راه اندازی نمایشگر OLED رنگی SSD1351 با STM32 – بخش دوم



تاریخ انتشار۲۲ خرداد, ۱۴۰۰ توسط آرش فتاحی

با عرض سلام خدمت همراهان سایت مایکروالکام. در بخش قبلی این آموزش، به توضیحات کلی درباره نمایشگرهای OLED، معرفی ماژول OLED RGB و پایه‌های مربوط به این نمایشگر پرداخته شد. همچنین تنظیمات و پیکربندیِ اولیه، برای راه‌اندازی واحد SPI در نرم افزار CUBEMX انجام گرفت.

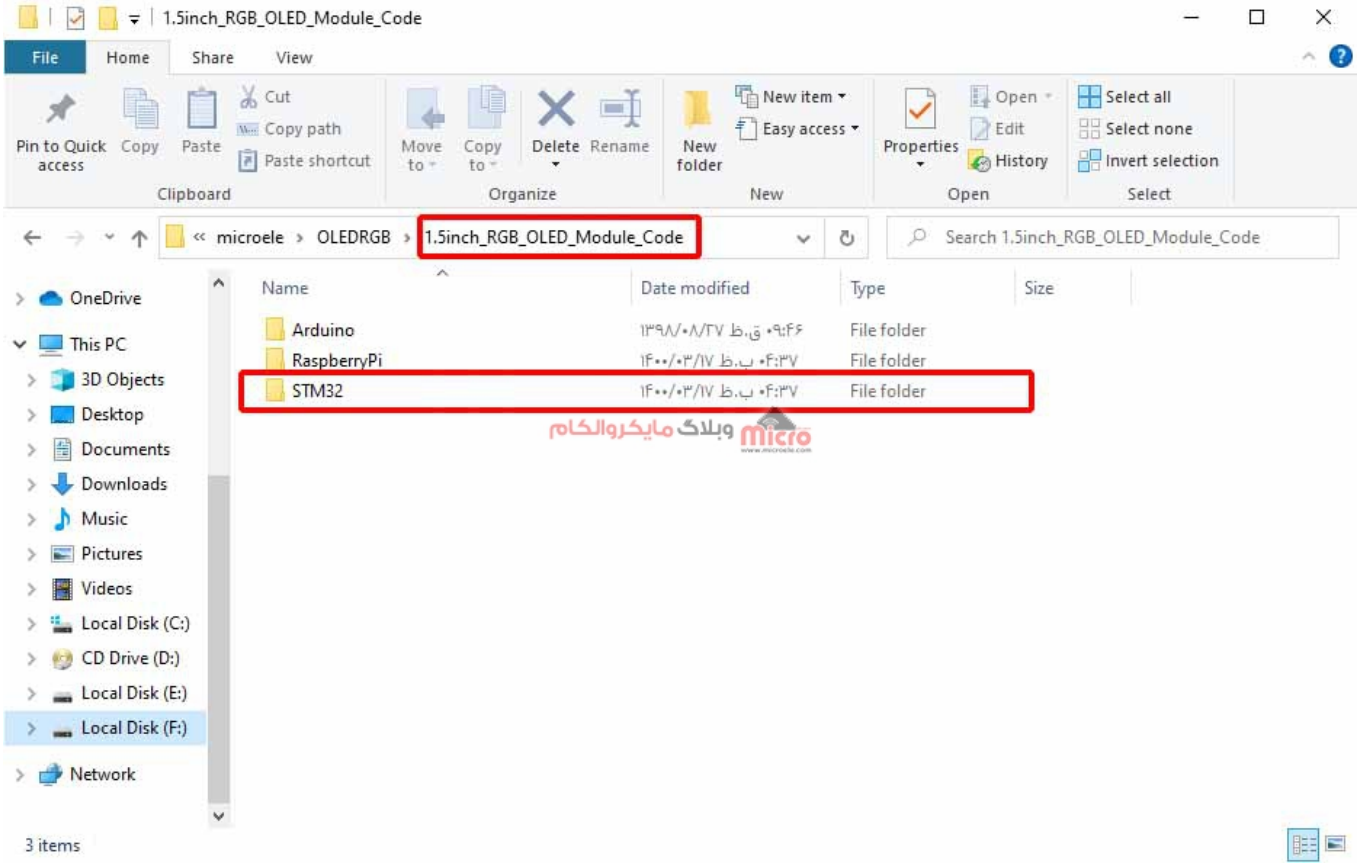
در این بخش به محیط Keil رفته و کتاب‌خانه‌های OLED رنگیِ Waveshare را اضافه کرده و یک برنامه ساده جهت نمایش بر روی ماژول OLED خواهیم نوشت. پس با من تا انتهای مطلب همراه باشید. همچنین میتونید سایر مطالب من رو از

این قسمت مطالعه کنید.

اضافه کردن کتاب‌خانه‌ها به محیط Keil

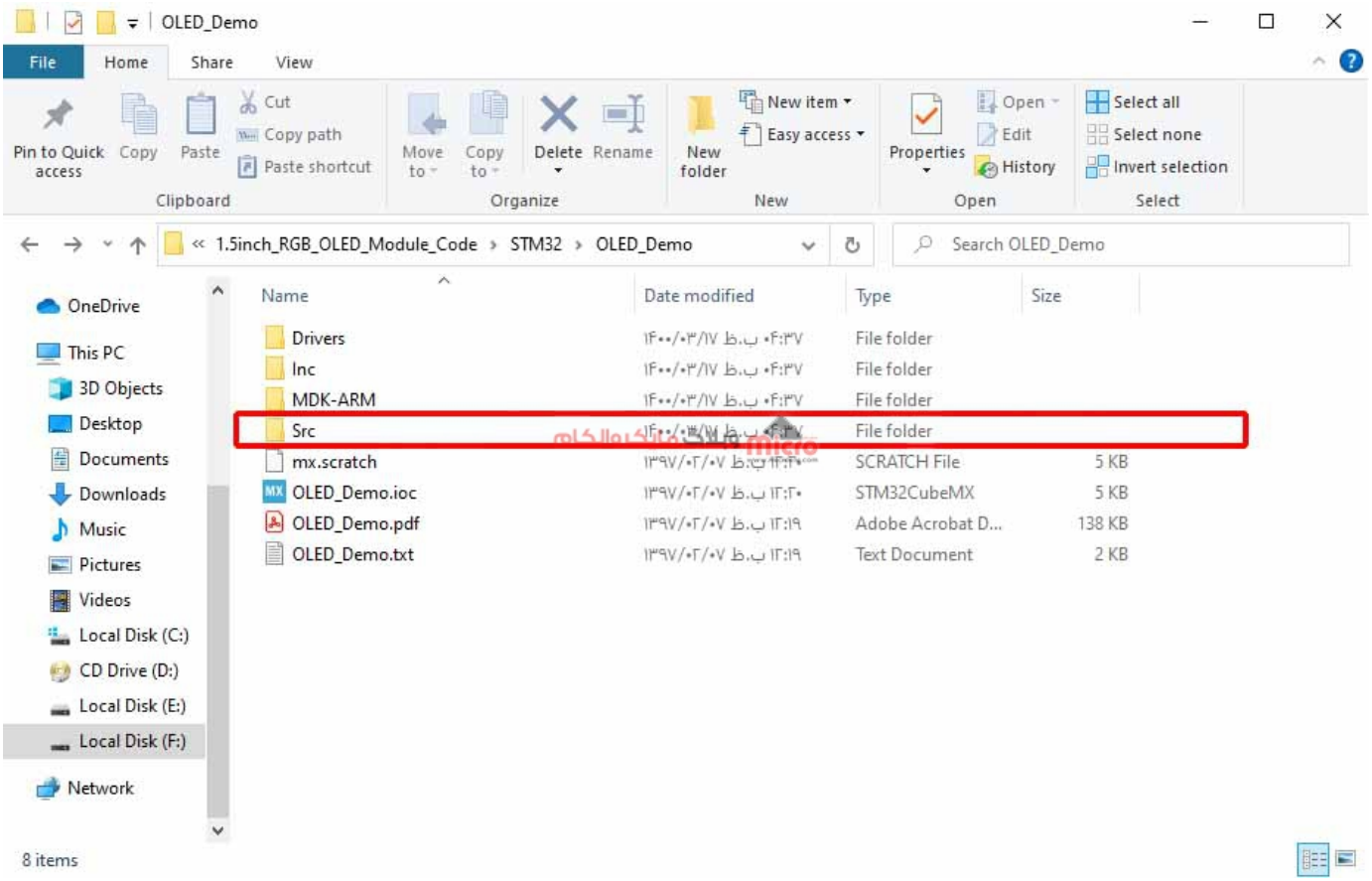
قسمت اول

در بخش قبل، از سایت Waveshare، اقدام به دانلود فایل‌های مورد نیاز جهت راه‌اندازی ماژول OLED کردیم. بعد از اکسترکت کردنِ فایل دانلود شده آن را باز کنید. داخل این پوشه سه پوشه دیگر وجود دارد. بر روی پوشه STM32 دوبار کلیک کرده و وارد آن شوید.



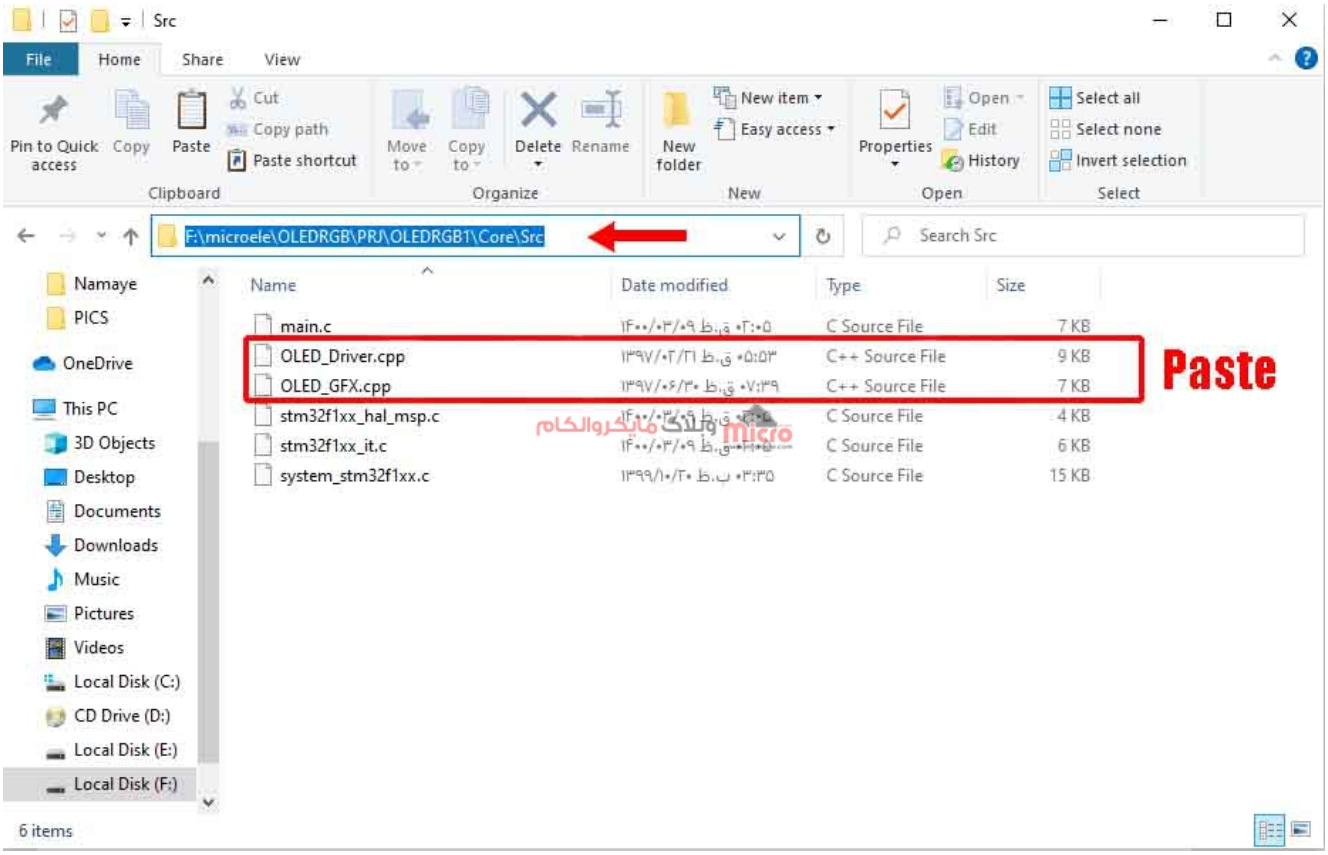
اضافه کردن کتاب‌خانه‌ها

پوشه OLED_DEMO را باز کنید. در این پوشه یک فایل پروژه Cube و همچنین یک فایل پروژه Keil به صورت آماده وجود دارد.



