آزمایش 5

طاها موسوی 98243058

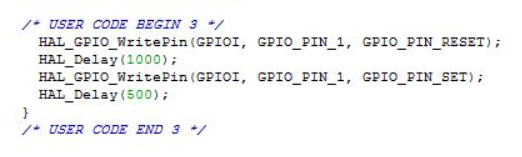
نیلوفر مرادی جم 97243063

گروه 2

# **سوالات تحلیلی:**

**1 - اگر تنظیمات میکرو خود را در میان کار تغییر دهیم، هنگام تولید مجدد کد در STM32CubeMX چگونه میتوان از از بین رفتن کدهای توسعه داده شده خود جلوگیری کنیم؟**

جواب: کد ها را بین USER CODE BEGIN و USER CODE END می گذاریم. سپس باید تعیین کنیم که این قسمت کد نگه داری یا بازنویسی شود. برای همین کار باید در قسمت Code Generator setting باید تعیین شود که کد بین دو کامنت گفته شده قرار گرفته است.   
به این صورت می­توانیم تنظیمات را حین کار تغییر دهیم بدون این که مشکلی پیش بیاید. برای مثال به این صورت:



**2 - واژه کلیدی \_\_weak در زبان C به چه معنی است و چرا برخی توابع CMSIS و HAL اینگونه تعریف شده اند؟**

از این واژه کلیدی برای تعریف کردن توابع و vector tables و default handlers و توابع وقفه ها یا اعلان توابع و متغیر ها استفاده می­شود. و باعث می­شود کامپایلر نماد ها را ضعیف ارسال کند.  
اگر تابعی از این نوع باشد، حتی در صورتی که از قبل در آن کدی نوشته شده باشد، قابلیت این را دارد تا برنامه نویس دوباره آن تابع را در فایل main بدون پیشوند weak بازنویسی کند.  
اما اگر تابع از نوع strong باشد، قابلیت این را ندارد که بیشتر از یک بار اعلان شود. که در این صورت باید از weak کمک گرفت.

اما در صورت تعریف چندین weak، لینکر خطا می­دهد. برای رفع این مشکل باید از گزینه muldefweak استفاده شود.

**3 - جهت به کارگیری وقفه ها در HAL چه سازوکاری تعبیه شده؟ چگونه باید از آن بهره برد؟ ارتباط آن با سازوکار تعبیه شده در CMSIS چیست؟**

در HAL به صورت کال بک این وقفه ها صدا زده می­شوند و نیازی به interrupt service routine نیست. زمانی که interrupt handler در سطح CMSIS فعال می­شود این متد ها هنگام اینتراپت در HAL صدا زده می­شود. چون این متد ها یک لایه بالاتر از interrupt service routine ها هستند و کار کردن با آن ها ساده تر است.

ابتدا کار هایی مانند clear pending bits و .. را انجام می­دهد و بعد آنها متد کال بک اجرا می­شود. پس می­توان کد های خود را درون متد کال بک نوشت تا اجرا شوند.

برای مثال در HAL می­توان از متد EXTIx\_HNADLER برای صدا زدن متد هندلر استفاده مرد و پین مربوط به آن Interrup line را مشخص کرد. و به کمک متدی که در HAL موجود است کار هایی که گفته شد و نیاز است را اجرا می­کند و بعد آن متد کال بک اجرا می­شود.   
مشابه همین در CMSIS از همین متد به عنوان یک متد هندلر برای interrupt service مربوط به EXTIx استفاده می­شود.

**رفرنس های سوالات تحلیلی**:  
- کلاس درس و اسلاید های درسی

STM32Cube-MX-HAL-MOOC

<https://melec.ir/%d8%b1%d9%88%d8%b4%d9%86-%da%a9%d8%b1%d8%af%d9%86-led-%d8%a8%d8%a7-%d9%85%db%8c%da%a9%d8%b1%d9%88-stm32/>

<https://stackoverflow.com/questions/35507446/what-are-weak-functions-and-what-are-their-uses-i-am-using-a-stm32f429-micro-co>

<https://sisoog.com/2019/01/06/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D9%85%DB%8C%DA%A9%D8%B1%D9%88%DA%A9%D9%86%D8%AA%D8%B1%D9%84%D8%B1-stm32f4-%D9%82%D8%B3%D9%85%D8%AA-%D8%B3%D9%88%D9%85-%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF-%D9%BE%D8%B1/>

