

## پیوست ۱: معرفی منابع

الف) منابع شرکت ST:

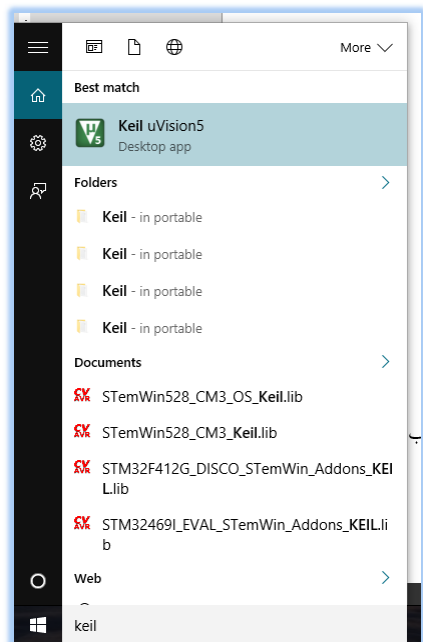
1. STM32F429xx datasheet ([link](#))
2. STM32F429 reference manual ([link](#))
3. Discovery kit with STM32F429ZI user manual ([link](#))
4. STM Cortex-M4 programming manual ([link](#))

ب) کتاب‌ها و منابع جانبی:

۱. میکروکنترلرهای ۳۲ بیتی ARM، ویرایش دوم، رضا سپاس‌یار، انتشارات فدک ایستاتیس
۲. مرجع کاربردی میکروکنترلرهای ARM، س. سهرابی، به صورت نسخه‌ی الکترونیکی رایگان از سایت [armkits.ir](http://armkits.ir) قابل دریافت است.
۳. معماری پردازشگر ARM، مصطفی پروین، به صورت نسخه‌ی الکترونیکی و رایگان قابل دریافت است.
۴. طراحی با میکروکنترلرهای STM32، حمید نجفی، مؤسسه فرهنگی هنری دیباگران تهران
۵. پروژه‌های عملی با میکروکنترلرهای AVR، علیرضا ملکی، حمزه نساج‌پور، انتشارات یاوریان
۶. کانال تلگرامی STM32: <https://t.me/STM32hal>
7. <http://www.iranmicro.ir/forum/>
8. <http://www.keil.com/>

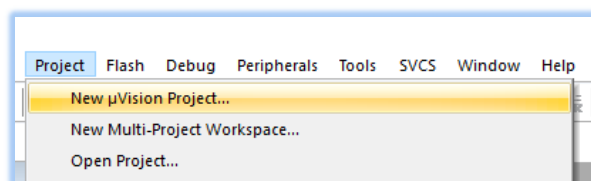
## پیوست ۲: آموزش ایجاد یک پروژه جدید در Keil

ابتدا بعد از نصب برنامه آن را اجرا کنید.



شکل ۲۱

سپس برای ایجاد پروژه‌ی جدید از طریق منوی Project گزینه‌ی New  $\mu$ Vision Project را انتخاب کنید.

