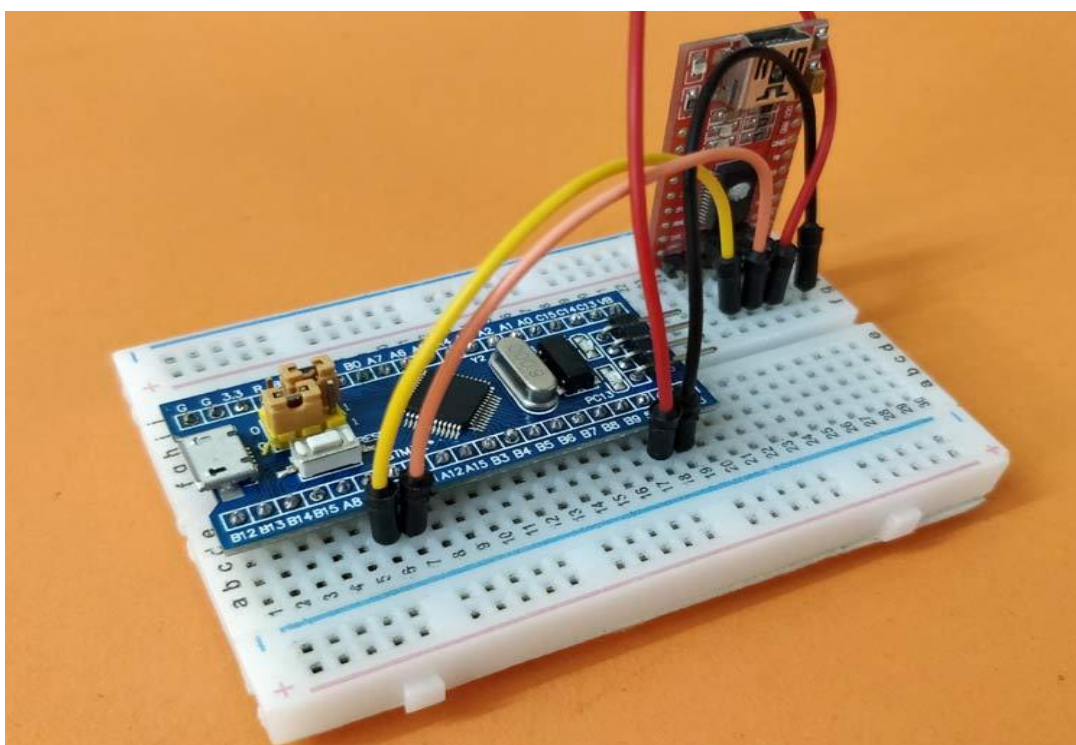


آموزش کار با میکروکنترلرهای STM32 با استفاده از Arduino IDE:

و پروژه LED چشمک زن

بوردهای آردوینو معمولا اولین انتخاب کسانی هستند که از روی سرگرمی و تفننی به سراغ پروژه‌های الکترونیکی می‌آیند. البته ناگفته نماند که انتخاب بسیاری مهندسان حرفه‌ای نیز در پروژه‌های سبک همین بوردها هستند. این که تاکید می‌کنیم که در پروژه‌های سبک و تفننی، به این علت است که هرچه وارد پروژه‌های جدی و عمیق‌تر مانند پروژه‌های عظیم صنعتی بشویم، خواهیم دید که CPU هشت بیتی این بوردها و سرعت بسیار پایین‌شان چیزی شبیه شوخی به نظر می‌رسد.

به عنوان جایگزین، بوردهای STM32F103C8T6 را داریم (مانند برد Blue pill که با CPU ای ۳۲ بیتی و معماری ARM Cortex M3، به مراتب عملکرد قابل قبول‌تری نسبت به آردوینوها دارند. خبر بسیار خوبی که در اینجا وجود دارد این است که برای کار کردن با این میکروهای STM32 و پروگرام کردن آنها، می‌توانیم با خیال آسوده از همان Arduino IDE استفاده کنیم که برای بوردهای آردوینو همیشه استفاده کرده و از بر هستیم. بنابراین، در این آموزش قصد داریم که ابتدا میکروکنترلر STM32 و برد Blue pill را معرفی کنیم و سپس با انجام یک پروژه ساده یعنی ایجاد LED چشمک زن (با استفاده از LED ای که بر روی خود برد وجود دارد)، پروگرام کردن این میکروها را با استفاده از Arduino IDE یاد بگیریم.