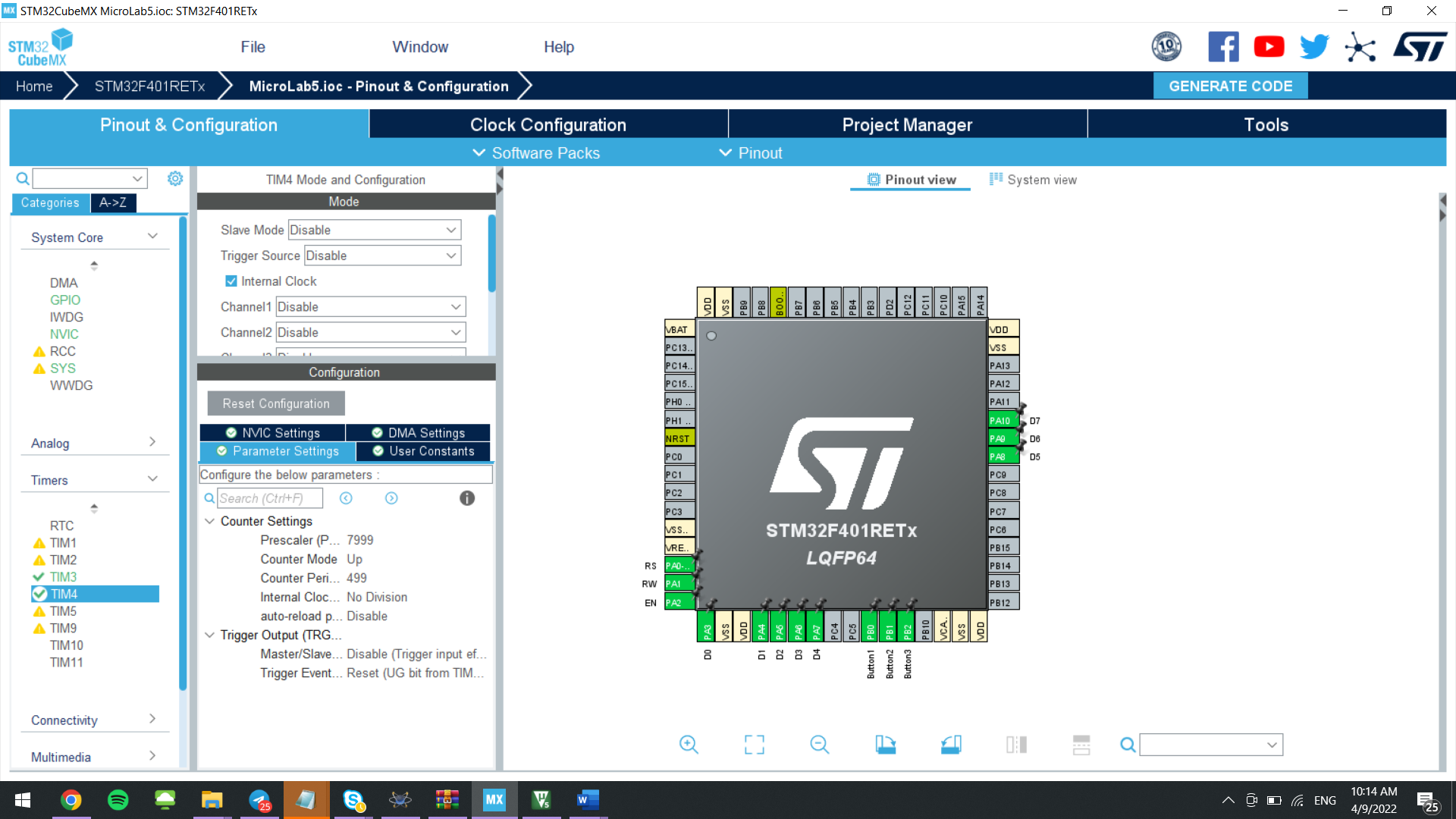
<http://freeelec.ir/?p=454>

# دستور کار:

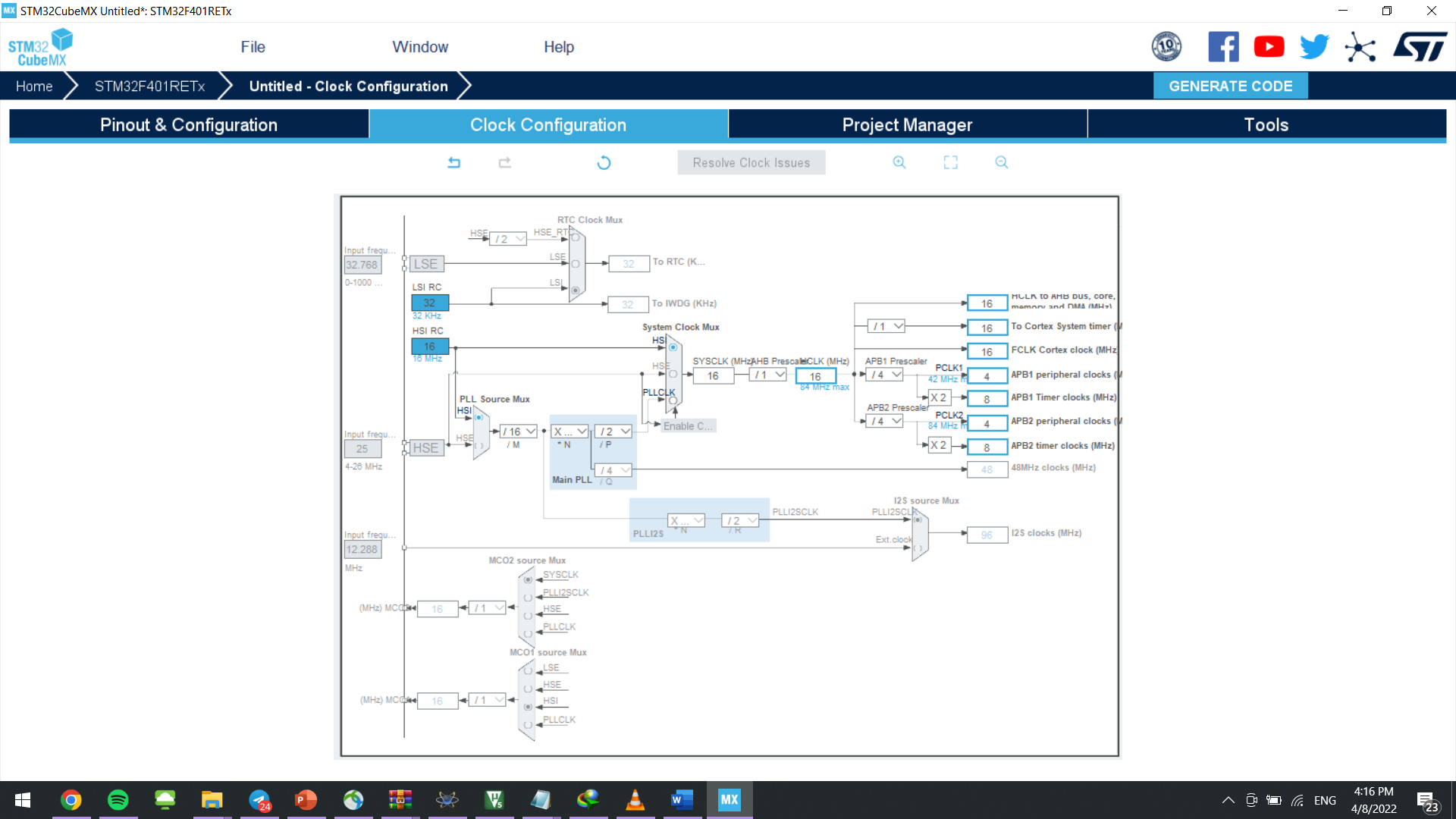
پورت ها را به صورت زیر کانفیگ میکنیم:

PA3 تا PA13 خروجی و متصل به lcd هستند. PA0 و PA1 و PA2 خروجی و متصل به پورت های EN RW RS هستند.

PB0و PB1 و PB2 به عنوان EXTI0\_GPIO, EXTI1\_GPIO, EXT12\_GPIO ست میشوند و متصل به سه دکمه شماره یک و دو و سه هستند.