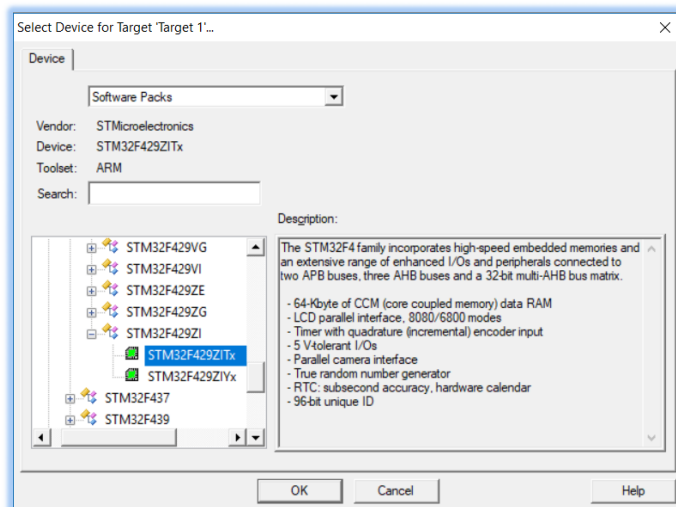


شکل ۲۲

در پنجره‌ی باز شده نام مناسبی برای پروژه‌ی خود انتخاب کرده و آن را در آدرس دلخواه از حافظه‌ی رایانه ذخیره کنید. پیشنهاد می‌شود برای هر پروژه یک فولدر جدید باز کنید تا انبوه فایل‌های تولید شده اشتباه نشوند. در مرحله‌ی بعد، پنجره‌ای باز شده و از شما می‌خواهد میکروکنترلر خود را انتخاب کنید. برای این کار از شاخه‌ی STMicroelectronics به ترتیب موارد زیر را انتخاب کنید:

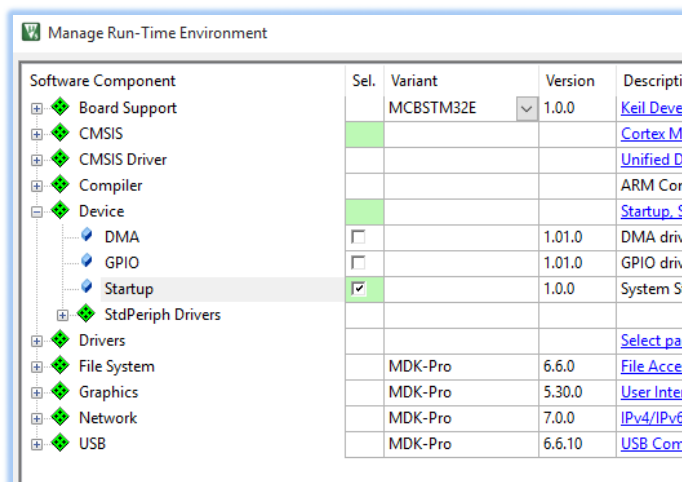
STM32F4 series  $\rightarrow$  STM32F429  $\rightarrow$  STM32F429ZI  $\rightarrow$  STM32F429ZITx

سپس روی OK کلیک کنید.



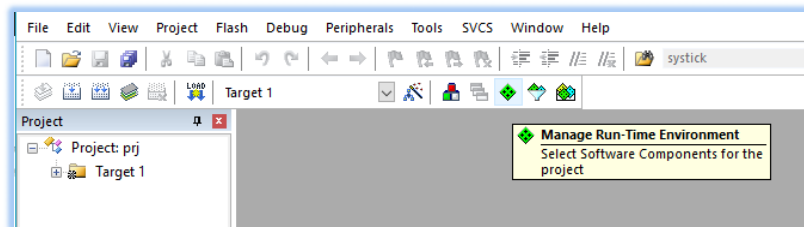
شکل ۲۳

پس از آن پنجره‌ی Manage Run-Time Environment باز می‌شود که می‌توانید کتابخانه‌های مورد نیاز خود را به پروژه اضافه نمایید. در اینجا از قسمت Device، تیک مربوط به Startup را فعال کنید. سپس کلید Resolve را بزنید تا کتابخانه‌هایی که Startup از آن‌ها استفاده می‌کند نیز به پروژه اضافه شوند. نهایتاً کلید OK را بزنید.



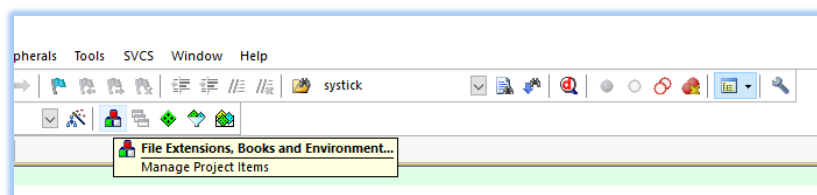
شکل ۲۴

در مراحل بعدی مجدداً می‌توانید این پنجره را از محیط Keil باز کرده و کتابخانه‌های مورد نیاز را به پروژه بیفزایید.