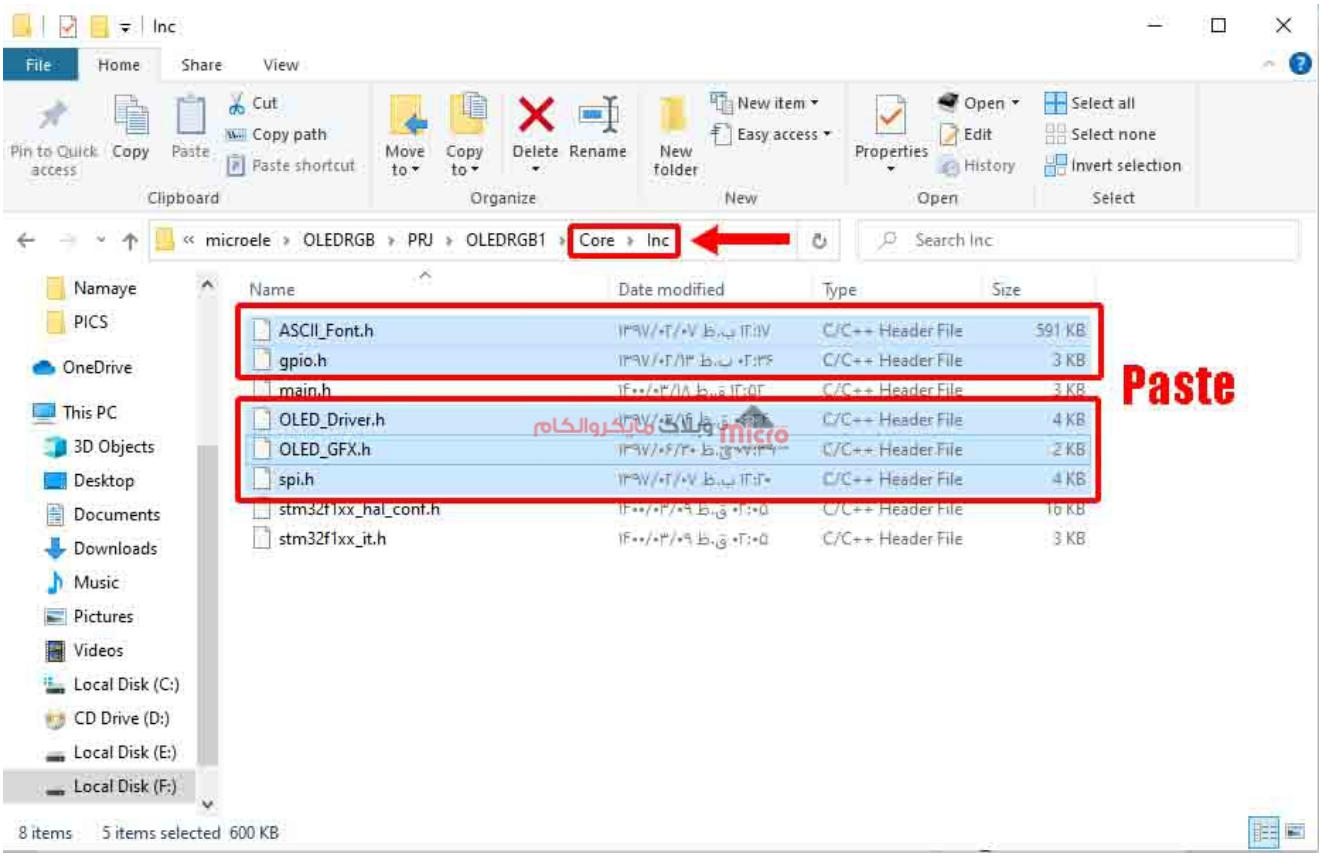
اضافه کردن کتاب‌خانه‌ها

از پوشه Demo، پوشه Src را باز کنید. فایل‌های OLED_Driver.cpp و OLED_GFX.cpp را به پوشه سورس در پروژه خود انتقال دهید. توجه شود که اگر برای ساخت پروژه، از ورژن‌های جدیدتر Cube استفاده می‌کنید. پوشه‌های Src و Inc، برای شما در پوشه Core ایجاد شده است.



اضافه کردن کتاب‌خانه‌ها در محیط *Keil*

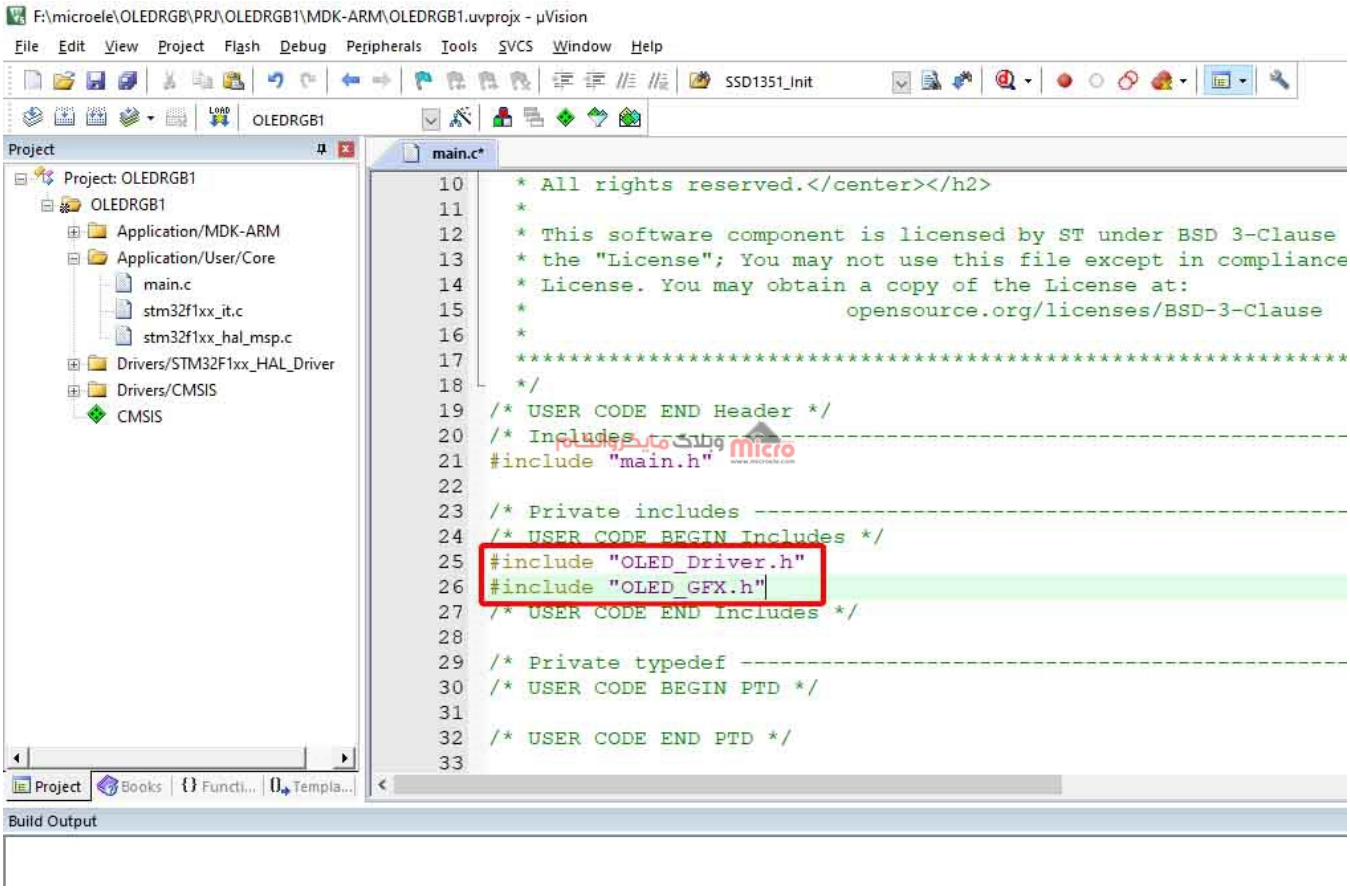
مجدد به پوشه Demo برگشته و این بار وارد پوشه Inc شوید. در اینجا فایل‌های OLED_Driver.h، ASCII_Font.h، spi.h، OLED_GFX.h، و gpio.h را کپی کرده و به درون پوشه Inc در پروژه خود منتقل کنید.



