



آزمایش مقدماتی

آزمایشگاه ریزپردازنده

فیض سال اول ۱۴۰۱ - ۱۴۰۲

هدف

هدف از این آزمایش آشنایی و مرور امکانات ابزارهای Proteus و شیوه راه اندازی آنها است.

پیش نیاز و مطالعه

نصب ابزارهای Keil و Proteus

مشاهده فیلم های آموزشی از [اینجا](#)

[/https://melec.ir/proteus-software-tutorials-setup-new-project](https://melec.ir/proteus-software-tutorials-setup-new-project)

[/https://irenx.ir/category/proteus](https://irenx.ir/category/proteus)

[/https://sariasan.com/featured/proteus-full-free-lessions-2](https://sariasan.com/featured/proteus-full-free-lessions-2)

سوالات تحلیلی

1. در چهار پاراگراف ویژگی های میکرو کامپیوتر، میکرو پروسسور و میکرو کنترلر را توضیح داده و با یکدیگر مقایسه نمایید.
2. در سه پاراگراف پردازنده های CISC و RISC را مقایسه نمایید.

دستور کار

بهره گیری از نرم افزار Proteus برای شبیه سازی یک مدار ساده مبتنی بر پردازنده با کد داده شده و بستن آن روی برد بورد

ابزار و تجهیزات مورد نیاز :

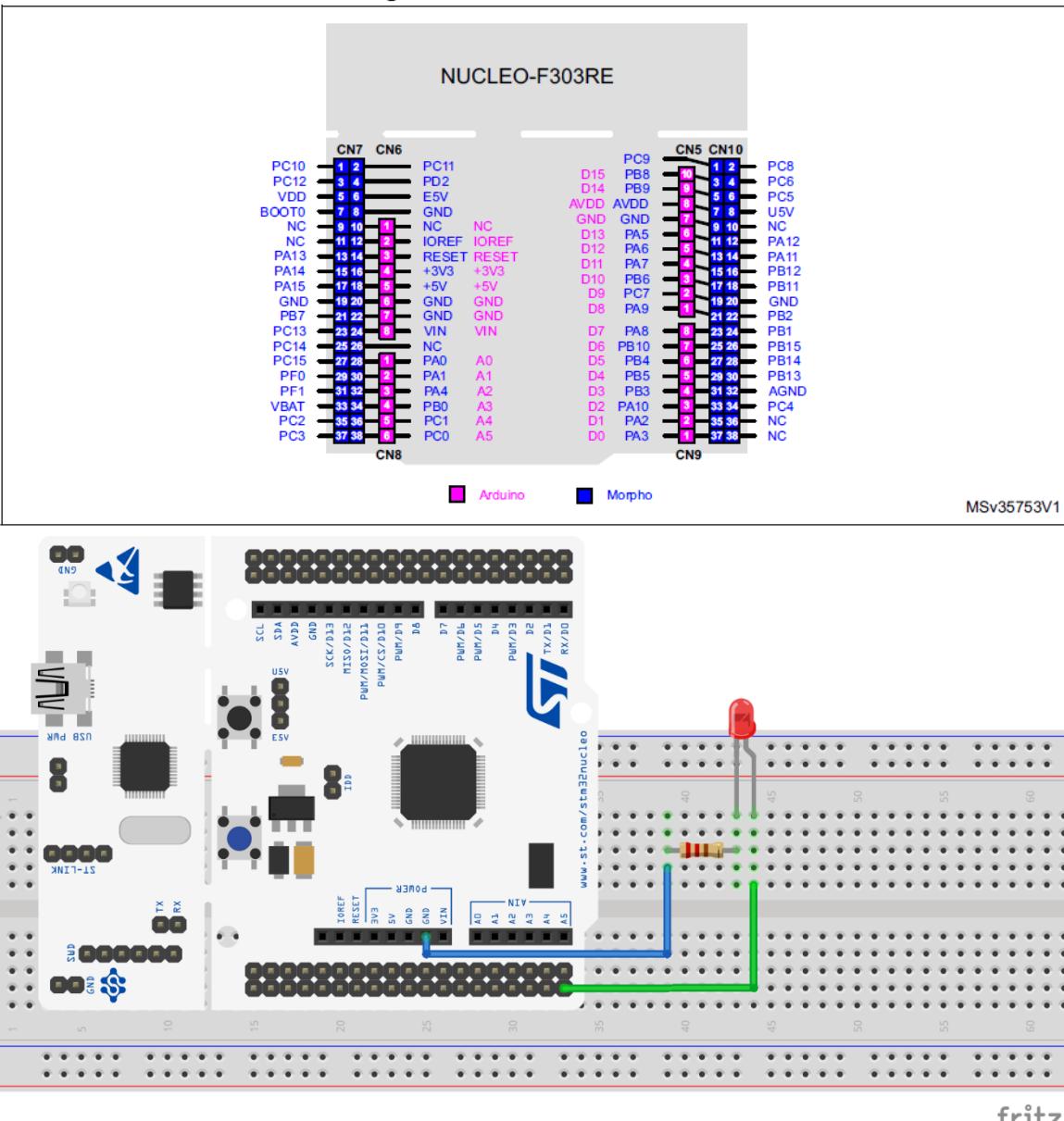
- 1 دانلود نرم افزار پروتئوس ویرایش 8.13 از [اینجا](#)
- 2 دانلود و نصب برنامه Keil uvision از [اینجا](#)
- 3 دانلود و نصب برنامه STM32CubeMX از [اینجا](#)
- 4 برد NUCLEO-F401RE
- 5 برد بورد
- 6 دیود نوری (LED)
- 7 مقاومت
- 8 کابل رابط برد و کامپیوتر

روش انجام آزمایش:

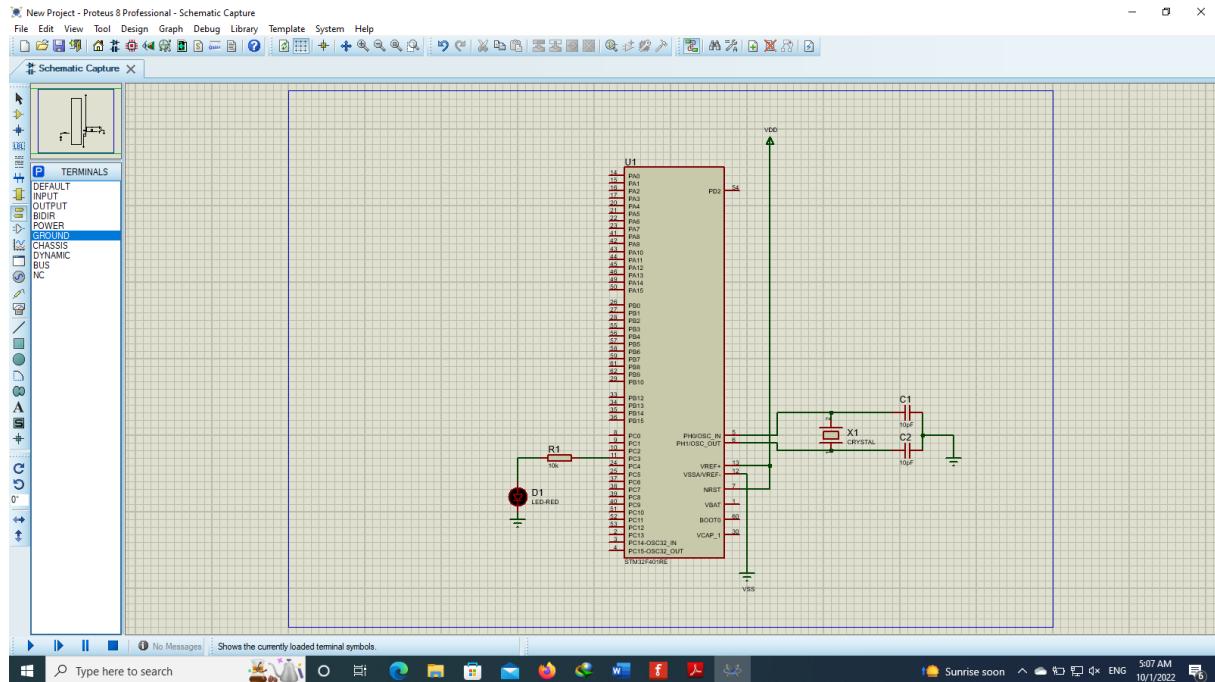
برای انجام این آزمایش ابتدا نرم افزار های مورد نیاز را دانلود و نصب کنید.

گام اول: سخت افزار مدار مورد نیاز را طراحی کنید. برای این کار می توانید از نرم افزار Fritzing استفاده نمایید.