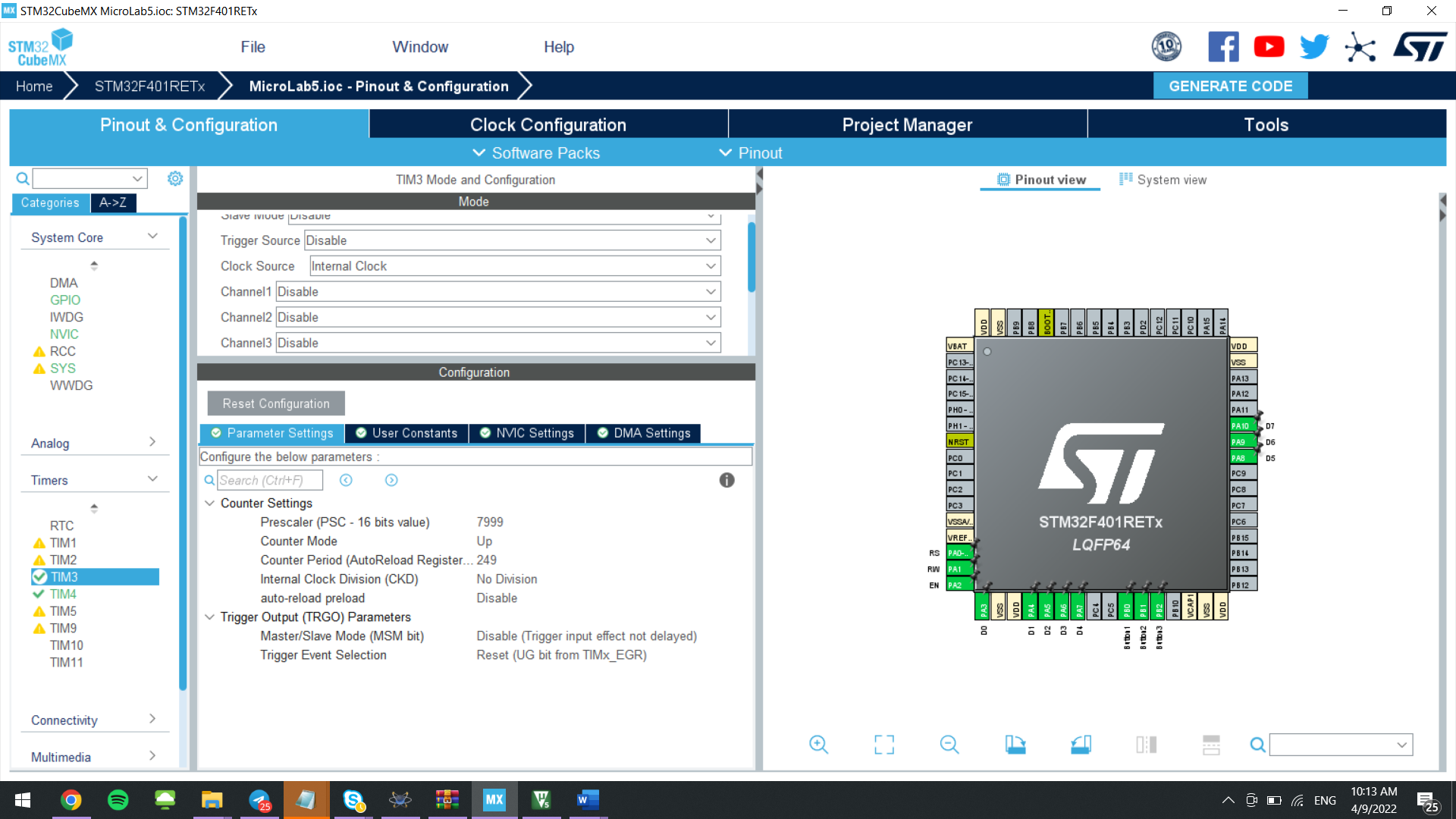
کالک تایمر 2 را که از باس APB1 تامین میشود را به صورت زیر تنظیم میکنیم:

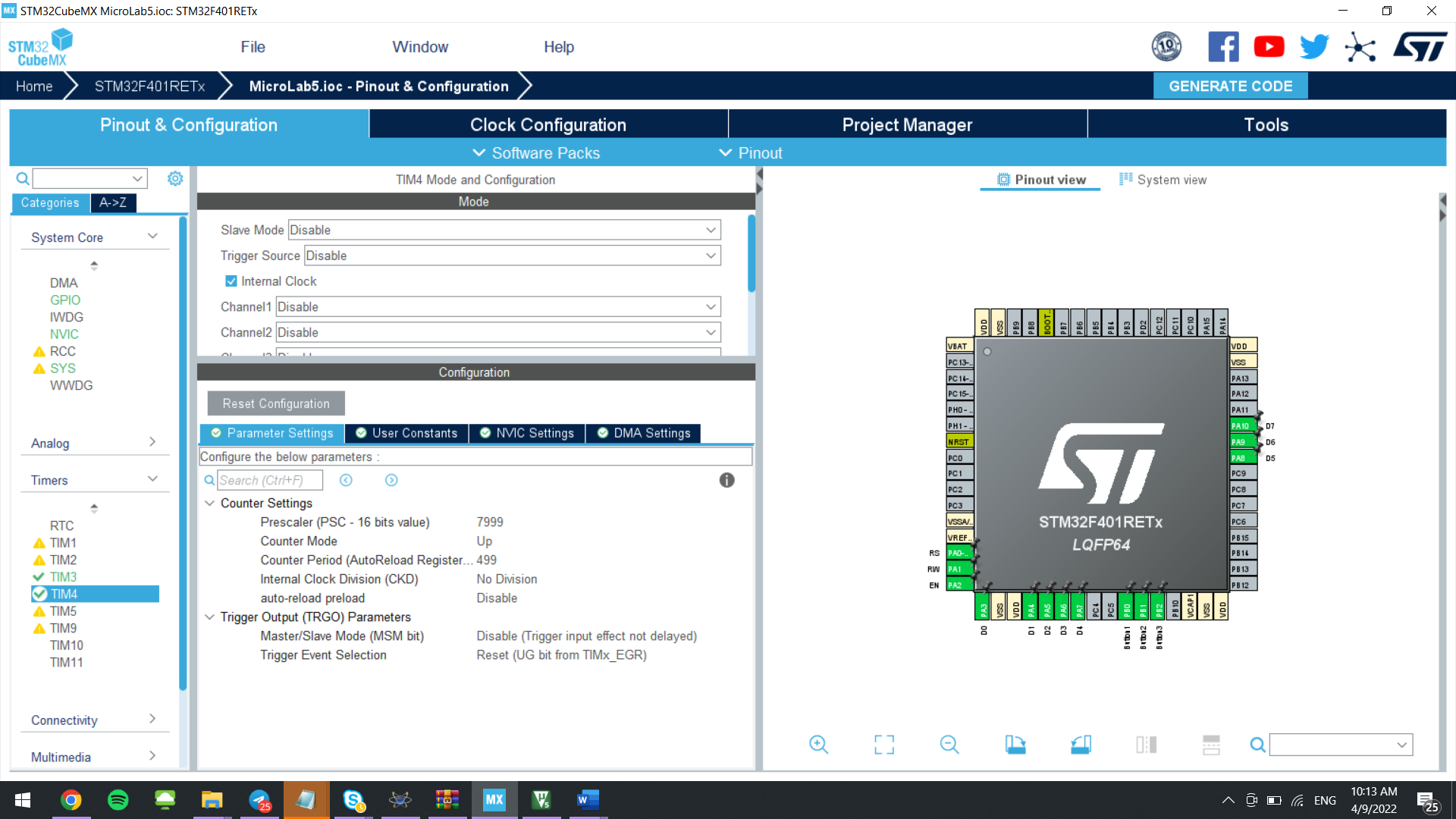


تایمر باید با فاصله های زمانی یک میلی ثانیه ای شمارش را انجام دهد. در تایمر ما از دو رجیستر prescaler و period counter برای تنظیم کردن این فاصله های زمانی هم استفاده میکنیم:

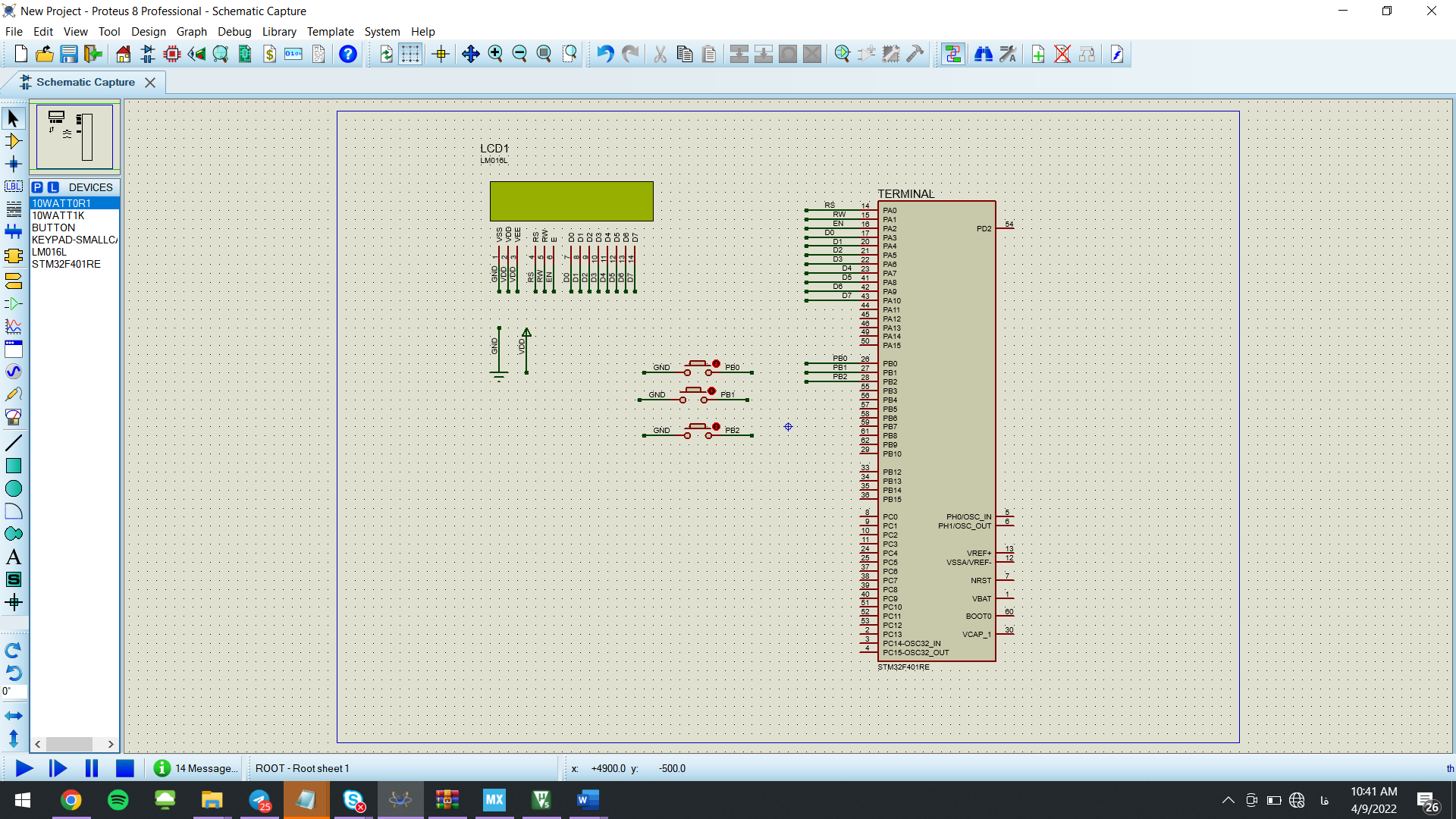
frequency = clock of timer / (prescaler \* counter period)

که با توجه به اینکه کلاک تایمر 16 میباشد باید برای تولید بازه های یک میلی ثانیه ای، به عنوان مثال، prescaler برابر 499 و period counter برابر 249 باشد تا تغییر هر 250 میلی ثانیه یک بار باشد





پروتئوس را مشابه بورد stm cubex طراحی میکنیم.

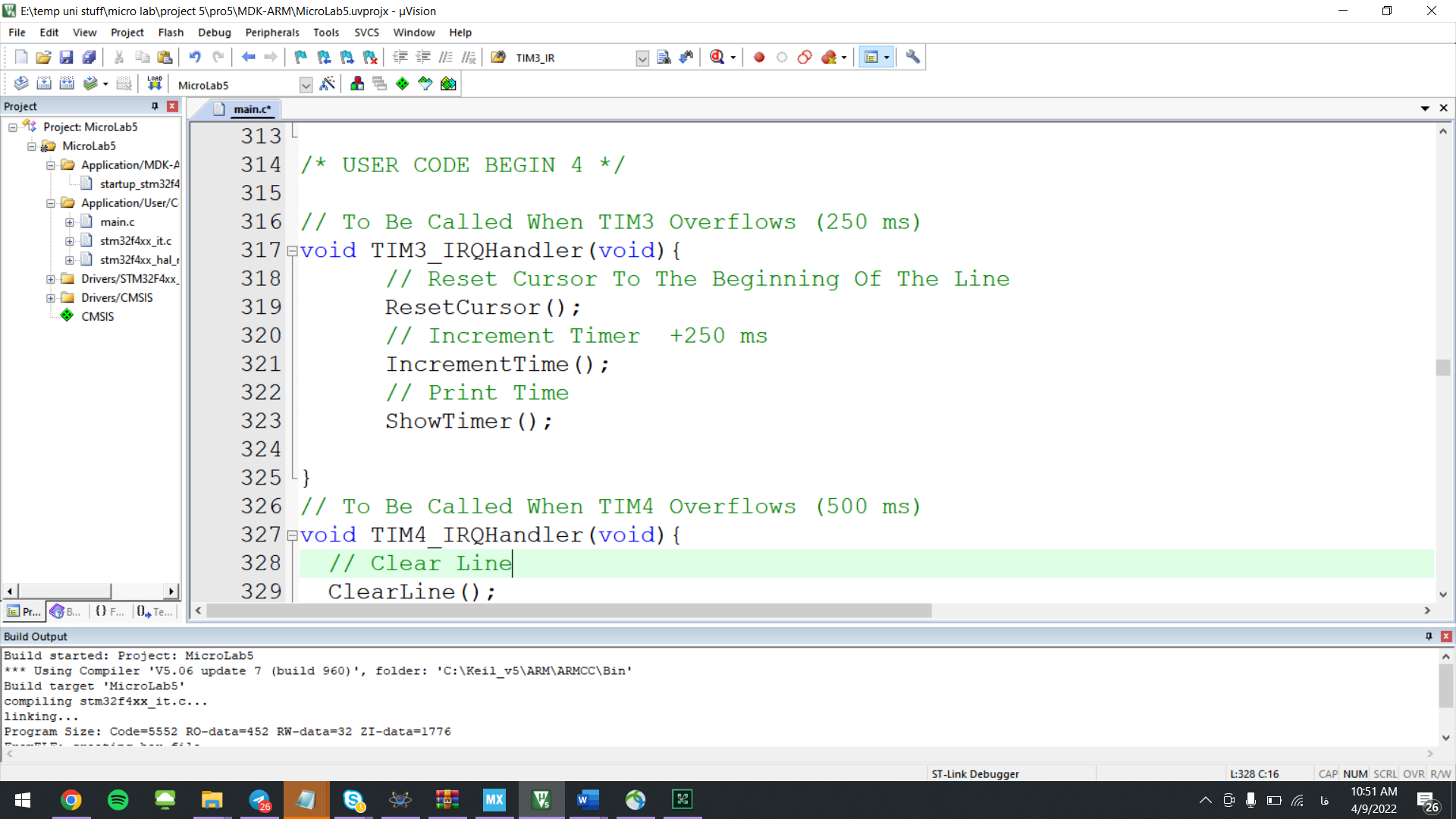


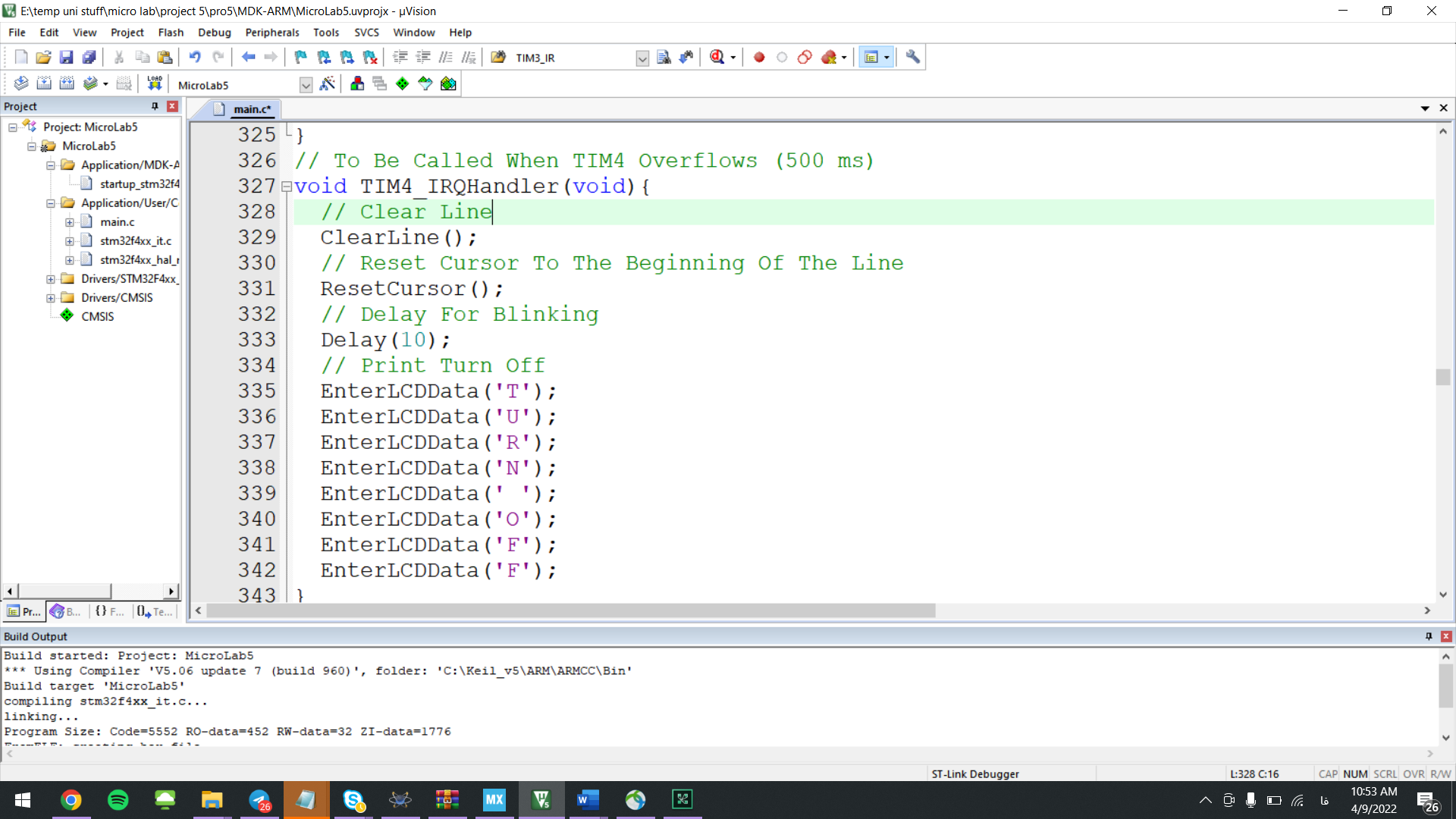
در ادامه با جنریت شدن کد توسط cubemx کد را طبق منطق صورت پروژه تغییر میدهیم.



بعد از تغییر در کلید متود بالا فراخوانی شده و با توجه به این که تغییر در کلید اول یا دوم یا سوم بوده است و طبق منطق صورت سوال تغییرات لازم اعمال میشود و خروجی نمایش داده میشود.