اضافه کردن کتاب‌خانه‌ها در محیط *Keil*

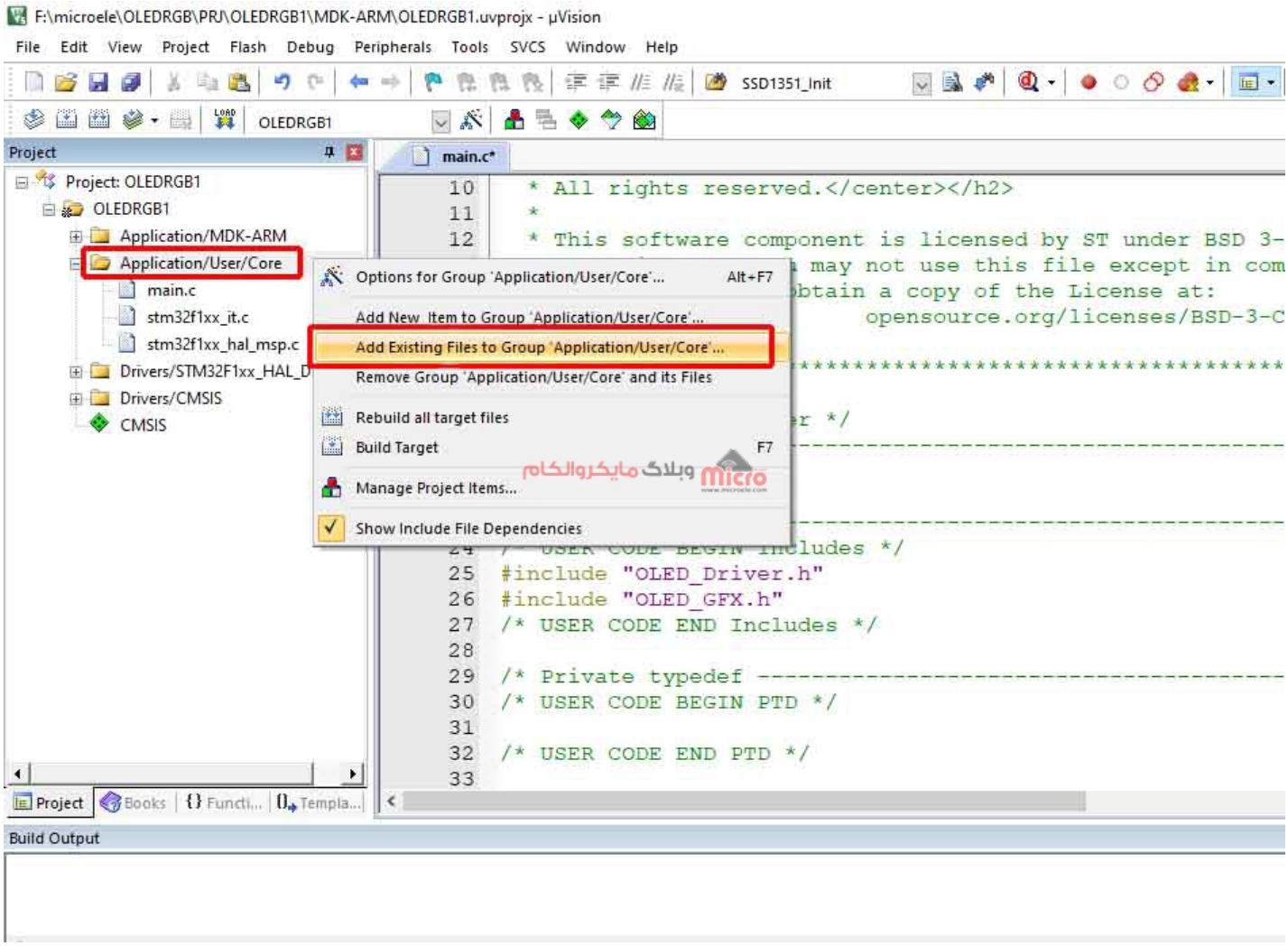
قسمت دوم

حال به محیط Keil برگشته و بعد از باز کردن پروژه‌ای که در بخش قبل ساختیم، فایل‌های h. را به پروژه خود اضافه می‌کنیم.



اضافه کردن کتاب‌خانه‌ها در محیط *Keil*

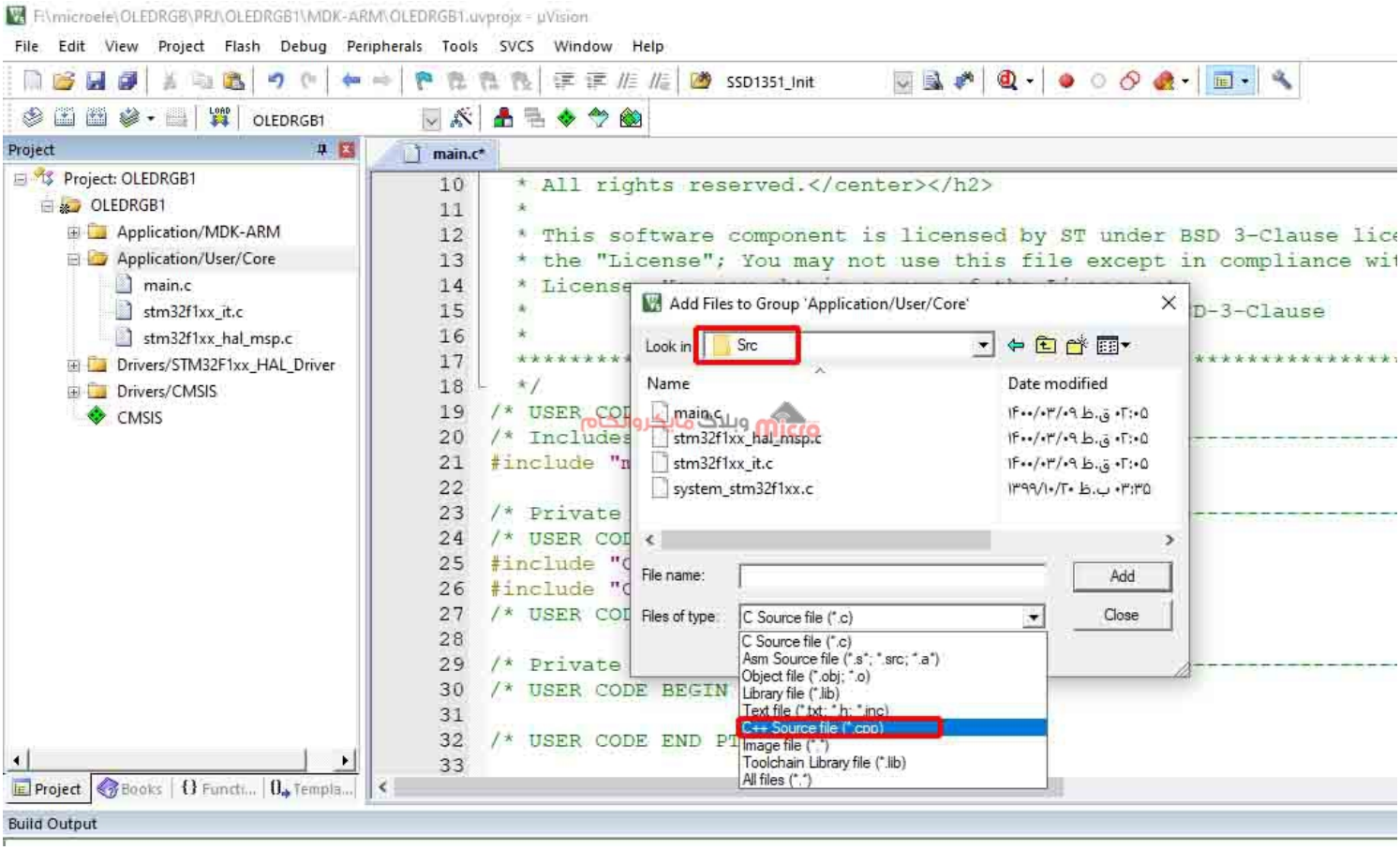
مرحله بعد اضافه کردن فایل‌های cpp که قبلتر در پوشه Src کپی کردیم می‌باشد. برای این کار بر روی Application/User/Core در بخش Project کلیک راست کرده و روی Add Existing Files to Group کلیک کنید.



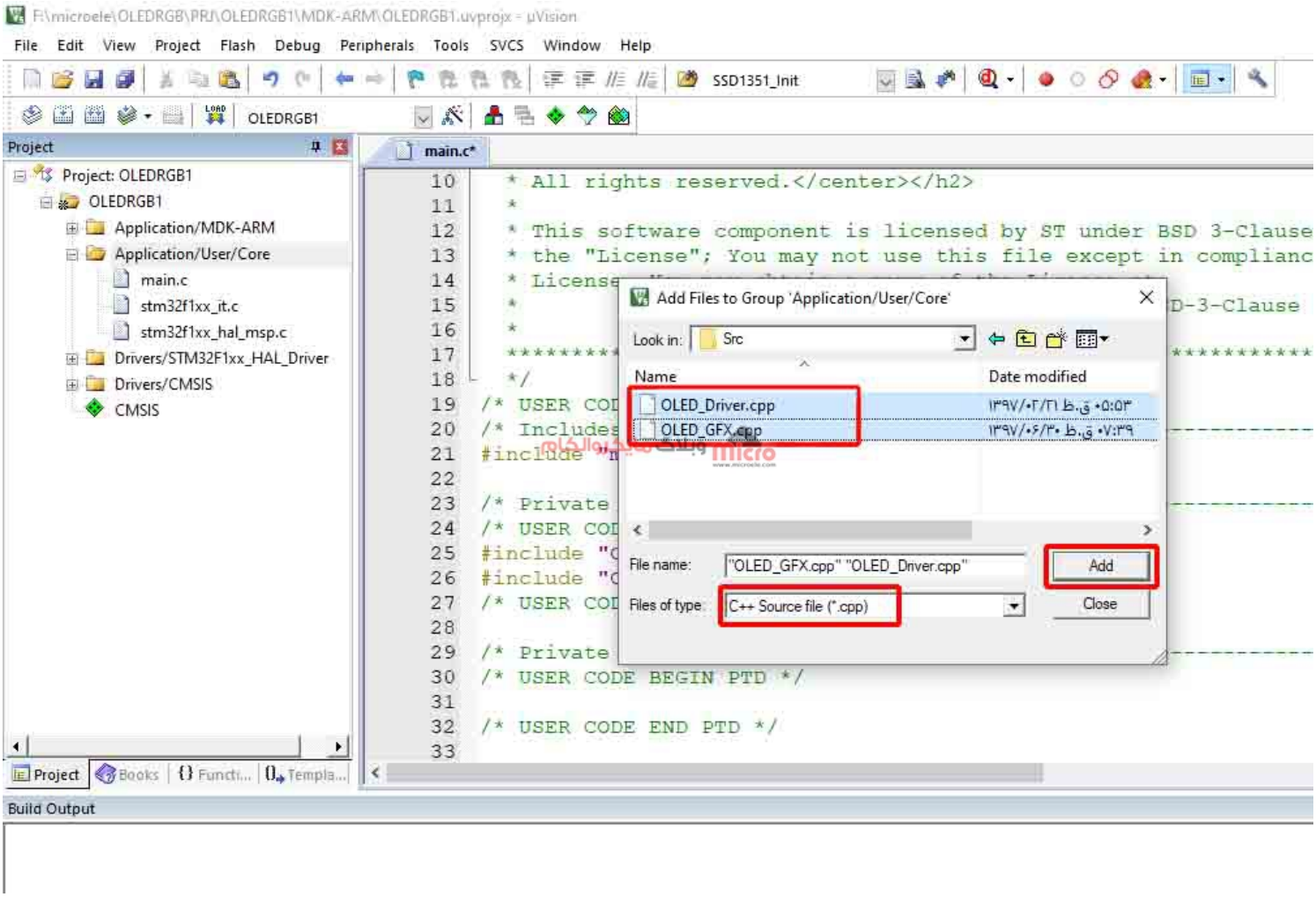
اضافه کردن کتاب‌خانه‌ها در محیط *Keil*

در پنجره Add Files، به پوشه Src در پروژه خود بروید. در اینجا باید دو فایل cpp که قبلتر در اینجا کپی کردیم را به برنامه خود Add کنیم. اما این فایل‌ها در حال حاضر نمایش داده نمی‌شوند.

در پایین پنجره، منوی Files of Types را باز کرده و گزینه C++ Source Files را انتخاب کنید تا دو فایل OLED_Driver.cpp و OLED_GFX.cpp نمایش داده شوند. هر دو فایل را انتخاب کرده و بر روی Add کلیک کنید.