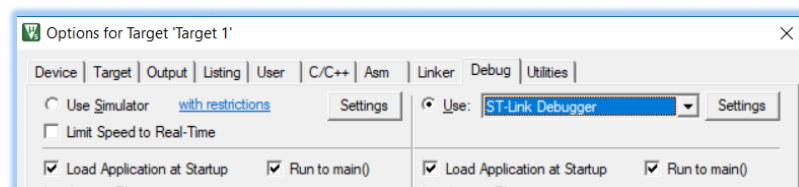
شکل ۲۵

در صورتی که فایل از پیش آماده شده‌ای برای برنامه خود ندارید می‌توانید از طریق منوی file و گزینه‌ی new فایل جدید ساخته و آن را با پسوند c. ذخیره کنید و سپس از طریق گزینه‌ی File Extension, ... آن را به پروژه اضافه نمایید.



شکل ۲۶

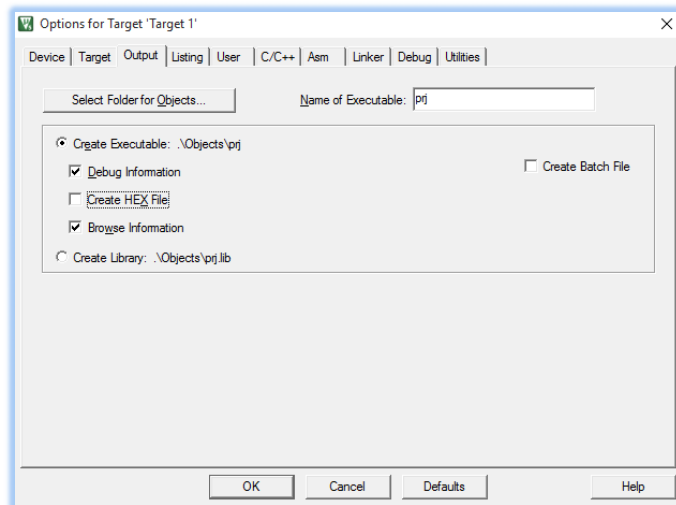
برای تنظیمات مورد نیاز از منوی Flash گزینه‌ی configure flash tools را انتخاب کرده و وارد تب Debug شوید و دیباگر خود را به ST-Link تغییر دهید.



شکل ۲۷

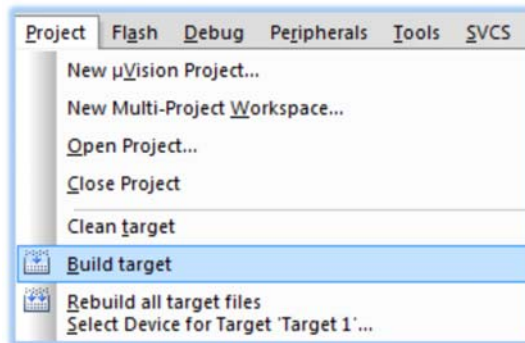
هم‌چنین در تب Target دقت کنید فرکانس کریستال وارد شده، با فرکانس کریستال روی برد یکسان باشد تا در مراحل دیباگ و برنامه‌ریزی مشکلی ایجاد نشود.

سپس وارد تب output شوید و تیک create hex file را بزنید. از این طریق فایل hex برنامه نوشته شده بعد از کامپایل، در اختیار شما قرار خواهد گرفت. از این فایل جهت پروگرام کردن میکرو با استفاده از سایر نرم‌افزارها می‌توان استفاده نمود. در صورتی که مستقیماً از داخل Keil جهت پروگرام کردن میکروکنترلر اقدام می‌نمایید، نیازی به اعمال این گزینه نخواهید داشت.