Figure 16. NUCLEO-F303RE



گام دوم: شبیه سازی مدار در پروتئوس
برای انجام اینکار نرم افزار پروتئوس را اجرا کنید. و مدار زیر را در آن رسم کنید (برای راهنمایی بیشتر از پیوست شماره 1 استفاده شود)

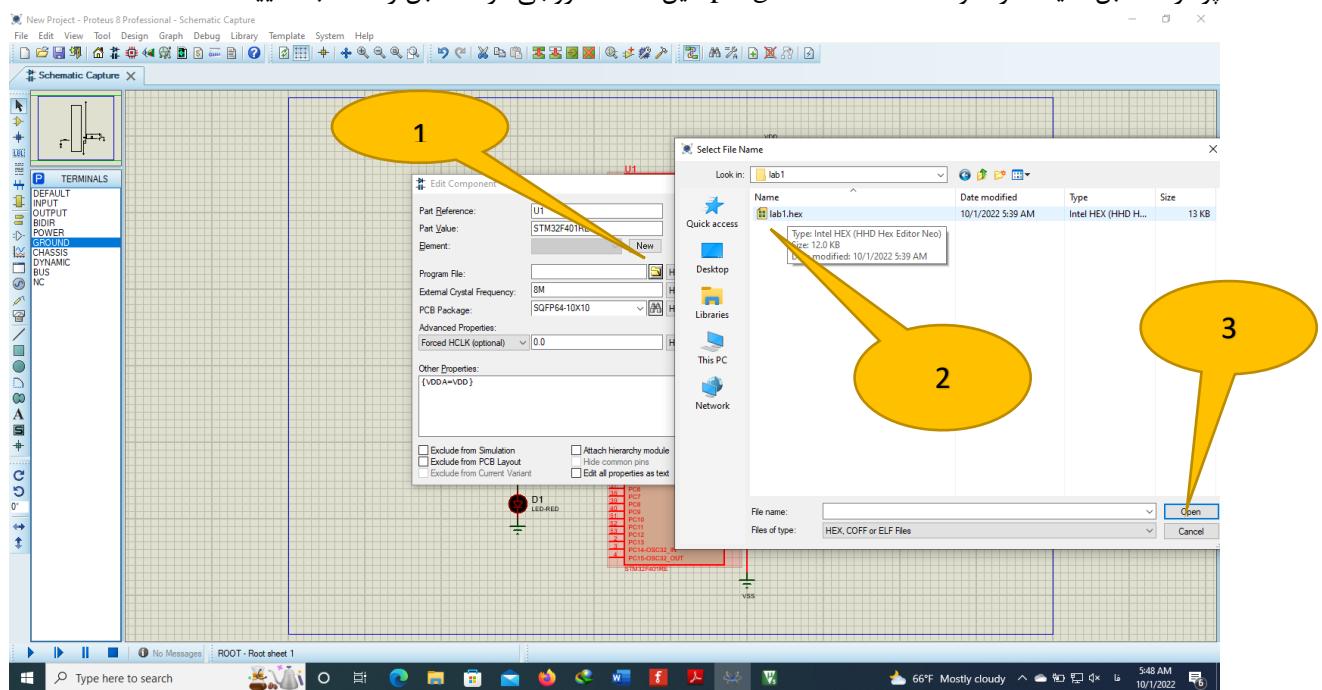


گام سوم : در برنامه STM32CubeMX سخت افزار سیستم را پیکربندی کنید و تنظیمات اولیه نرم افزار keil را خروجی بگیرید. (برای راهنمایی بیشتر از پیوست شماره 2 استفاده شود)

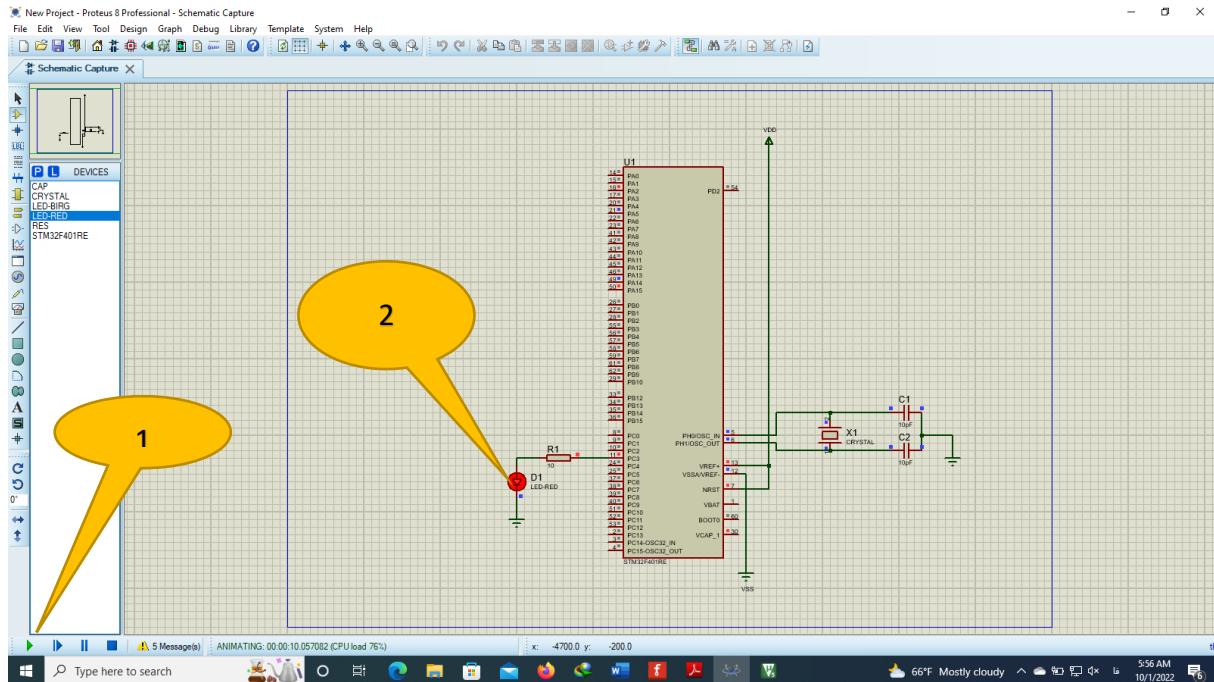
گام چهارم : در نرم افزار Keil فایل خروجی گام قبل را باز کرده و قطعه کد زیر را به حلقه اجرایی main تابع اضافه کنید (برای راهنمایی بستر از پیوست 3 استفاده شود)

```
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_3);
HAL_Delay(1000);
```

گام پنجم : فایل HEX خروجی نرم افزار keil را بر روی پردازنده در نرم افزار Proteus بارگزاری کنید. برای اینکار بر روی پردازنده دابل کلیک کرده و قسمت program file فایل Hex خروجی مرحله قبل را انتخاب نمایید.



برای انجام شبیه سازی دکمه run را کلیک کنید و نتیجه کار را ببینید.



موارد تحويل دادنی

- سوالات تحلیلی را به صورت کامل پاسخ و تحويل دهید.
- سورس کد تمام بخش‌های ذکر شده را به صورت کامل تحويل دهید. برای خوانایی بیشتر حتماً بایست بخش‌های مختلف کد کامنت‌گذاری شود.
- گزارشی کامل و واضح از بخش‌های مختلف انجام شده در طی اجرای دستور کار تحويل شود. اگر در بخشی قطعه کدی توضیح داده می‌شود حتماً کپی آن بخش از کد در گزارش آورده شود.

تذکرهای مهم

- در صورتی که مدار پیاده‌سازی شده یا هر قسمی را از منبعی استخراج نموده‌اید، حتماً ارجاع دهید. در غیراینصورت، به دلیل تخلف، نمره‌ای تعلق نخواهد گرفت.

نکات حائز اهمیت

- بخش‌های مختلفی که باید تحويل داده شوند همگی در یک فایل فشرده باشند و نام فایل فشرده حتماً به فرمت زیر باشد:

<گروه درسی-نام-نام خانوادگی-شماره دانشجویی>

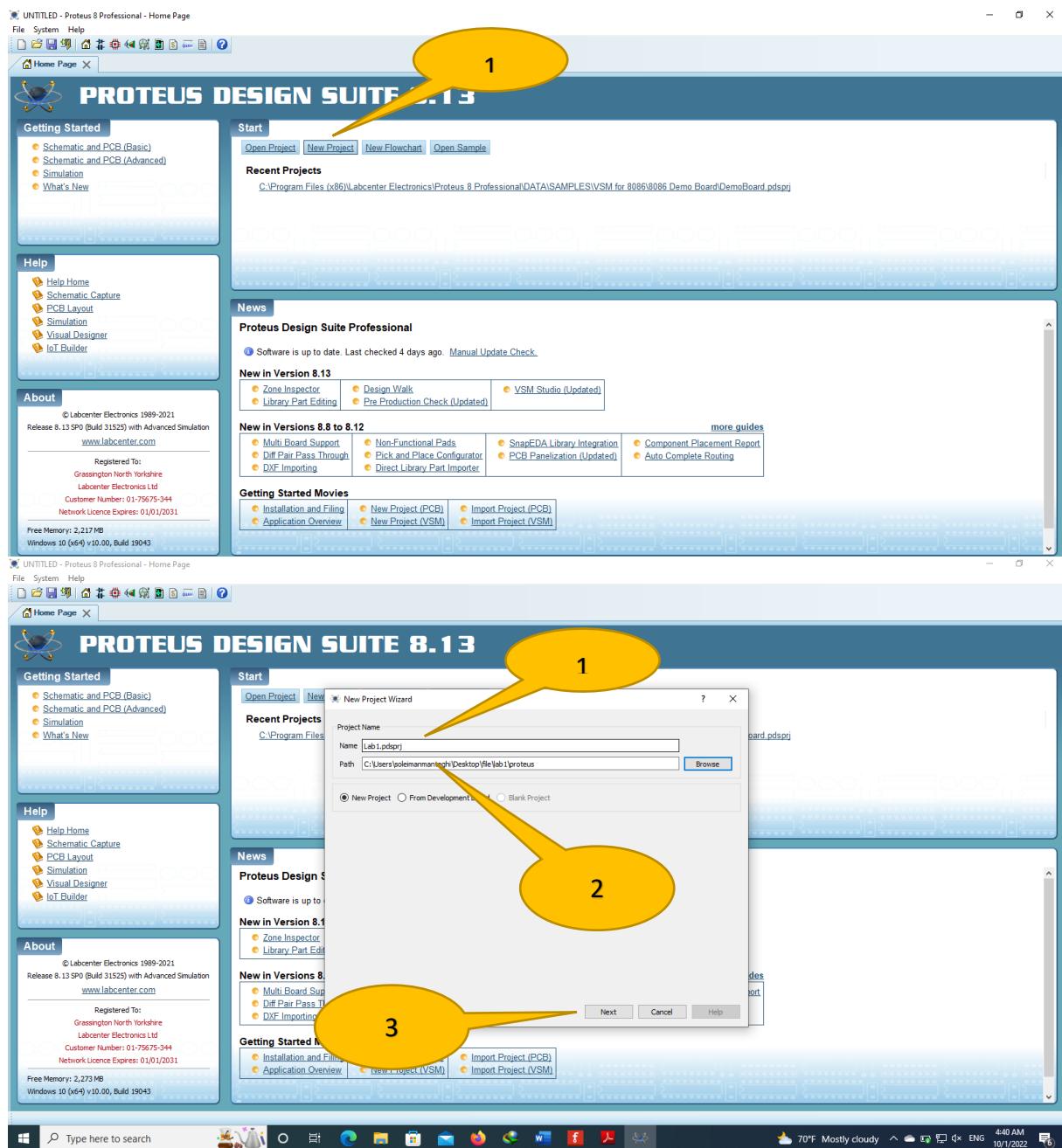
- به ازای هر روز تاخیر، روز اول 15٪، روز دوم 25٪ و روزهای سوم و چهارم 30٪ از نمره کسر خواهد شد و در روز پنجم نمره‌ای تخصیص نمی‌گردد.
- حتماً در گزارش نام اعضاء، شماره دانشجویی و گروه درسی ذکر گردد.
- آزمایش‌های ریزپردازnde به صورت گروههای دو نفره انجام داده شده و تحويل می‌شوند.
- نکته مهم این است تمامی افراد گروه باید به همه جوانب و جزئیات آزمایش‌ها مسلط باشند که این نکته توسط مدرسین هنگام تحويل به دقت بررسی خواهد شد.