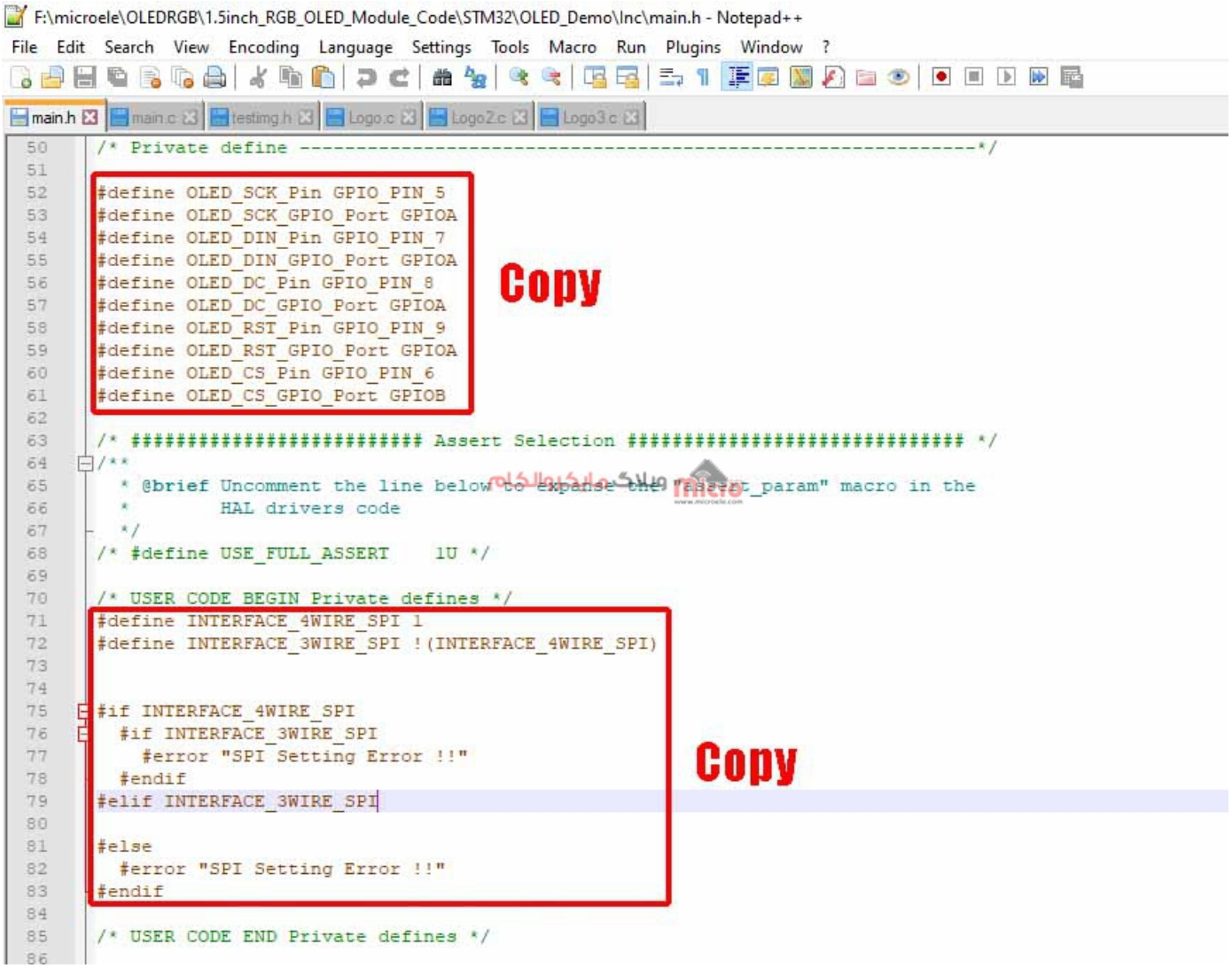
اضافه کردن فایل‌های *cpp* به *keil*



اضافه کردن فایل‌های *cpp* به *keil*

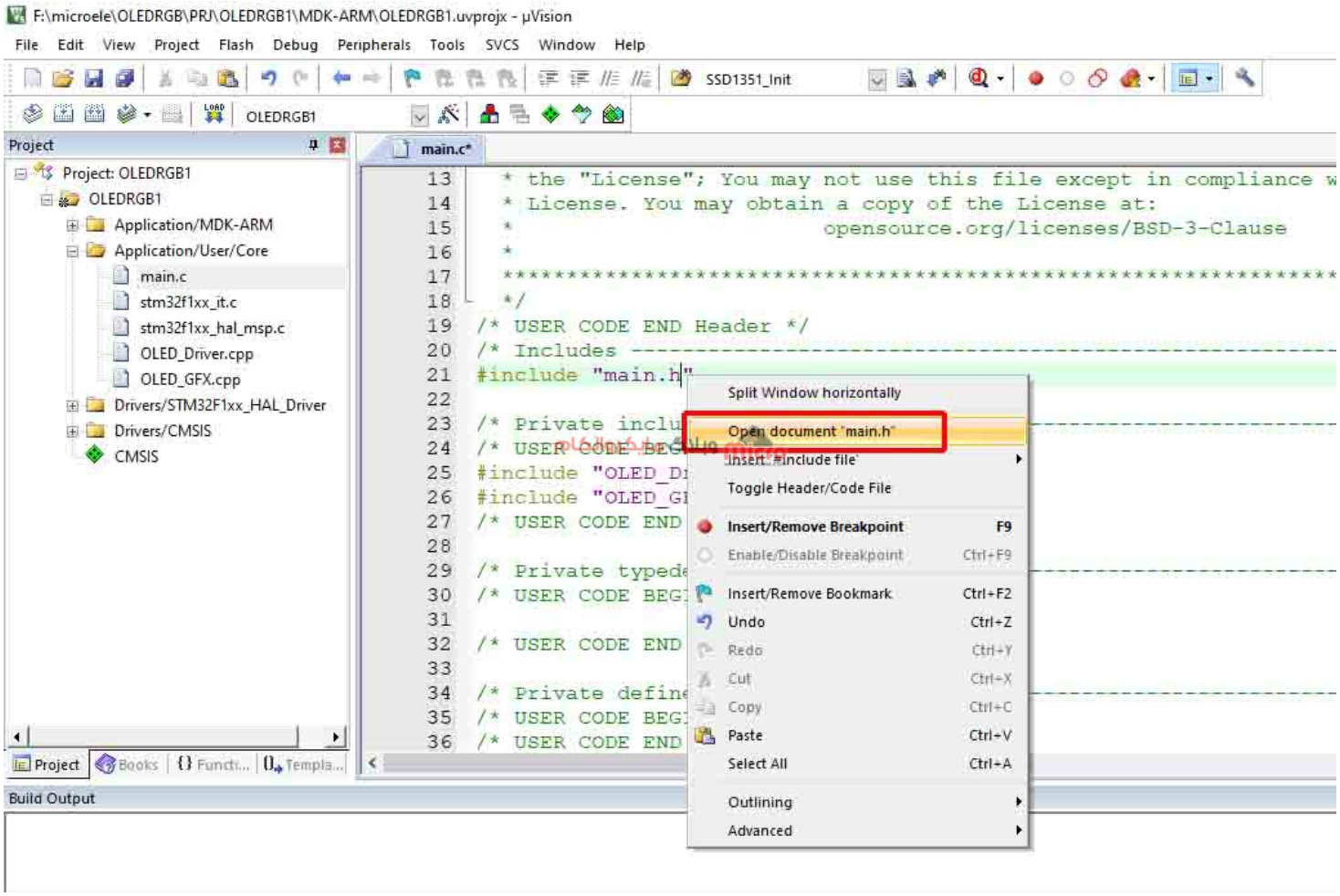
قسمت سوم

در پوشه Demo، فایل main.h را با یک ادیتور متن مانند notepad++ باز کنید. در این‌جا باید define های مربوط به دو بخشِ پایه‌های OLED و مشخص کردنِ کار با واحد SPI به صورت 3 سیم یا 4 سیم را کپی کرده و بردارید.



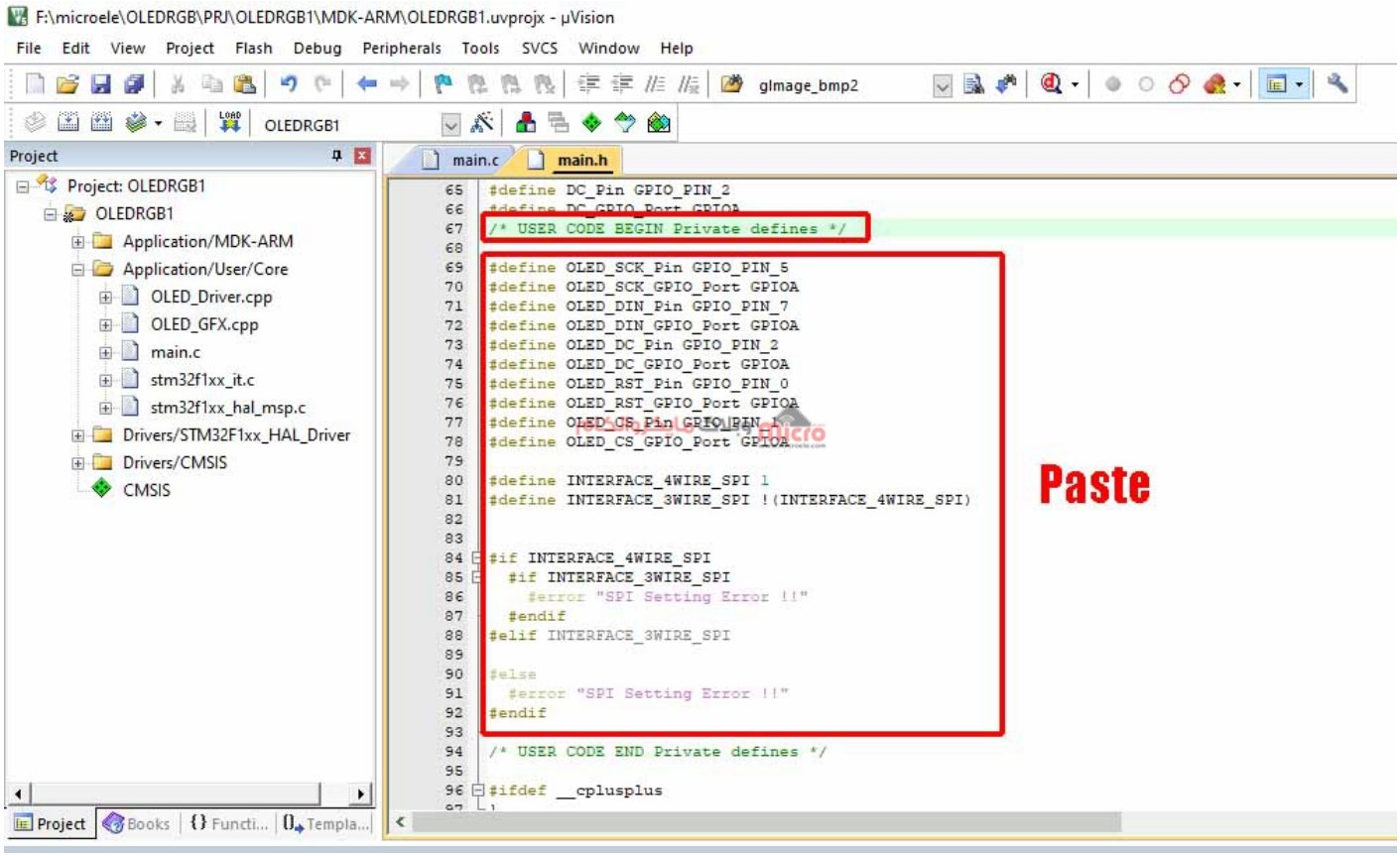
برداشتن مقادیر مورد نیاز از فایل *main.h*

قسمتی‌هایی را که کپی کردیم باید در فایل main.h پروژه خود اضافه کنیم. به محیط Keil برگشته و بر روی include main.h کلیک راست کنید. در منوی باز شده بر روی Open document main.h کلیک کرده تا فایل main.h پروژه شما باز شود.



