آزمایش 6

طاها موسوی 98243058

نیلوفر مرادی جم 97243063

گروه 2

# **سوالات تحلیلی:**

**1- مشخصات یک سیگنال PWM چیست؟**

از مشخصات pwm میتوانیم به موارد زیر اشاره کنیم:

فرکانس: نشان میدهد چند پالس در هر ثانیه رخ میدهد همچنین این فرکانس ثابت است.

دوره تناوب: از طریق فرکانس بدست میاید یک تقسیم بر فرکانس دوره تناوب را به ما میدهد.

On time: مدت زمانی که پالس روشن است.

Duty-cycle: تقسیم on time بر دوره تناوب هر چقدر duty-cycle بزرگتر باشد یعنی توان بیشتری منتقل شده.

Adjust on-time : نشان دادن مقادیر انالوگ

**2 - تفاوت مد یک و دو PWM در STM32F4 چیست؟**

در مد اول در upcounting کانال اول فعال است تا زمانی که TIMx\_CNT<TIMx\_CCR1 باشد در غیر این صورت غیر فعال است و در downcounting کانال اول غیر فعال است تا زمانی که TIMx\_CNT>TIMx\_CCR1 در غیر این صورت فعال است.

در مد دوم برعکس است یعنی چنل اول غیر فعال است تا زمانی که TIMx\_CNT<TIMx\_CCR1 در غیر این صورت فعال است و در downcounting کانال اول فعال است تا زمانی که TIMx\_CNT>TIMx\_CCR1 در غیر این صورت غیر فعال است.

همچنین فاز شکل موج وقتی duty-cycle تغییر بدیم در مود اول تغییر نمیکند و در دومی تغییر میکند.

**3 - چه تفاوتی در ساختار LEDها با رنگهای متفاوت وجود دارد؟ آیا می­توان در هر مداری آنها را جایگزین یکدیگر نمود؟ شرح دهید.**

TIMx\_CCMR1(capture compare mode register 1) برای تنظیم کانال های یک و دو استفاده میشه و TIMx\_CCMR2(capture compare mode register 2) برای تنظیم کانال های سه و چهار استفاده میشود.

capture compare mode register 1 هشت بیت سمت راست آن برای تنظیمات کانال یک و هشت بیت سمت چپ برای تنظیمات کانال دو میباشد.

capture compare mode register 2 هشت بیت سمت راست آن برای تنظیمات کانال سه و هشت بیت سمت چپ برای تنظیمات کانال چهار میباشد.

دو مد مختلف input capture و output compare دارند که

Input capture برای گرفتن سیگنال ورودی داده شده به میکرونترلر و اندازه گیری فرکانس و عرض پالس آن کاربرد دارد.

Output capture برای کنترل شکل موج خروجی یا نشان دادن دوره زمانی که سپری شده و همچنین برای تولید پالس های خروجی کاربرد دارد.

**رفرنس های سوالات تحلیلی**:

- کلاس درس و اسلاید های درسی

<https://resources.pcb.cadence.com/blog/2020-pulse-width-modulation-characteristics-and-the-effects-of-frequency-and-duty-cycle#:~:text=PWM%20Characteristics&text=There%20are%20two%20primary%20components,a%20full%20ON-OFF%20cycle>.

https://en.wikipedia.org/wiki/Pulse-width\_modulation

http://www.ledified.com.au/understanding-the-colour-temperature-of-led-lights/#:~:text=LED%20lights%20come%20in%20a,light%20appears%20crisp%20and%20white.

# دستور کار:

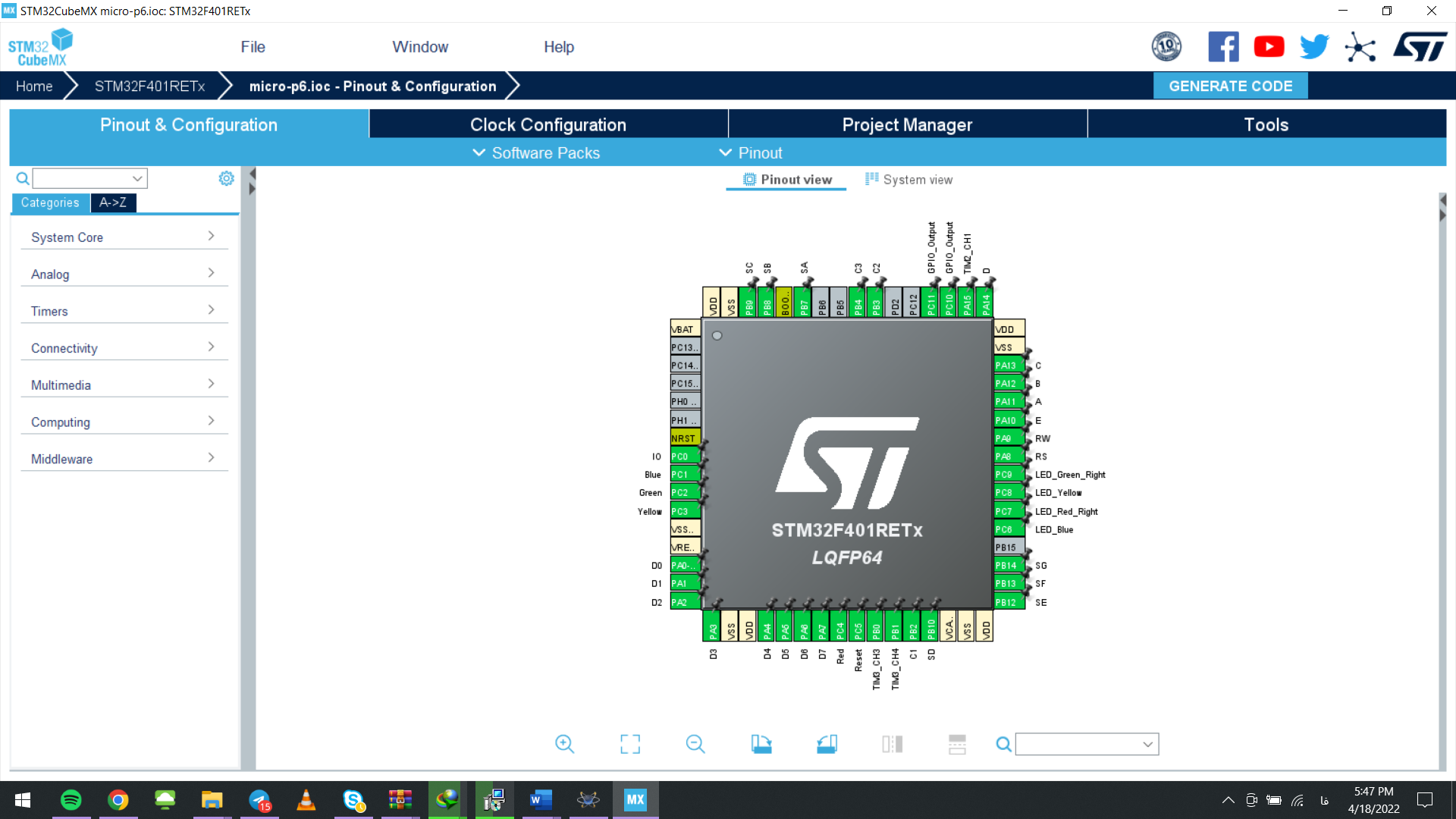
ورودی خروجی­ها:‌

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PA0 | PA1 | PA2 | PA3 | PA4 | PA5 | PA6 | PA7 | PA8 | PA9 | PA10 | PA11 | PA12 | PA13 | PA  14 | PA15 | pin |
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | RS | RW | E | A | B | C | D | RESET | **symbol** |
| O | O | O | O | O | O | O | O | O | O | O | O | O | O | O | IC | **type** |