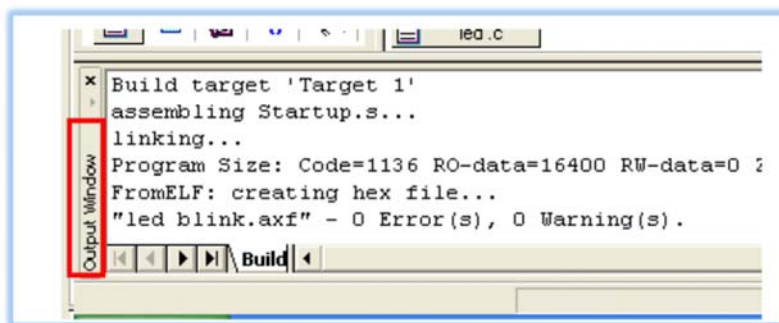
شکل ۲۸

بر روی OK کلیک کرده و تغییرات اعمال شده را ذخیره نمایید. برای کامپایل برنامه‌ی خود از منوی Project گزینه‌ی build target را انتخاب کنید یا از کلید میان‌بر F7 استفاده کنید تا عملیات کامپایل اجرا گردد.



شکل ۲۹

خطاها و اخطارهای برنامه‌ی نوشته شده بعد از کامپایل در قسمت output window به همراه جزئیات نمایش داده خواهد شد. صفر بودن تعداد خطاهای برنامه برای برنامه ریزی تراشه ضروری است.

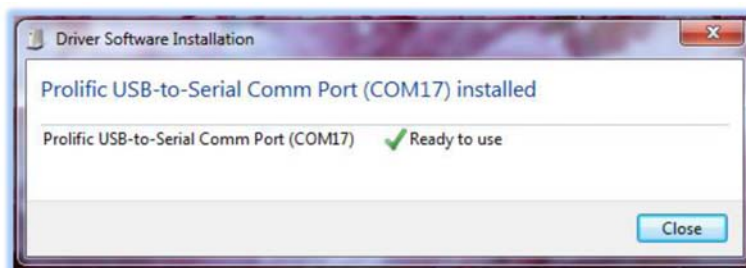


شکل ۳۰

### پیوست ۳: آموزش پروگرام کردن با استفاده از Flash Loader Demonstrator

نرم افزار Flash Loader Demonstrator به منظور برنامه ریزی میکروکنترلرهای شرکت ST از طریق بوت لودر آن ها از طریق USART ارائه شده است. سورس اصلی برنامه به زبان C# نوشته شده و در دسترس عموم قرار دارد. در ادامه آموزش استفاده از آن آورده شده است.

جهت برنامه ریزی میکروکنترلر، بایستی ابتدا پورت سریال رایانه را به میکروکنترلر متصل نمود. در اینجا به دلیل عدم وجود پورت سریال بر روی بسیاری از لپ تاپ ها، از یک مبدل serial2usb استفاده می کنیم. این مبدل به منزله ی یک پورت COM مجازی بر روی سیستم عمل خواهد کرد. ابتدا درایور مربوط به این مبدل<sup>۱</sup> را بر روی سیستم خود نصب کنید. سپس کابل مبدل را به رایانه ی خود وصل نمایید. پیغامی مانند شکل زیر ظاهر می شود که شماره پورت COM مجازی ایجاد شده را نشان می دهد (علاوه بر این، می توانید شماره ی پورت را از قسمت Device Manager ویندوز نیز پیدا کنید).