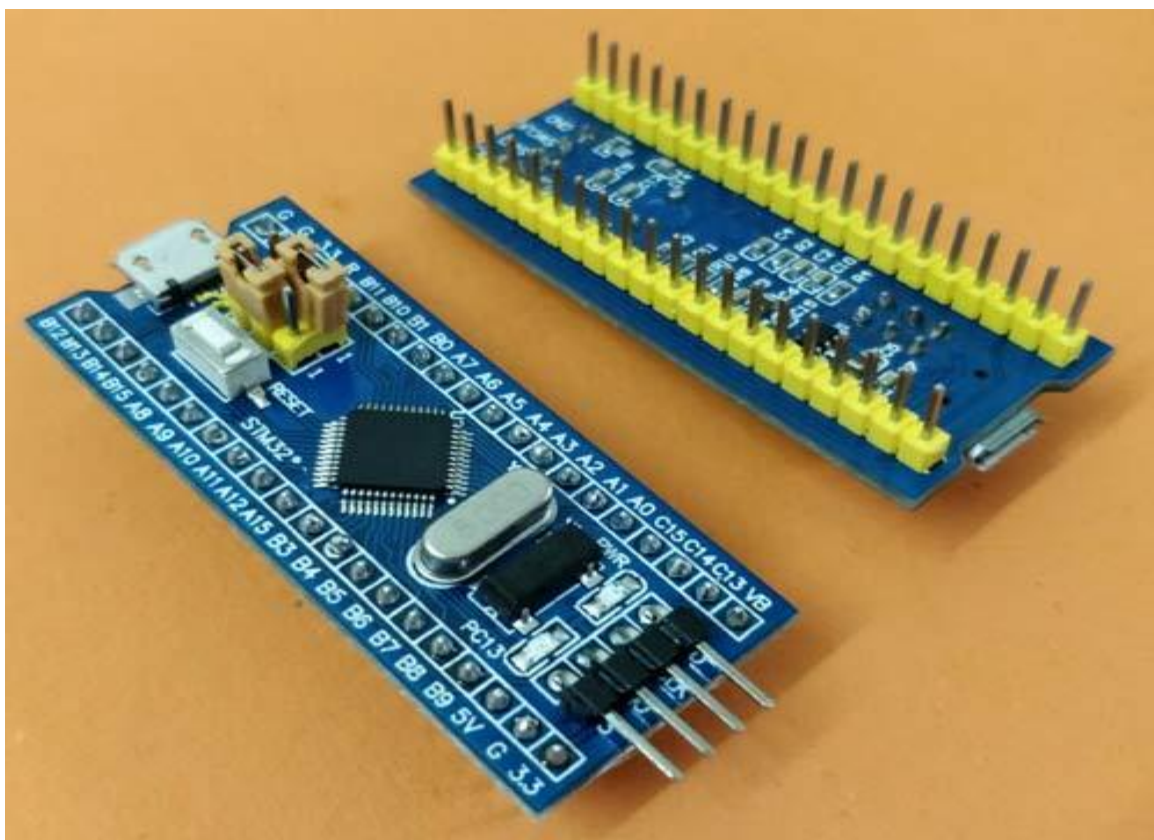


آنچه برای انجام پروژه نیاز داریم

- برد توسعه‌ی (Blue pill) STM32F103C8T6
- پروگرامر FTDI
- برد برد
- سیم برد بردی
- لپ‌تاپ متصل به اینترنت

مقدمه‌ای برای آشنا شدن با برد Blue Pill

برد STM32 یا همان Blue Pill ، یک میکروکنترلر ARM Cortex M3 است. این برد تا حدود زیادی به برد Arduino Nano شبیه است اما ساختار یکپارچه‌تری نسبت به آن دارد. تصویر این برد را در عکس زیر می‌توانید ببینید.



خوبی دیگری که بوردهای STM32 نسبت به بوردهای متداول آردوینو دارند قیمت پایین و معقول آنها و سخت افزار این سورسی است که دارند. میکروی مورد استفاده در آنها ساخته‌ی شرکت STMicroelectronics است و به جز کریستالی که خود میکرو دارد، برد نیز دارای دو کریستال اسیلاتور جداگانه است که دارای فرکانس ۸ MHz و ۳۲ KHz هستند. با استفاده از این اسیلاتور ها می‌توان (Real Time Clock) RTC را درایو کرد. در نتیجه میکرو می‌تواند در deep sleep mode هم عمل کند که در سیستم‌هایی که از محدودیت و حساسیت مصرف توان برخوردار هستند، امتیاز بسیار مهمی محسوب می‌شود.

از آنجایی که میکرو با ولتاژ ۳.۳ ولت کار می‌کند، بر روی برد دو آی‌سی ولتاژ رگولاتور ۳.۳ ولت و ۵ ولت نیز وجود دارند. رگولاتور ۳.۳ ولت تغذیه میکرو و رگولاتور ۵ ولت تغذیه پین‌ها را تامین می‌کند. تمام پین‌های میکروکنترلر در برد قابلیت دسترسی دارند و بر روی هر کدام label نام آن پین نیز وجود دارد. برد همچنین دارای دو عدد LED نیز می‌باشد. یکی به رنگ قرمز که نشان دهنده‌ی روشن و خاموش بودن برد است و یکی به رنگ سبز که برای GPIO ها استفاده دارد و به پین PC13 متصل است. دو عدد header pin نیز بر روی برد تعبیه شده‌اند که کمک می‌کنند میکرو بین دو مود مختلف programming و operation جابه‌جا شود. در ادامه‌ی آموزش در مورد این دو مود بیشتر توضیح خواهیم داد.

اما اگر شما هم جز آن دسته افرادی هستید که برایتان سوال شده چرا نام این برد را Blue pill گذاشته‌اند، درست‌ترین پاسخی که می‌توانیم به شما بدهیم این است که ما هم از علت دقیق آن بی‌خبر هستیم! تنها حدسی که می‌توانیم سر هم کنیم این است که شاید چون رنگ برد آبی است و چون برد بسیار قدرتمند و کار راه اندازی است، سازندگان آن به این نتیجه رسیده‌اند که چنین نامی برایش انتخاب کنند. بهر حال مهم نیست، خیلی روی آن حساس نشوید.

مشخصات برد STM32F103C8T6

همان‌طور که گفتیم، در این برد از میکروکنترلر ARM Cortex M3 STM32F103C8 استفاده شده است. برخلاف نام Blue pill که گفتیم فلسفه‌ی دقیق نام‌گذاری آن بر کسی واضح نیست، پشت انتخاب نام STM32F103C8T6 یک فلسفه‌ی دقیق وجود دارد.

- STM از ابتدای نام شرکت سازنده‌ی این میکروها گرفته شده است، STMicroelectronics.
- عدد ۳۲ نشان دهنده‌ی ۳۲ بیتی بودن میکروکنترلرهای ARM مورد استفاده است.
- F103 به منظور نشان دادن معماری مورد استفاده یعنی ARM Cortex M3 است.
- C نماد داشتن ۴۸ پین است.
- عدد ۸ از وجود یک ۶۴ KB Flash memory خبر می‌دهد.
- T نوع پکیج استفاده شده برای میکرو را نشان می‌دهد، LQFP :
- و عدد ۶ محدودی دمای کاری میکرو را که از -۴۰ درجه سلسیوس تا +۸۵ درجه سلسیوس است.