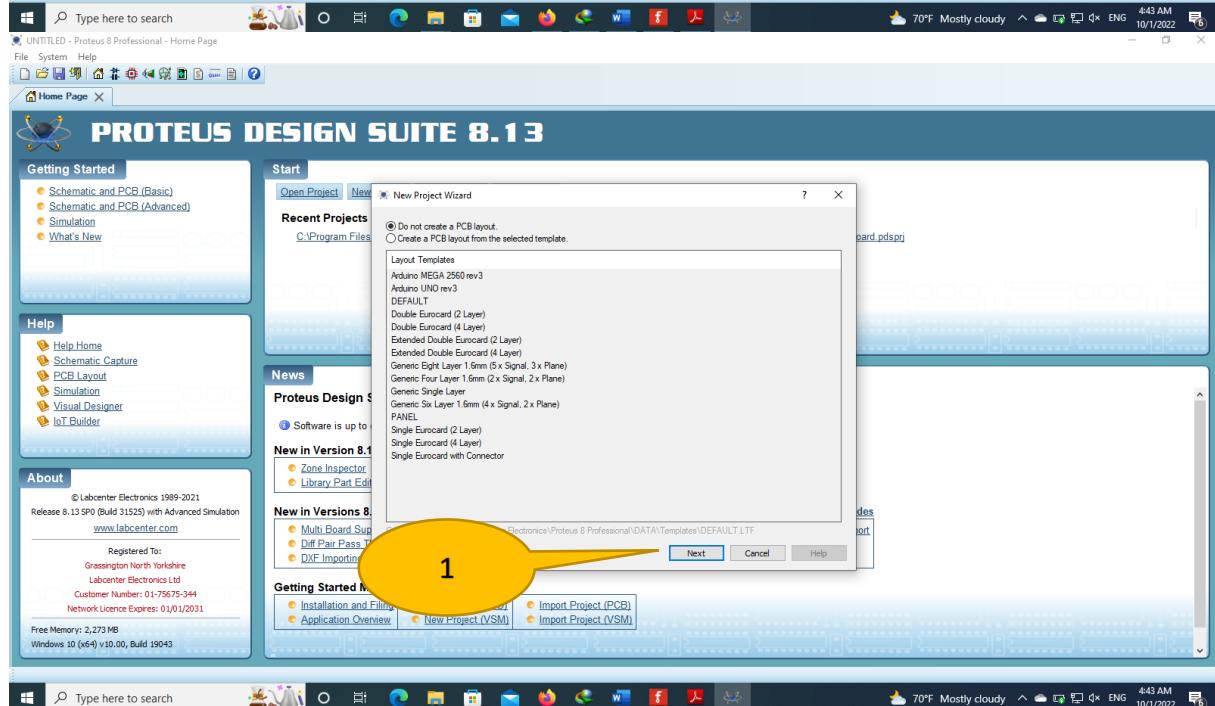
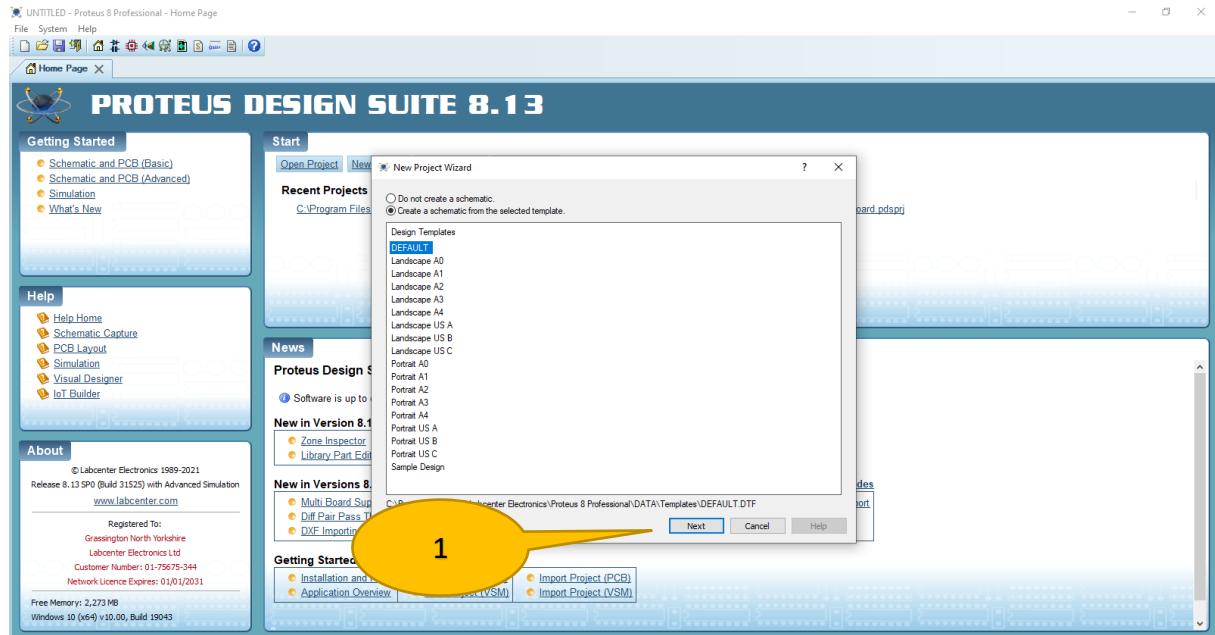
- هر گروه باید به صورت مجزا آزمایش را انجام دهد و کپی نتایج آزمایش گروه‌های دیگر تخلف است.
- به منظور ایجاد شرایط یکسان برای تمامی گروه‌ها و فاصله داشتن زمان آپلود و تحويل، به هنگام تحويل، اعضای گروه، در همان زمان پاسخ آزمایش خود را از درس‌افزار دانلود کرده و روی سیستم خود تحويل می‌دهند.

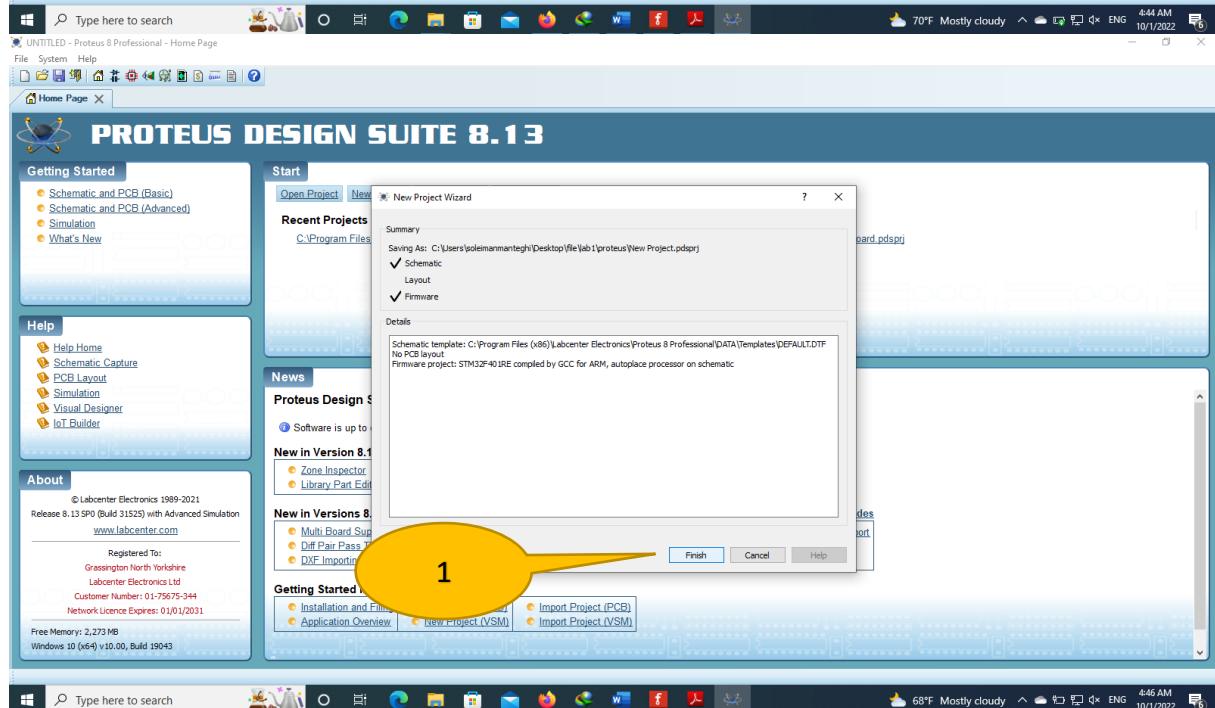
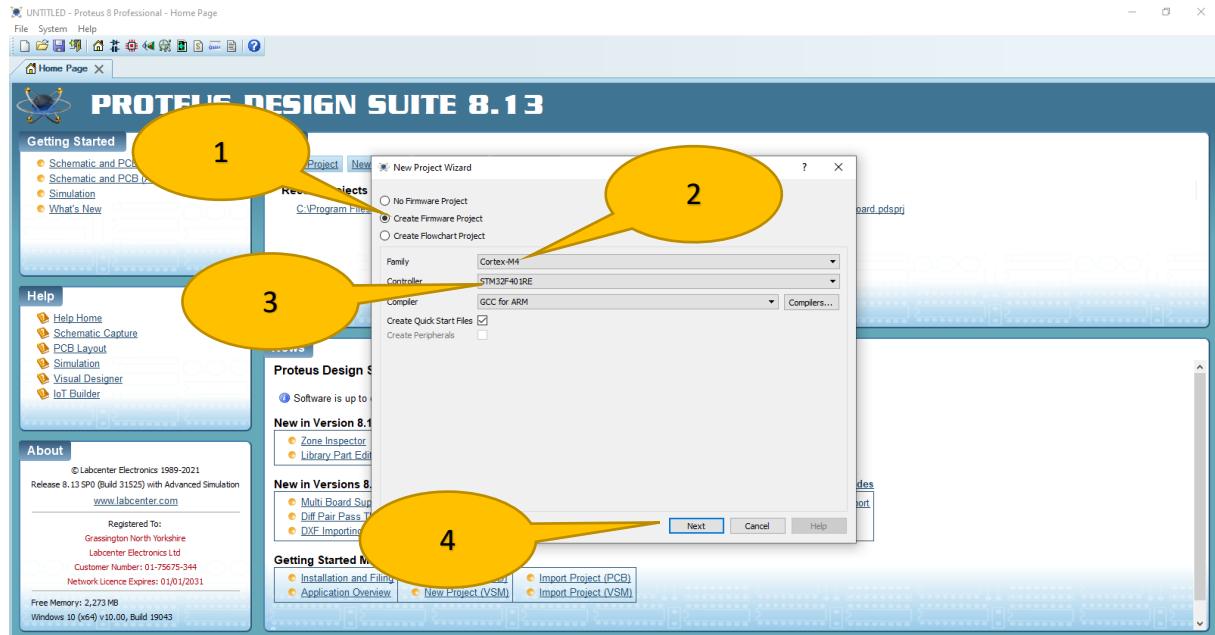
موفق باشد