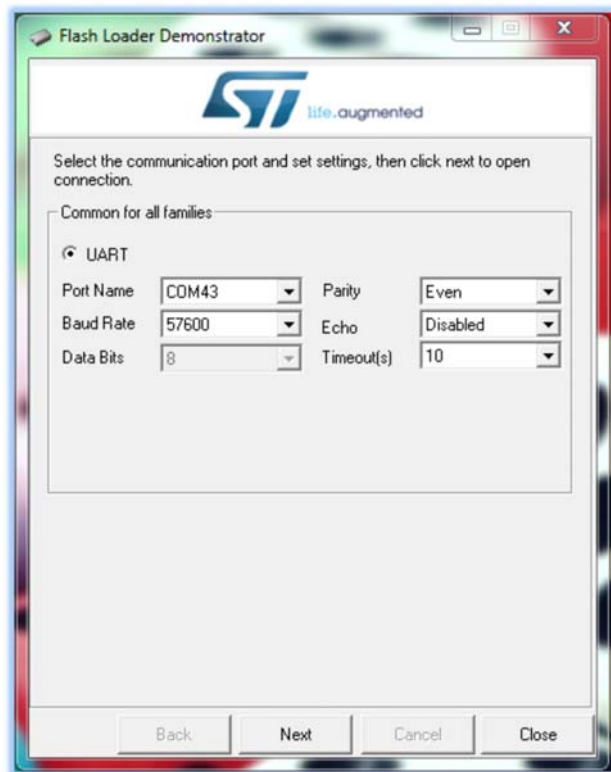
شکل ۳۱

میکروکنترلر را در حالت اجرای برنامه ی بوت لودر روشن کنید.

پورت USB بخش مبدل USB به سریال (USB2TTL) را به کامپیوتر متصل نمایید .

اکنون نرم افزار Flash Loader Demonstrator را باز کنید. نمای اولیه ی این برنامه در شکل زیر آورده شده است.

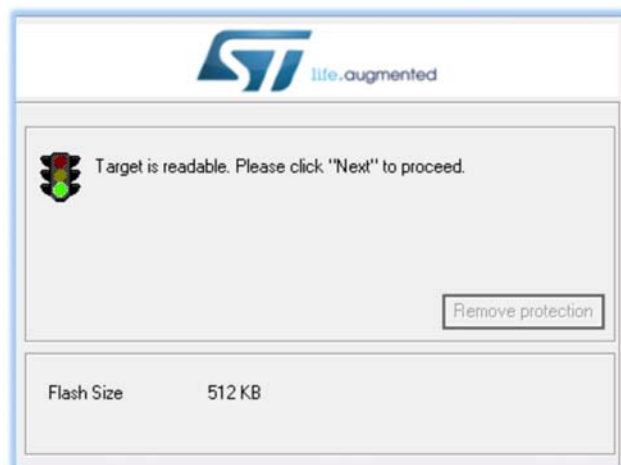
<sup>۱</sup> در اینجا از آی سی مبدل PL2303TA استفاده شده است.



شکل ۳۲ - نرم افزار Flash loader demonstrator

شماره‌ی پورت صحیح را وارد کرده و روی Next کلیک کنید.

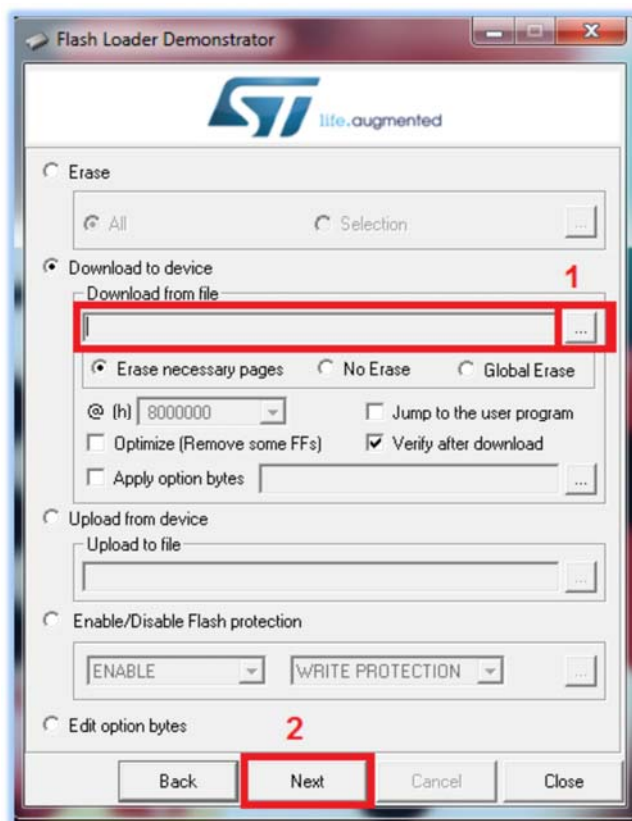
در صورتی که مراحل قبلی به درستی انجام شوند با صفحه زیر مواجه خواهید شد که در آن ظرفیت حافظه Flash میکروکنترلر شناسایی شده است. در غیر این صورت مراحل قبلی را دوباره تکرار کنید.