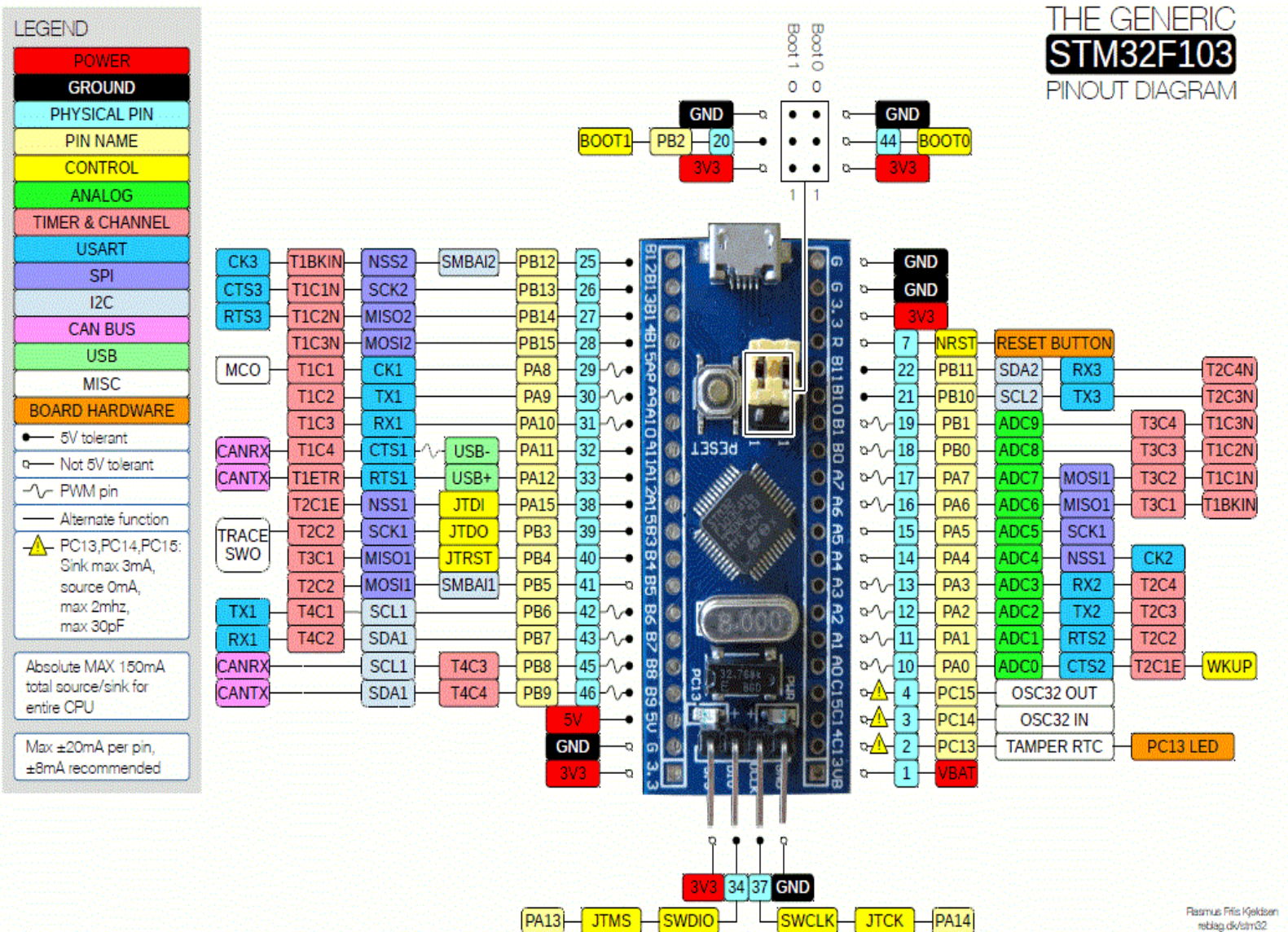
بسیار خوب، پس از پرداختن به فلسفه‌ی نام گذاری میکرو؛ بهتر است به مشخصات آن بپردازیم.

- معماری مورد استفاده: ۳۲ bit ARM Cortex M3
- ولتاژ کاری: ۲.۷V تا ۳.۶V
- فرکانس CPU: ۷۲ MHz
- تعداد پین‌های GPIO: سی و هفت عدد
- تعداد پین‌های PWM: دوازده عدد
- پین‌های آنالوگ ورودی: ۱۰ (۱۲ بیتی)
- تعداد پریفرال‌های USART: سه عدد
- تعداد پریفرال‌های I2C: دو عدد
- تعداد پریفرال‌های SPI: دو عدد
- تعداد پریفرال‌های CAN2.0: یک عدد
- تایمر: ۲ عدد (۱۶ بیتی)، یک عدد (PWM)
- حافظه ۶۴ KB: FLASH
- حافظه ۲۰ KB: RAM

بسیار خوب، تا همین جا کفایت می‌کند. اما اگر علاقه‌مند بودید بیشتر با مشخصات فنی این برد آشنا شوید به دیتاشیت آن مراجعه کنید. همچنین ممکن است سر زدن به این لینک و این لینک هم شما را با اطلاعات جالب و بیشتری در مورد این برد مواجه کند؛ البته باز هم در صورتی که علاقه‌مند هستید.

جزئیات و نقشه پایه‌های STM32

در تصویر زیر، نقشه‌ی کامل پایه‌های برد Blue pill را می‌بینید. همان‌طور که می‌بینید، برچسب نام هر پایه در مقابل آن زده شده است. شیوه‌ی نام گذاری پایه‌ها تقریباً مشابه بردهای آردوینو است. مثلاً از G برای نمایش ground یعنی پایه‌ی زمین استفاده شده است. و یا مثلاً از ۳.۳ ولت برای نشان دادن خروجی رگولاتور ۳.۳ ولتی. پایه‌ی ۵ ولت هم می‌تواند به عنوان ورودی برای دریافت تغذیه‌ی ۵ ولتی برد استفاده شود و هم به عنوان خروجی به عنوان یک ولتاژ ۵ ولتی مورد استفاده قرار گیرد. حالت دوم در صورتی است که برد از طریق USB تغذیه‌رسانی شود. LED ای که بر روی خود برد تعبیه شده است نیز به پایه‌ی PC13 متصل است.



Rasmus Fris Kjeldsen
reblog.dk/stm32



V1.0

برخلاف بوردهای آردوینو، برد STM32 باید به صورت دستی در مود پروگرم شدن قرار بگیرد. این کار به وسیله‌ی جامپرهای boot 0 و boot 1 انجام می‌شود.

البته معمولاً با وضعیت boot 1 کاری نداریم و کفایت که boot 0 jumper را در وضعیت اتصال به ۳.۳ ولت قرار دهیم تا میکرو در مود پروگرم قرار گیرد. چنانچه آن را به زمین متصل کنیم نیز در مود عملکردی (operation) خواهد بود. در بخش آپلود کردن کد بر روی برد مجدداً این موضوع را مرور خواهیم کرد.

نحوه پروگرم کردن برد blue pill

از آنجا که STM32 از تولیدات شرکت STMicroelectronics است، بسیاری از قواعد پروگرم کردن آن بسیار مشابه به فرآیند مشابه در میکروهای ARM است.

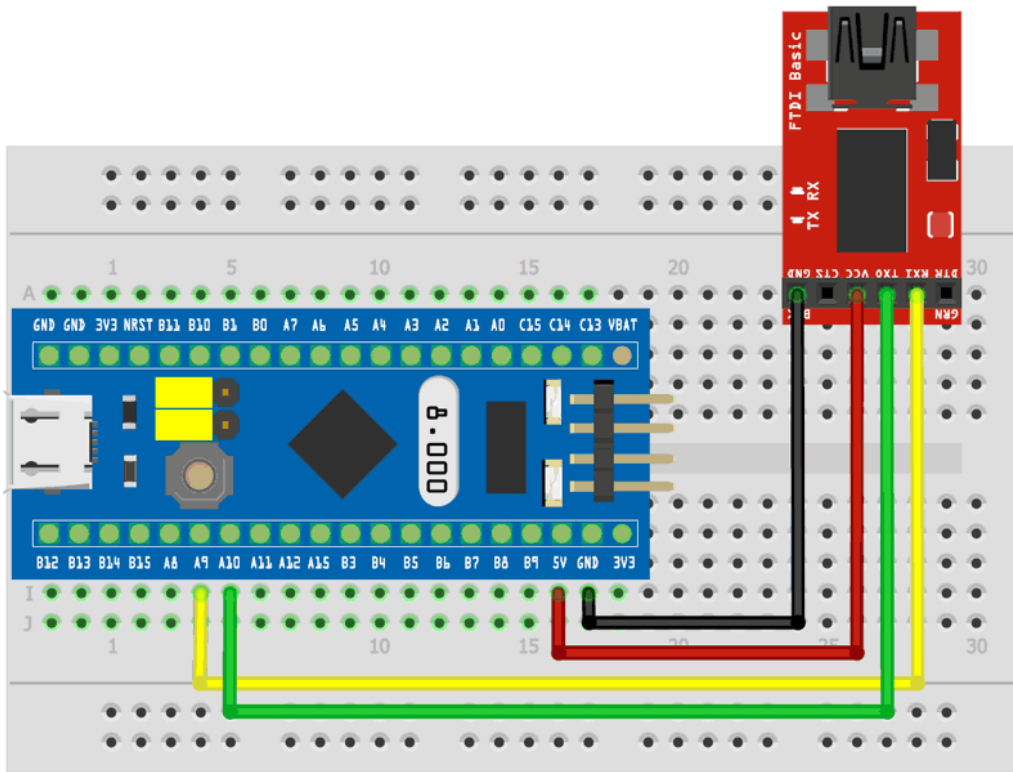
یکی از IDE های مشهور برای پروگرم کردن این بوردها، Keil ARM MDK است. همچنین می‌توانیم از موارد دیگری هم به این منظور استفاده کنیم؛ برای مثال موارد زیر را می‌توان نام برد.

Ride 7, Crossworks ARM, MicroC Pro ARM, Atollic TrueStudio, IAR workbench, PlatformIO+STM32 و ...

اما مهم‌ترین علت محبوبیت برد STM32، این است که پروگرم کردن آن علاوه بر تمام پلتفرم‌های بالا از طریق Arduino IDE نیز امکان‌پذیر است. به این ترتیب در صورتی که کار کردن با زبان ساده‌ی بوردهای آردوینو و IDE محبوب و کاربرپسند آنها را از قبل بدانیم، دیگر نیازی نخواهد بود که برای آموختن نحوه‌ی پروگرم کردن بوردهای STM32 نیز زمان مجزایی صرف کنیم. تمام کتابخانه‌ها نیز از قبل آماده هستند. در ادامه‌ی این آموزش نیز ما از Arduino IDE استفاده خواهیم کرد.