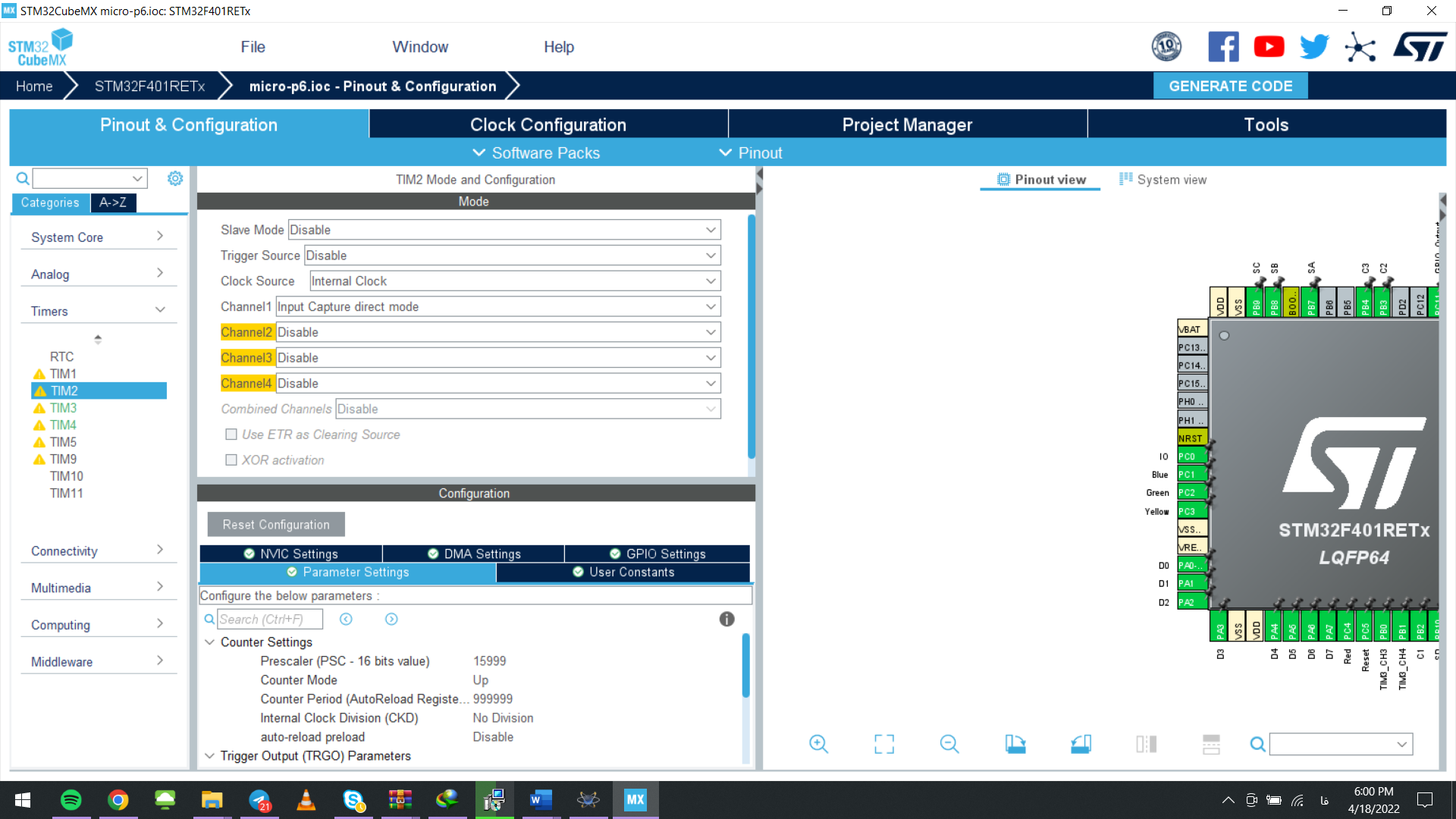
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PB0 | PB1 | PB2 | PB3 | PB4 | PB7 | PB8 | PB9 | PB10 | PB12 | PB13 | PB14 | **PIN** |
| GREEN  PWM | RED  PWM | C1 | C2 | C3 | SA | SB | SC | SD | SE | SF | SG | **SYM** |
| TIM3-CH4-PWM | TIM3-CH3-PWM | I | I | I | O | O | O | O | O | O | O | **TYPE** |

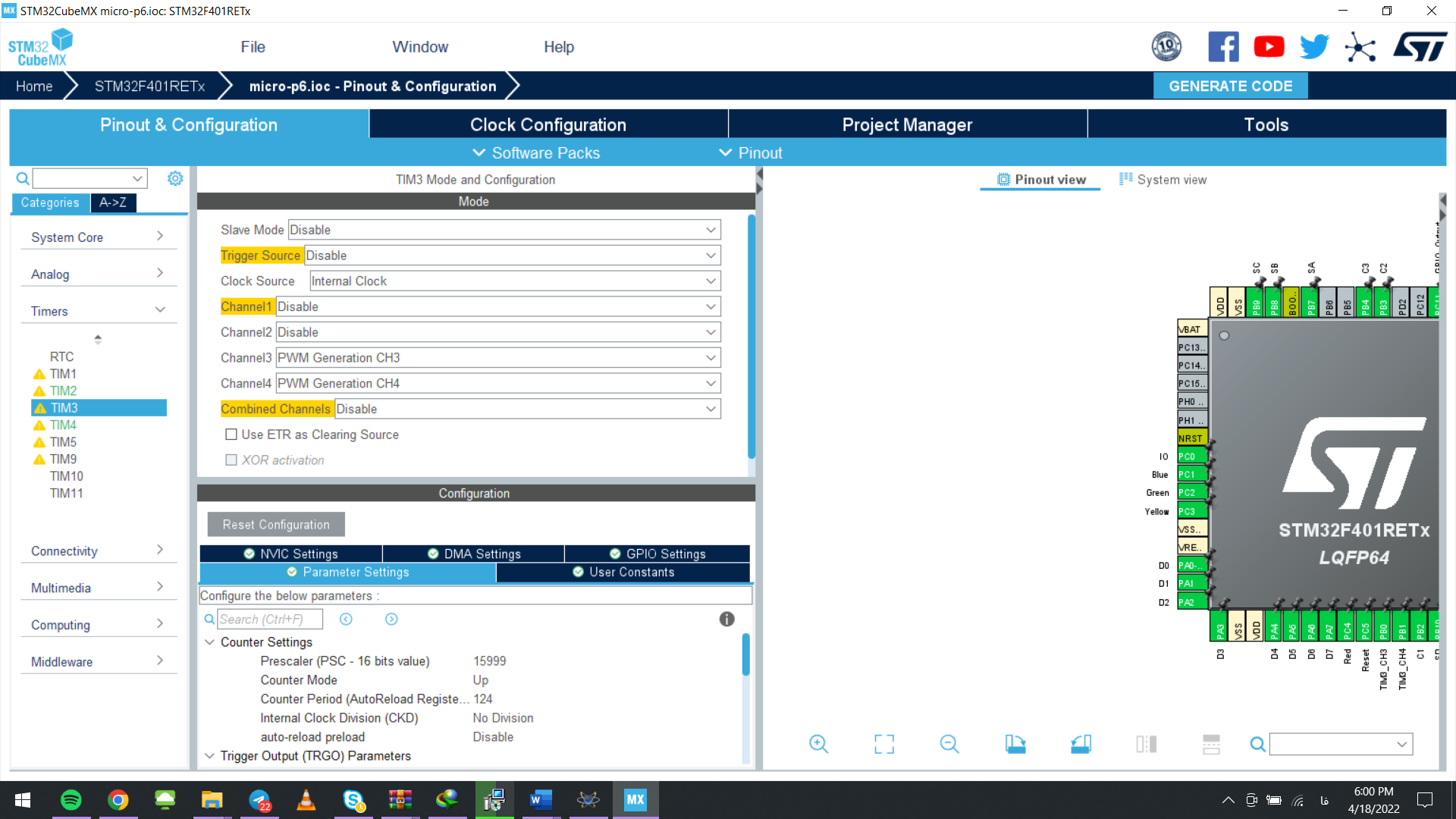
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PC0 | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PC8 | PC9 | PC10 | PC11 | **PIN** |
| I0 | BLUE | GREEN | YELLOW | RED | RESET | LED\_BLUE | LED\_  GREEN | LED\_  YELLOW | LED\_  RED | GREEN\_PWM\_  EN | RED\_  PWM\_  EN | **SYM** |
| interrupt | I | I | I | I | I | O | O | O | O | O | o | **TYPE** |