شکل ۳۳

کلید Next را در این صفحه و صفحه بعدی فشار دهید تا به صفحه اصلی برنامه منتقل شوید.

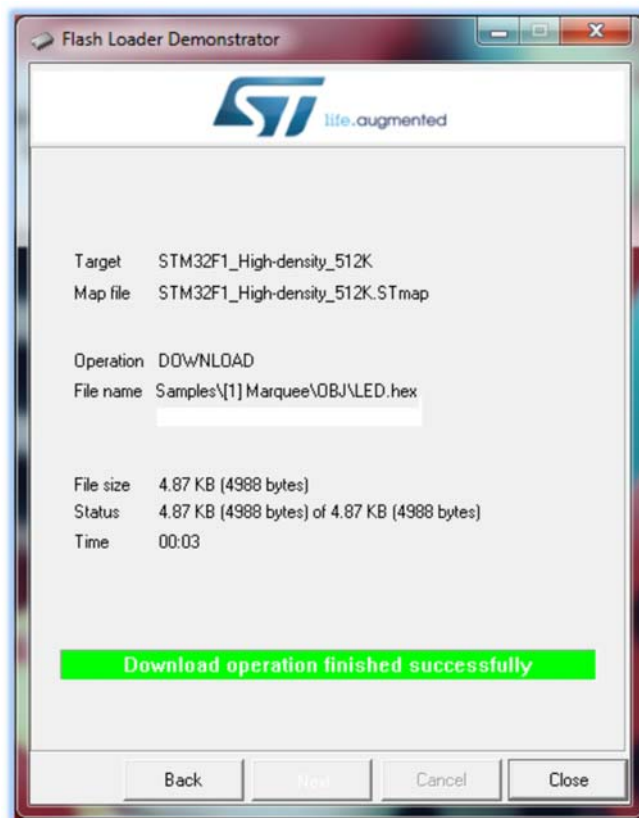
مسیر فایل HEX مورد نظر را مطابق شکل زیر با فشردن کلید ... مشخص نمایید و برای بازبینی پروگرام صحیح میکروکنترلر تیک گزینه Verify after download را بزنید.



شکل ۳۴

کلید Next را فشار دهید تا عملیات انتقال فایل شروع شود. پس از چند ثانیه پیغام زیر نشان می‌دهد که عملیات به خوبی انجام شده است.





شکل ۳۵

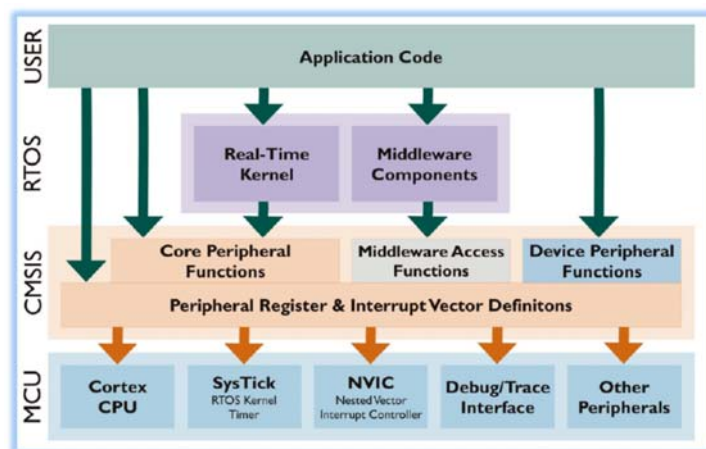
## پیوست ۴: نحوه‌ی برنامه‌نویسی Keil در STM32F4xx

برنامه‌نویسی میکروکنترلرهای سری Cortex بر اساس استاندارد انجام می‌شود که در قالب چند کتابخانه مشهور به CMSIS<sup>۱</sup> ارائه شده است. این استاندارد ابتدا توسط خود ARM ایجاد شده است و سپس توسط شرکت‌های سازنده‌ی میکروکنترلر مبتنی بر هسته‌ی ARM توسعه و پشتیبانی یافته است. هدف از تولید این استاندارد، ساده‌سازی برنامه‌نویسی برای استفاده‌کنندگان و کاهش پیچیدگی برنامه‌ها و هزینه‌های تولید و همچنین قابلیت تبدیل سریع کدهای نوشته شده برای یک محصول جهت استفاده در سایر محصولات است. CMSIS به سه لایه تقسیم می‌شود:

Core peripheral access layer (CPAL): این لایه در پایین‌ترین سطح قرار دارد و جهت ارتباط با رجیسترهای هسته و دیباگ نمودن و NVIC تعبیه شده است. این لایه توسط خود شرکت ARM ارائه می‌گردد.

Middleware access layer (MAL): این لایه در حال حاضر در دست توسعه است.

Device peripheral access layer (DPAL): آدرس رجیسترهای سخت افزاری و سایر تعاریف در این لایه وجود دارند. این لایه را شرکت سازنده هر میکروکنترلر به طور مجزا ارائه می‌کند.



شکل ۳۶

ساختار نام فایل‌ها در CMSIS به شکل زیر است که هنگام ایجاد پروژه‌ی جدید، باید موارد مورد نیاز را به پروژه اضافه نمود.

core\_cm3.h, core\_cm3.c, <device>.h, system\_<device>.h, system\_<device>.c,

<sup>۱</sup> Cortex Microcontroller Software Interface Standard

که به طور خاص برای میکروکنترلر مورد استفاده در آزمایشگاه به جای <device> باید stm32f4xx قرار گیرد.

چگونه با رجیسترها کار کنیم؟ برنامه‌ریزی رجیسترهای سخت‌افزاری را می‌توان با اضافه کردن کتابخانه‌های مربوط به لایه‌ی DPAL انجام داد. اما در اینجا این کار را بدون استفاده از این کتابخانه‌های اضافی انجام می‌دهیم. این کار را با یک مثال آموزش می‌دهیم. می‌خواهیم رجیستر x را که در بخش y قرار دارد مقداردهی 0x1F کنیم، به صورت زیر عمل می‌کنیم.

```
Y->X = 0x1F;
```

توجه کنید که در عبارت فوق y و x با حروف بزرگتر نوشته می‌شوند. نام رجیسترهای هر بخش در منوال میکروکنترلر آورده شده است. مثال: چهار پایه‌ی اول از پورت A را یک کنید:

```
GPIOA->BSRR = 0x0000000F;
```

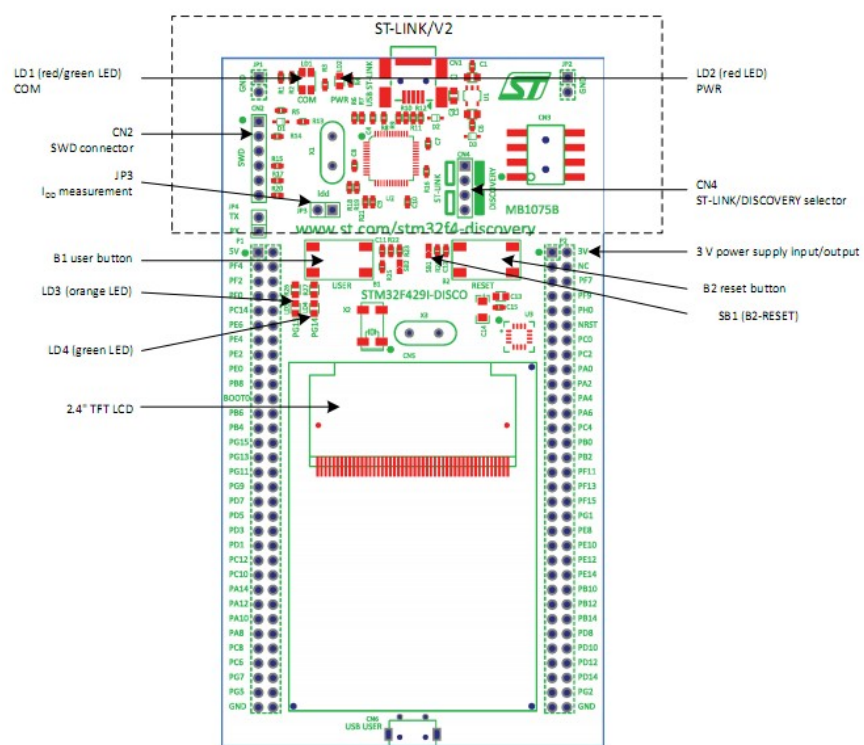
از منظر برنامه‌نویسی، با توجه به نماد >- می‌توان فهمید که در عبارت فوق، GPIOA یک اشاره‌گر<sup>۱</sup> به یک ساختار<sup>۲</sup> است که BSRR یک متغیر تعریف شده در آن ساختار است.

---

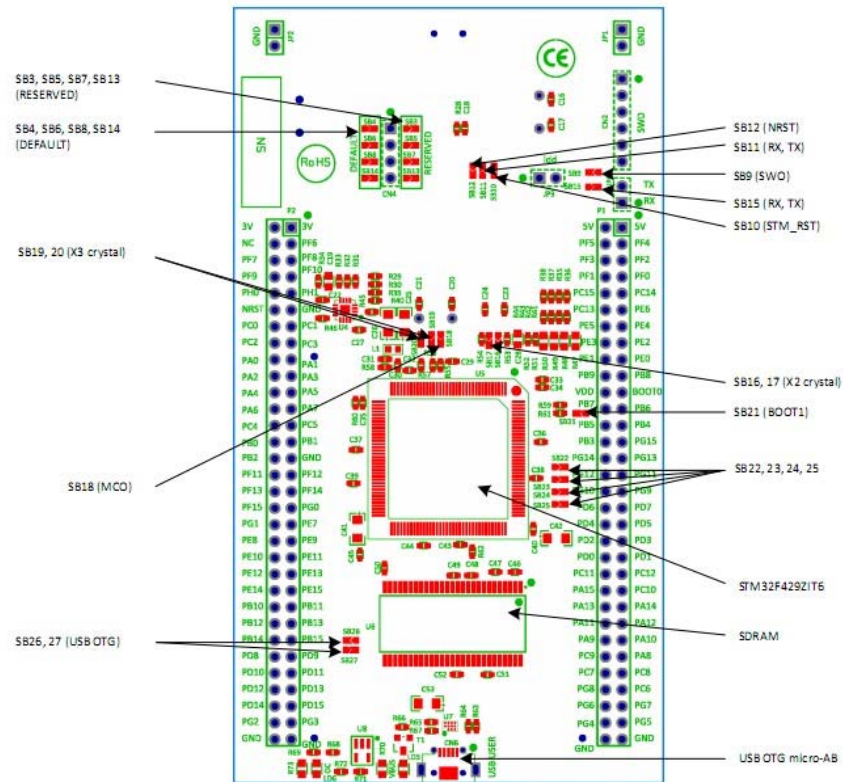
<sup>1</sup> Pointer

<sup>2</sup> Structure

## پیوست ۵: مشخصات برد آموزشی



شکل ۳۷ - نمای روبه‌رویی برد discovery-F429



شکل ۳۸ - نمای پشتی برد discovery-F429