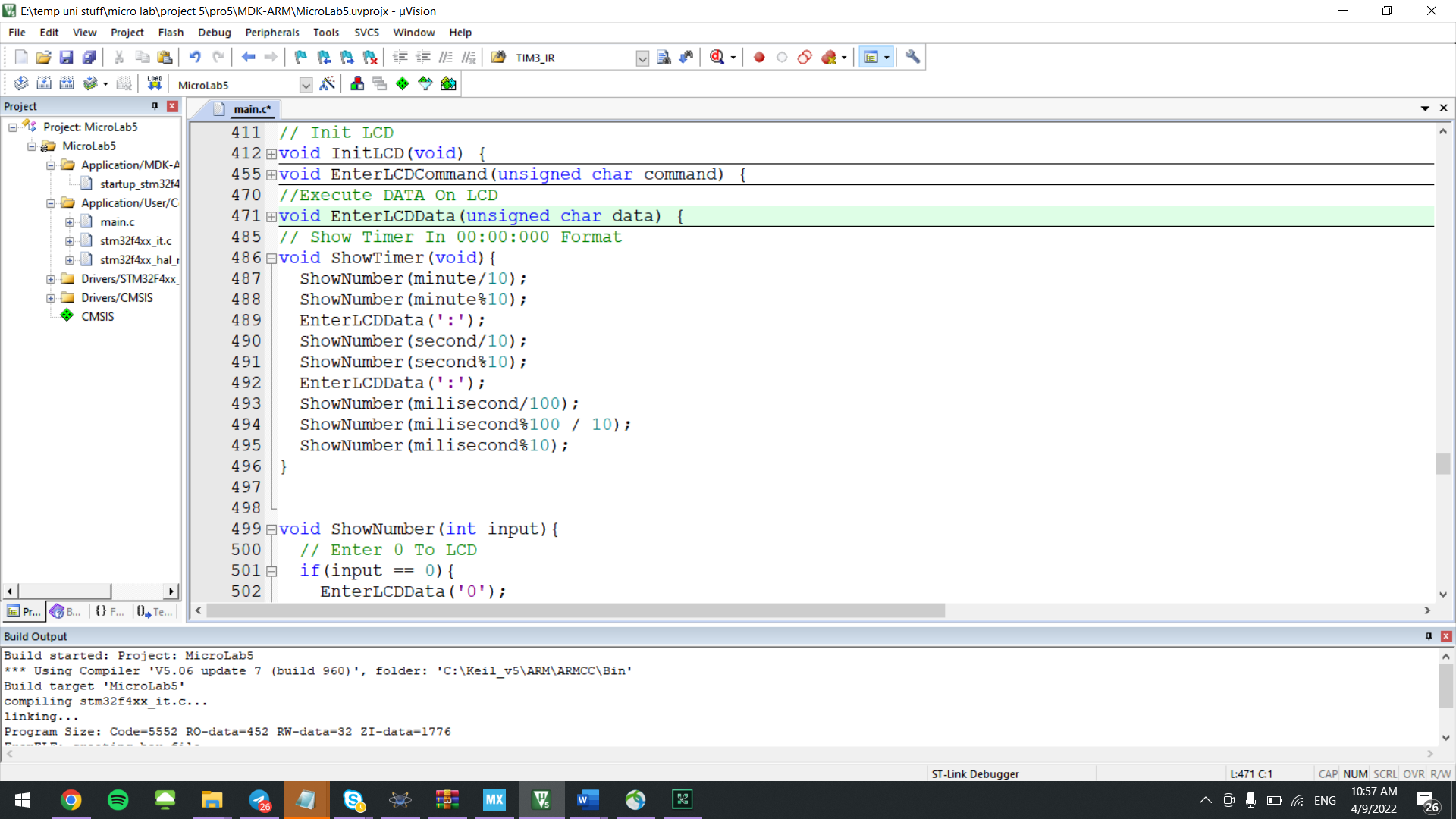
تایمر هر 250 میلی ثانیه یک بار آپدیت شده و زمان جدید روی lcd نمایش داده میشود.