باز کردن فایل *main.h*

در قسمت USER CODE BEGIN Private defines در فایل main.h، define های کپی شده را پیست کنید.



جایگزینی مقادیر در *main.h*

با توجه به انتخاب پایه‌ها در محیط Cube، شماره و پورت آن‌ها را در این بخش اصلاح می‌کنیم.

```
#define OLED_SCK_Pin GPIO_PIN_5
#define OLED_SCK_GPIO_Port GPIOA
#define OLED_DIN_Pin GPIO_PIN_7
#define OLED_DIN_GPIO_Port GPIOA
#define OLED_DC_Pin GPIO_PIN_2
#define OLED_DC_GPIO_Port GPIOA
#define OLED_RST_Pin GPIO_PIN_0
#define OLED_RST_GPIO_Port GPIOA
#define OLED_CS_Pin GPIO_PIN_1
#define OLED_CS_GPIO_Port GPIOA
```

برای مثال، ما در Cube، در قسمت قبل پایه‌های Rst ،DC و CS را بر روی پورت A و پایه‌های PA0 تا PA2 قرار دادیم که در بالا این موارد را اصلاح کردیم. حال با استفاده از کلیدهای ctrl + s فایل را ذخیره کنید.

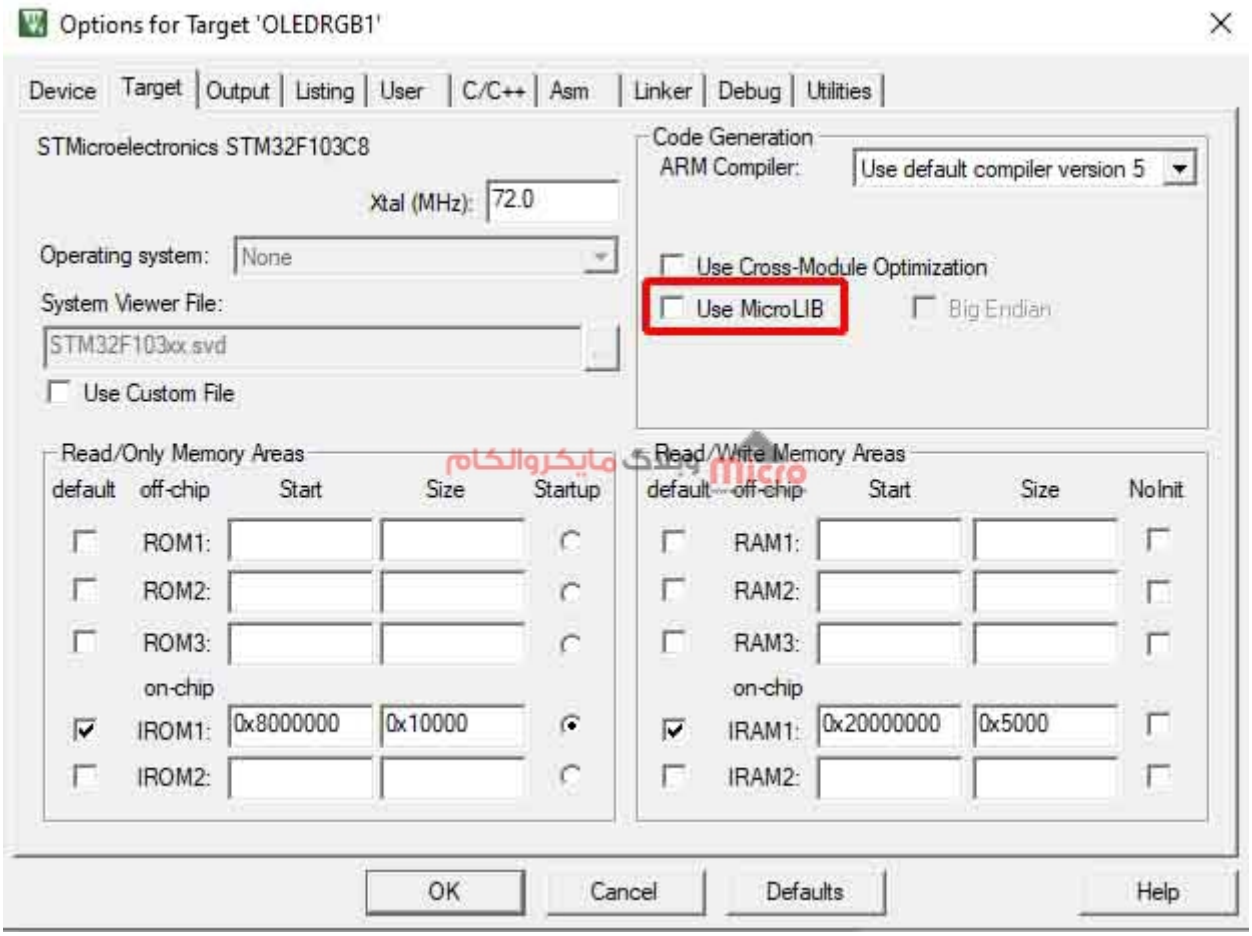
اضافه کردن امکان کامپایل فایل‌های C++ به Keil

با توجه به این که در این پروژه از فایل‌های C++ استفاده شده است، نیاز است که امکان کامپال کردن این فایل‌ها را به keil اضافه کنیم. برای این کار به منوی Project رفته و Options for Target را انتخاب کنید.



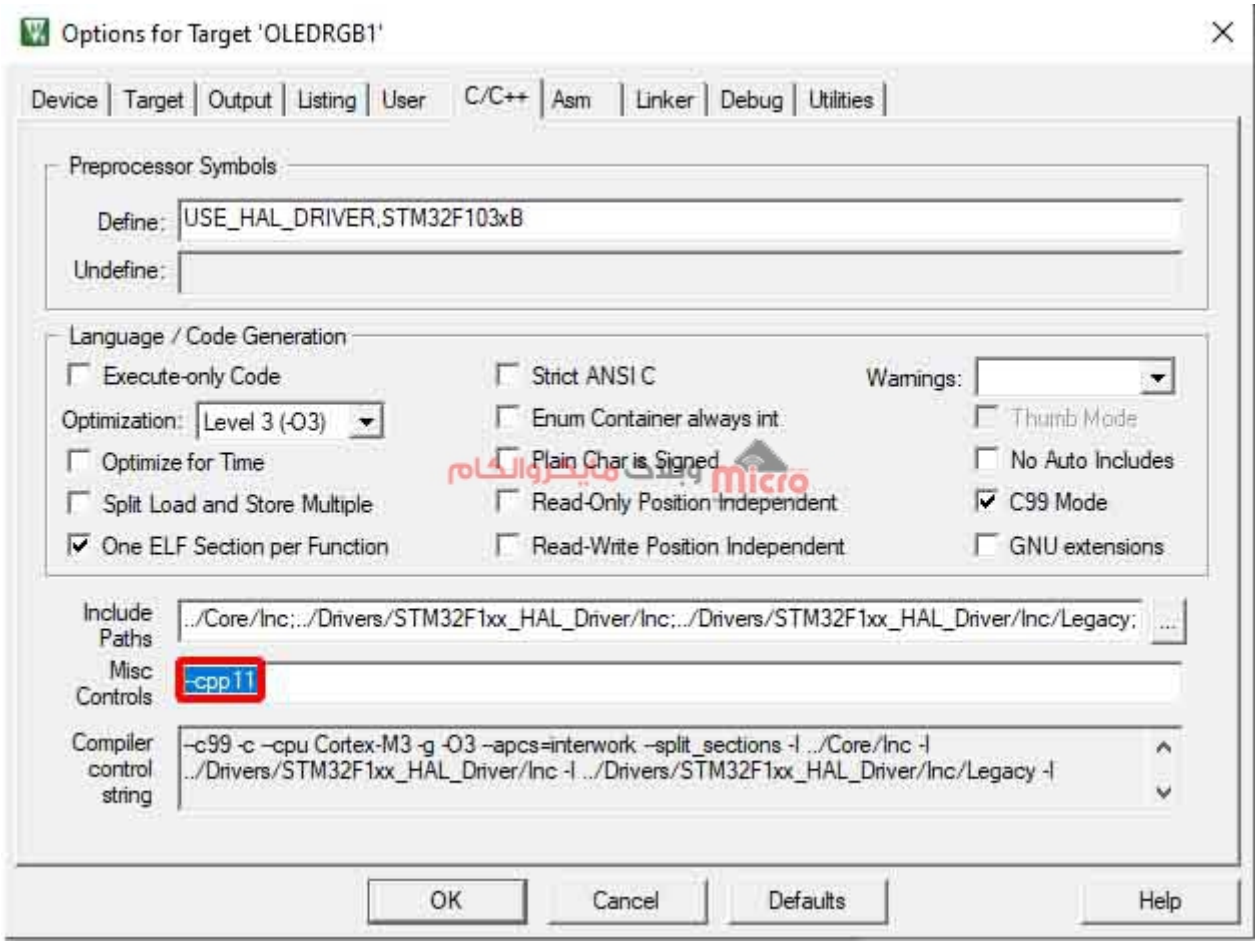
اعمال تنظیمات جدید در پروژه ساخته شده در *keil*

به منوی Project رفته و Options for Target را انتخاب کنید. از پنجره باز شده، سربرگ Target را انتخاب و از آن‌جا تیک گزینه Use MicroLIB را بردارید و آن را غیر فعال کنید.



اعمال تنظیمات جدید در پروژه ساخته شده در *keil*

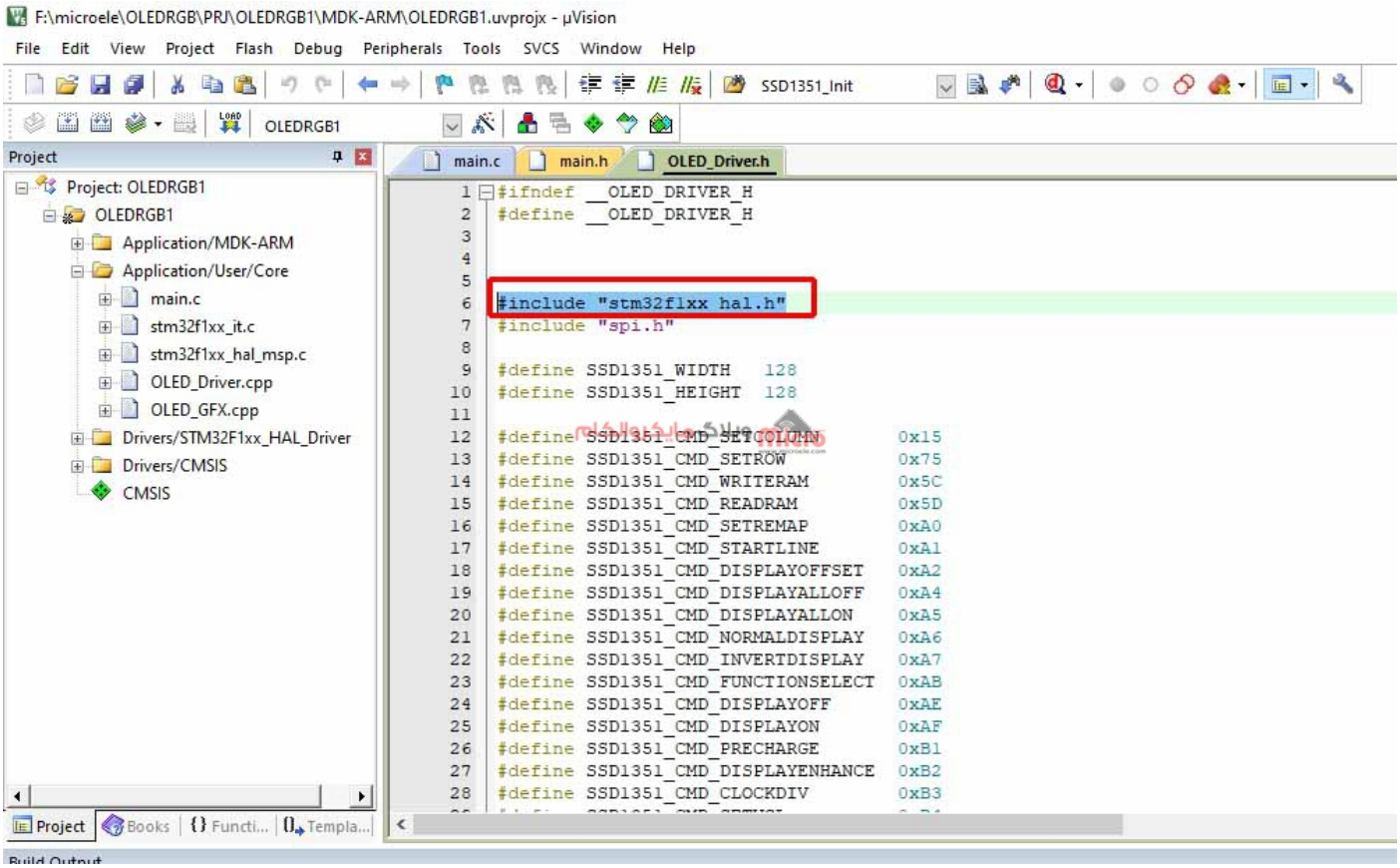
در مرحله بعد به سربرگ C/C++ رفته و در قسمت Misc Controls عبارت cpp11-- را تایپ کرده و در نهایت بر روی OK کلیک کنید تا تنظیمات اعمال و پنجره Options بسته شود.



اعمال تنظیمات جدید در پروژه ساخته شده در *keil*

```
--cpp11
```

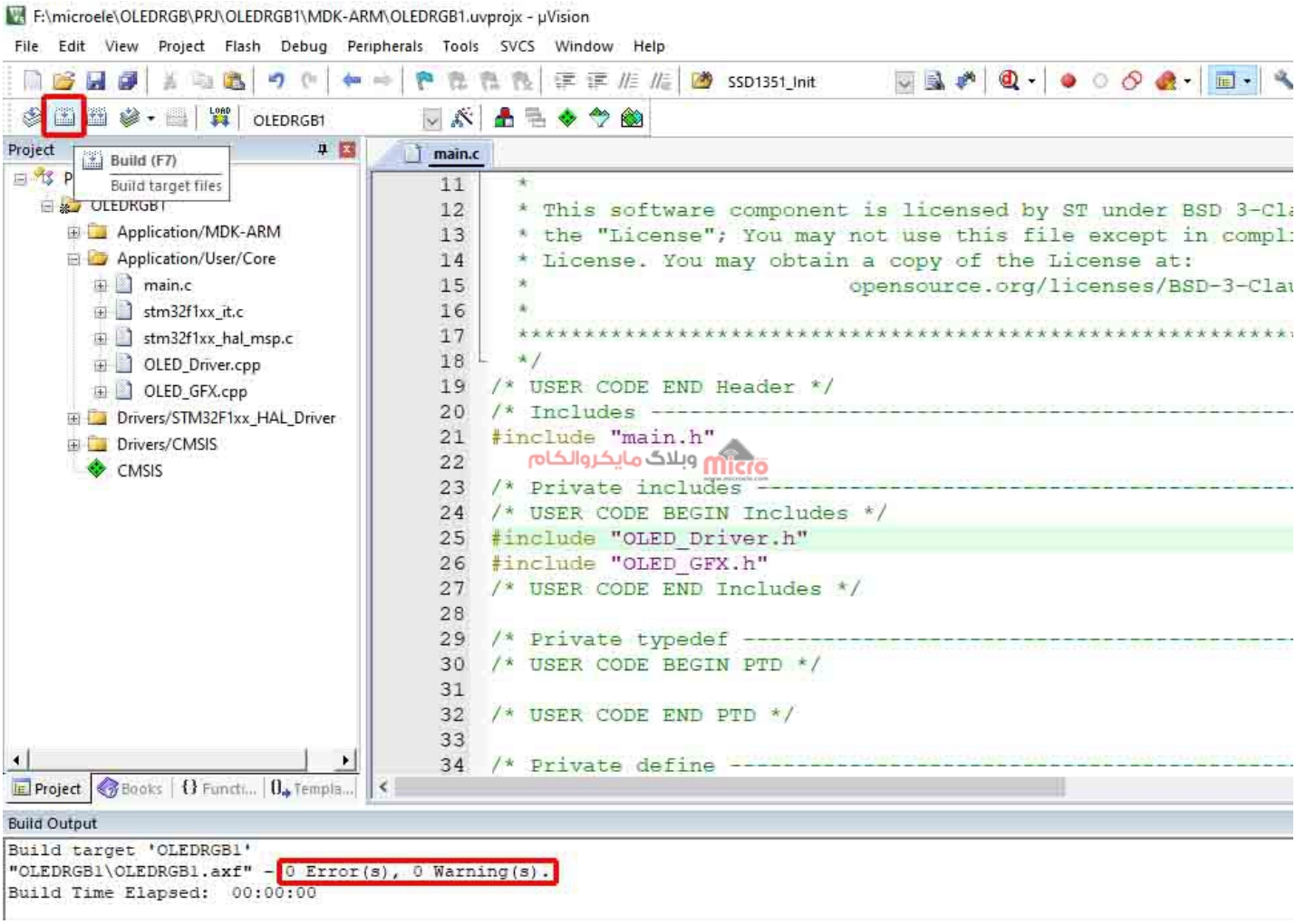
توجه شود، در صورتی که می‌خواهید از کتاب‌خانه‌ها برای سری‌های دیگر از میکروکنترلرهای STM32 استفاده کنید، باید فایل‌های h. را باز کرده و کتاب‌خانه میکرو مدنظر خود را با stm32f1xx.h جایگزین کنید.



جایگزینی کتاب‌خانه برای سری مورد نظر *STM32*

قسمت چهارم

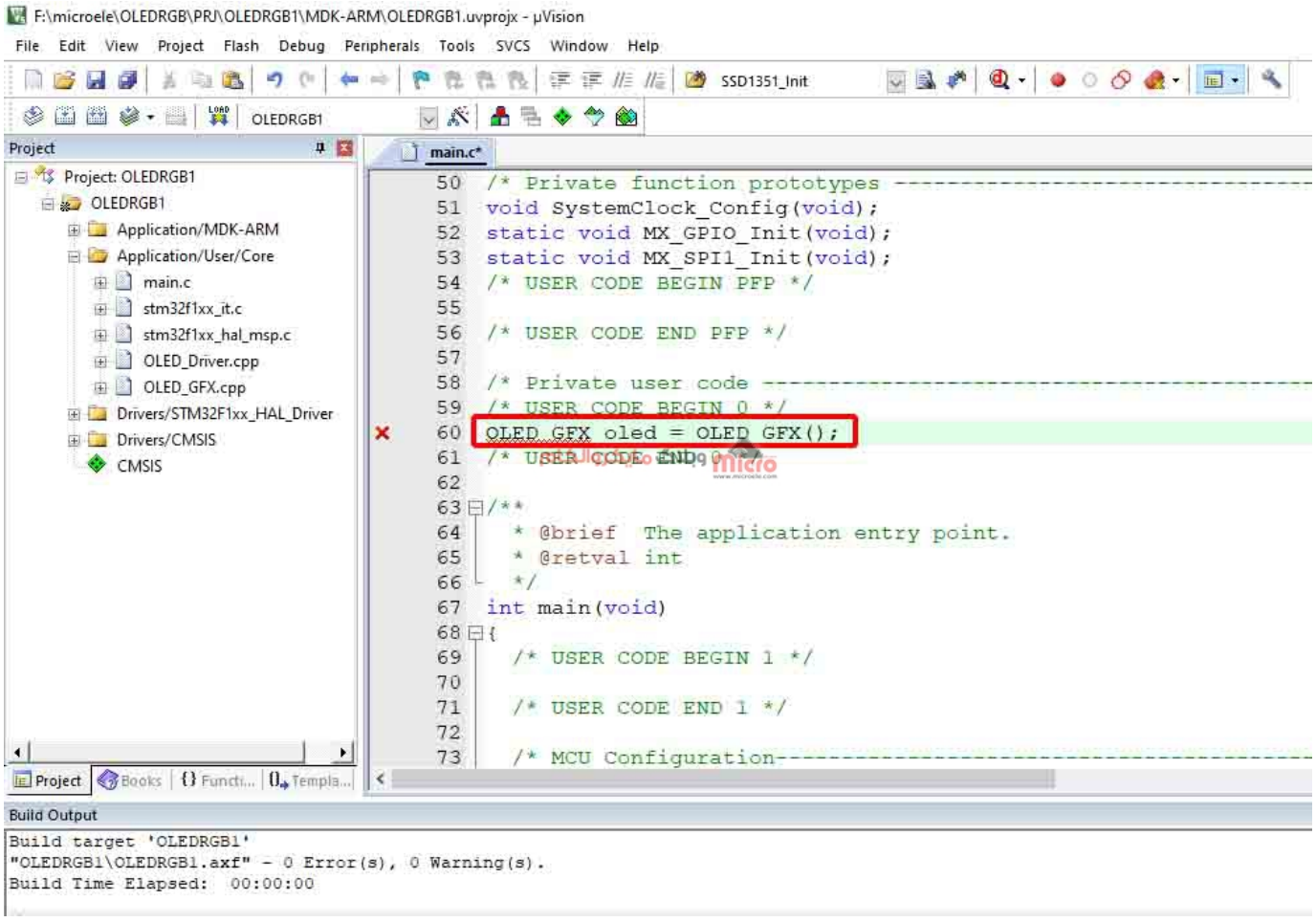
تا این‌جا، کارهای مربوط به پیکربندیِ پروژه تمام شده است و کافیست برنامه را Build کنیم. در صورت عدم وجود خطا، می‌توانیم نوشتن برنامه برای نمایش تصاویر و متون بر روی نمایشگر را انجام دهیم.



Build کردن برنامه در *keil*

شروع کدنویسی برای ایجاد تصاویر بر روی نمایشگر OLED

برای این که بتوانیم از کلاس‌های تعریف شده استفاده کنیم، ابتدا باید یک آبجکت برای OLED تعریف کنیم. برای این کار قبل از تابع اصلی (main)، آبجکتِ OLED را با استفاده از تابع OLED_GFX ایجاد می‌کنیم. نام آبجکتِ مورد نظر را به عنوان مثال OLED می‌گذاریم.

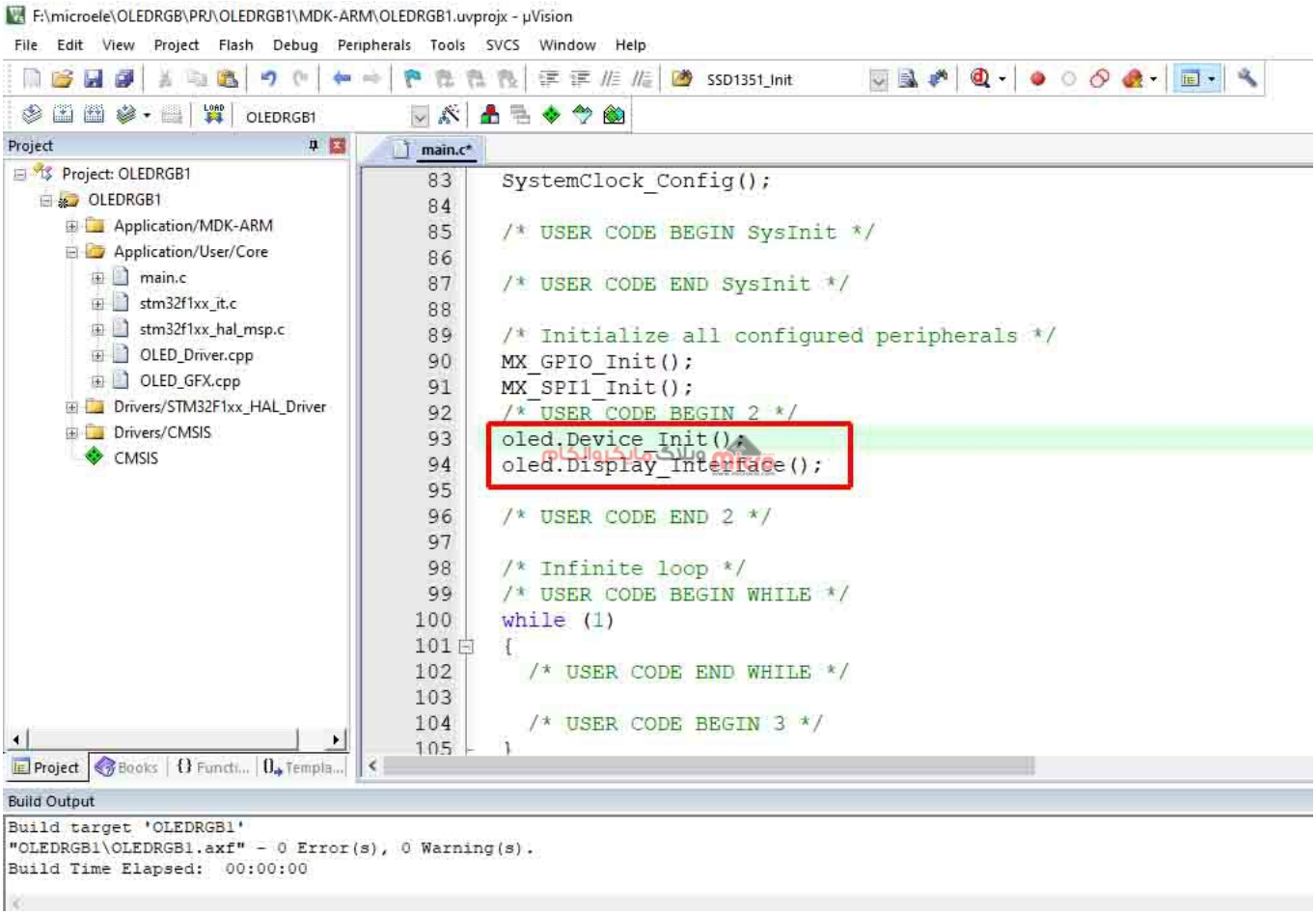


اضافه کردن آبجکت در فایل *main.c*

ممکن است بعد از اعمال این قسمت، از طرف keil با خطا در ادیتور مواجه شوید، که مهم نبوده و می‌توانید آن را نادیده بگیرید.

در داخل تابع main و قبل از while، ابتدا باید ماژول OLED توسط برنامه Initialize گردد. برای این کار از دستور Device_Init استفاده می‌کنیم.

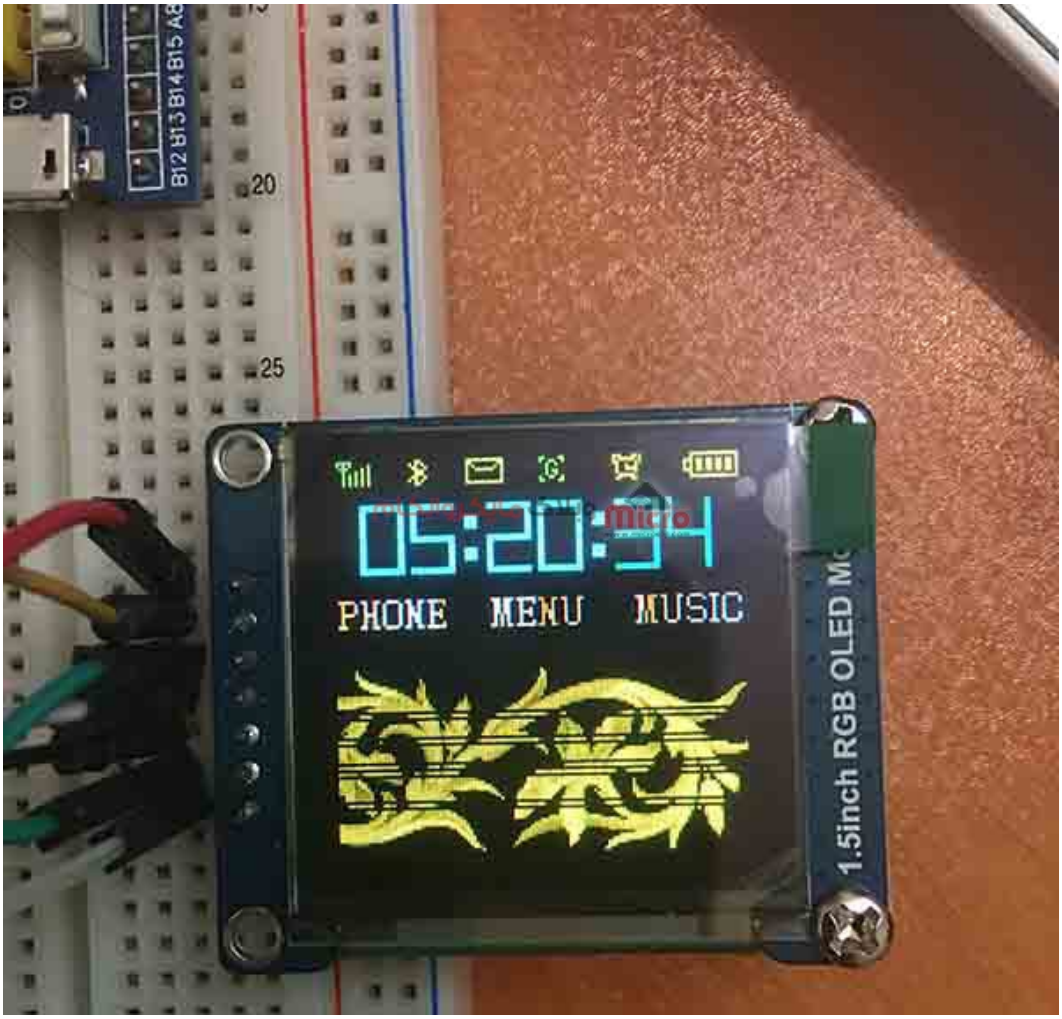
در کتاب‌خانه استفاده شده، یک تصویرِ آماده جهت نمایش وجود دارد که آن را نیز می‌توان با دستور Display_Interface فراخوانی و نمایش داد.



initialize کردن *OLED*

```
oled.Device_Init();
oled.Display_Interface();
```

برای تست ماژول OLED، برنامه را Build کرده و بر روی برد، Load کنید. مطابق عکس زیر، باید این تصویر بر روی OLED ایجاد شده باشد.



لود کردنِ *interface* آماده بر روی *OLED*

پاک کردن صفحه نمایشگر OLED

با دستور Clear_Screen می‌توان کل صفحه را پاک کرد.

```
oled.Clear_Screen();
```

تعیین رنگ متن در نمایشگر OLED

برای تعیین رنگ نوشته و شکل، کافیست قبل از فراخوانی دستورات مربوط به آن‌ها، با نوشتن دستورِ Set_Color رنگ مدنظر را انتخاب کنید.

به عنوان مثال برای ترسیم یک نقطه به رنگ سبز در مختصات x=50 و y=50 داریم:

```
oled.Set_Color(GREEN);
oled.Draw_Pixel(50,50);
```



روشن کردن یک پیکسل به رنگ سبز بر روی OLED

نوشتن متن برروی نمایشگر OLED

دستور print_String برای نوشتن متن بر روی OLED می‌باشد. دو پارامتر اول برای انتخاب مختصاتِ محل درج نوشته، پارامتر سوم خود متن و پارامتر چهارم انتخاب فونت نوشته است.

مثال برای نوشتن متن بر روی OLED:

```
oled.Set_Color(RED);
oled.print_String(0, 8, "microele.com",FONT_8X16);
```

همچنین کتاب‌خانه مورد استفاده از دو فونتِ **FONT_5X8** و **FONT_8X16** پشتیبانی می‌کند.



نمایش تصویر برروی نمایشگر OLED

برای درج تصویر، ابتدا باید یک عکس را به سایز 128 در 128 برده و با برنامه‌ای همچون LCD Vision آن را به صورت کد هگز تبدیل کنید.

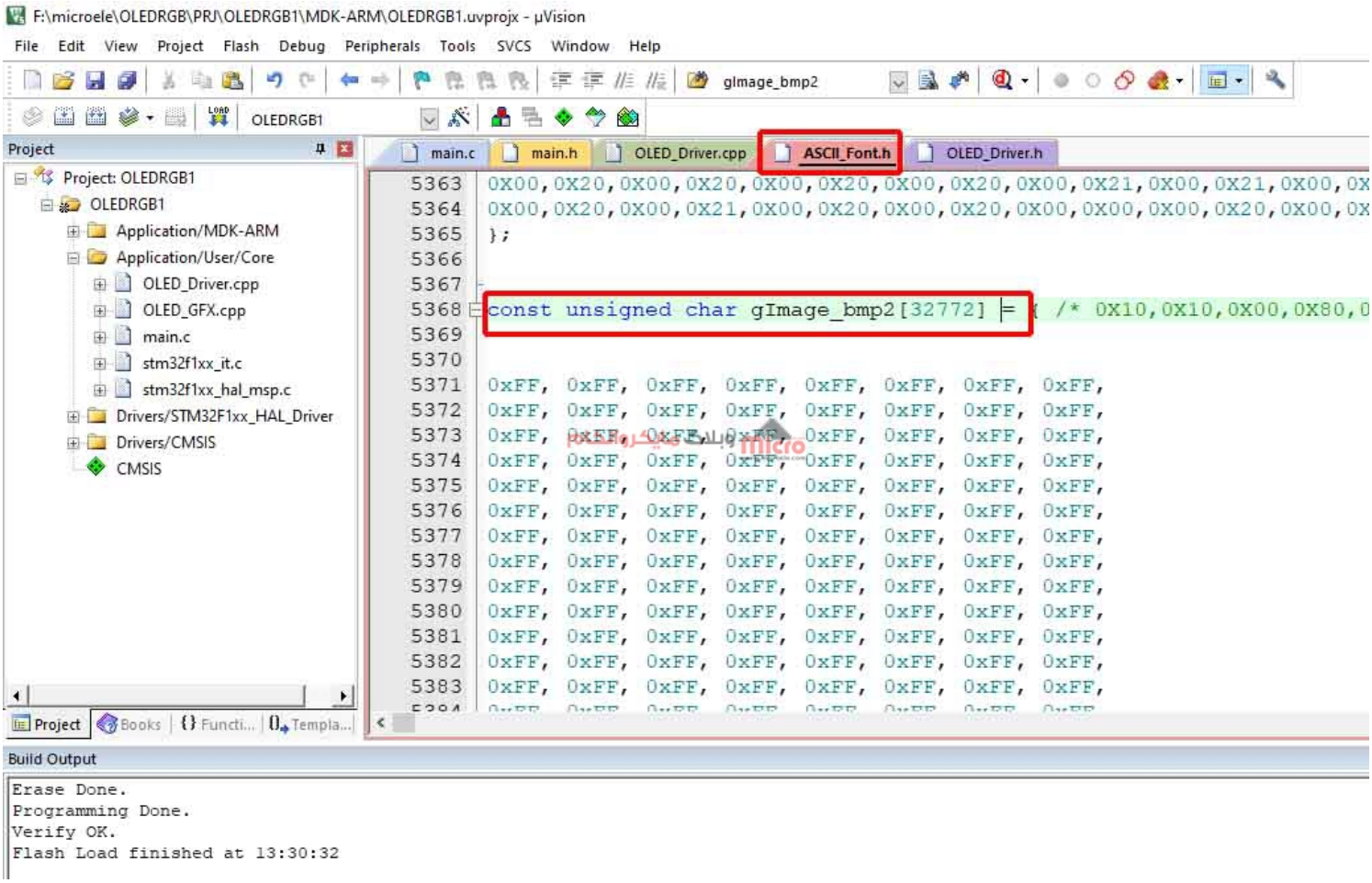
سپس تصویر خود را در فایل ASCII_FONT.h به صورت یک آرایه ذخیره کنید.

برای نمایش، تابع Display_bmp را در فایل OLED_Driver.cpp پیدا کرده و به جای gImage_bmp2، نام آرایه تصویر خود را جایگزین کنید.

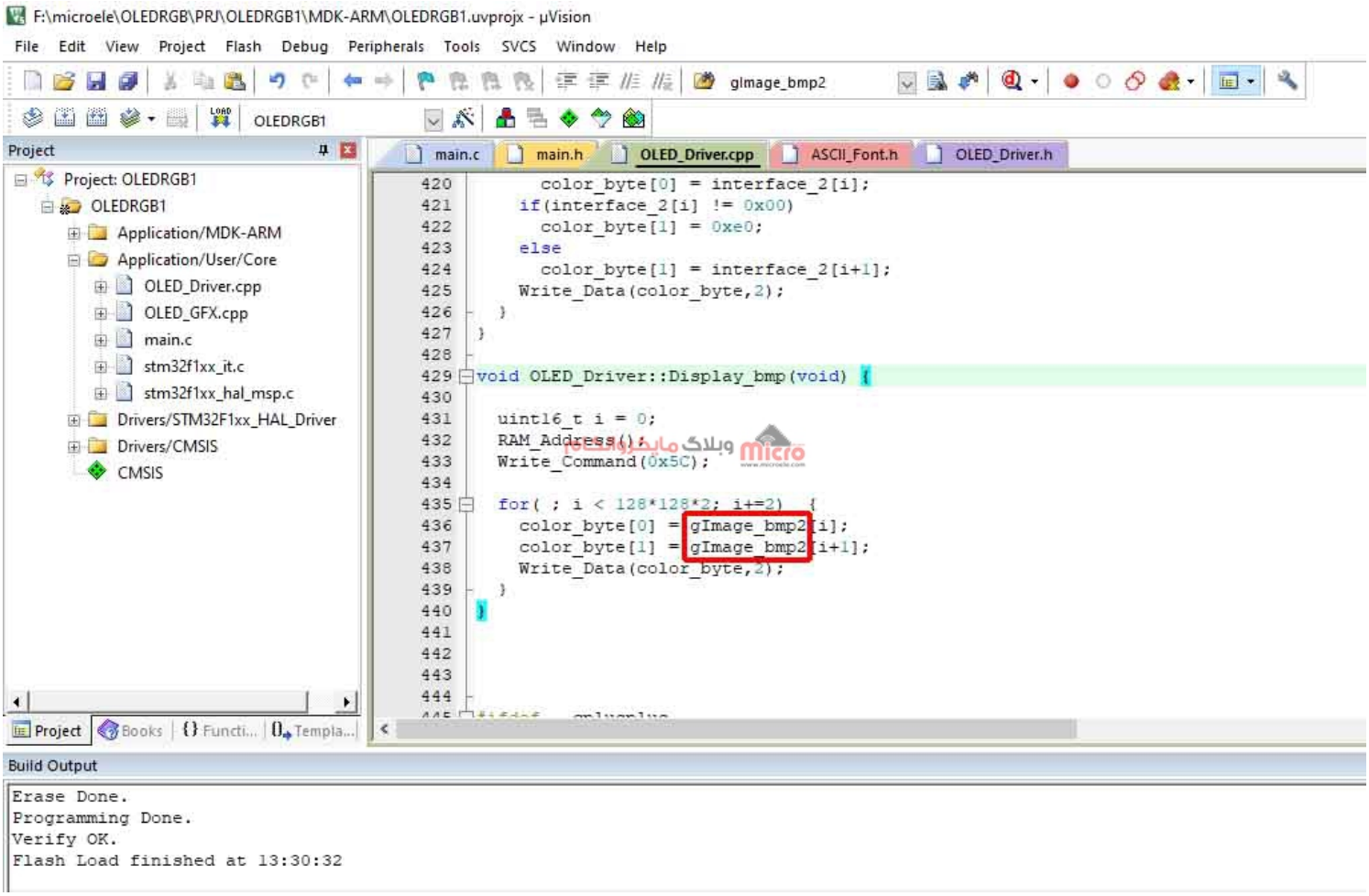
در فایل main با اعمال دستور Display_bmp تصویر شما نمایش داده خواهد شد.



درج تصویر بر روی OLED

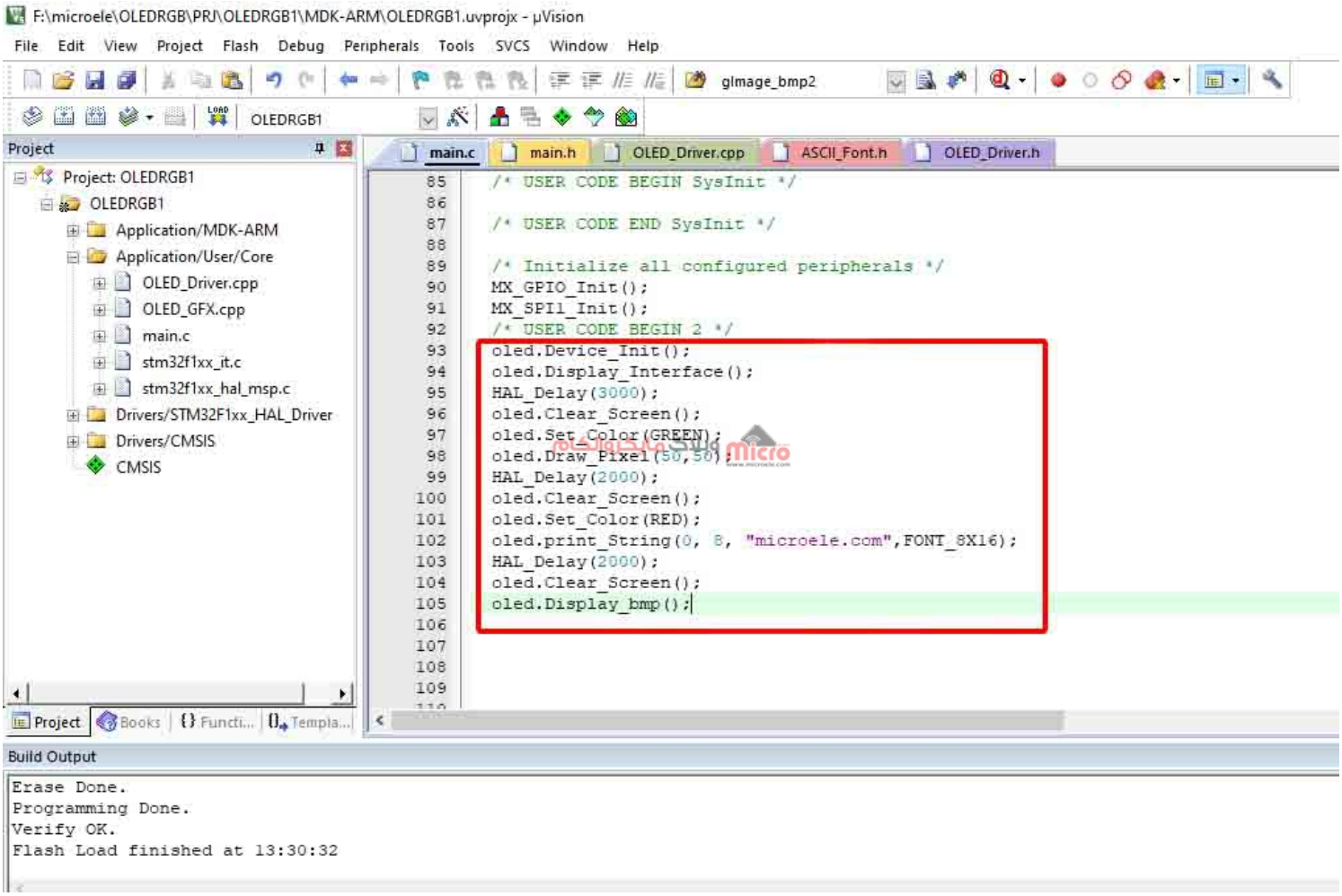


آرایه حاوی کد تصویر جهت نمایش



اعمال تنظیمات در کتاب‌خانه جهت نمایش آرایه مد نظر

در نهایت کدهایی که در فایل main.c نوشته شده به صورت زیر خواهد بود:



کد کاملا نوشته شده در فایل *main.c*

```
oled.Device_Init();
oled.Display_Interface();
HAL_Delay(3000);
oled.Clear_Screen();
oled.Set_Color(GREEN);
oled.Draw_Pixel(50,50);
HAL_Delay(2000);
oled.Clear_Screen();
```

```
oled.Set_Color(RED);
oled.print_String(0, 8, "microele.com",FONT_8X16);
HAL_Delay(2000);
oled.Clear_Screen();
oled.Display_bmp();
```

علاوه بر توابع گفته شده در این مطلب، توابع کاربردیِ دیگری نیز در کتاب‌خانه استفاده شده وجود دارد. علاقه‌مندان می‌توانند با باز کردن فایل‌های OLED_Driver و OLED_GFX اقدام به مطالعه و استفاده از آن‌ها نمایند. همچنین می‌توانید از فایل main.c در پوشه Demo نیز برای مطالعه بیشتر استفاده کنید.

نتیجه گیری

در این مطلب نحوه راه‌اندازیِ نمایشگر OLED رنگی برای میکروهای STM32 توضیح داده شد. در بخش قبل، پیکربندیِ اولیه در محیط STM32CubeMX انجام شد. در این بخش نحوه اضافه کردن کتاب‌خانه‌های OLED در محیط Keil و پیکربندی، جهت استفاده از توابع آن و نحوه کامپایل فایل‌های C++ در محیط Keil بررسی گردید. همچنین برخی از توابع موجود در این کتاب‌خانه، برای ترسیم شکل و درج نوشته و تصویر بر روی ماژول OLED همراه با مثال توضیح داده شد.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوال داشتید درباره این آموزش لطفا اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ microelecom# به اشتراک بگذارید و پیج مایکروالکام (microelecom@) رو هم منشن کنید.