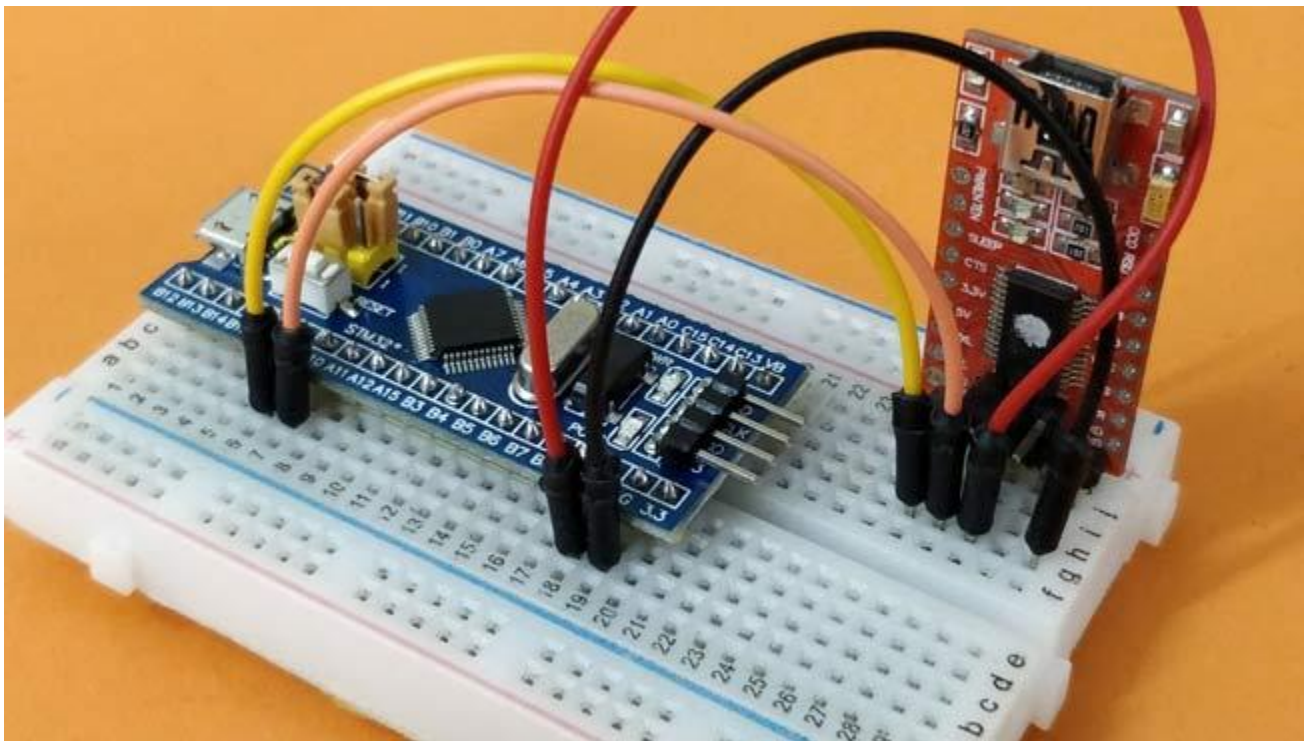
نمودار مدار

برای پروگرم کردن مستقیم برد STM32 از طریق Arduino IDE، لازم است که از ارتباط سریال با FTDI استفاده کنیم. پروگرامر FTDI، مانند تصویر زیر به پین‌های Rx و Tx میکرو متصل می‌شود. پین Vcc پروگرامر را نیز به پین ۵ ولت STM32 وصل می‌کنیم و زمین هر دو برد را نیز به یکدیگر متصل می‌کنیم. پین‌های Rx و Tx نیز به ترتیب به پین‌های (A9 پین Tx میکرو) و (A10 پین Rx میکرو) میکروکنترلر متصل می‌شوند.



FTDI >> STM32
 Gnd >> Gnd
 Vcc >> 5V
 Rx >> A9
 Tx >> A10

fritzing



پروگرم کردن STM32 بدون استفاده از ارتباط سریال و از طریق Micro USB

اگر این سوال به ذهن‌تان رسیده است که چرا مانند بوردهای آردوینو در اینجا نیز از طریق ارتباط USB میکرو را پروگرم نمی‌کنیم؟ پاسخ این است که بله، به آن طریق نیز امکان‌پذیر است اما ما در این جلسه نمی‌خواهیم به روش و چگونگی آن بپردازیم. نکته این است که برد STM32 به صورت پیش‌فرض و زمانی که آن را خریداری می‌کنیم، bootloader ندارد که بتواند با Arduino IDE ارتباط برقرار کند. اما می‌توان به صورت دستی این بوت لودر را بر روی برد فلش کرد و از آن به بعد پورت usb برای ارتباط با کامپیوتر و پروگرم مستقیم آماده خواهد بود.

آماده‌سازی Arduino IDE به منظور پروگرم کردن STM32

برای دانلود (اگر از قبل ندارید) و آماده‌سازی این IDE به منظور پروگرم کردن برد STM32 مراحل زیر را انجام دهید.

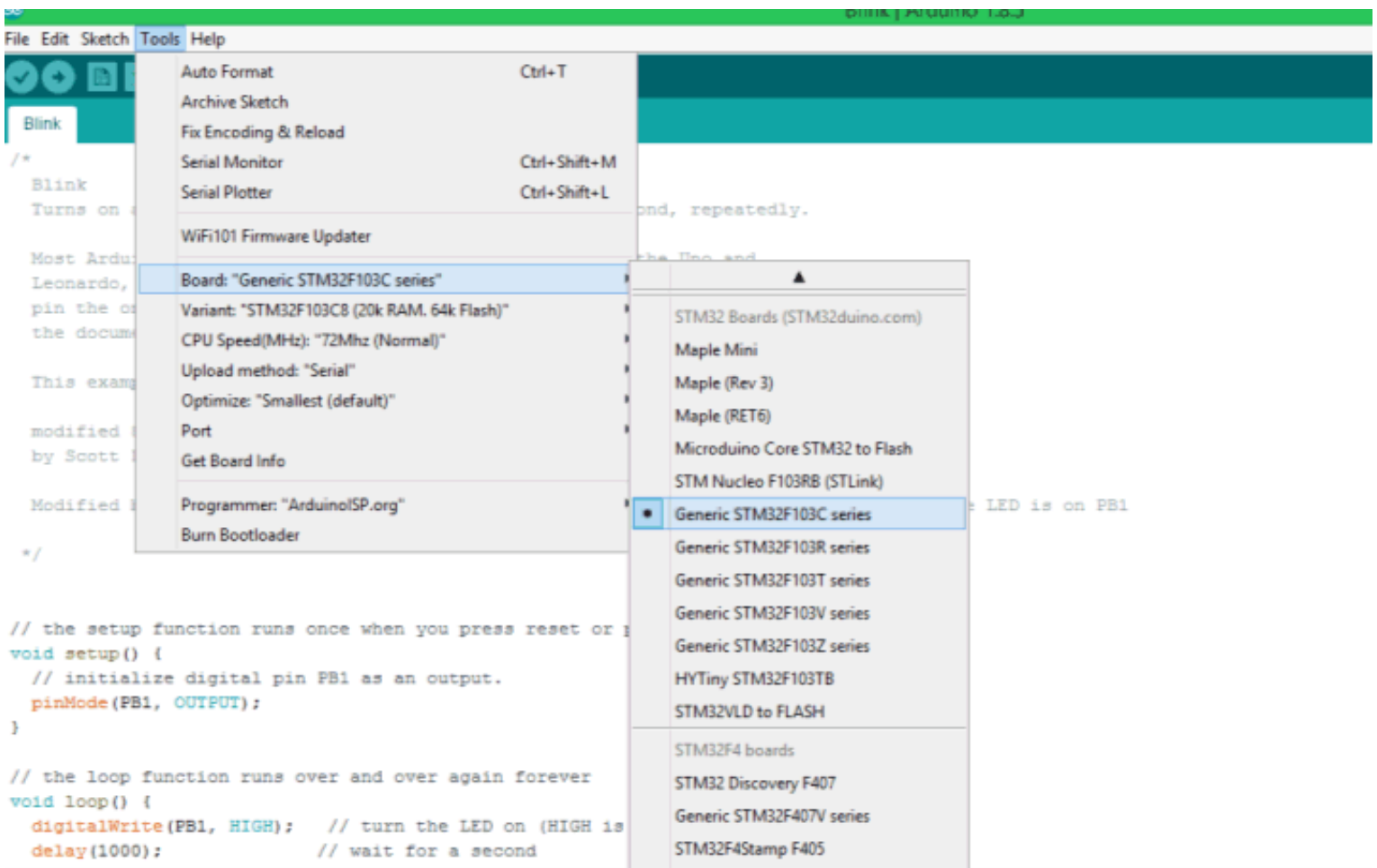
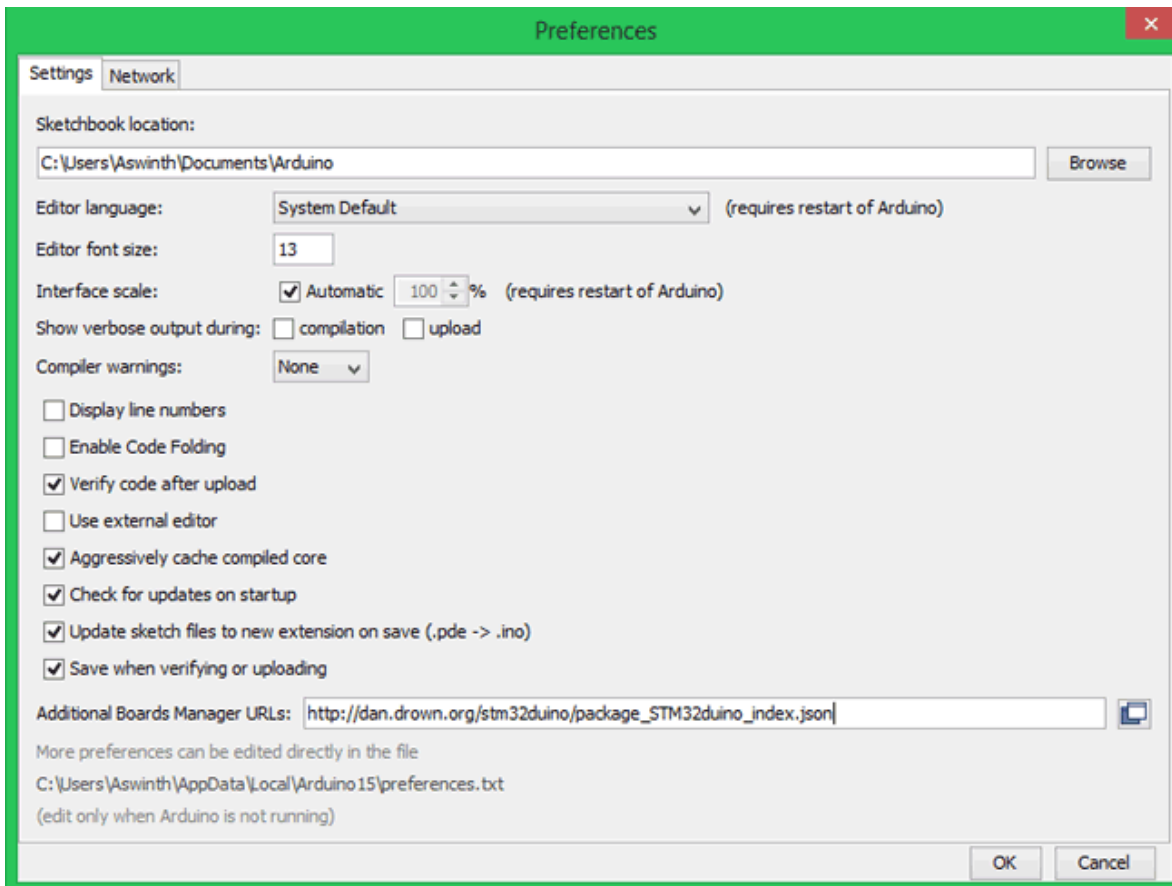
مرحله اول: در صورتی که Arduino IDE را نصب نکرده‌اید، آن را از این لینک دانلود و نصب کنید. فقط دقت داشته باشید که نسخه‌ی متناسب با سیستم عامل خودتان را دانلود کنید.

مرحله دوم: پس از نصب IDE، پکیج‌های لازم برای STM32 را نیز دانلود و نصب کنید. (از این مسیر: selecting File -> Preferences)

مرحله سوم: اگر بر روی Preferences کلیک کنید، پنجره‌ای مانند تصویر زیر باز خواهد شد. در بخش additional Boards Manager URL که در قسمت پایین این پنجره می‌بینید؛ لینک زیر را کپی کنید.
http://dan.drown.org/stm32duino/package_STM32duino_index.json
و OK کنید.

مرحله چهارم: حالا به Board Manager -> Boards -> Tool بروید و در پنجره‌ی باز شده STM32F1 را جستجو کنید. پکیج‌های نشان داده شده را نصب کنید.

مرحله پنجم: پس از نصب شدن پکیج‌ها، به Tools بروید و از آنجا صفحه را آنقدر پایین بکشید تا STM32F103C را بیابید (مانند تصویر زیر) سپس در آنجا ببینید که نوع (variant) حتماً روی ۶۴kFlah، سرعت CPU بر روی ۷۲MHz و روش آپلود بر روی Serial باشد.



مرحله ششم: حالا بورد FTDI را متصل کنید و از طریق device manager چک کنید که به کدام پورت COM متصل شده است. همان شماره پورت را در Tools->Port وارد کنید.

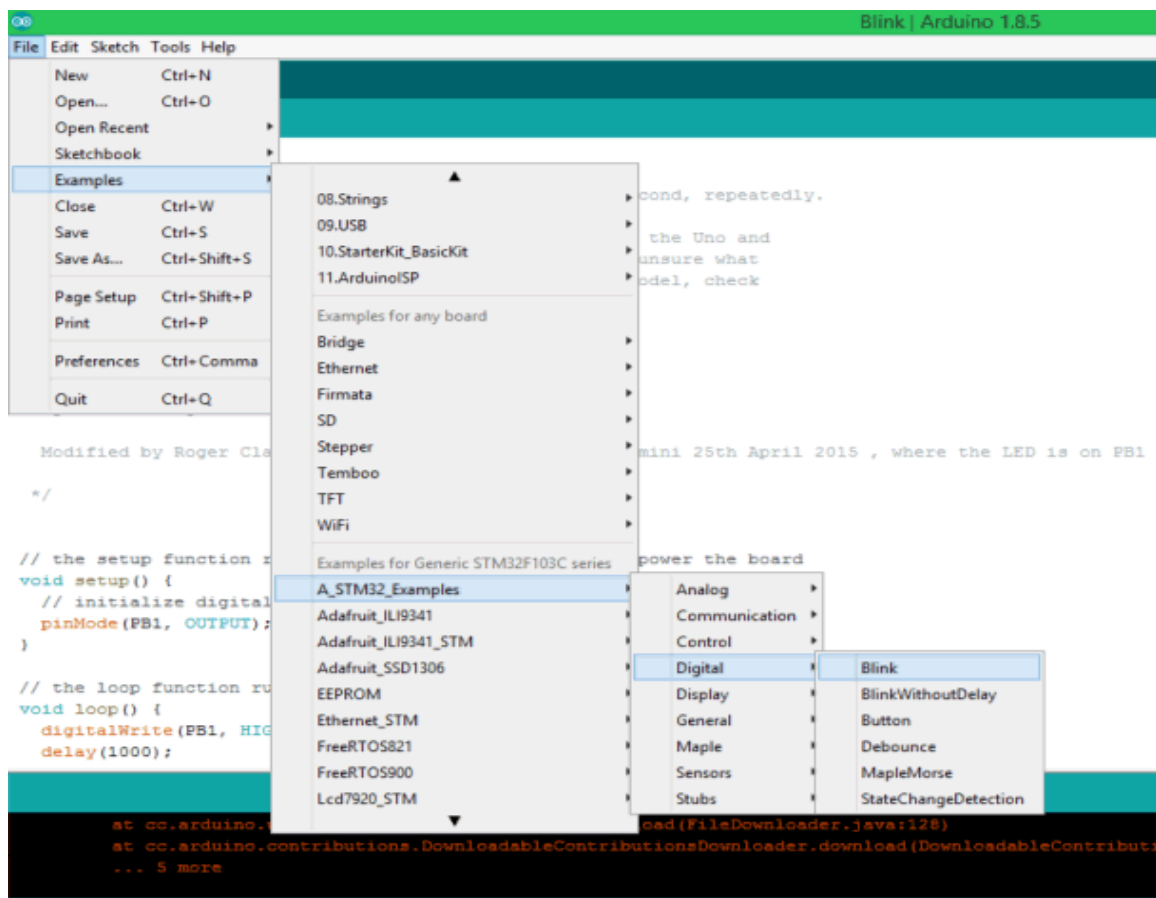
مرحله هفتم: پس از انجام تمام این تغییرات، گوشه پایین سمت راست صفحه‌ی Arduino IDE را ببینید که آیا پیغامی مانند تصویر زیر ظاهر شده است یا خیر. پورت ما COM7 بوده است و پورت استفاده شده‌ی شما ممکن است متفاوت باشد.



در اینجا Arduino IDE کاملاً برای پروگرام کردن STM32 آماده است.

آپلود کردن کد پروژه‌ی LED چشمک زن بر روی STM32

برای آنکه مطمئن شویم که تمام مراحل را درست انجام داده‌ایم، یک پروژه‌ی ساده را مانند LED چشمک زن در نظر می‌گیریم و تلاش می‌کنیم آن را بر روی بورد STM32 با استفاده از Arduino IDE پیاده کنیم. این برنامه‌ی ساده را در مسیر نشان داده شده در تصویر زیر می‌توانیم پیدا کنیم.



وقتی که برنامه باز می‌شود، ابتدا باید تغییرات کوچکی را در آن ایجاد کنیم. چیزی که باید تغییر دهیم پورت اتصال LED است. این پورت به صورت پیش‌فرض در برنامه‌ی موجود PB1 است اما ما باید آن را به PC13 که LED برد STM32 به آن متصل است تغییر دهیم. کد کامل این برنامه را که اصلاح فوق نیز در آن انجام شده است، در انتهای این آموزش می‌توانید ببینید.

در بخش زیر، کدی که در قسمت تابع حلقه (loop) وجود دارد را می‌توانیم ببینیم. در این کد تعیین می‌کنیم که این پین برای هر ۱۰۰۰ میلی ثانیه یک بار، در وضعیت HIGH و LOW ولتاژ قرار بگیرد تا به این ترتیب LED هر یک ثانیه یک بار روشن و خاموش شود (در مجموع حالت چشمک زن خواهد داشت) این فرآیند به صورت بی‌نهایت تکرار خواهد شد چون در تابع حلقه نامحدود هستیم.

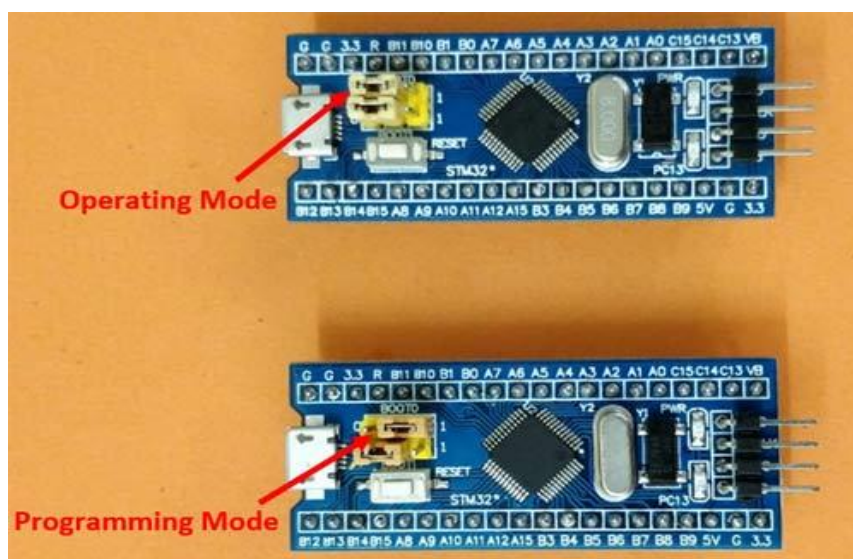
```
digitalWrite(PC13, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000);           // wait for a second

digitalWrite(PC13, LOW); // turn the LED off by making the voltage LOW

delay(1000);           // wait for a second
```

خب، همان‌طور که در اوایل جلسه گفتیم، برای پروگرام کردن STM32 ابتدا باید آن را در مود پروگرام قرار دهیم. به این منظور boot 0 jumper را در وضعیت low قرار می‌دهیم. این مود و مود عملکردی را برای این jumper در تصویر زیر می‌توانید ببینید.



پس از انجام عمل فوق و قرار دادن میکرو در مود پروگرم، یک بار کلید RESET را فشار می‌دهیم. به محض فشار دادن این کلید، برد وارد مود پروگرم شده و چراغ سبز رنگ آن خاموش می‌شود. حالا همه چیز آماده است که برنامه بر روی برد آپلود شود.

در IDE، گزینه‌ی Upload را بزنید تا برنامه یک دور کامپایل و سپس بر روی برد آپلود شود. اگر همه چیز به درستی پیش برود تصویر زیر را در IDE خواهید دید.

```
Done uploading.
Wrote address 0x08002300 (71.04%)
Wrote address 0x08002400 (73.07%)
Wrote address 0x08002500 (75.10%)
Wrote address 0x08002600 (77.13%)
Wrote address 0x08002700 (79.16%)
Wrote address 0x08002800 (81.19%)
Wrote address 0x08002900 (83.22%)
Wrote address 0x08002a00 (85.25%)
Wrote address 0x08002b00 (87.28%)
Wrote address 0x08002c00 (89.31%)
Wrote address 0x08002d00 (91.34%)
Wrote address 0x08002e00 (93.37%)
Wrote address 0x08002f00 (95.40%)
Wrote address 0x08003000 (97.43%)
Wrote address 0x08003100 (99.46%)
Wrote address 0x08003144 (100.00%) Done.
Starting execution at address 0x08000000... done.
```

نکته: گاهی اوقات ممکن است پروسه‌ی پروگرم شدن به درستی تکمیل نشود اما IDE پیغام Done successful بدهد. بنابراین در تصویر فوق که کنسول IDE است، همواره چک کنید که خطوط تا تکمیل ۱۰۰ درصدی را نشان داده باشند.

عملکرد پروژه‌ی LED چشمک زن بر روی STM32

چنانچه همه چیز به خوبی پیش رفته باشد، خواهید دید که LED سبز رنگ روی برد در فاصله‌های یک ثانیه‌ای شروع به روشن و خاموش شدن خواهد کرد. این عملکرد را در ویدئوی بخش زیر هم می‌توانید ببینید. اگر حوصله داشتید می‌توانید کد را بیشتر هم دستکاری کنید و مثلاً فاصله‌های بین چشمک زدن‌ها را کم و زیاد کنید.

نکته‌ی مهمی که وجود دارد، اگر می‌خواهید که دفعه‌ی بعدی که این برد را روشن می‌کنید همین برنامه بر روی آن اجرا شود، لازم است که پس از اتمام بارگذاری کد بر روی آن، boot 0 را مجدداً به مود عملکردی برگردانید.