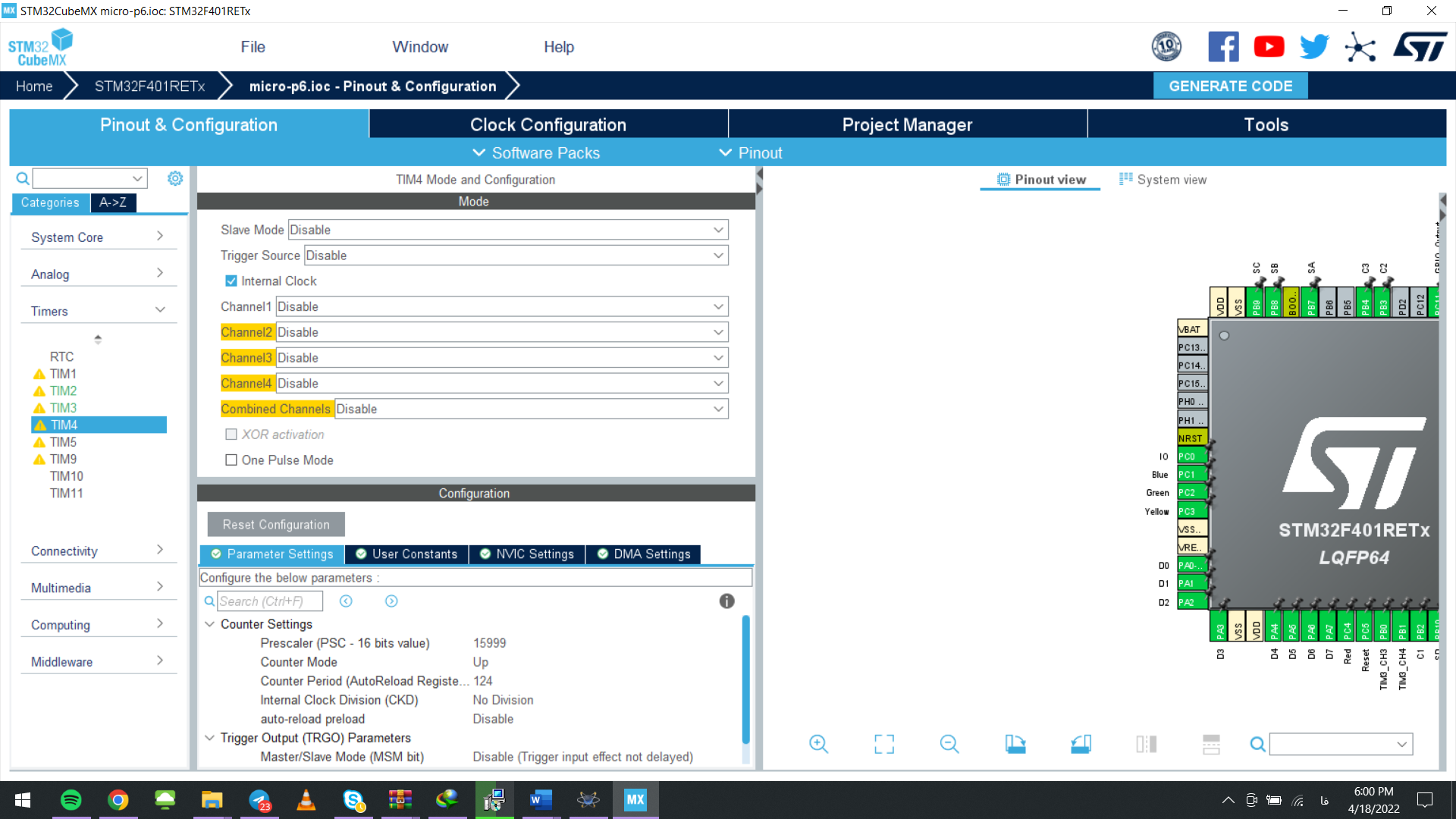
طرح کلی:



تایمر ۴ برای محاسبه فاصله زمانی میان toggle کردن نام است و از چنل اول تایمر ۲ نیز برای input-capture‌ روی خط reset و از چنلهای ۳ و ۴ تایمر را برای pwm‌ استفاده میشود.







numberOfAllowedMisses تعداد خطای مجاز که برابر 3 است.

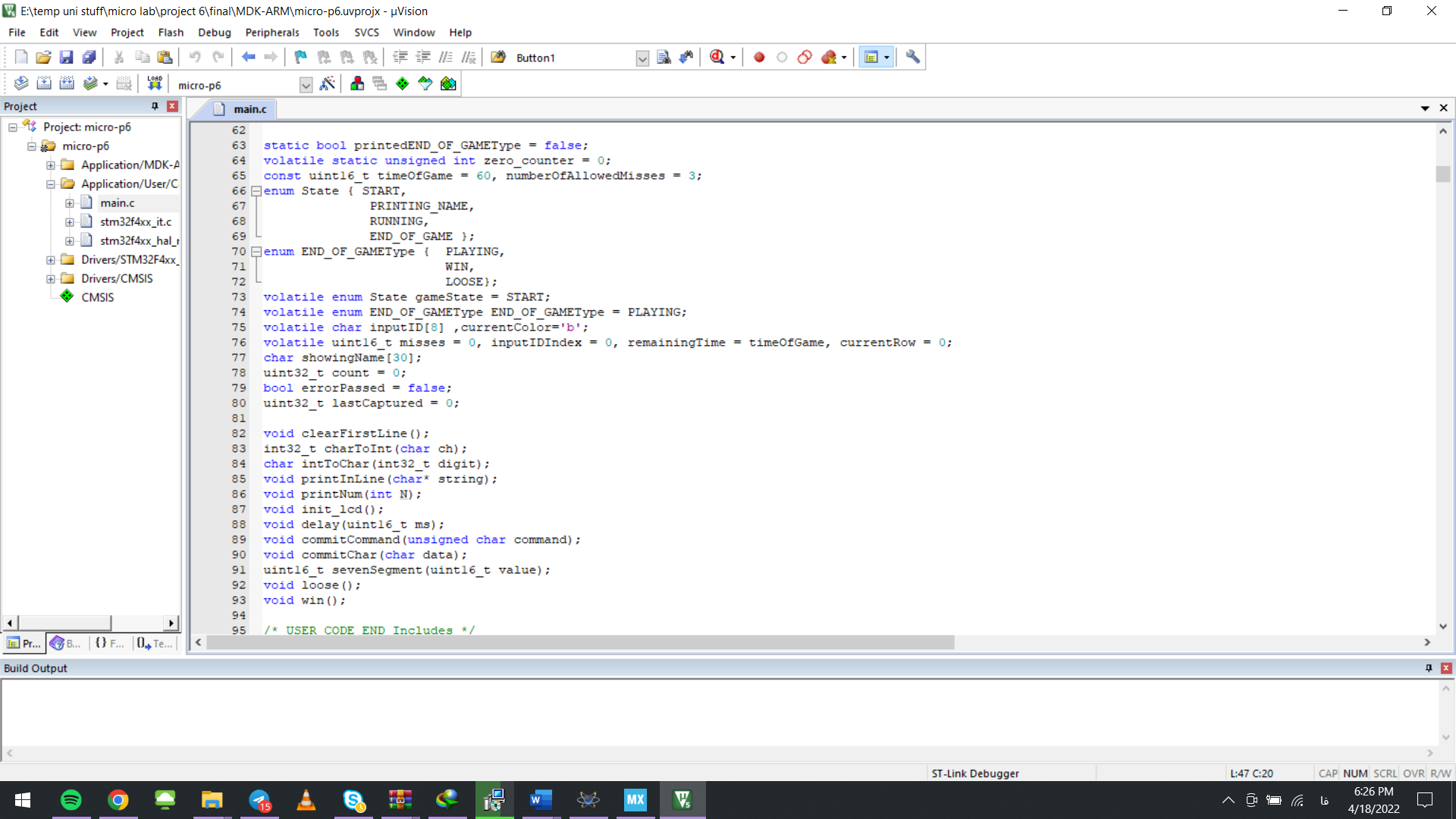
timeOfGame تایم کلی بازی را نشان میدهند.

آرایه inputID برای شماره دانشجویی میباشد.