گروه آزمایشگاه‌های ریزپردازند

پیوست 1 _ نرم افزار پروتئوس







New Project - Proteus 8 Professional - Source Code

File Project Build Edit Debug System Help

Schematic Capture Source Code

Projects STM32F401RE(U1)

Source Files main.c startup_stm32f401xx.s system_stm324xx.c

Header Files core_cm4.h core_cmfunc.h core_cmisr.h core_cmmd.h stm324xx.h stm324xx_i.h system_stm324xx.h

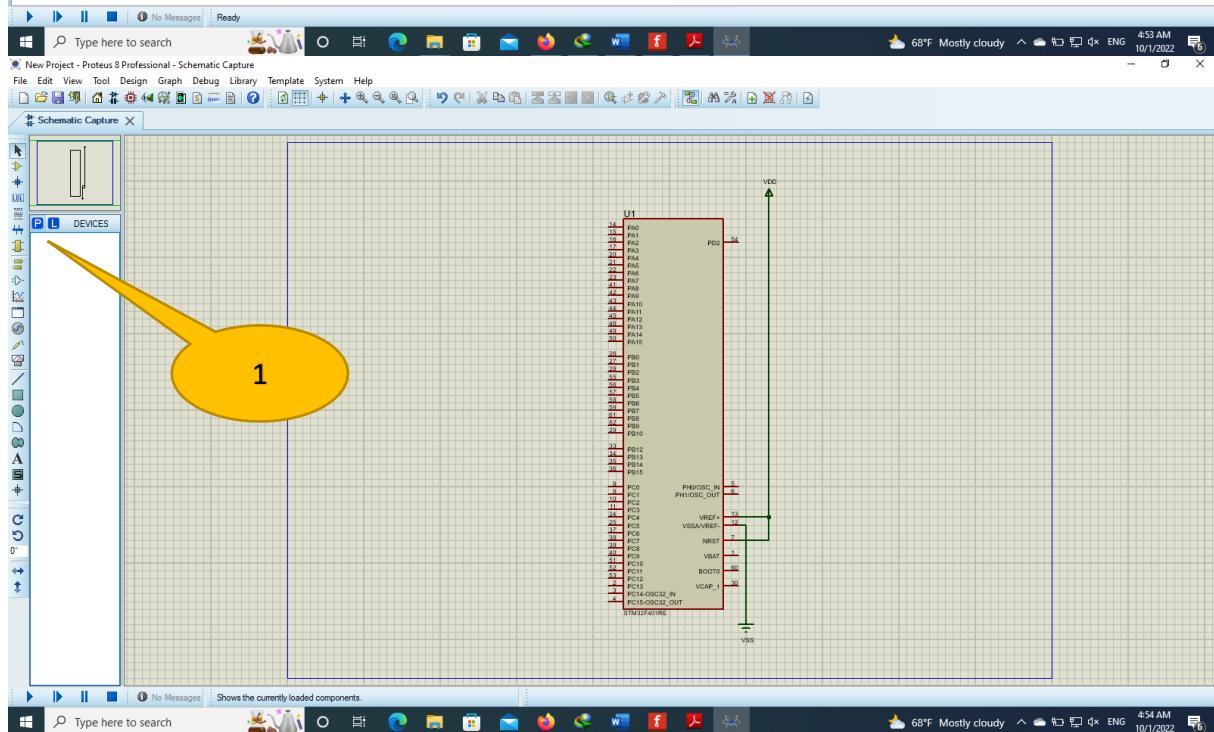
Linker Script Files STM32F401xE.ld

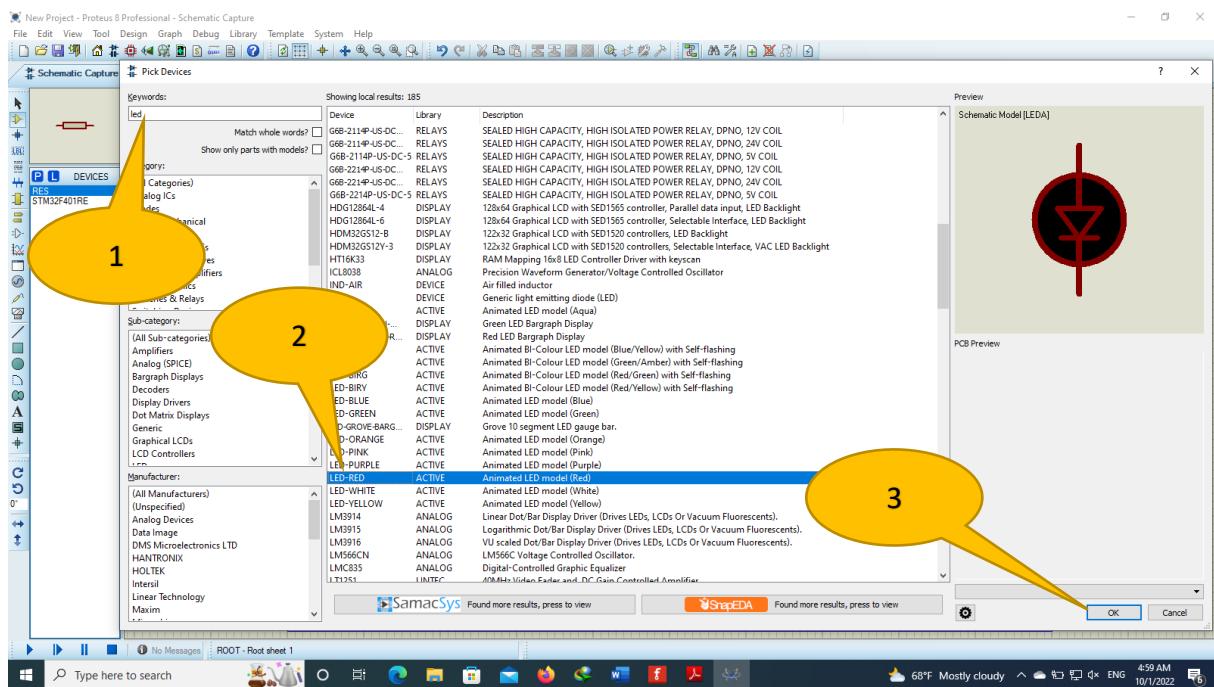
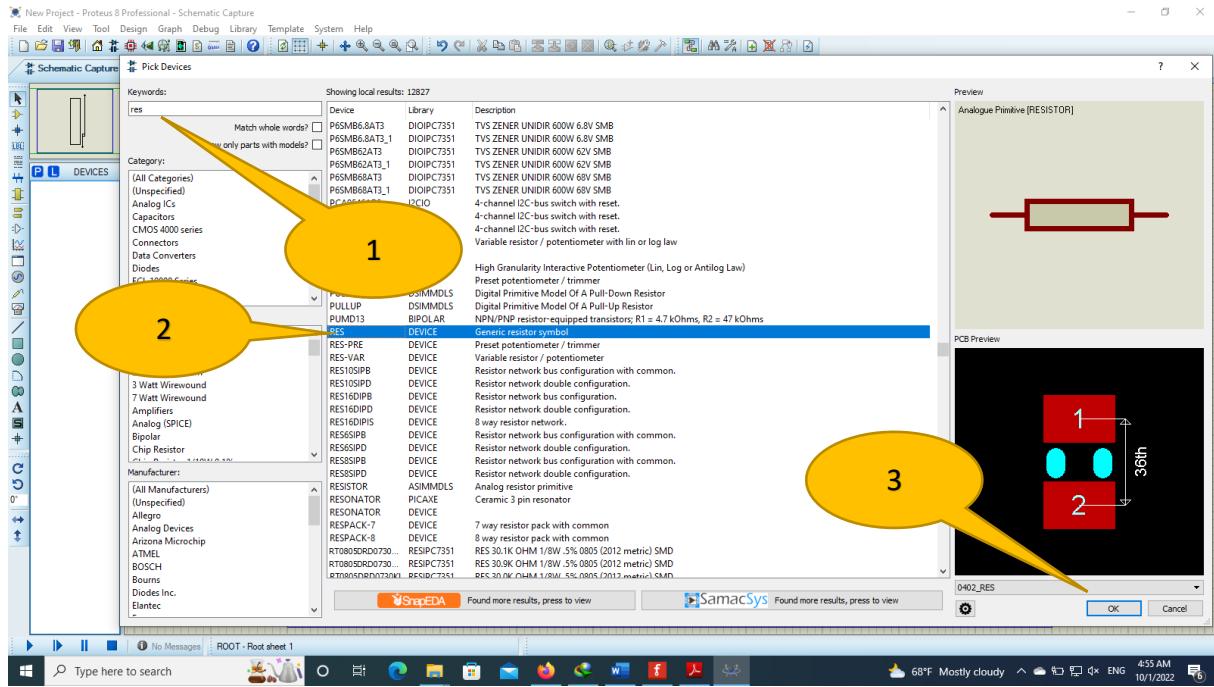
main.c

