به نام خداوند جان و خرد

درس ابزار دقیق

گروه کنترل

نیمسال اول ۱۴۰۱-۱۴۰۰ | مینی پروژه سوم | مدرس: محمدرضا نیری

در این مینی پروژه مراحل لازم جهت شروع کار و برنامه نویسی با میکرو کنترلر ARM را دنبال خواهید کرد. برای این منظور از نرم افزار STM32CubeIDE و پروتئوس استفاده خواهید کرد. این پروژه نیاز به گزارش نخواهد داشت و تنها ارسال فایل های خواسته شده در هر بخش کفایت می کند. در صورت لزوم پروژه انجام شده با استفاده از اسکایپ ارائه خواهد شد. با مشاهده ویدئوهای مربوط به درس و توضیحات ارائه شده در پروژه نیازی به منابع دیگر نخواهید داشت اما در صورت لزوم می توانید به ویدئوهای کانال زیر نیز مراجعه فرمایید:

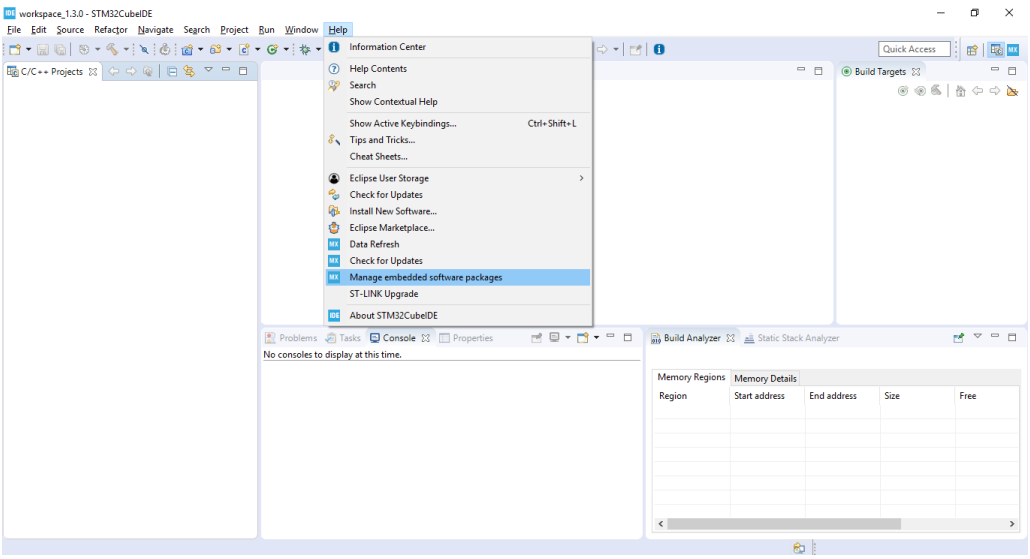
https://www.youtube.com/channel/UC-CuJ6qKst9-8Z-EXjoYK3Q/videos

➢ نصب و آماده سازی نرم افزارها

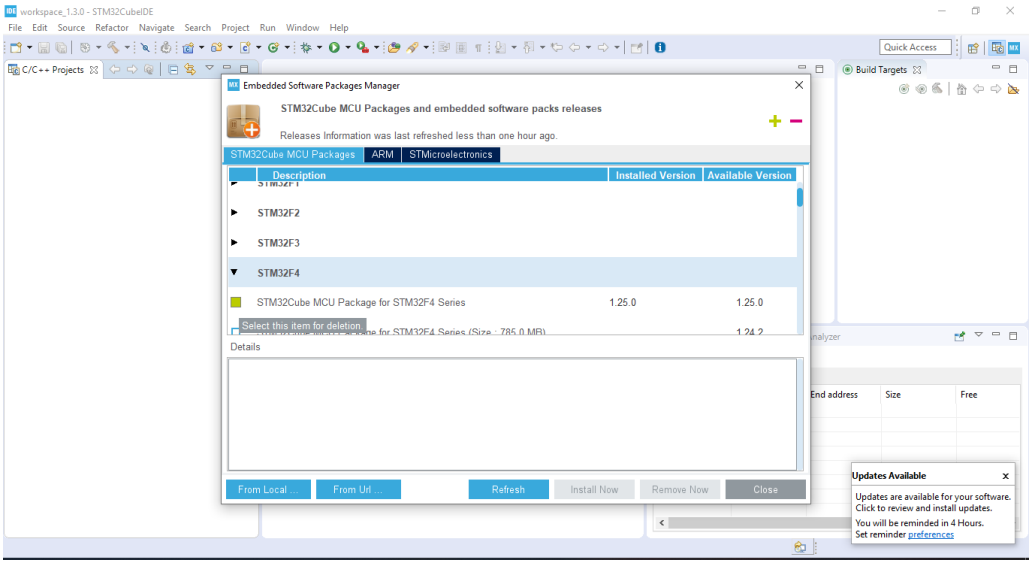
مراحل زیر را پی بگیرید

۱- نرم افزار STM32CubeIDE را نصب کنید (ویدئو مربوط به چگونگی نصب در سامانه درس بارگذاری شده است)

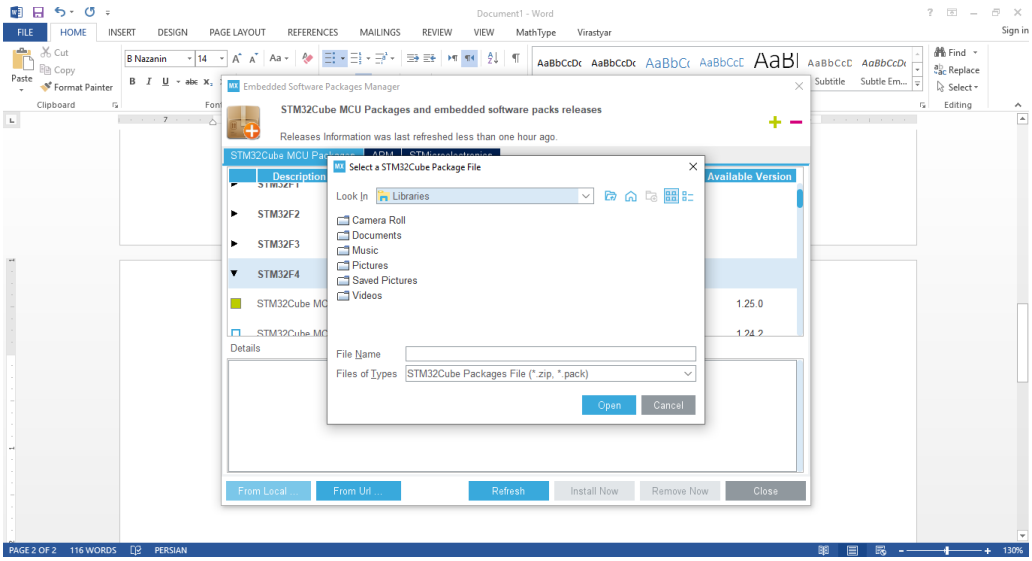
۲- پس از نصب نرم افزار را باز کنید و مطابق شکل زیر پکیج STM32F1 را به یکی از دو روش زیر نصب کنید