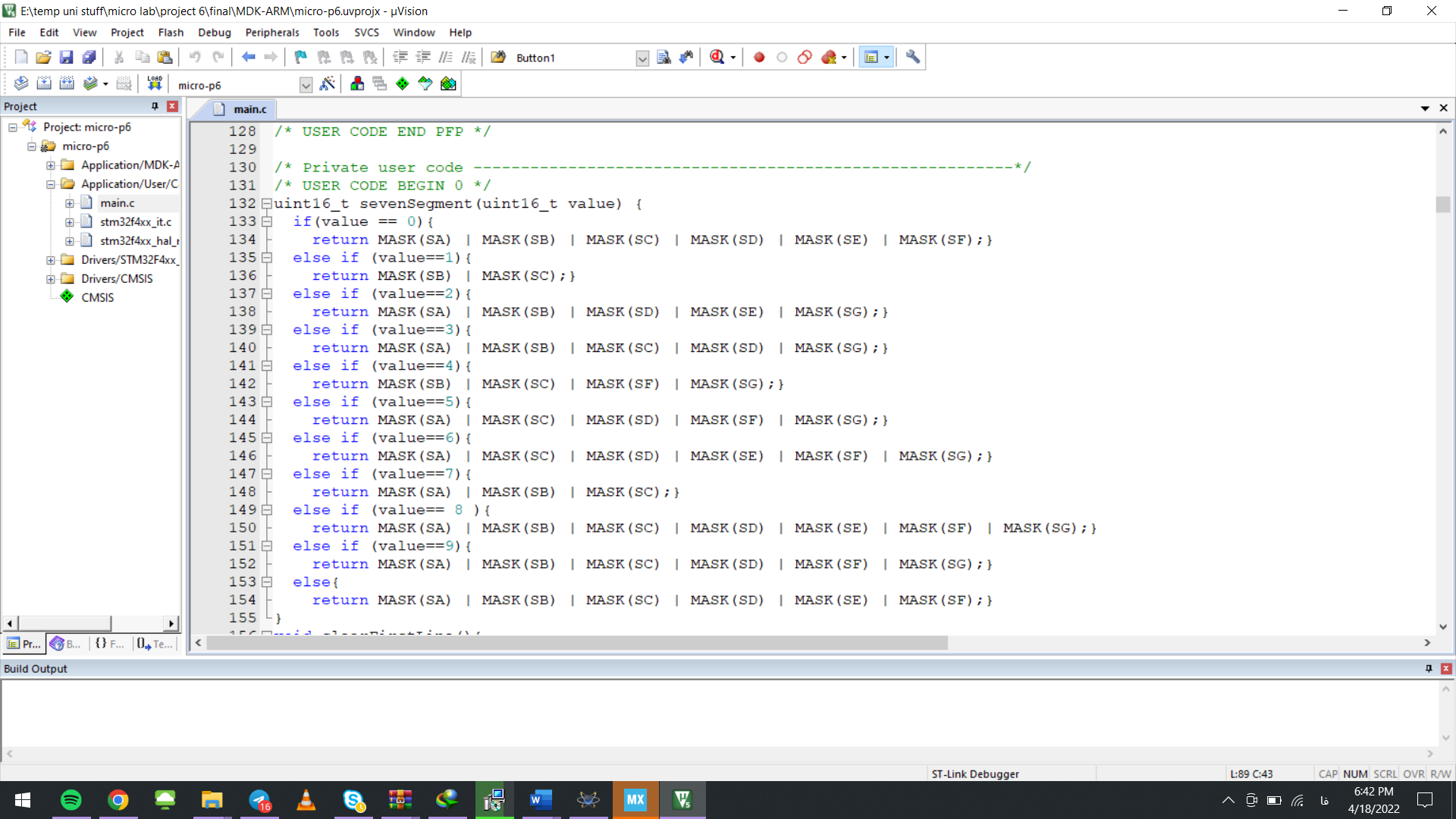
تعدادخطاها، متغیری برای نشاندادن و حفظ اسم متناظر با شماره دانشجویی ورودی، حالت برنامه، تایم باقیمانده، و شکل نتیجه نیز تعریف میشوند.

اینام هایی برای مشخص کردن استیت بازی و همچنین برنده و بازنده و در حال بازی بودن تعریف شده است.

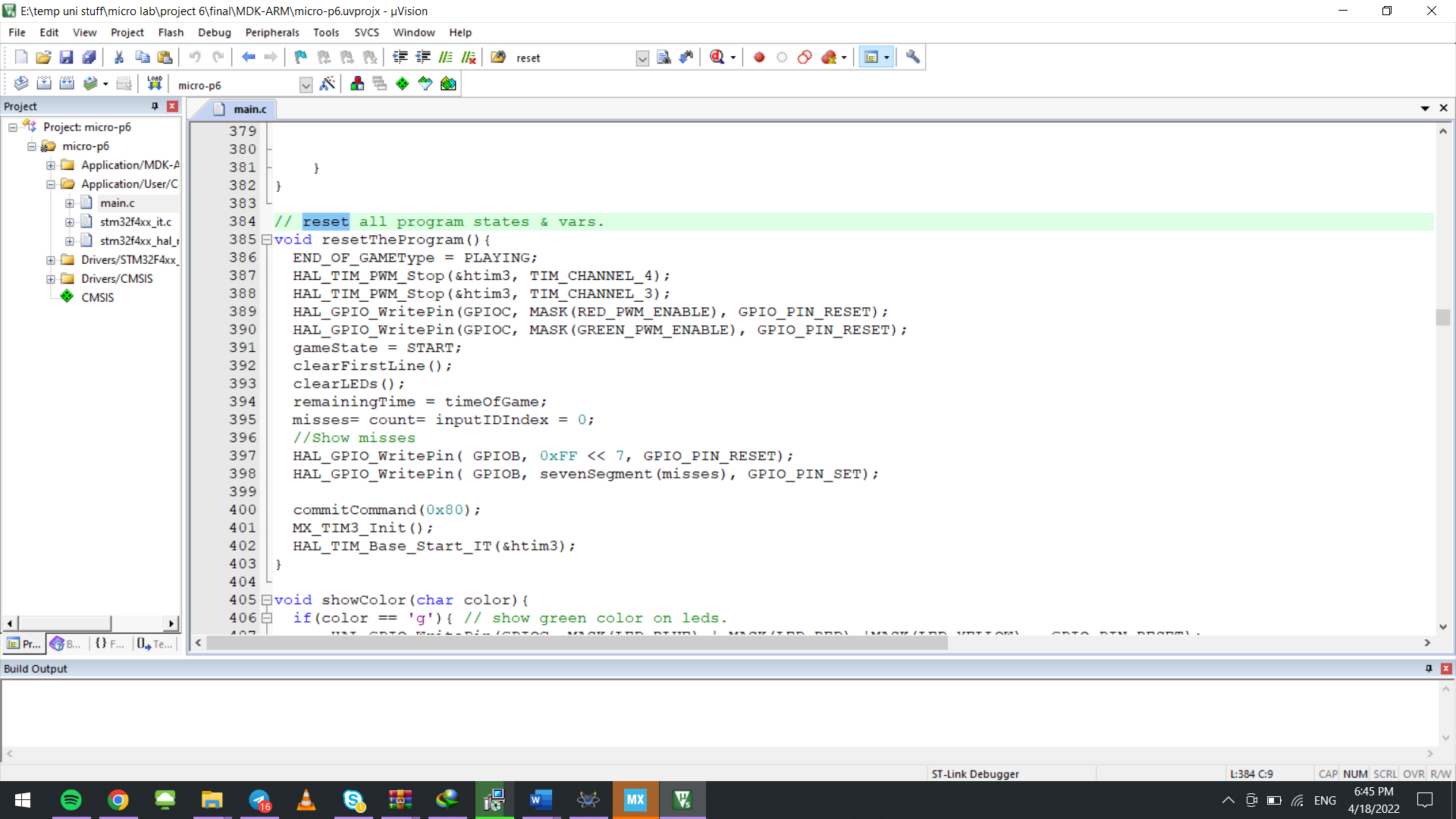
Current color نیز رنگ ال ای دی است که در حال حاضر روشن است.



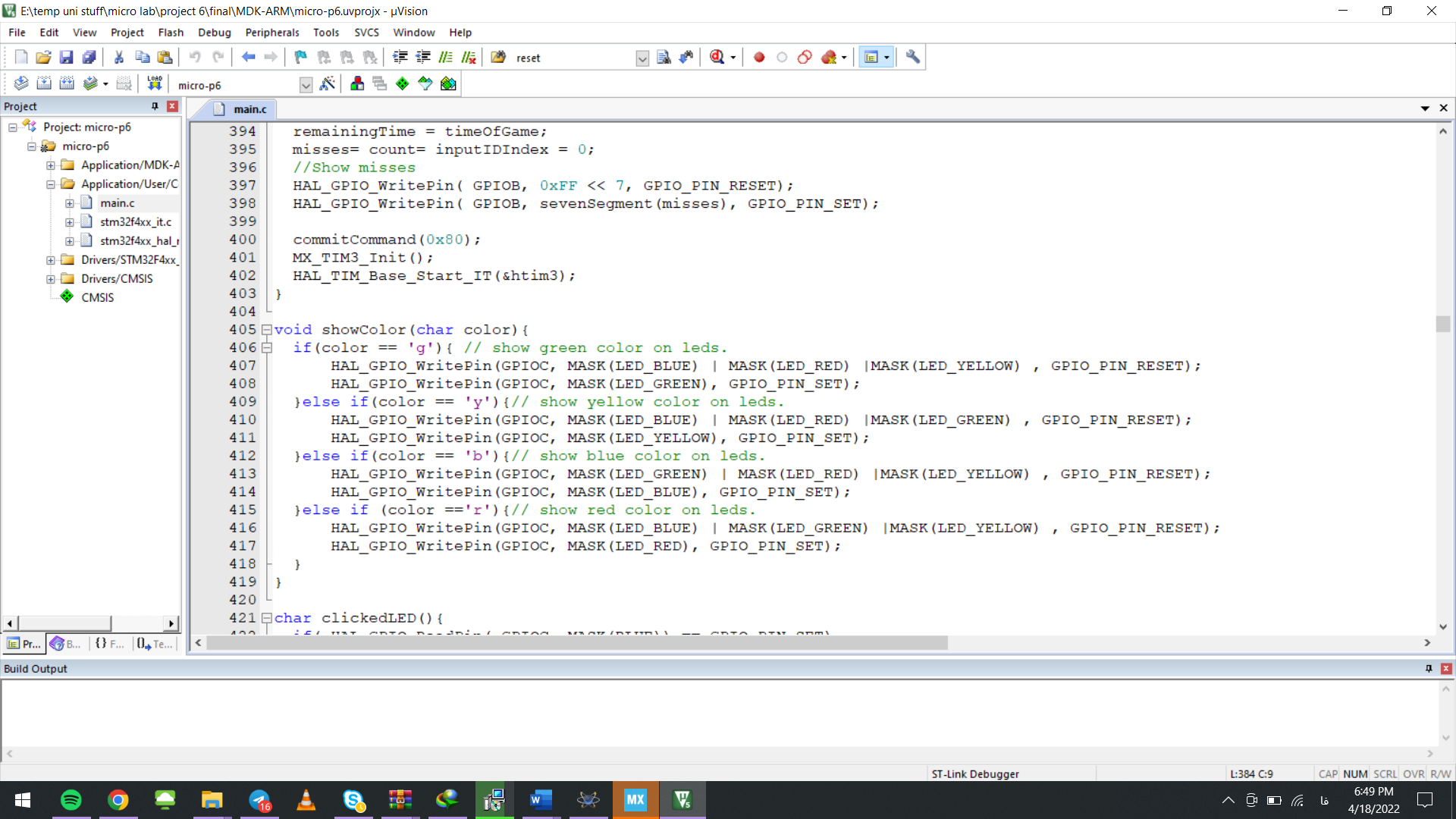
تابع مربوط به آپدیت سون سگمنت:



این تابع برنامه و متغیرها را ریست میکند و درواقع شرایط را به حالت اولیه میبرد:



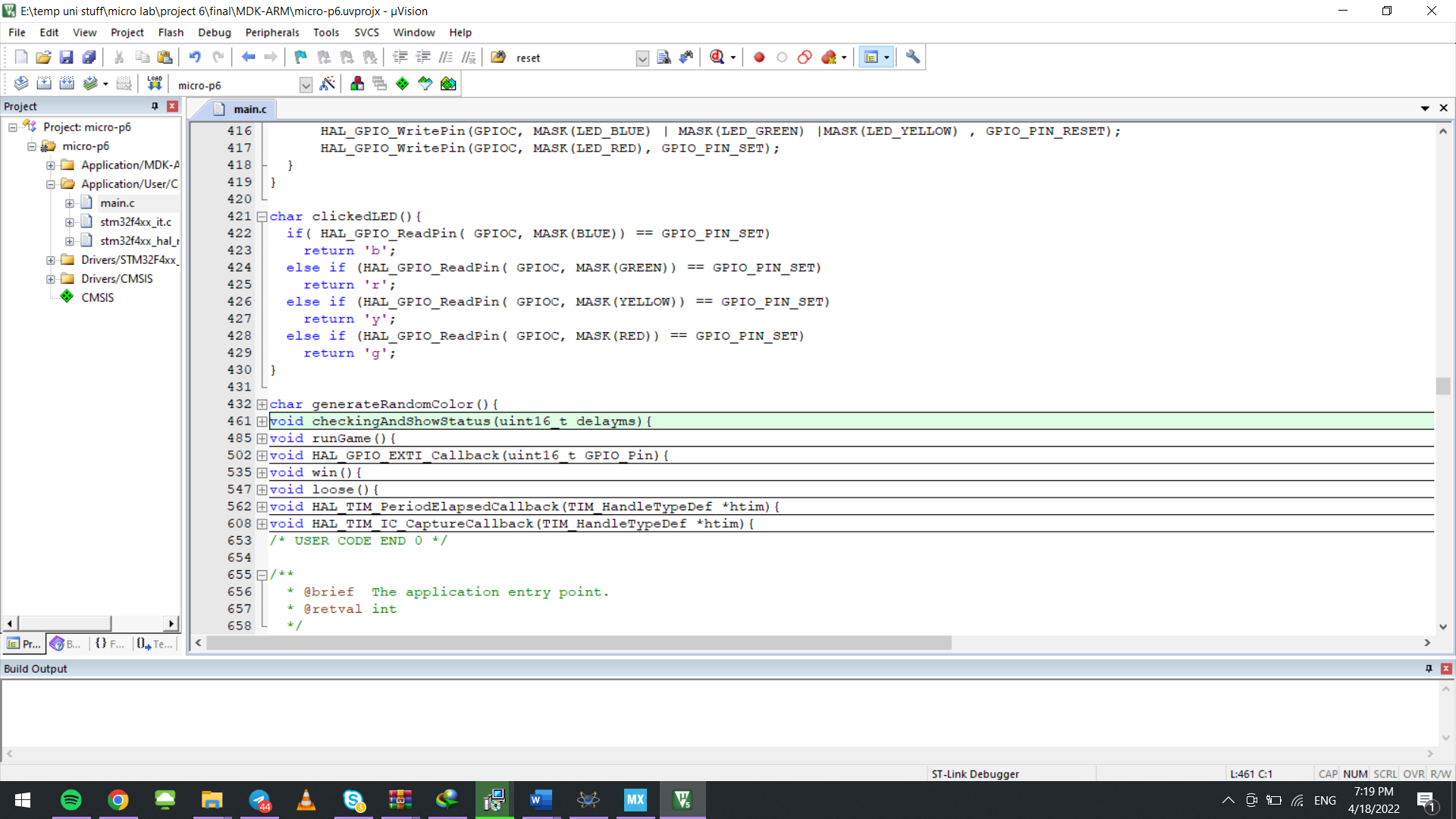
در این تابع ال ای دی با رنگ فعلی روشن و بقیه ال ای دی ها خاموش میشوند:



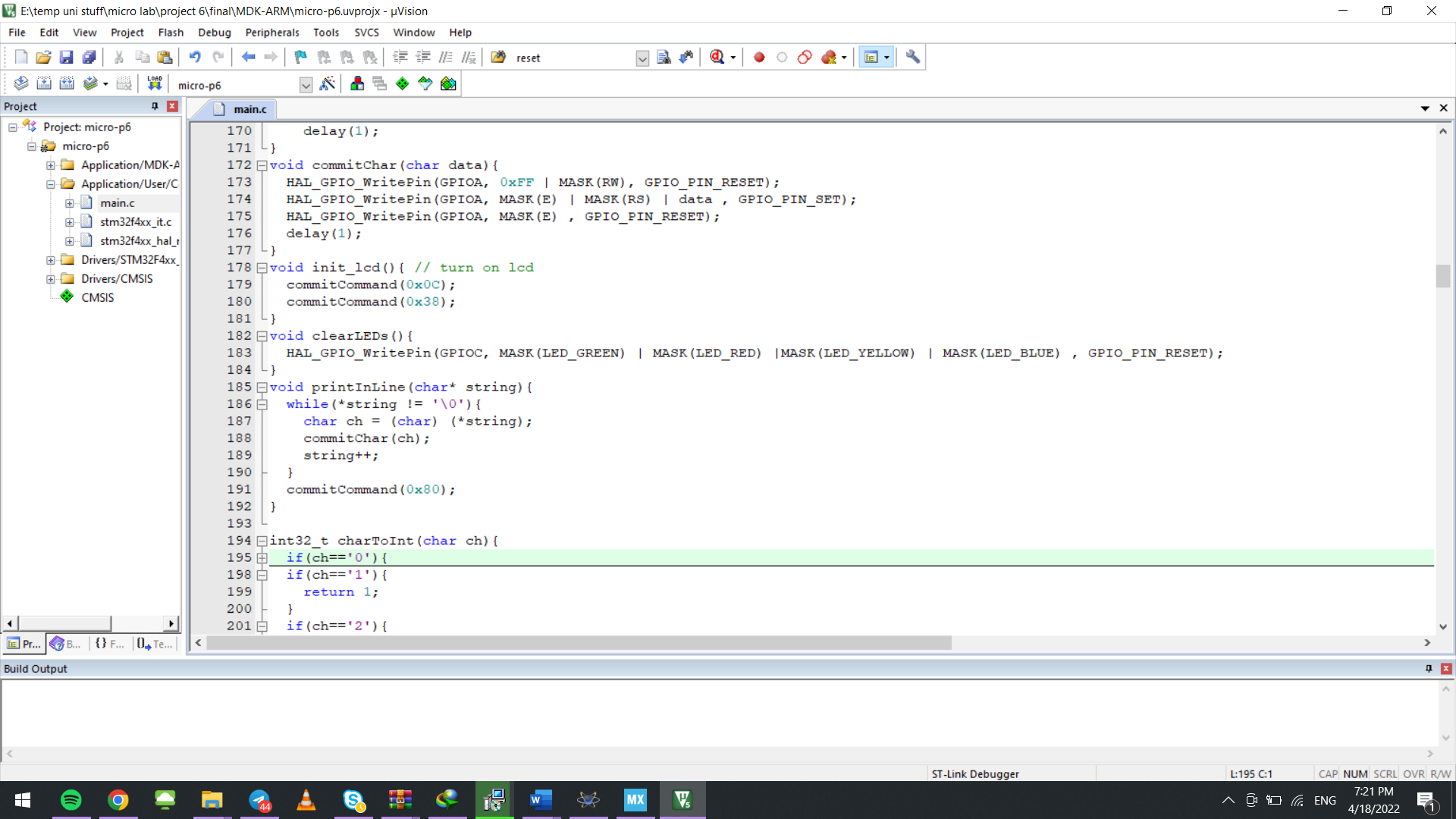
این تابع یک کارکتر رندوم برای رنگ ال ای دی به ما برمیگرداند:



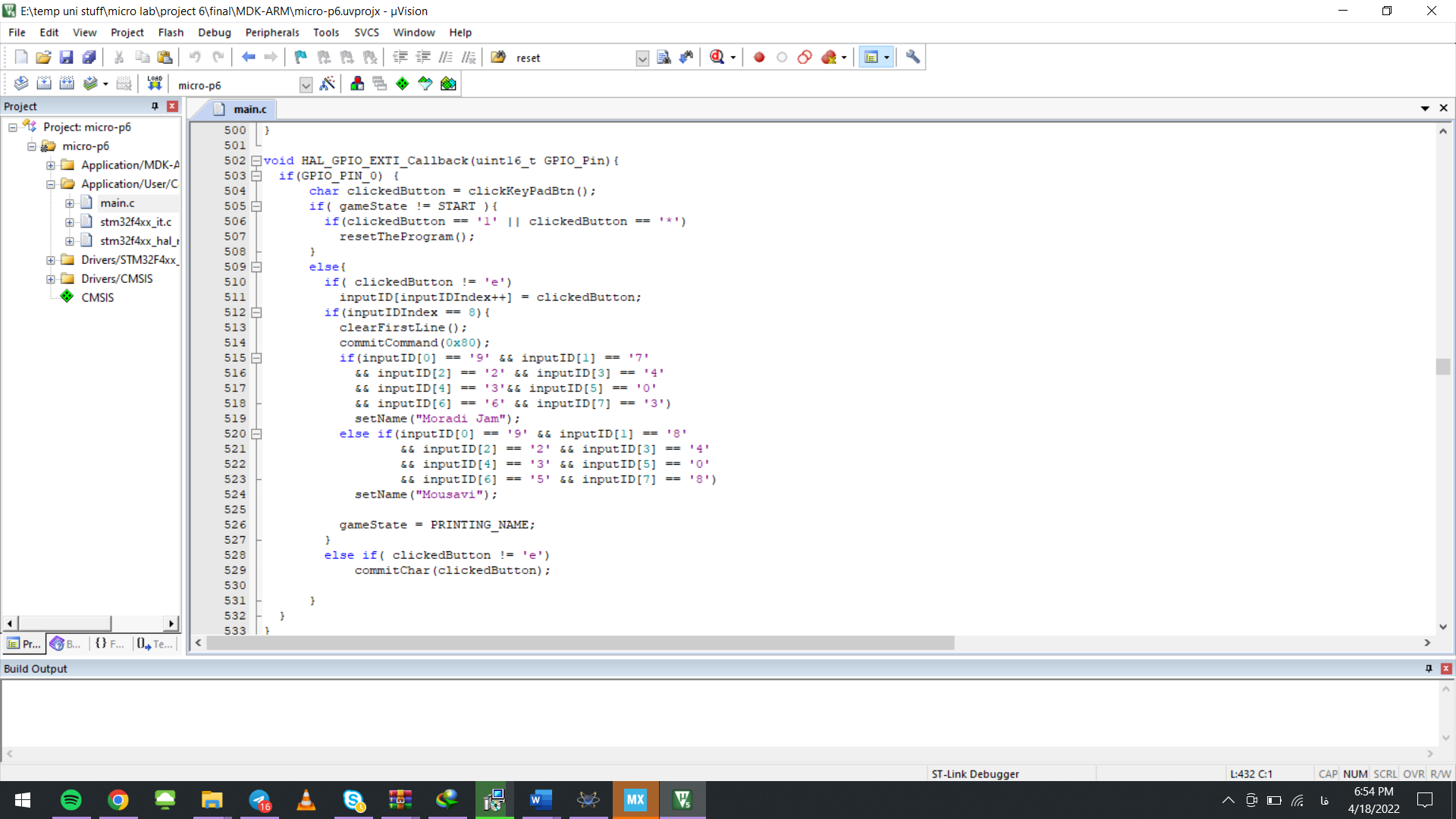
این تابع بررسی میکند که کاربر کلید کدام رنگ فشرده است:



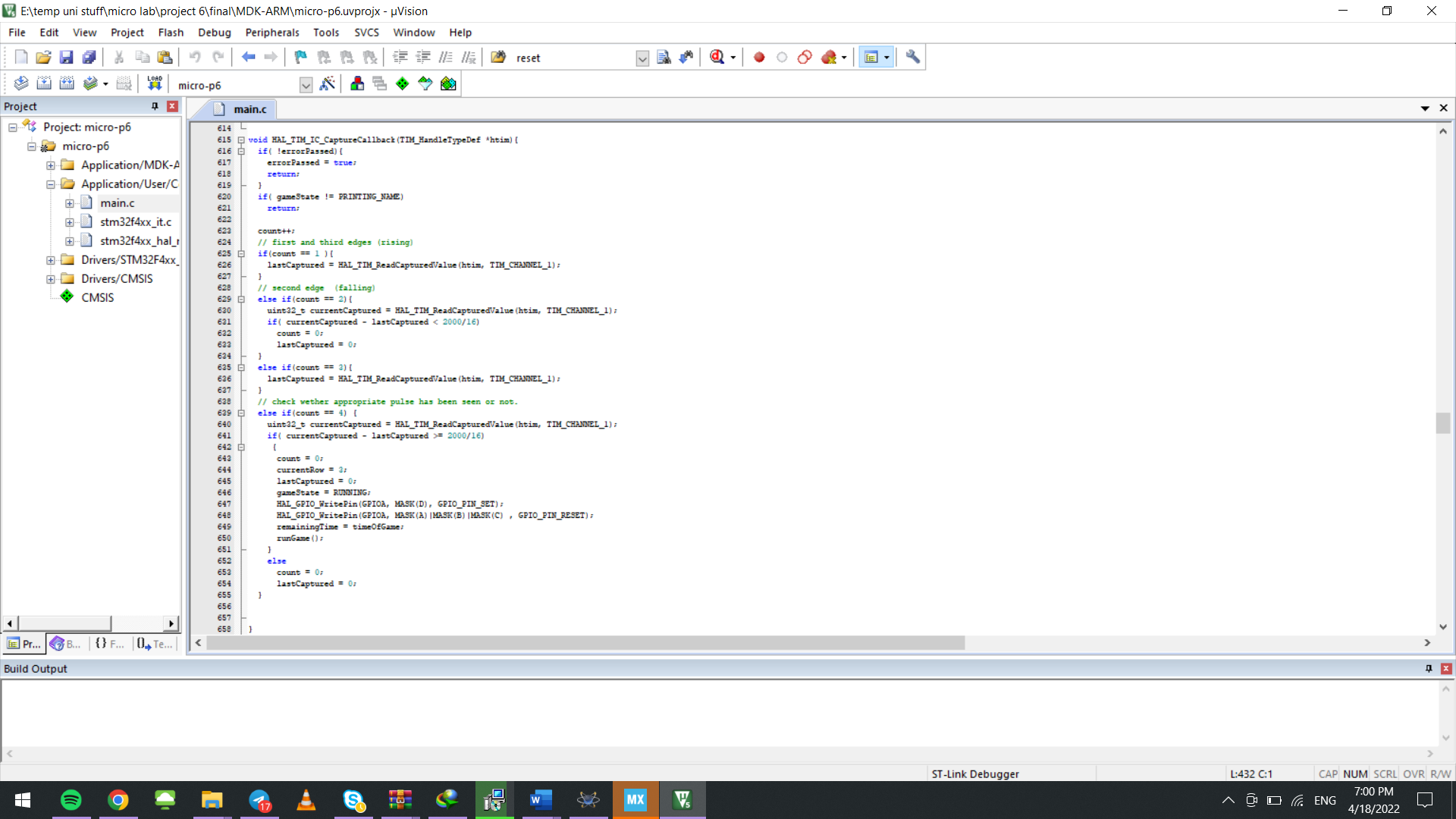
این تابع همه LED هارا خاموش میکند:



اینتراپت درصورت فشردن کیپد یا Reset اجرا میشود که در حالت START صرفا ورودی­هارا نشان میدهیم وقتی ۸ کاراکتر وارد شد، نام متناظر با آن را متعاقب ورود به حالت PRINTING\_NAME به­صورت چشمک زن نشان می­دهیم:

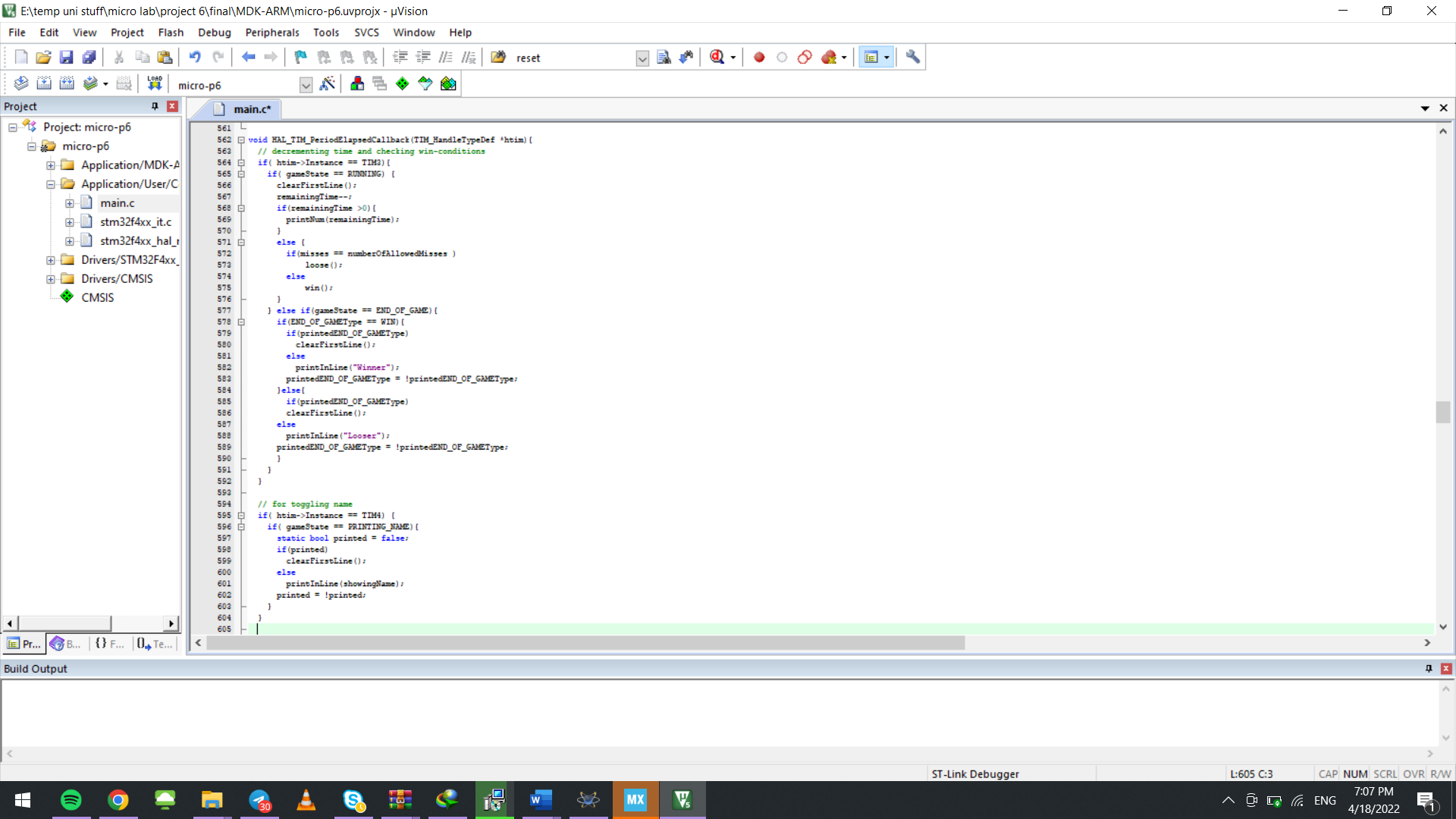


در این تابع زمانیکه ۴ لبه (پایین رونده و بالارونده) دیده شود، به­طوریکه اگر فاصله بین لبه­های بالارونده بیش از ۲ ثانیه باشد، سیستم وارد حالت RUNNING شده و بازی اجرا می­شود (runGame ) :