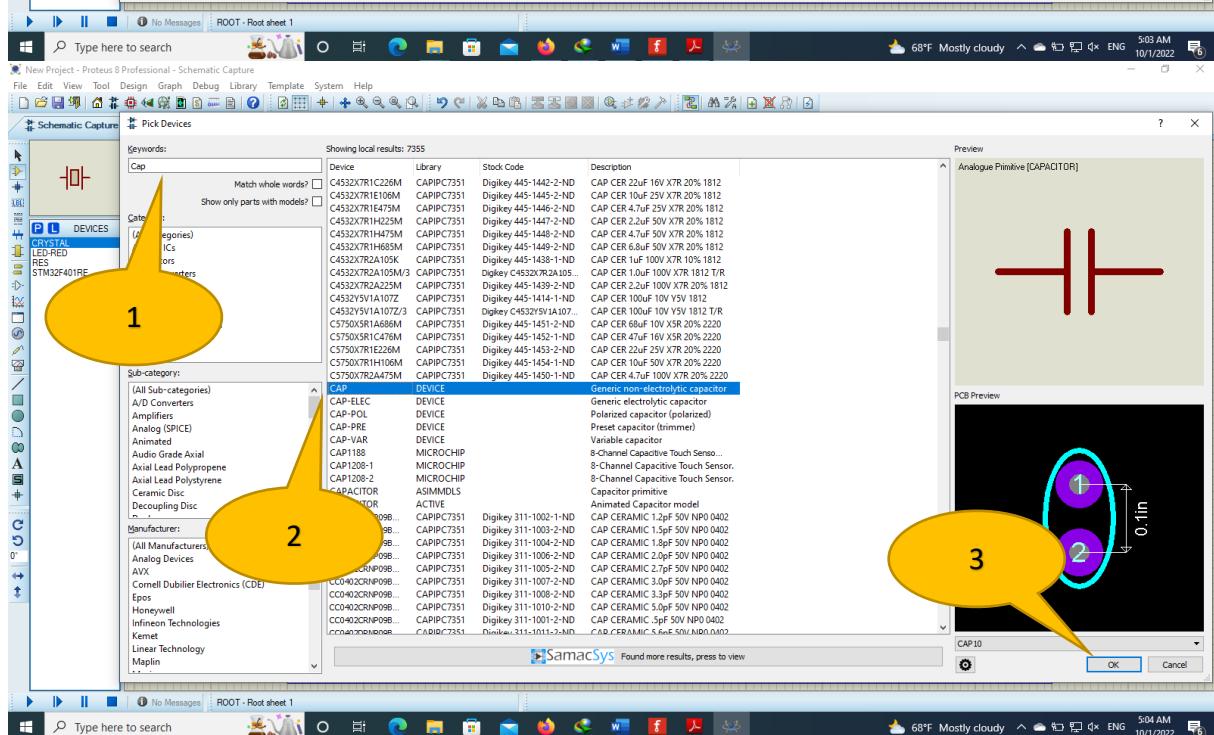
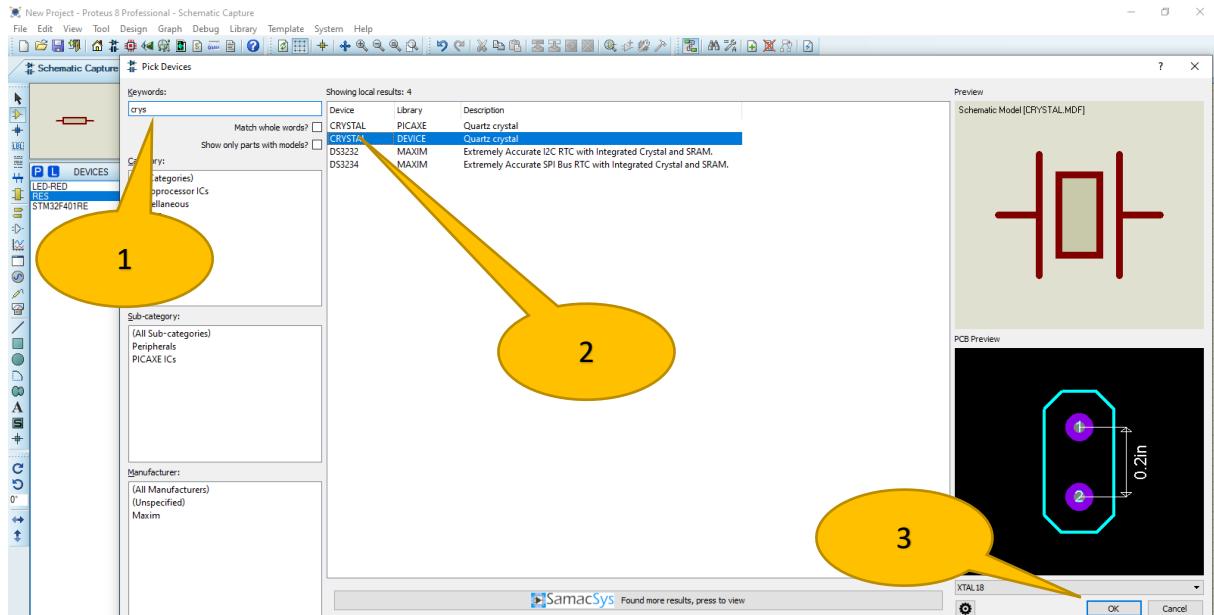
```
1 /* File generated by New Project wizard
2 *
3 * Created: 1 Oct 1 2022
4 * Processor: STM32F401RE
5 * Compiler: GCC
6 */
7
8 #include <stm32f4xx.h>
9
10 int main (void)
11 {
12     // Write your code
13     while (1)
14     {
15         ;
16     }
17     return 0;
}
```