- دانلود و نصب به صورت آنلاین: که باید به صورت شکل زیر ابتدا تیک مربوط به پکیج مورد نظر را فعال کرده و سپس بر روی گزینه Install Now کلیک کنید

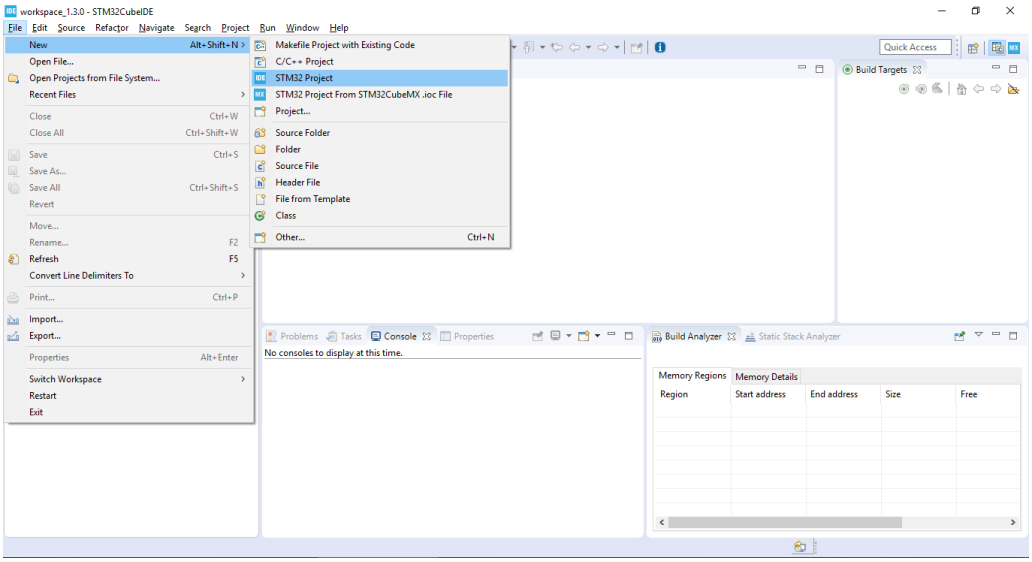


- نصب به صورت آفلاین: در صورتی که پکیج مورد نظر رو قبلا دانلود کرده اید کافیست از From Local فایل مورد نظر را بر روی هارد خود پیدا کرده و انتخاب کنید.



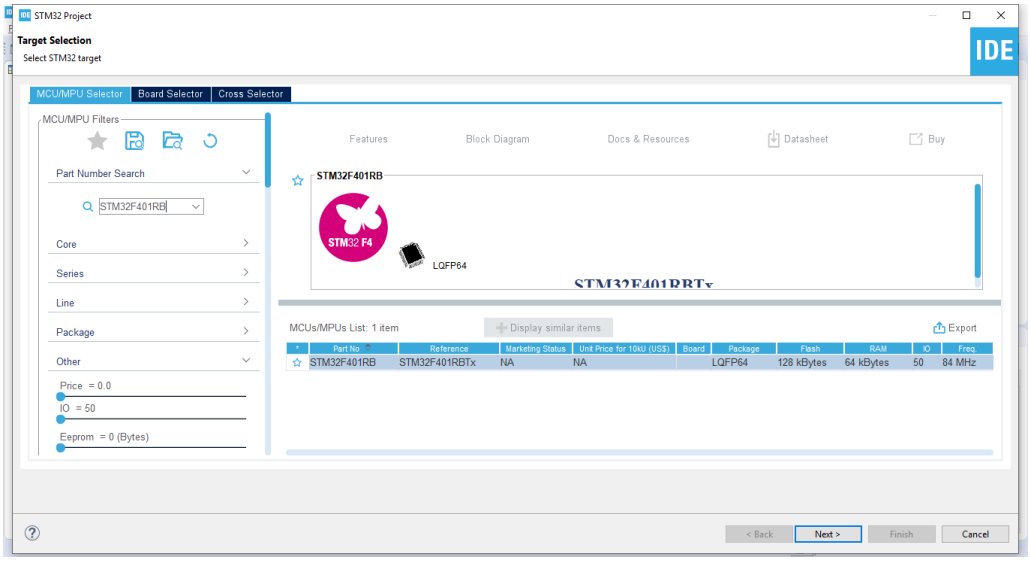
پس از نصب درست پکیج مربع کنار پکیج مورد نظر سبز رنگ خواهد شد.

۳- از مسیر شکل زیر یک پروژه جدید بسازید

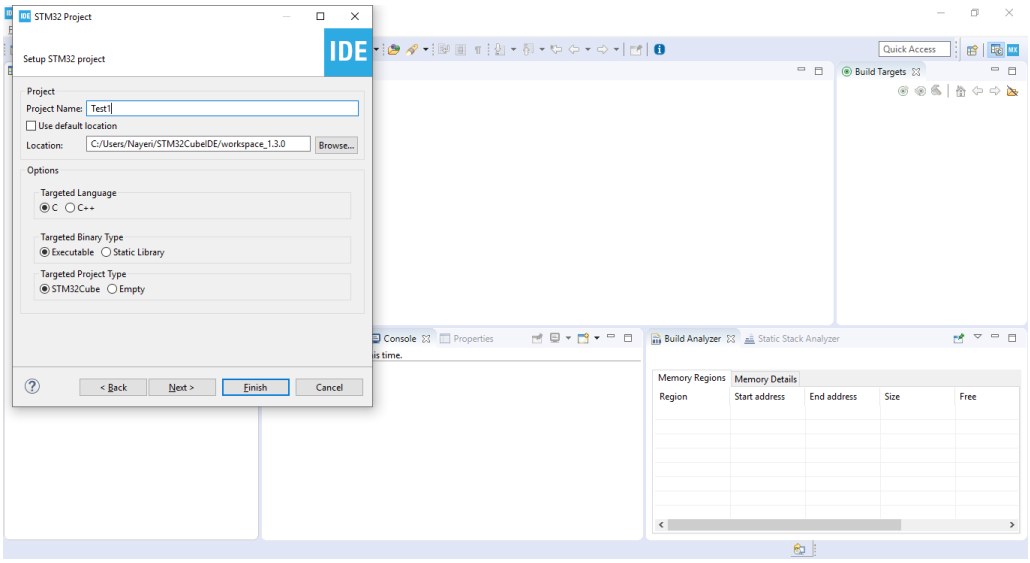


سپس صفحه ای به صورت شکل زیر باز خواهد شد. در قسمت Part Number Search میکرو کنترلر مطابق جدول زیر را تایپ کرده در پنجره وسط آن را انتخاب کنید و بر روی Next کلیک کنید

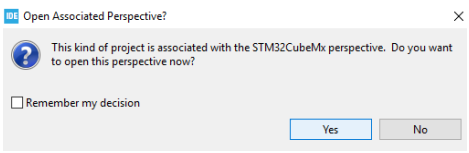
باقیمانده شماره دانشجویی بر ۳	میکروکنترلر
۰	STM32F103C6
۱	STM32F103R6
۲	STM32F103T6



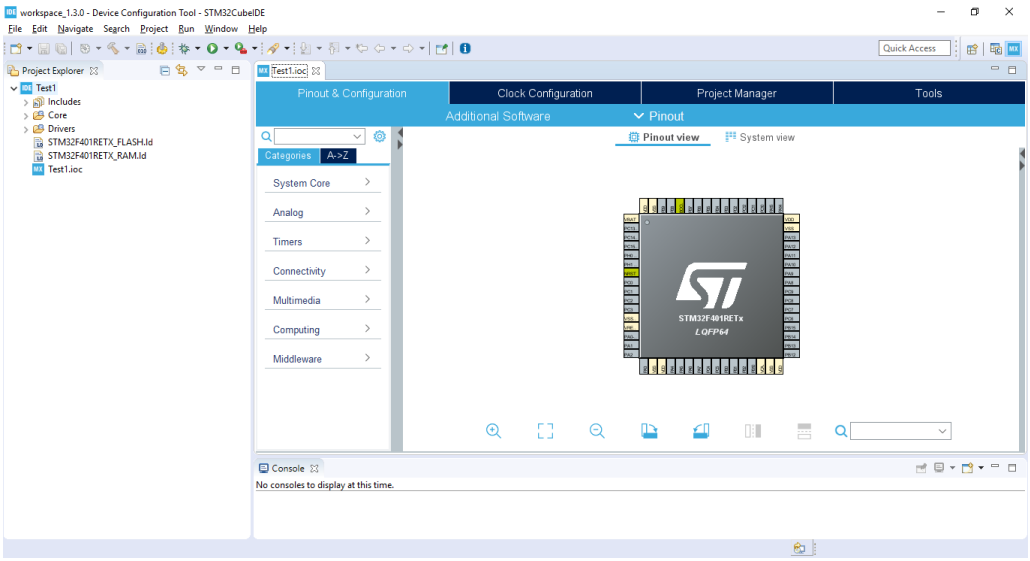
سپس پنجره ای به صورت شکل زیر باز خواهد شد که در آن میبایست یک نام برای پروژه انتخاب کرده و سپس مسیری را که پروژه در آن ذخیره می شود انتخاب کنید:



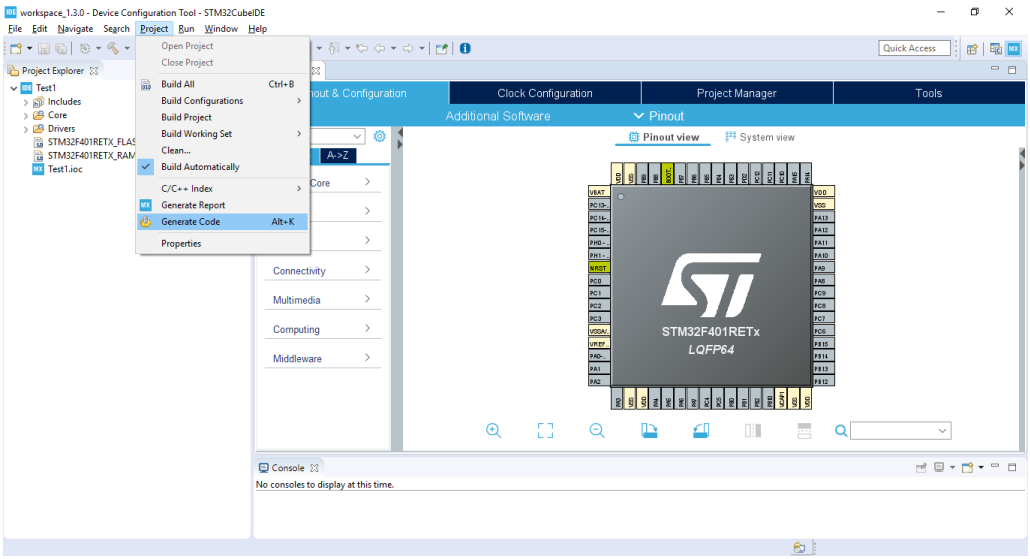
سپس بر روی گزینه Finish کلیک کنید. در صورتی که پیامی به صورت شکل زیر ظاهر شد، بر روی گزینه Yes کلیک کنید



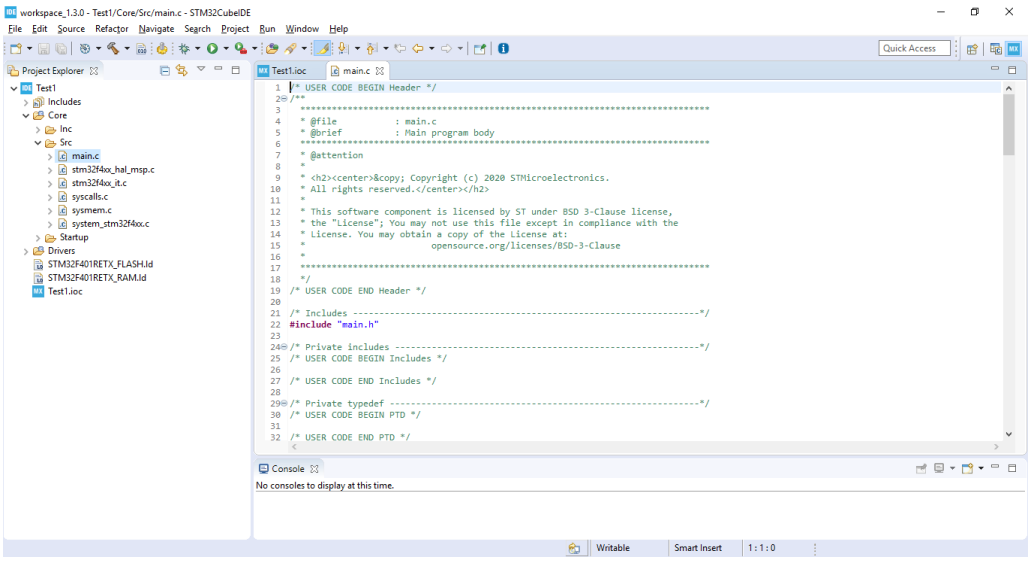
صفحه ای مطابق با شکل زیر ظاهر خواهد شد که شبیه به نرم افزار CubeMX است و به راحتی می توانید تنظیمات مربوط به بخش های مختلف میکروکنترلر را روی آن انجام دهید.



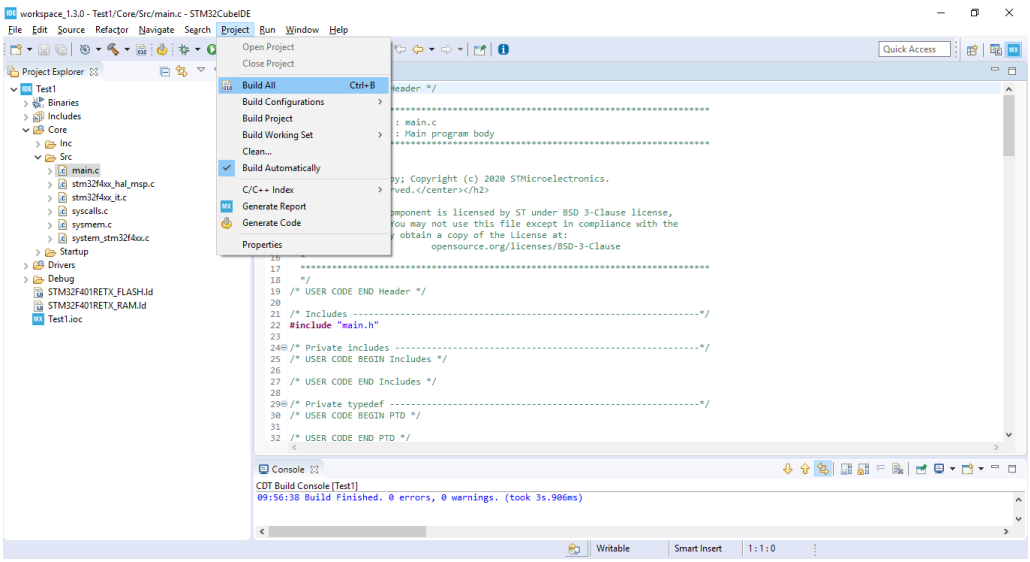
پس از انجام تنظیمات مورد نظر مطابق شکل زیر روی Generate Code کلیک کنید:



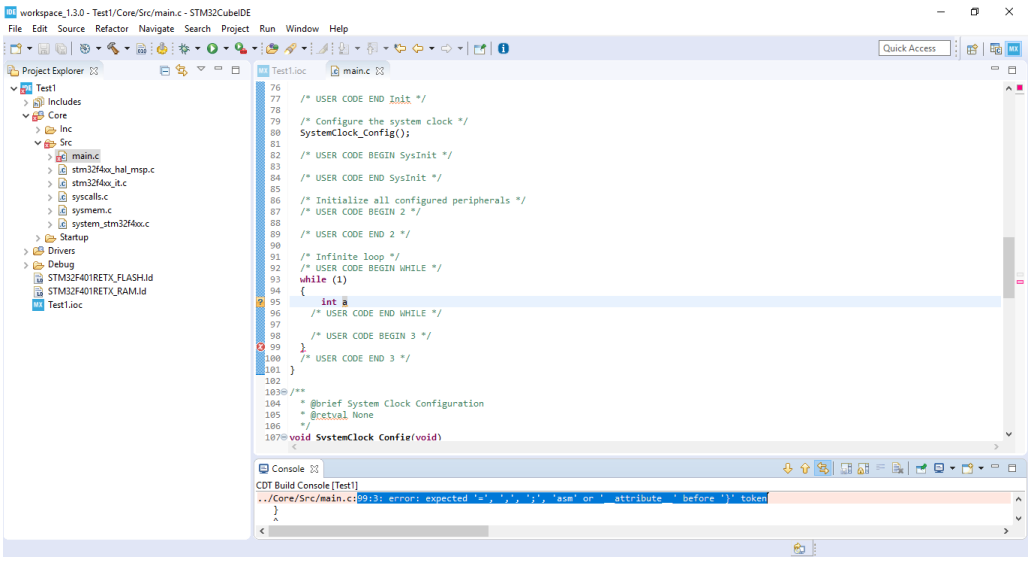
سپس مطابق شکل زیر از ستون سمت چپ فایل main.c را انتخاب کنید. صفحه ای مطابق شکل زیر باز خواهد شد که در واقع بدنه اصلی برنامه است و مانند نرم افزار Keil می توانید در آن کدنویسی کنید.



پس از اینکه کدهای مورد نظر را نوشتید می بایست کد مورد نظر را Build کنید تا از خطاها و هشدارهای احتمالی مطلع شوید. برای این منظور مطابق شکل زیر Build All را انتخاب کنید. همانطور که مشاهده می شود در Console پایین نرم افزار تعداد خطاها و هشدارها مشخص است.

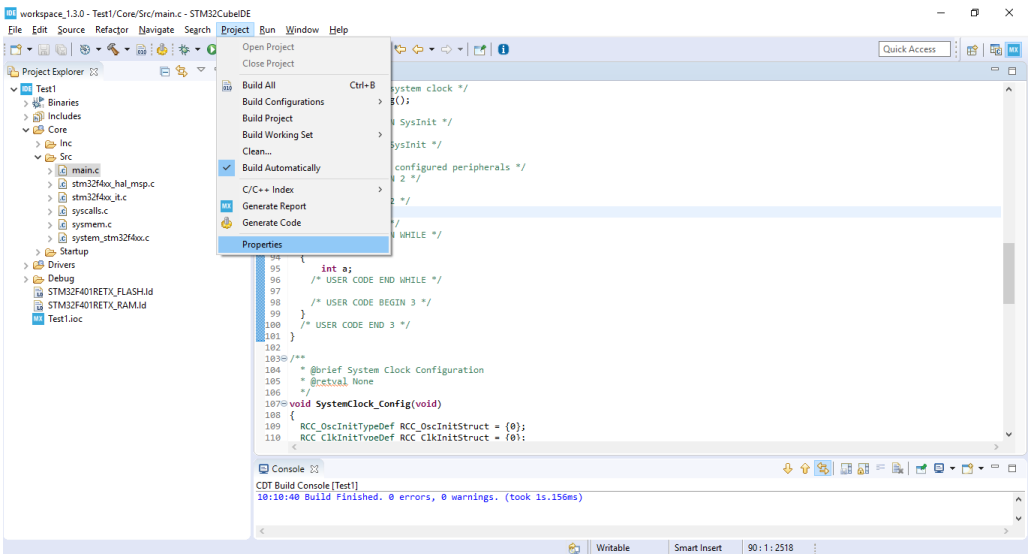


در صورتی که خطا یا هشداری در برنامه نویسی وجود داشته باشد، مطابق شکل زیر شماره خط و علت آن در Console قابل مشاهده است.

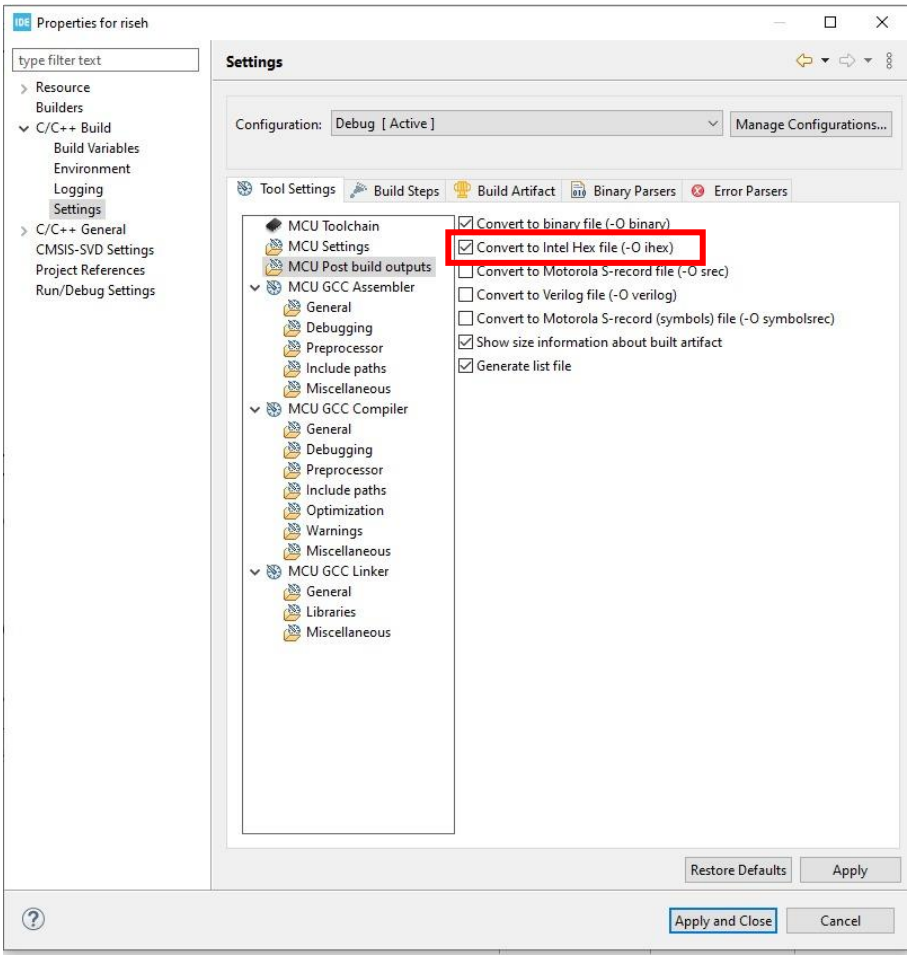


۴- ساخت فایل hex.* جهت استفاده در نرم افزار Proteus

مطابق شکل های زیر عمل کنید

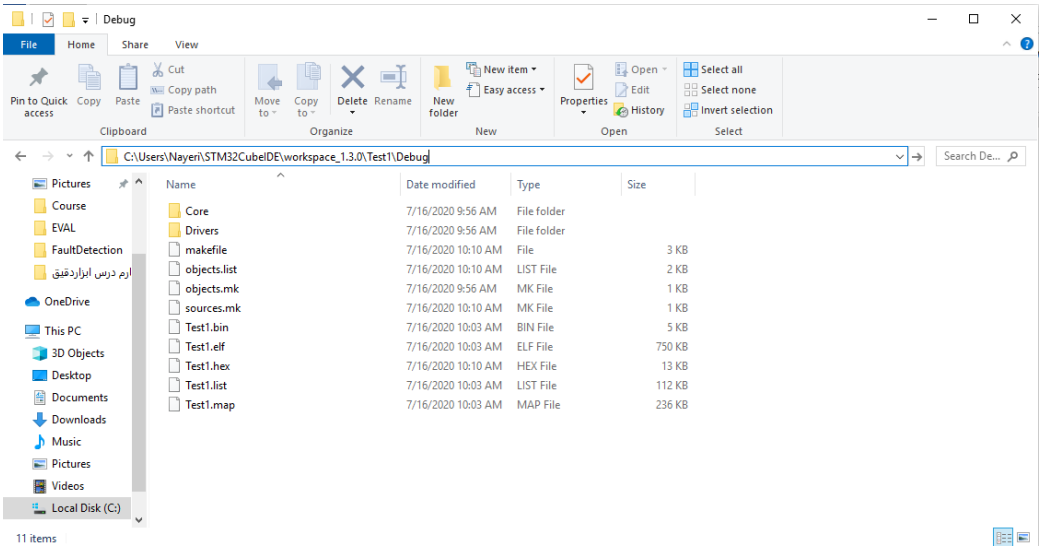


در بخش MCU Post-Build outputs تیک مربوط به Convert to intel Hex file و بر روی Apply and Close کلیک کنید.



سپس پروژه را مجددا Build All کنید.

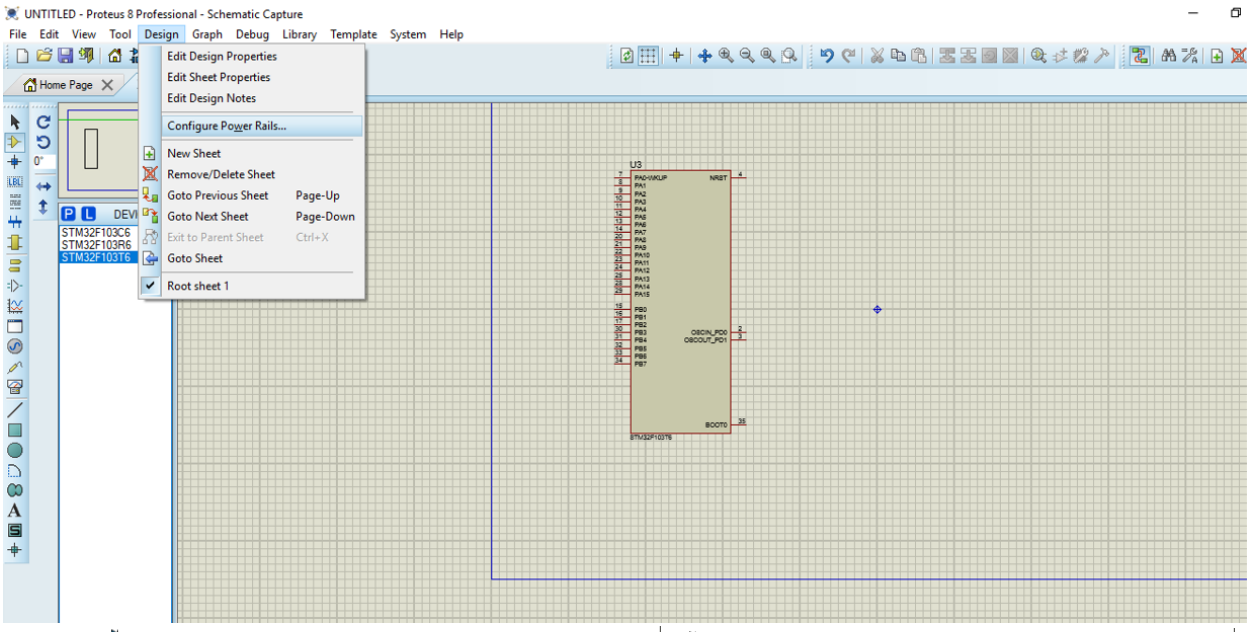
برای دسترسی به فایل hex.* به آدرس ساخت پروژه مطابق شکل زیر بروید:

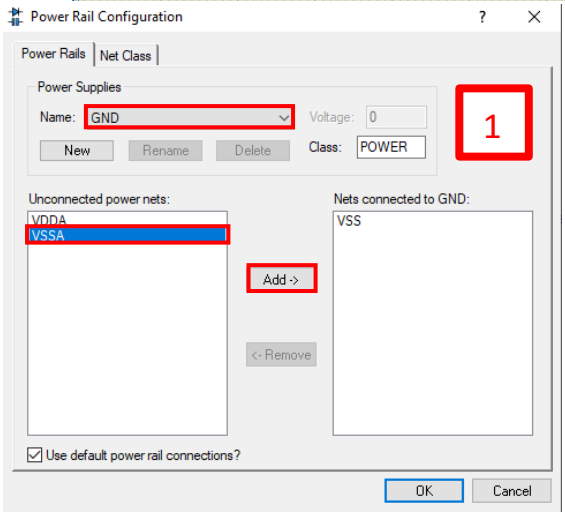


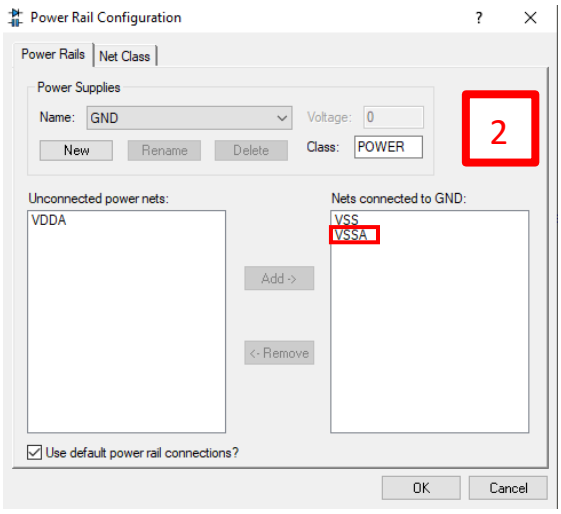
۵- نرم افزار Proteus را نصب کنید. (ورژن 8.9 به بالا را نصب کنید)

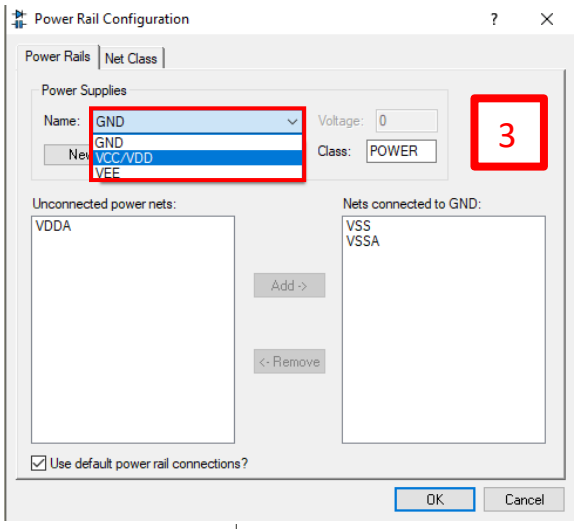
توجه: باتوجه به اینکه این نرم افزار کرک شده است و بسته به کامپیوتر شما و نوع کرک، ممکن است پس از مدتی که نرم افزار باز است، خود به خود بسته شود بنابراین پس از هر تغییر در محیط شبیه سازی بلافاصله پروژه را Save کنید تا تغییرات از دست نروند.

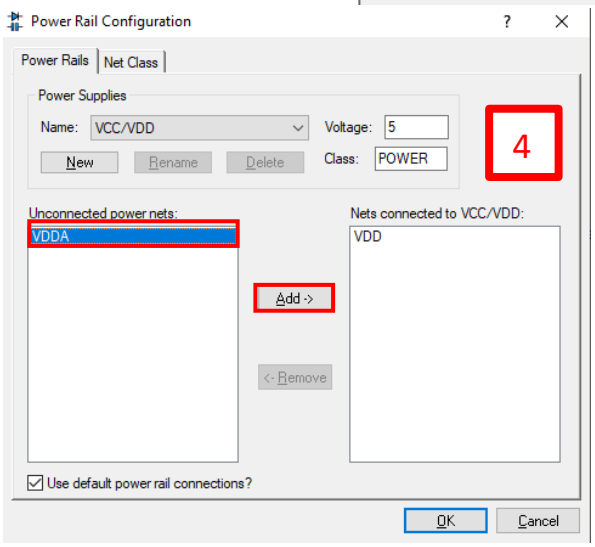
۶- برای شبیه سازی نیاز است که پایه های VDDA و VSSA را به ترتیب به تغذیه و زمین میکروکنترلر متصل کنید. برای این منظور مطابق شکل های زیر عمل کنید:

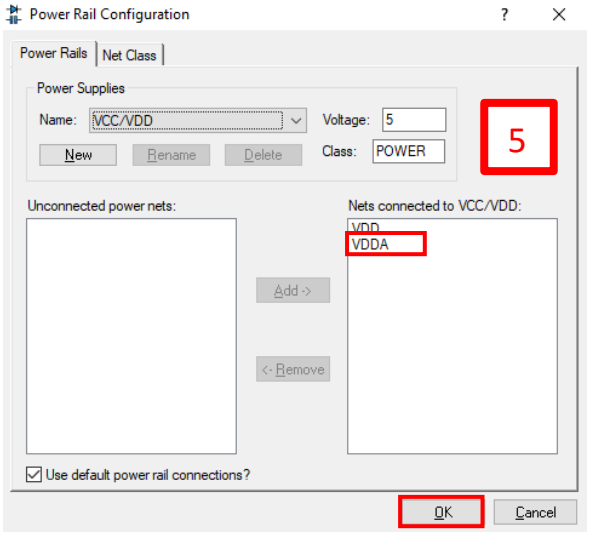












۷- مقدار کلاک اصلی میکرو کنترلر را به صورت زیر تنظیم کنید

$HCLK(MHz)$	**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

➢ روشن و خاموش کردن پایه دیجیتال

- بر روی PORTA پایه های X1 و X2 را به عنوان خروجی دیجیتال به صورت زیر تعریف کنید.

باقیمانده شماره دانشجویی بر ۵ = X1

باقیمانده شماره دانشجویی بر ۷+ ۵= X2

- یک LED سبز و یک LED قرمز به پایه های فوق متصل کنید.

توجه: در اتصال LED به صورت سخت افزاری، برای روشن کردن LED نیاز است که حداکثر جریان قابل تحمل آن را بدانیم که این مقدار با مراجعه به دیتاشیت آن مشخص می شود. در صورتی که LED مستقیم به پایه میکروکنترلر متصل شود حداکثر جریان خروجی به آن متصل می گردد و احتمال سوختن LED و یا پایه خروجی دیجیتال میکرو کنترلر زیاد است. به منظور محدود کردن جریان یک مقاومت با توجه به حداکثر جریان قابل تحمل با LED سری می شود. برای مثال اگر حداکثر جریان قابل تحمل LED ۵ میلی آمپر باشد و بخواهیم آن را به میکروکنترلر ARM (که ولتاژ خروجی دیجیتال آن در حالت معمول ۳/۳ ولت است) متصل کنیم، از یک مقاومت ۱ کیلو اهم جهت محدود کردن جریان (به مقدار حداکثر ۳/۳ میلی آمپر) استفاده می شود. (این موضوع با فرض مقاومت ناچیز LED در نظر گرفته شده است).

- می خواهیم برنامه ای بنویسیم که به صورت پیوسته عملیات زیر انجام شود:
  I. LED سبز روشن شود.
  II. t میلی ثانیه بعد LED قرمز روشن شود.
  III. t میلی ثانیه بعد LED سبز خاموش شود.
  IV. t میلی ثانیه بعد LED قرمز خاموش شود.
  V. ۱ ثانیه بعد مجددا عملیات فوق اجرا شوند.

که در آن مقدار t به صورت زیر است:

$$t = 50 \times \left(1 + 5 \text{ باقیمانده تقسیم شماره دانشجویی بر}\right)$$

- با استفاده از دستور HAL_GPIO_WritePin و دستور HAL_Delay برنامه ای بنویسید که بتواند عملیات بخش قبل را محقق کند.
- برنامه بخش قبل را با استفاده از پروتئوس شبیه سازی کنید.
- برنامه میکروکنترلر (فایل های main.c ، *.hex و *.ioc) و برنامه پروتئوس(فایل *.pdsprj) را در یک فولدر با نام Q1 ذخیره کنید.

➢ روشن و خاموش کردن پایه دیجیتال با استفاده از ورودی های دیجیتال

- فرض کنید LED های سبز و قرمز متصل به پایه های تعریف شده در بخش قبل در این بخش وجود دارند.
- بر روی PORTB پایه های Y1 و Y2 را به عنوان ورودی دیجیتال به صورت زیر تعریف کنید.

باقیمانده شماره دانشجویی بر ۳ = Y1

باقیمانده شماره دانشجویی بر ۲+ ۴= Y2

- به پایه های B.Y1 و B.Y2 بخش قبل دو Push button که خروجی آن ها Pull up است را متصل کنید.
- برنامه ای بنویسید که با فشردن Push button مربوط به پایه B.Y1 ، LED سبز روشن و LED قرمز خاموش و با فشردن Push button مربوط به پایه B.Y2 ، LED سبز خاموش و LED قرمز روشن شود. برای این منظور به دستور HAL_GPIO_ReadPin نیاز خواهید داشت.
- برنامه نوشته شده را با استفاده از پروتئوس شبیه سازی کنید.
- برنامه های میکروکنترلر (فایل های main.c ، *.hex و *.ioc) و برنامه پروتئوس(فایل *.pdsprj) را در یک فولدر با نام Q2 ذخیره کنید.

➢ وقفه خارجی

- پایه B.Y1 را به عنوان ورودی دیجیتا تعریف کنید.
- پایه های B.Y2 بخش قبل را به عنوان وقفه خارجی حساس به لبه بالارونده تعریف کنید.
- به پایه های بخش قبل دو Push button که خروجی آنها Pull up است را متصل کنید.
- برنامه ای بنویسید که سناریوی زیر پیاده سازی شود

I. در حالت عادی سناریو بخش "روشن خاموش کردن پایه دیجیتال" اجرا شود.

II. در صورت فعال شدن وقفه B.Y2 ، LED سبز خاموش و LED قرمز هر ۱۰۰ میلی ثانیه روشن و خاموش شود. برای این منظور از دستور HAL_GPIO_TogglePin استفاده کنید.

III. با فشردن کلید متصل به B.Y2 برنامه به حالت عادی برگردد.

توجه: پس از فعال کردن وقفه در Cubemx ، از مسیر

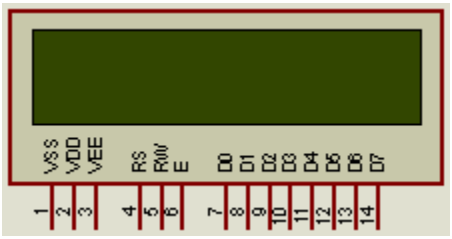
Core->Src->stm32f1xx_it.c

به روتین وقفه دسترسی خواهید داشت و می توانید کد های مورد نظر درهنگام فعال شدن وقفه را در این روتین بنویسید.

- برنامه نوشته شده را با استفاده از پروتئوس شبیه سازی کنید.
- برنامه های میکروکنترلر (فایل های main.c ، hex.* و ioc.*) و برنامه پروتئوس(فایل pdsprj.*) را در یک فولدر با نام Q3 ذخیره کنید.

➢ اتصال نمایشگر به میکروکنترلر

هدف از اجرای این بخش اتصال 16x2 Alphanumeric LCD به صورت شکل زیر به پورت B میکروکنترلر است. برای این نمایشگر کتابخانه های متنوعی تولید شده است.



- با جست و جو در اینترنت برنامه ای بنویسید که بتوان با استفاده از آن در خط اول نام و در خط دوم شماره دانشجویی شما را نمایش دهد.
- برنامه نوشته شده را با استفاده از پروتئوس شبیه سازی کنید. (توجه: برای راه اندازی LCD در پروتئوس پایه مربوط به RW را به زمین متصل کنید)
- برنامه های میکروکنترلر (فایل های main.c ، hex.* و ioc.*) و برنامه پروتئوس(فایل pdsprj.*) را در یک فولدر با نام Q4a ذخیره کنید.
- یک پروژه برنامه نویسی جدید ایجاد کنید. ۳ رقم سمت راست شماره دانشجویی خود را در متغیر a ذخیره کنید. سپس متغیر a را بر عدد ۷ تقسیم کرده و در متغیر b ذخیره کنید. سپس مقدار متغیر a را در خط اول و مقدار متغیر b را با ۴ رقم اعشار در خط دوم نمایش دهید. (برای این منظور می بایست از دستور sprintf استفاده کنید)

توجه: چنانچه دستور مورد نظر در cubeide کار نکرد راه حل را با جست و جو در اینترنت بیابید.

- برنامه های میکروکنترلر (فایل های main.c ، hex.* و ioc.*) و برنامه پروتئوس(فایل pdsprj.*) را در یک فولدر با نام Q4b ذخیره کنید.

➢ تایمر

- برنامه ای بنویسید که با استفاده از یک تایمر یک متغیر از مقدار اولیه ۰ هر ۵۰ میلی ثانیه یکبار، ۱ واحد به مقدار آن افزوده شود و هنگامی که به مقدار ۵۰ رسید، ریست شود. مقدار این متغیر را بر روی نمایشگر نشان دهید.

توجه: زمان شبیه سازی پروتئوس بر اساس زمان واقعی نیست بنابراین برای بررسی صحت عملکرد برنامه نوشته شده می بایست Animating Time را در نظر بگیرید.

- برنامه های میکروکنترلر (فایل های main.c ، *.hex و *.ioc) و برنامه پروتئوس(فایل *.pdsprj) را در یک فولدر با نام Q5 ذخیره کنید.

➢ کانتر

- می خواهیم از واحد تایمر میکروکنترلر به منظور شمارش تعداد فشرده شدن یک Push button که خروجی آن Pull up است، استفاده کنیم. برای این منظور Clock Source باید بر روی حالت External قرار گیرد. مقدار کانتر نیز بر روی نمایشگر نشان داده می شود.

توجه : در این بخش نباید از وقفه خارجی استفاده کنید.

- برنامه های میکروکنترلر (فایل های main.c ، *.hex و *.ioc) و برنامه پروتئوس(فایل *.pdsprj) را در یک فولدر با نام Q6 ذخیره کنید.

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

- تنها به پروژه هایی که با STM32CubeIDE انجام شده اند نمره تعلق خواهد گرفت.
- فولدرهای مربوط به برنامه ها را به صورت یک فایل zip تجمیع و با نام student_number.zip ارسال شوند .
- به تمرین هایی که به صورت مشابه حل شده اند نمره ای تعلق نخواهد گرفت.

همواره موفق باشید