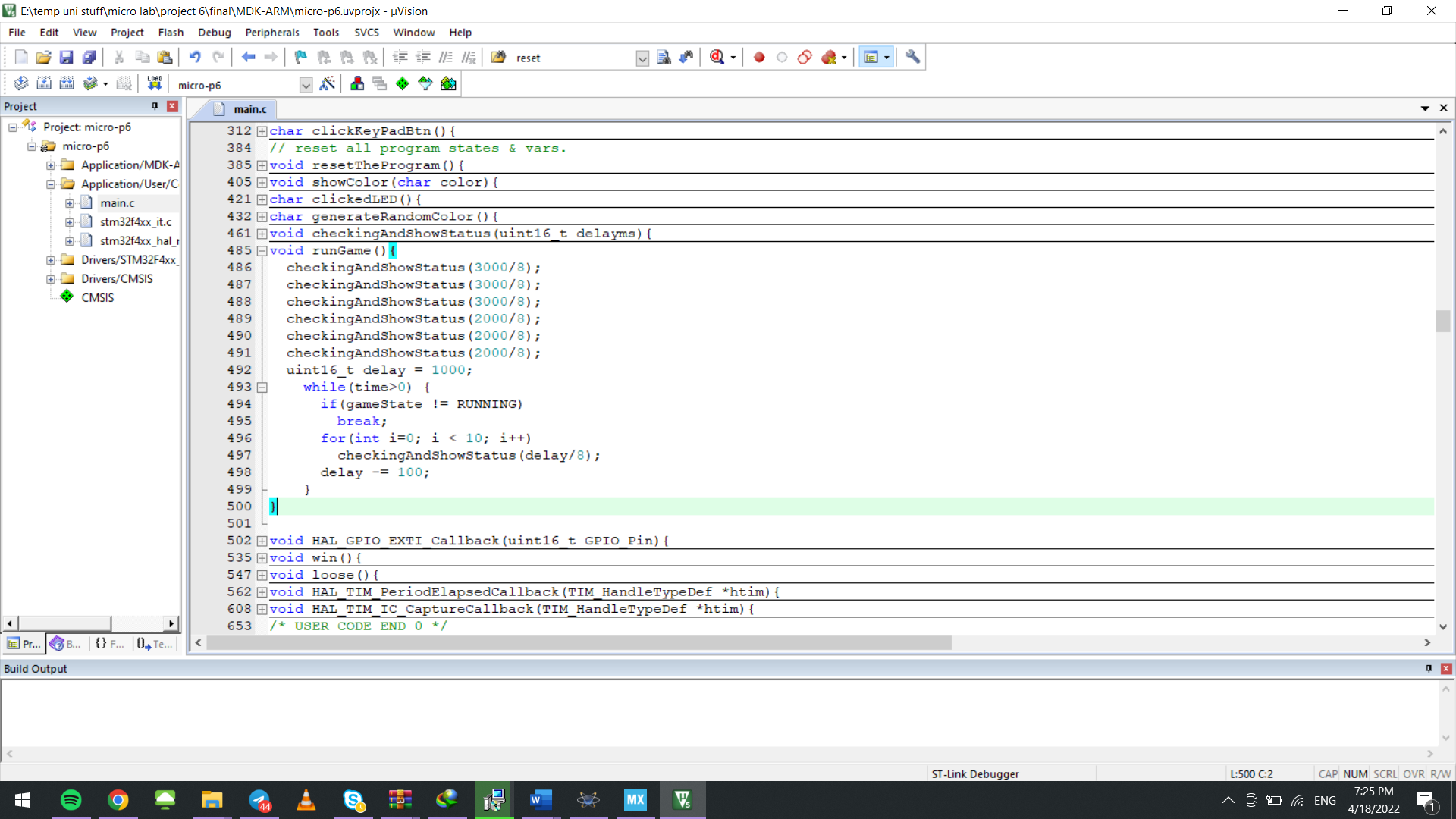


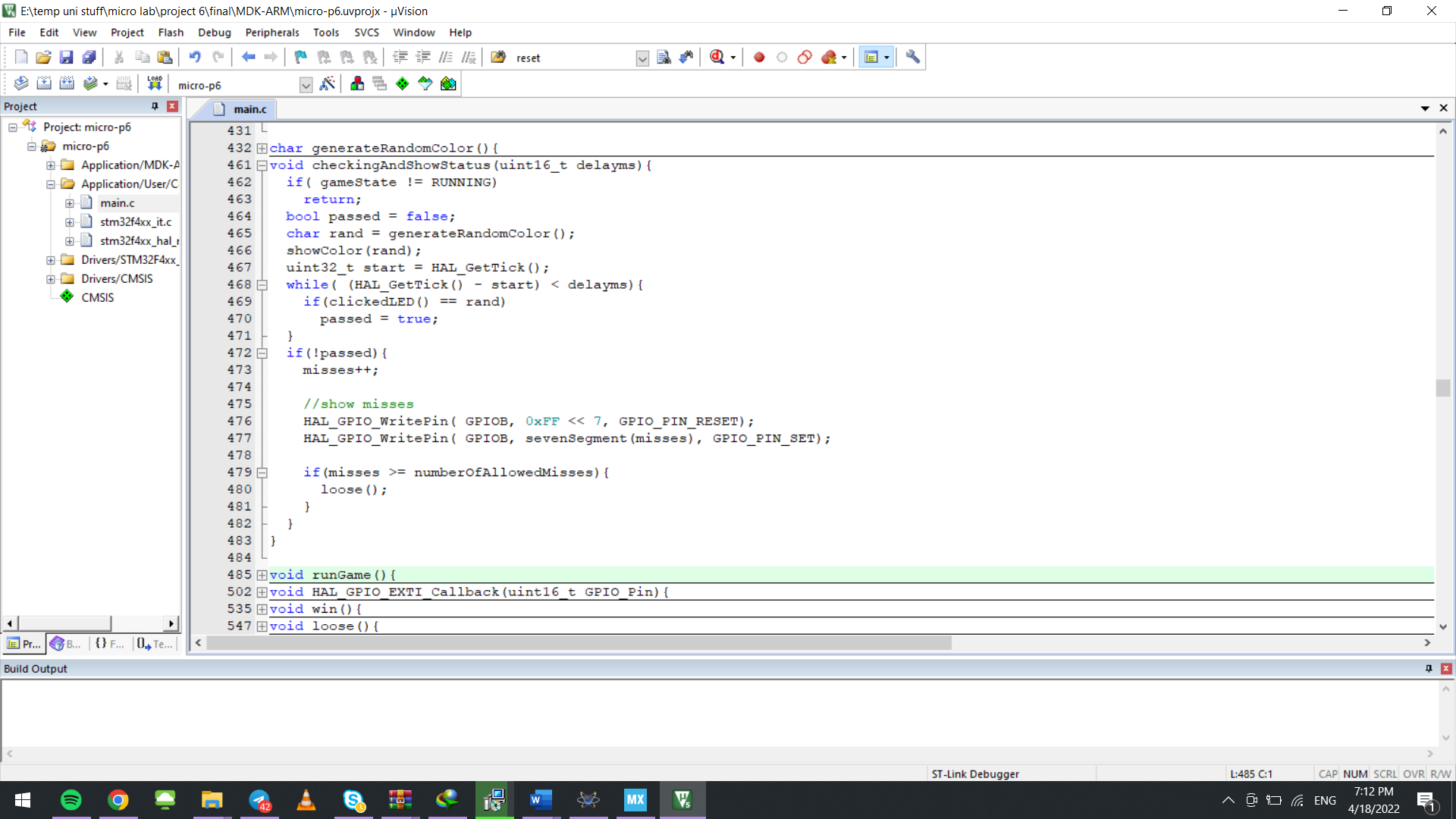
تایمر 4 که تایمر Base است مسئولیت چشمک زدن نام را دارد.

تایمر 3 نیز در حالت Base اگر سیستم درحالت RUNNING باشد و تایم تمام نشده باشد، در هرثانیه آنرا Decrement میکند و نشان می­دهد. اگر تایم تمام شود و تعداد خطاها کمتر از آستانه تعریف شده باشد؛ بازی با برد و در غیر اینصورت با باخت خاتمه یافته است:



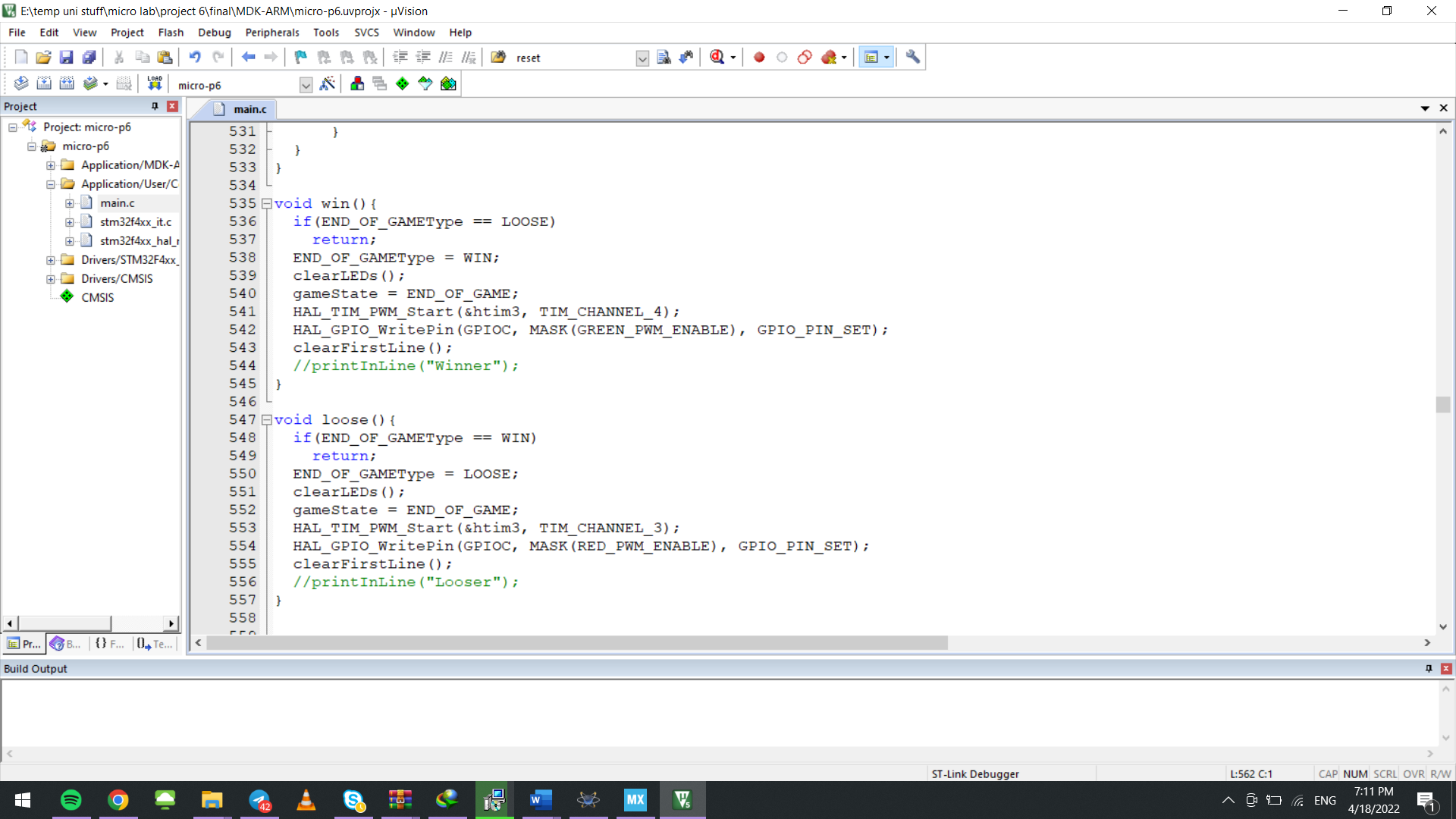
با توجه به دستورکار، در تابع runGame ۳ مرتبه ۳ ثانیه­ای LED روشن و ورودی چک می­شود و سپس ۳ مرتبه ۲ثانیه­ای و پس آن طبق دستورکار ۱۰ مرتبه ۱۰ مرتبه با کسر ۱۰۰ میلی­ثانیه روال انجام می­شود. در تابع checkingAndShowStatus نیز یک رنگ رندوم ایجاد و LED‌ مربوطه را روشن میکنیم و زمان مشخصی مرتبا ورودی کاربر را چک میکنیم درصورتیکه کاربر کلید درست در تایم مربوطه وارد نکند misses اضافه میشود، اگر به تعداد مجاز خطا برسد، loose کال می­شود:





در تابع loose() اگر بازی به حالت WIN باشد برای جلوگیری از اختلال زمانی loose نبایستی اجرا شود. در غیر اینصورت این تابع حالت نتیجه را LOOSE و حالت سیستم را در حالت END\_OF\_GAME میگذارد. سپس PWM مربوط به باخت (قرمز) را فعال میکن و عبارت لوزر را چاپ میکند.

متقارن همین تابع WIN وجود دارد.



نتیجه کار در پروتئوس:

