    Lcd_string(&lcd, "Hello");

    Lcd_cursor(&lcd, 1, 0);
    char number_string[20];
    sprintf(number_string, "%d", 5);
    Lcd_string(&lcd, number_string);

}
```

حال این تابع را در while فراخوانی می کنیم.

سوال 4)

در بسیاری از سیستم های اندازه گیری صنعتی مانند دیتالاگرها، مولتی مترهای قابل برنامه ریزی و ایستگاه های آزمایشگاهی، یک میکروکنترلر میتواند با تشخیص ولتاژ تولید شده توسط سنسورهای مختلف، به طور خودکار نوع سنسور متصل شده را شناسایی کرده و مقدار اندازه گیری شده را نمایش دهد. در این سوال شما باید سیستمی را طراحی کنید که با خواندن ولتاژ آنالوگ از یک پتانسیومتر که به عنوان شبیه ساز سنسور استفاده میشود، تشخیص دهد که چه نوع سنسوری متصل است و سپس مقدار متناسب با آن سنسور را روی یک LCD نمایش دهد. (ولتاژ مرجه را نیز ۳.۳ ولت فرض کنید)

محدوده ولتاژها و سنسورهای متناظر به صورت زیر می باشند:

محدوده ولتاژ(ولت)	نوع سنسور تشخیص داده شده	مقدار نمایش داده شده
۰.۰۰–۰.۹۹	سنسور دما (LM35)	دما (درجه سانتی گراد) = ۱۰۰ * ولتاژ
۱.۰۰–۱.۹۹	سنسور رطوبت (HIH–4000)	رطوبت (درصد) = ۳۳.۳ * ولتاژ
۲.۰۰–۲.۹۹	سنسور فشار (MPX4115)	فشار (هکتوپاسکال) = ۱۰۰ * ولتاژ
۳.۰۰–۳.۳۰	سنسور گاز (MQ–135)	کیفیت هوا (درصد) = ۱۰۰ * (۳.۳ / ولتاژ)

به عنوان مثال اگر ولتاژ ۰.۵ ولت باشد، یعنی سنسور دما متصل است و دمای ۵۰ درجه سانتی گراد را اندازگیری میکند.

الف) در مورد واحد ADC تحقیق کنید و چگونگی اضافه کردن آن به stm32 را بررسی کنید. رزولوشن ADC را چند بیت انتخاب میکنید و چرا؟ و درنهایت یک پایه برای ADC جهت گرفتن ورودی آنالوگ تعیین کنید.

ب) برنامه ای بنویسید که به صورت پیوسته از ADC نمونه برداری کند و ولتاژ لحظه ای را با دو رقم اعشار محاسبه کند. (ADC ۱۲ بیت است).

ج) اکنون براساس ولتاژ خوانده شده، نوع سنسور را مطابق جدول بالا تشخیص دهید و سپس مقدار فیزیکی متناسب با آن سنسور را محاسبه کنید.

د) اطلاعات را در قالب زیر روی یک LCD Alphanumeric 2 × 16 نمایش دهید:

در خط اول باید نوع اندازه گیری مشخص شود (دما، رطوبت، فشار و یا کیفیت هوا) و در خط دوم نیز مقدار فیزیکی به همراه واحد اندازه گیری آن نمایش داده شود.

برنامه های میکروکنترلر(main.c, *.hex , *.ioc) را به همراه برنامه پروتئوس(.pdsprj) در پوشه Q4 ذخیره کنید.

لطفا در ارسال تمرینات به موارد زیر توجه بفرمایید ، در صورت عدم رعایت هر یک از موارد زیر تمرین شما تصحیح نخواهد شد :

- **تشابه در حل سوالات به صورت جدی بررسی خواهد شد. در صورت تشخیص تمرین مشابه نمره تقسیم خواهد شد.**
- **در صورت دست نویس بودن تمرین ، نوشته ها خوانا باشند و کیفیت اسکن آن ها مناسب باشد.**
- **پاسخ ها در قالب یک فایل *pdf* تجمیع و ارسال شوند. همچنین تمامی کد ها با ذکر اینکه مربوط به کدام سوال هستند در پوشه ای با نام *Codes* ذخیره شده سپس تمامی فایل ها در قالب یک فایل *zip* تجمیع و با نام *student_number.zip* ارسال شوند .**
- **در صورت هر گونه ابهام در سوالات برای هر یک از سوالات به دستیار مربوطه تنها از طریق ایمیل دانشگاهی با موضوع *Series#-Q#* (که در آن # شماره سری تمرین و سوال مورد نظر است) ایمیل زده و ایمیل دستیار ارشد را نیز *CC* کنید.**

  سوال1) آقای روزبهانی ایمیل *samyar.rouzbahani@ut.ac.ir*

  سوال2) خانم شمائی ایمیل *negar.shamaie@ut.ac.ir*

  سوال3) آقای محمودی ایمیل *momahdimahmoodi@ut.ac.ir*

  سوال4) آقای زارع ایمیل *ali.zare.bilondy@ut.ac.ir*

  ایمیل دستیار ارشد *bayan.mahdiyar@ut.ac.ir*
- **همچنین تمامی دستیاران داخل گروه بله عضو هستند و ترجیحا سوالات در آنجا مطرح شود تا بقیه دوستان هم استفاده کنند.**