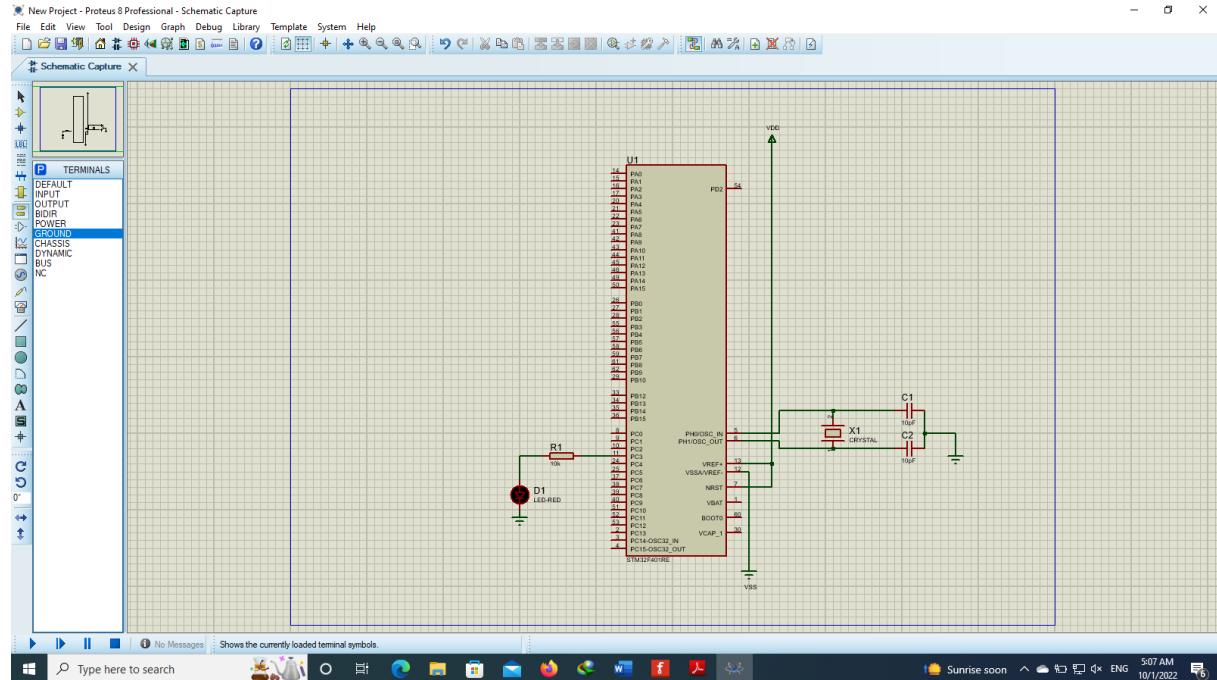
VSM Studio Output

Firmware file missing from the project directory: STM32F401RE_1\Debug\Debug.elf

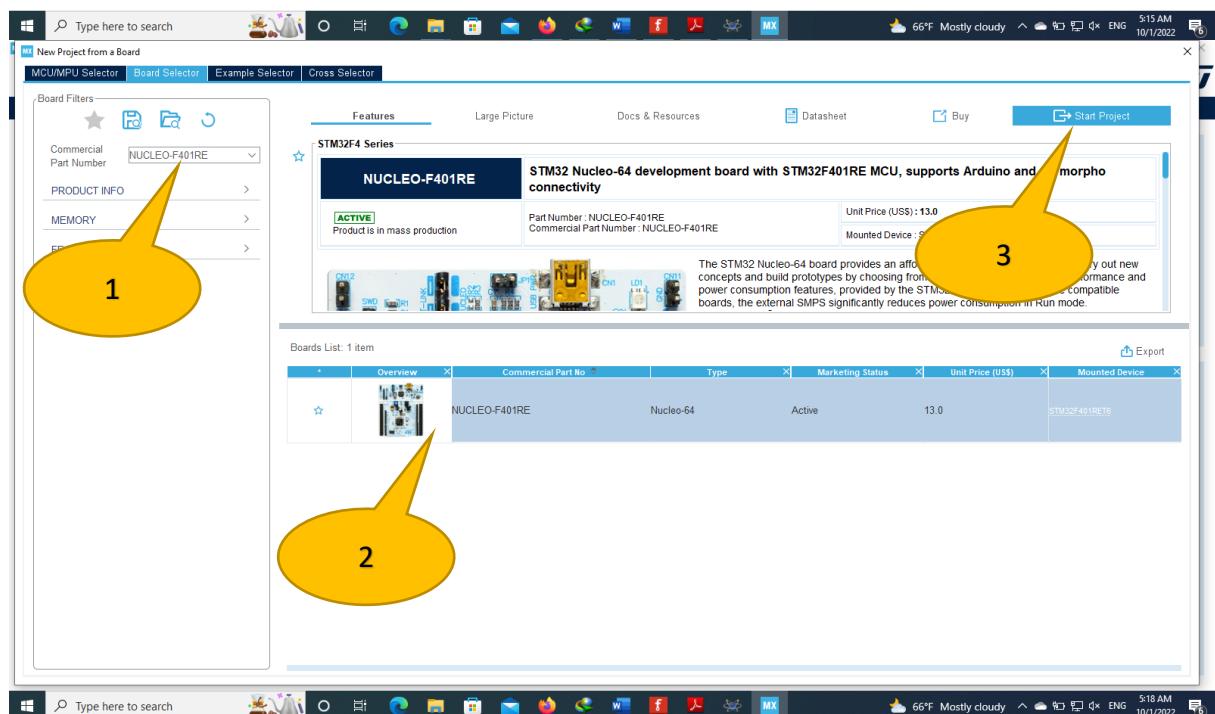
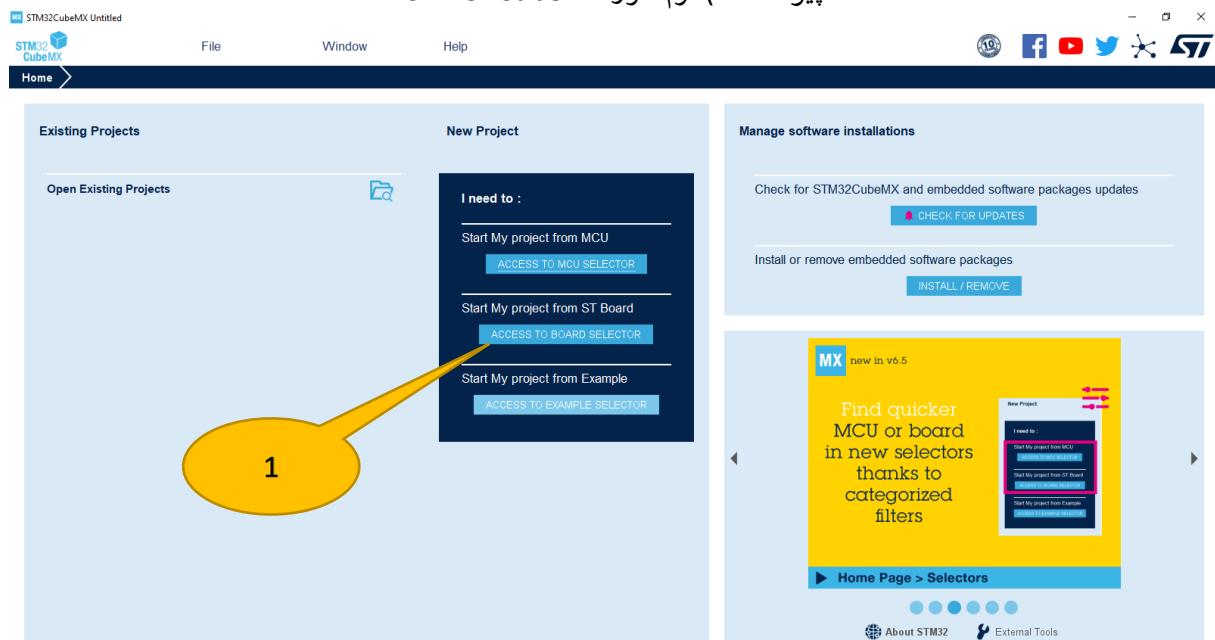


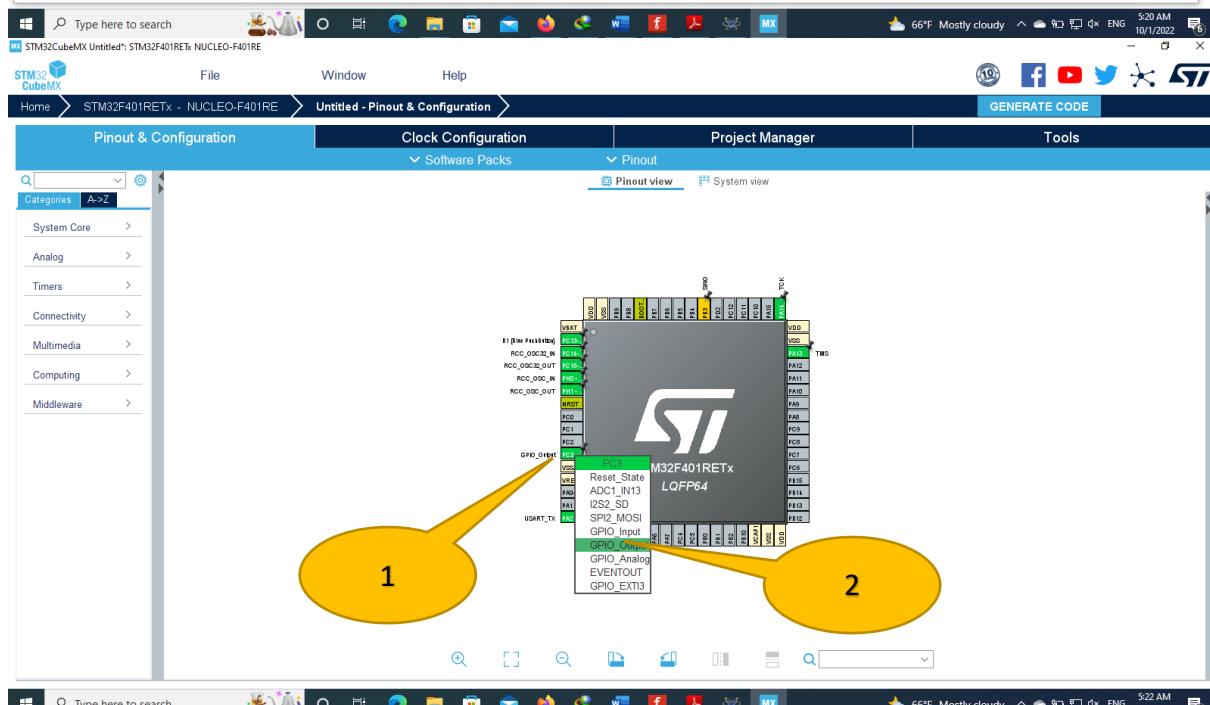
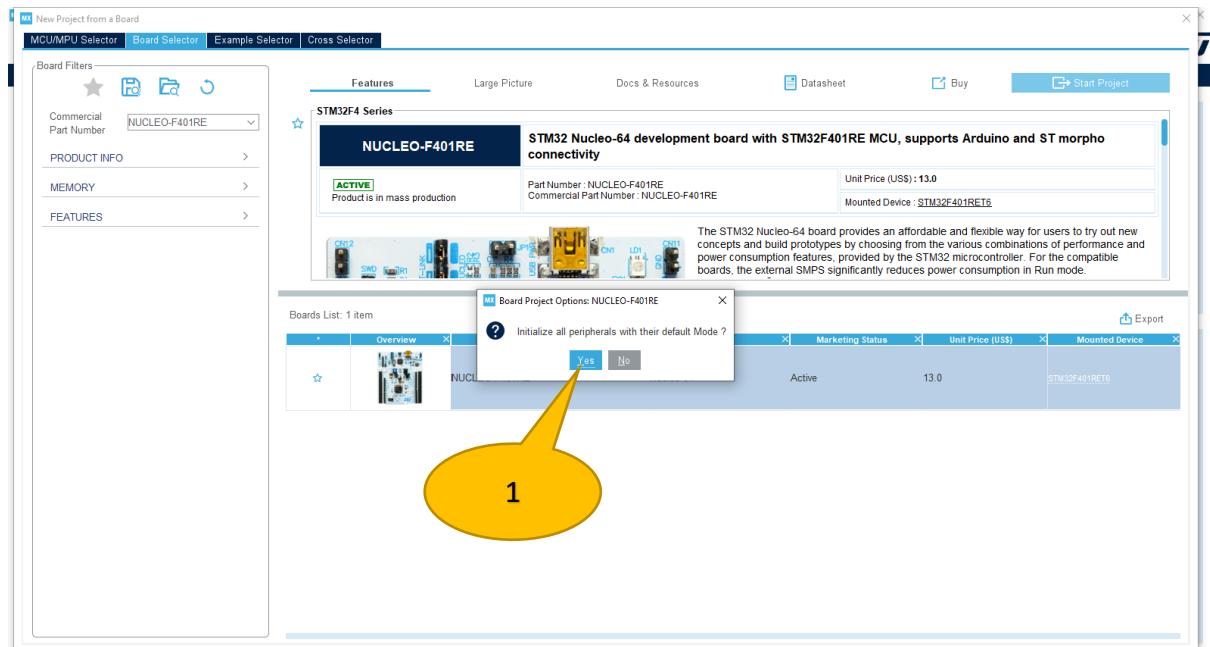


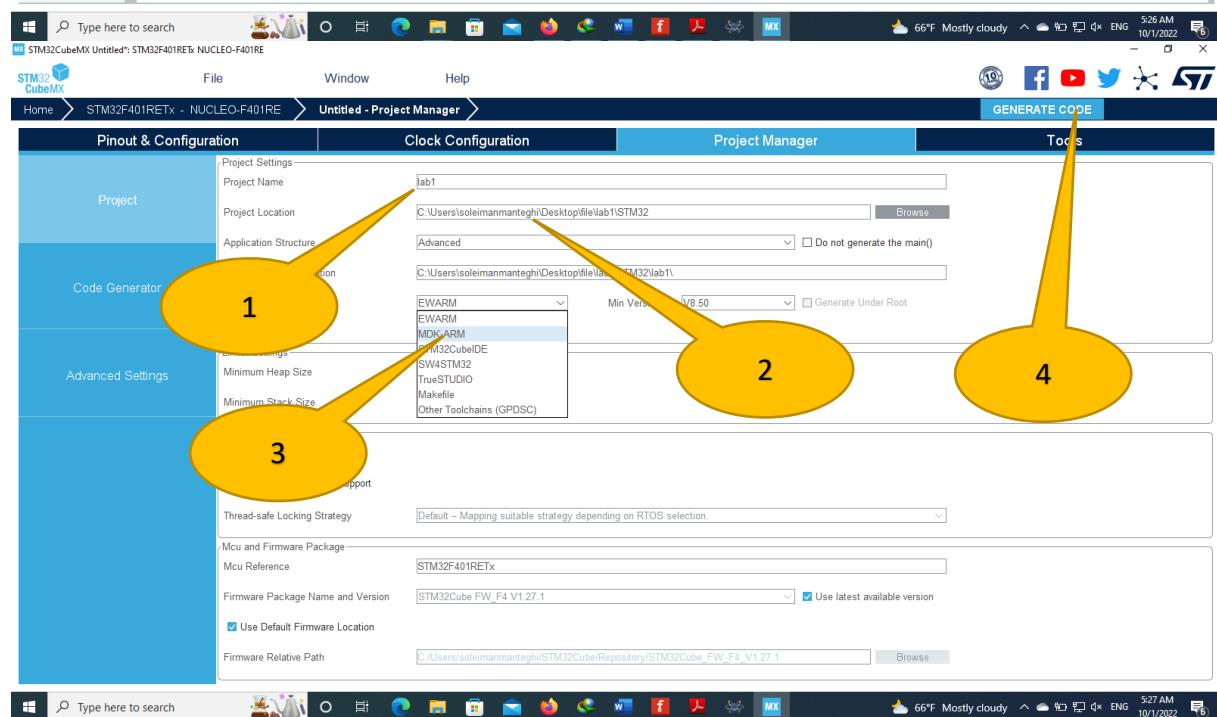
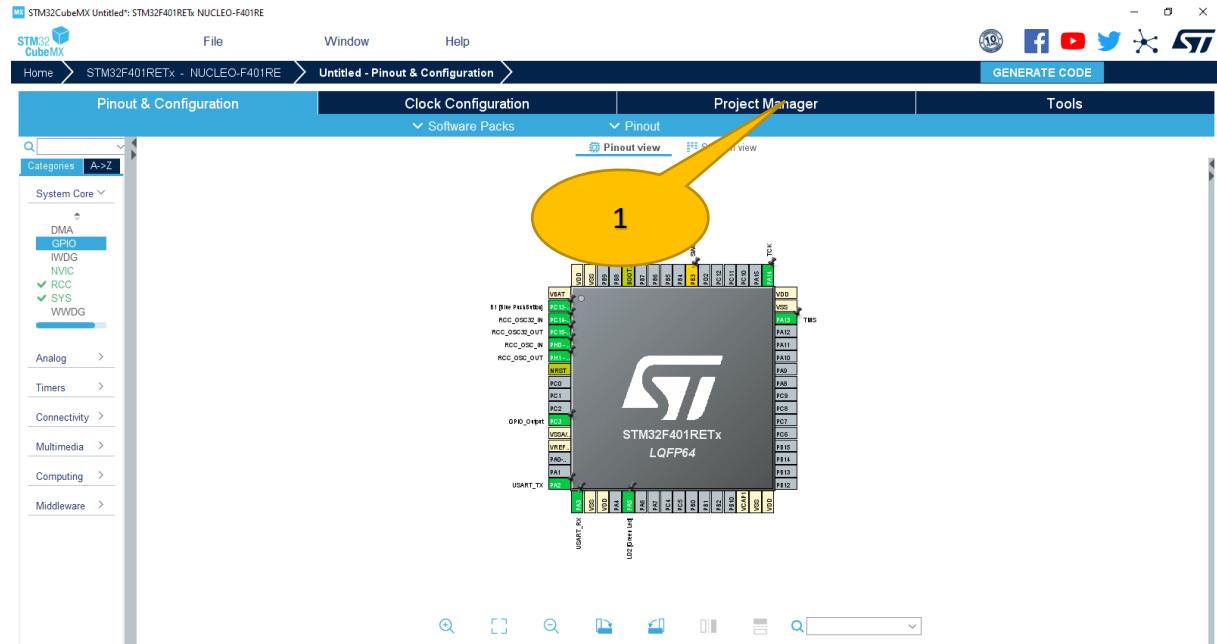


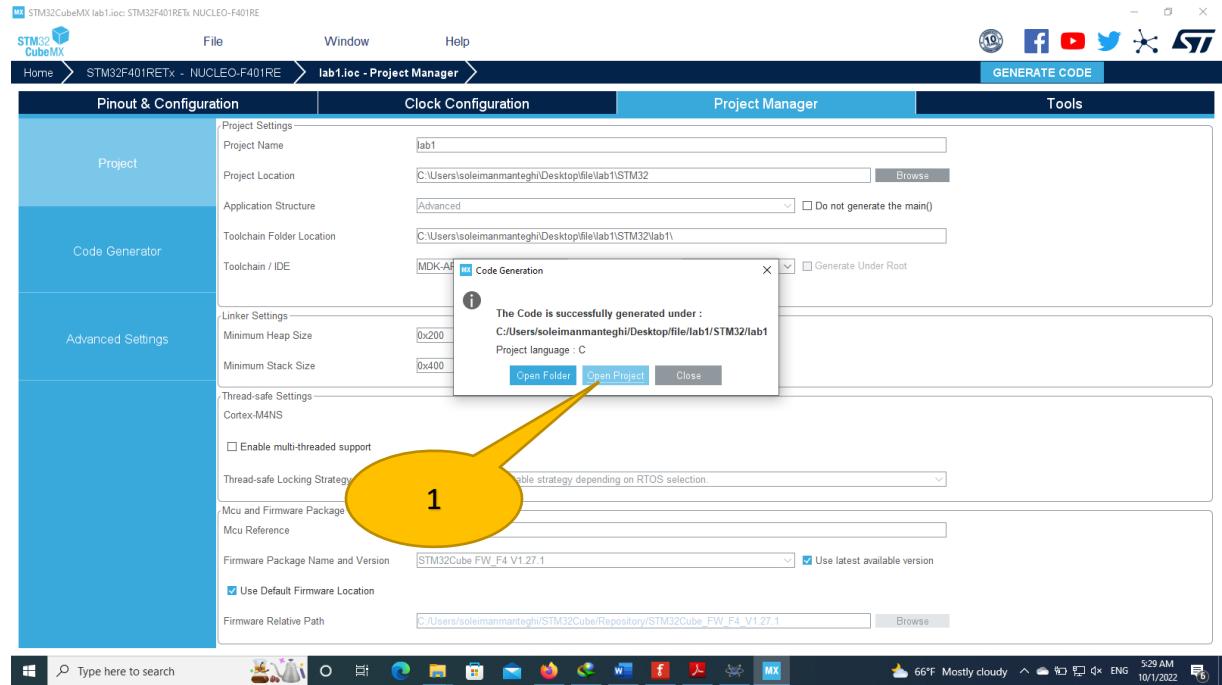


پیوست 2) نرم افزار STM32CubeMX









پیوست شماره 3_ نرم افزار keil

```

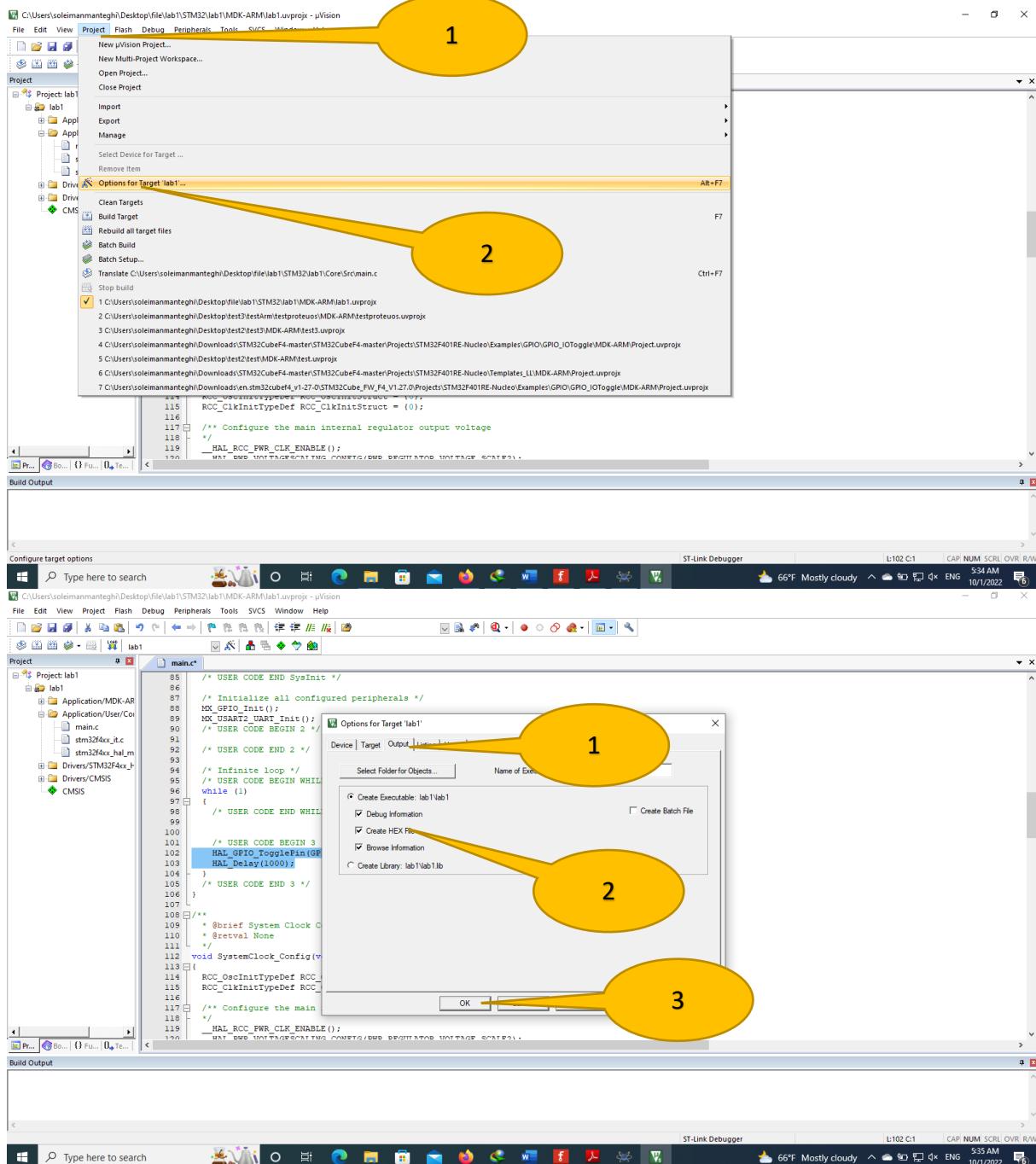
85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_GPIO_Init();
89  MX_USART2_UART_Init();
90  /* USER CODE BEGIN 2 */
91
92  /* Infinite loop */
93  /* USER CODE BEGIN WHILE */
94  while (1)
95  {
96      /* USER CODE END WHILE */
97
98      /* USER CODE BEGIN 3 */
99
100     /* USER CODE END 3 */
101
102 }
103
104
105  /* @brief System Clock Configuration
106  * @retval None
107  */
108 void SystemClock_Config(void)
109 {
110     /* @brief RCC_OscInitTypeDef RCC_OscInitStruct = {0}; */
111     /* @brief RCC_ClkInitTypeDef RCC_ClkInitStruct = {0}; */
112
113     /* Configure the main internal regulator output voltage
114     */
115     /* HAL_RCC_PWR_CLK_ENABLE(); */
116     /* HAL_PWRE_LL_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2); */
117
118     /* Initializes the RCC Oscillators according to the specified parameters
119     * in the RCC_OscInitTypeDef structure.
120 */

```

```

85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_GPIO_Init();
89  MX_USART2_UART_Init();
90  /* USER CODE BEGIN 2 */
91
92  /* Infinite loop */
93  /* USER CODE BEGIN WHILE */
94  while (1)
95  {
96      /* USER CODE END WHILE */
97
98      /* USER CODE BEGIN 3 */
99
100     /* HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_3);
101     HAL_Delay(1000);
102     */
103
104     /* USER CODE END 3 */
105
106 }
107
108 /* @brief System Clock Configuration
109 * @retval None
110 */
111 void SystemClock_Config(void)
112 {
113     /* @brief RCC_OscInitTypeDef RCC_OscInitStruct = {0}; */
114     /* @brief RCC_ClkInitTypeDef RCC_ClkInitStruct = {0}; */
115
116     /* Configure the main internal regulator output voltage
117     */
118     /* HAL_RCC_PWR_CLK_ENABLE(); */
119     /* HAL_RCC_HSE_CLK_ENABLE(); */

```



1

```

85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_USART2_UART_Init();
89
90  /* USER CODE BEGIN 2 */
91
92  /* Infinite loop */
93
94  /* USER CODE BEGIN WHILE */
95  while (1)
96  {
97
98  /* USER CODE END WHILE */
99
100
101  /* USER CODE BEGIN 3 */
102  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_3);
103  HAL_Delay(1000);
104
105  /* USER CODE END 3 */
106
107
108  /*
109  * @brief System Clock Configuration
110  * @retval None
111  */
112  void SystemClock_Config(void)
113  {
114      RCC_OscInitTypeDef RCC_OscInitStruct;
115      RCC_ClkInitTypeDef RCC_ClkInitStruct;
116
117  /* Configure the main internal regulator output voltage
118  */
119  /* HAL_RCC_PWR_CLK_ENABLE(); */
120  /* HAL_DCD_VOTINGSCALING_CONFIG(); */
121  /* HAL_DCD_REGULATOR_VOLTAGE_SCALE1(); */

```

2

3

Build Output

```

'Manage Project Items - Folders/Extensions' to manage ARM Compiler Versions.
'Options for Target - Target' to select an ARM Compiler Version for the target.
*** Build aborted.
Build Time Elapsed: 00:00:00

```

ST-Link Debugger

L102 C1 CAP NUM SCR LVR R/W

66°F Mostly cloudy

5:37 AM 10/1/2022

1

Build target files

Type here to search

C:\Users\soleimanmanteghi\Desktop\file\lab1\STM32\lab1\MDK-ARM\lab1.uvproj - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project

main.c

1

```

85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_USART2_UART_Init();
89
90  /* USER CODE BEGIN 2 */
91
92  /* Infinite loop */
93
94  /* USER CODE BEGIN WHILE */
95  while (1)
96  {
97
98  /* USER CODE END WHILE */
99
100
101  /* USER CODE BEGIN 3 */
102  HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_3);
103  HAL_Delay(1000);
104
105  /* USER CODE END 3 */
106
107
108  /*
109  * @brief System Clock Configuration
110  * @retval None
111  */
112  void SystemClock_Config(void)
113  {
114      RCC_OscInitTypeDef RCC_OscInitStruct;
115      RCC_ClkInitTypeDef RCC_ClkInitStruct;
116
117  /* Configure the main internal regulator output voltage
118  */
119  /* HAL_RCC_PWR_CLK_ENABLE(); */
120  /* HAL_DCD_VOTINGSCALING_CONFIG(); */
121  /* HAL_DCD_REGULATOR_VOLTAGE_SCALE1(); */

```

Build Output

```

'Manage Project Items - Folders/Extensions' to manage ARM Compiler Versions.
'Options for Target - Target' to select an ARM Compiler Version for the target.
*** Build aborted.
Build Time Elapsed: 00:00:00

```

ST-Link Debugger

L102 C1 CAP NUM SCR LVR R/W

Sunrise

5:38 AM 10/1/2022

Build target files

Type here to search

C:\Users\soleimanmanteghi\Desktop\file\lab1\STM32\lab1\MDK-ARM\lab1.uvproj - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Project

main.c

The screenshot shows the µVision IDE interface with the main.c file open. A yellow callout points to line 102 of the code, which contains the instruction `HAL_Delay(1000);`. The code is part of a user-defined loop.

```
85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_GPIO_Init();
89  MX_USART2_UART_Init();
90  /* USER CODE BEGIN 2 */
91
92  /* Infinite loop */
93  /* USER CODE BEGIN WHILE */
94  while (1)
95  {
96      /* USER CODE END WHILE */
97
98      /* USER CODE BEGIN 3 */
99
100     /* USER CODE BEGIN 3 */
101    HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_3);
102    HAL_Delay(1000);
103
104 }
105 /* USER CODE END 3 */
106
107
108 /**
109  * @brief System Clock Configuration
110  * @retval None
111  */
112 void SystemClock_Config(void)
113 {
114     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
115     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
116
117     /* Configure the main internal oscillator output voltage
118     */
119     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
120     RCC_OscInitStruct.HSEState = RCC_HSE_ON;
121     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
122     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
123     RCC_OscInitStruct.PLL.PLLM = 2;
124     RCC_OscInitStruct.PLL.PLLN = 84;
125     RCC_OscInitStruct.PLL.PLLP = 2;
126     RCC_OscInitStruct.PLL.PLLQ = 2;
127     RCC_OscInitStruct.PLL.PLLR = 2;
128
129     if(HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
130     {
131         /* Initialization Error */
132         /* User can check ERROR_LG flag in RCC寄存器 */
133     }
134
135     /* Configure the HSE clock source, PLL configuration and activation
136     */
137     if(HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
138     {
139         /* Initialization Error */
140         /* User can check ERROR_LG flag in RCC寄存器 */
141     }
142 }
```

Build Output:

```
Program Size: Code=9872 RO-data=460 RW-data=12 ZI-data=1708
FrcmSLF: creating hex file...
"lab1\lab1.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:06
```