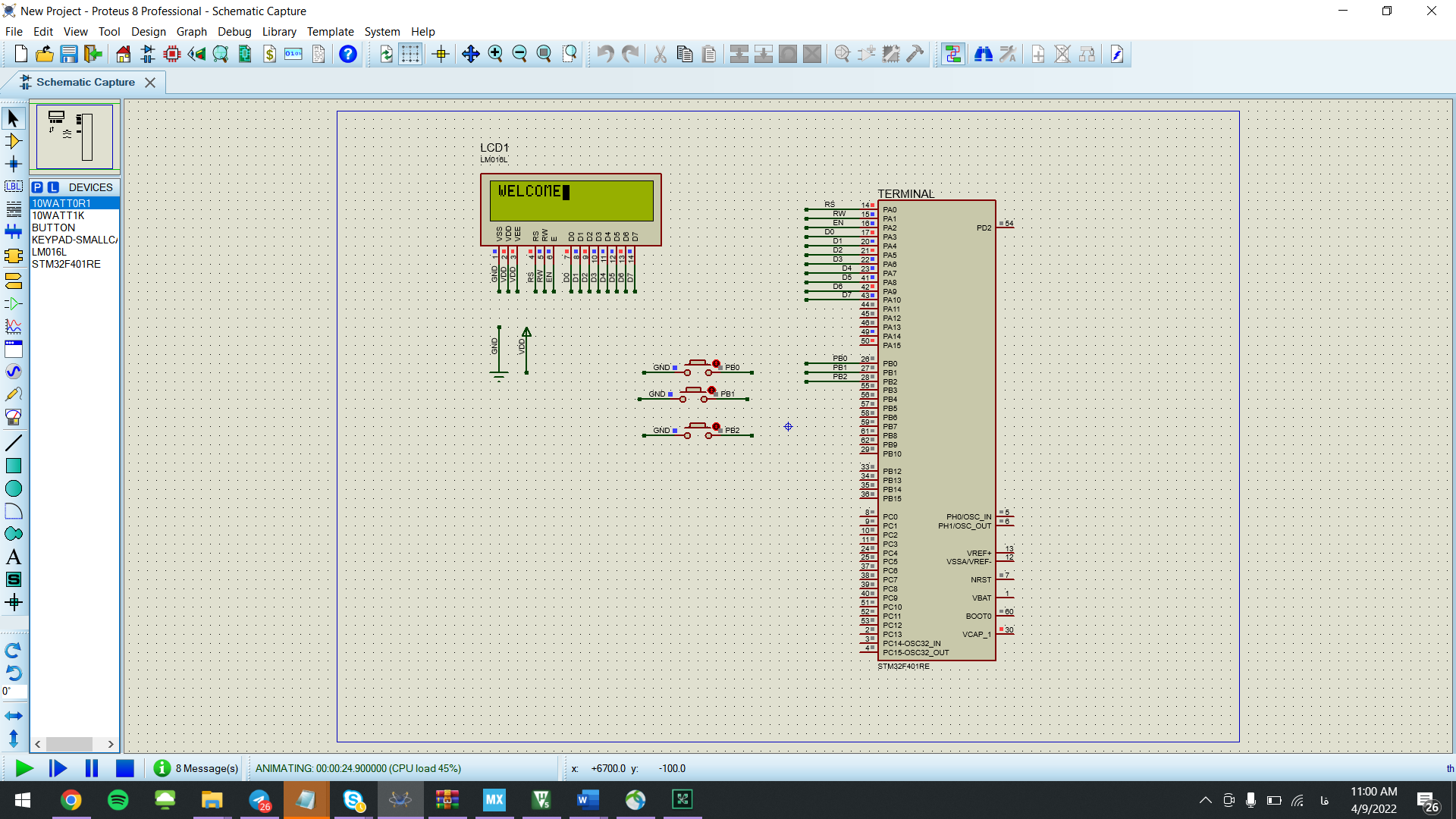


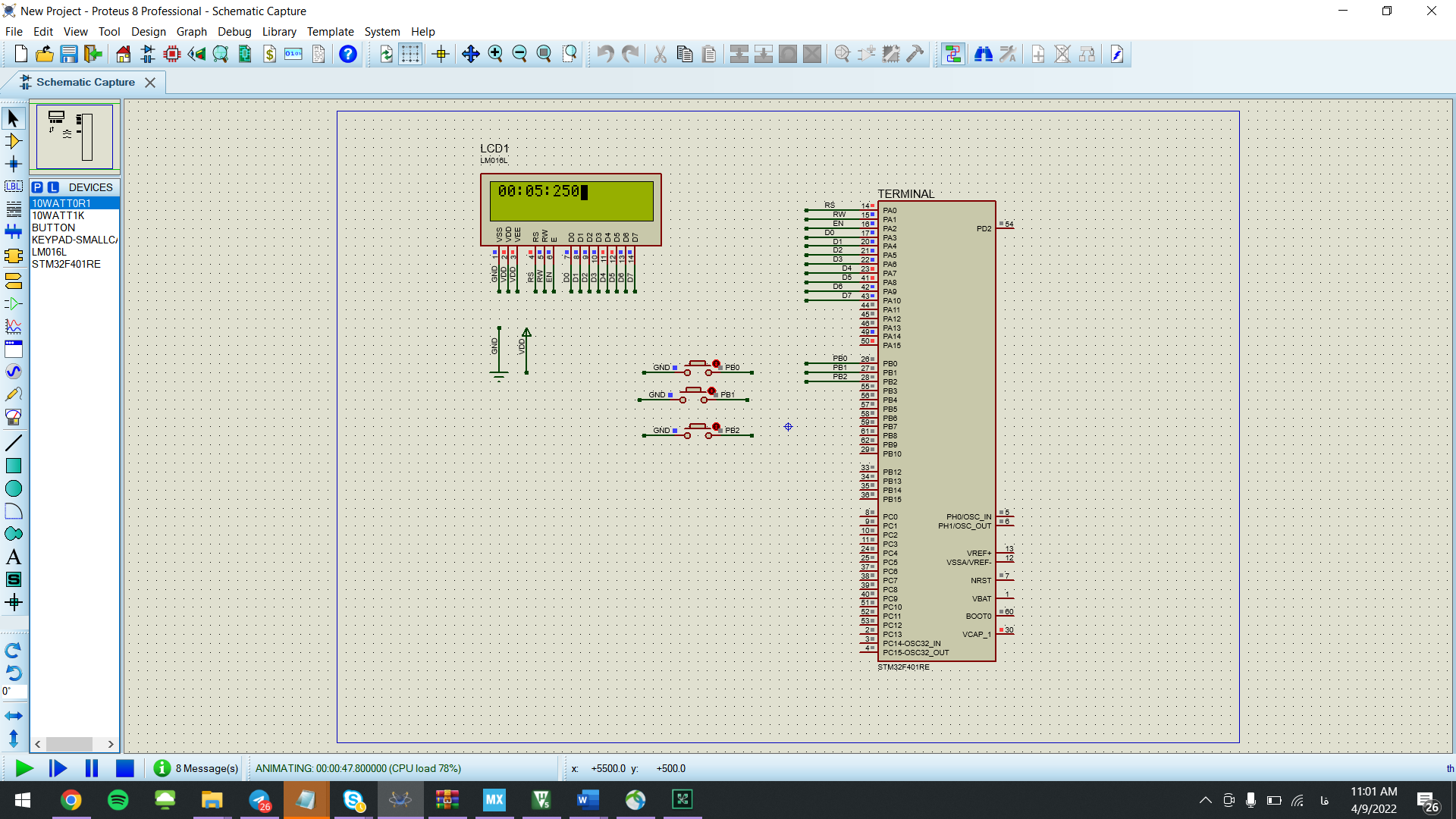
به سراغ Callback های مربوط به تایمر ها می رویم. از آنجاکه Callback مربوط به تایمر یعنی PeriodElapsedCallBack\_TIM\_HAL در هنگام سرریز شدن تایمرها صدا زده نمیشد, از Handler Interrupt های پیش فرض مربوط به این دو تایمر استفاده کرده ایم.

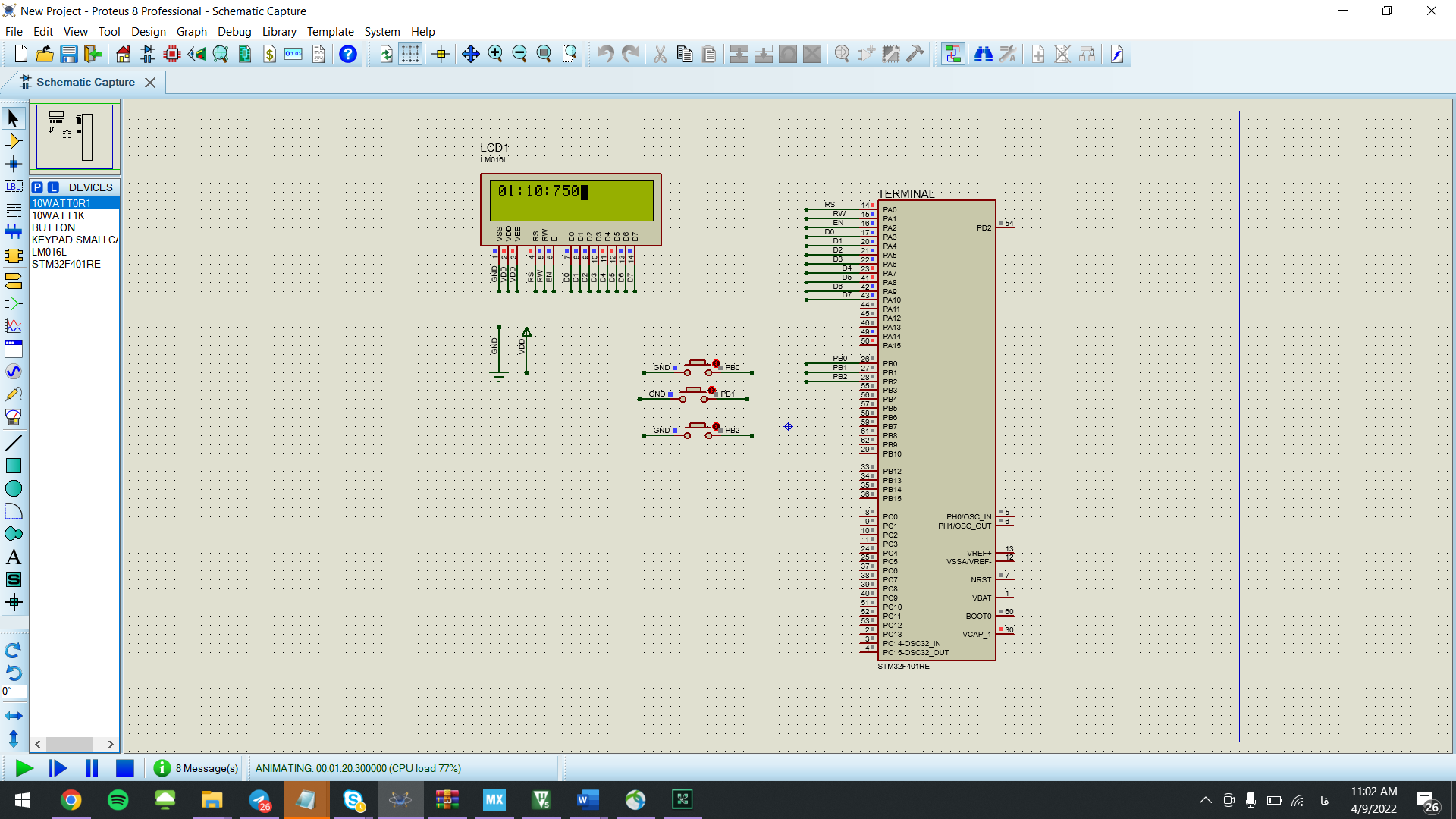
برای نمایش از زمان نری متد ShowTimer و با استفاده از متغریهای تعریف شده, به صورت زیر استفاده میکنیم تا به فرمت mmm:SS:MM نمایش داده شود.

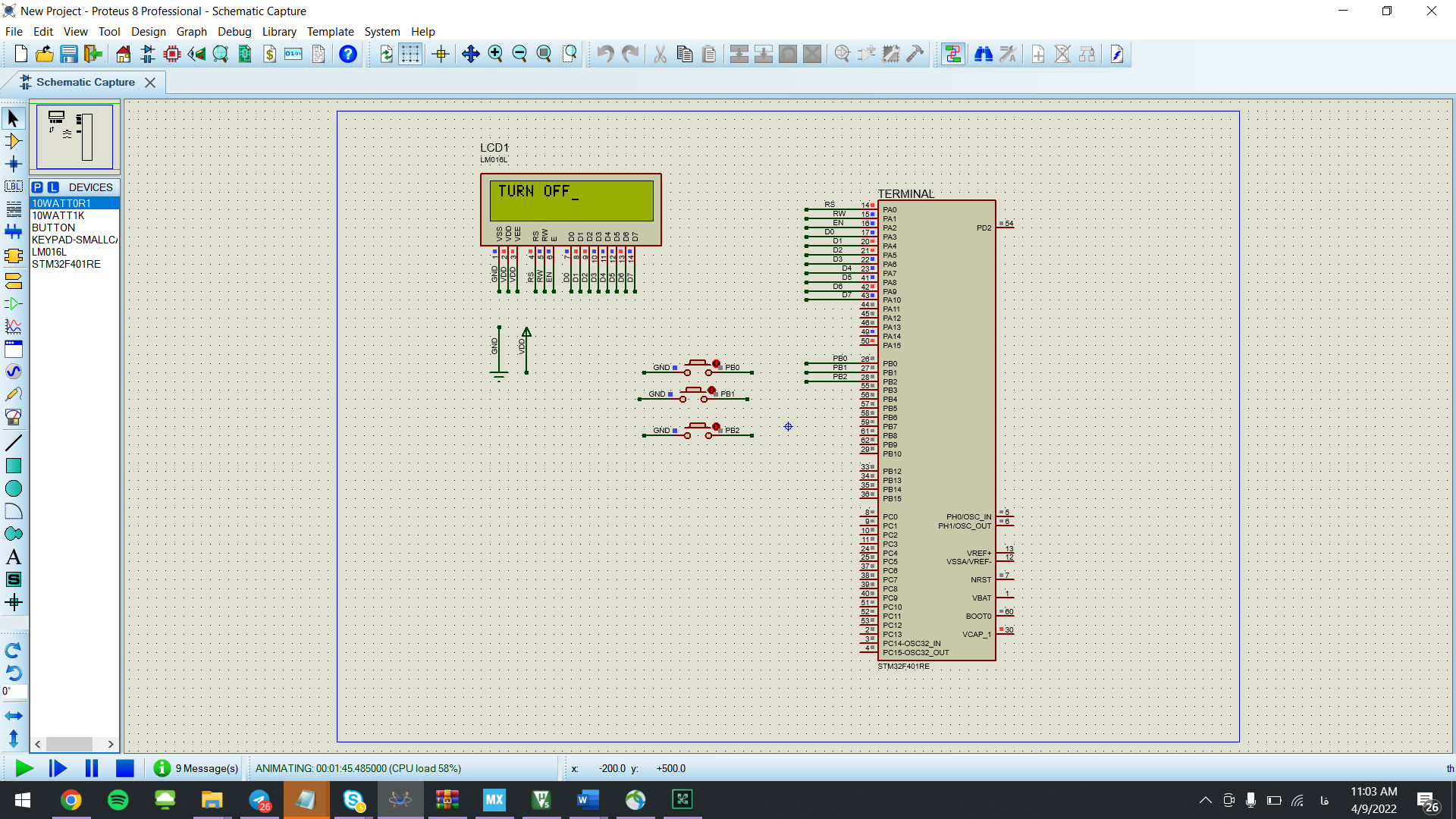


در نهایت مشاهده خروجی در پروتئوس:









**رفرنس دستور کار:**

کلاس درس و اسلاید های درسی

دیتا شیت و رفرنس منوآل

<https://www.keil.com/pack/doc/cmsis/DSP/html/index.html>

<https://www.st.com/en/embedded-software/stsw-stm32065.html>

<https://www.keil.com/dd/docs/arm/st/stm32f4xx/stm32f4xx.h>

<https://www.keil.com/dd/docs/arm/st/stm32f4xx/stm32f4xx.h>