Lab Series

Jedzia

ARM Development Boards

Reference Document for Electronic Laboratory Work

First Edition

ARM Development Boards

Reference Document for Electronic Laboratory Work

ARM Development Boards

Reference Document for Electronic Laboratory Work

First Edition

Edited by

Jedzia

Here goes ISBN, Author Info, publishing info

Preface to the First Edition

The algorithmic way of life is best.
Hermann Weyl

Lorem IpsumOnsequossit que eriorem elenime ndaecum re vellam sam, tem eosae porem. Tatempe rferianda quam re, occum elliquati volores tiberchitiam que non rem quo occabor eiciae culparionsed ea simust alic teniminvelit que voloreiur alignat urerate cus et dunt ea voloris excerum et laboreped ent aut qui dis dipsum doluptatquis eum es nullori busdam rem quo earuptatem dolore opturibus ulpa volorpo stotatis des ipsam, sitae nus eum sum aribear chillest illat volecus, inctus voloreped ut vollit at es dolorec uptati nis am quis velis excerferia ni con eossusda doluptiatis aut volum am nos aspelendaes ea nonsedigene landi dignatio di qui ad quis nost ius dit, sitatur, explign atemporro vellabo reptia parions ectate ducillu ptatur ad esequia volorume pernatur ra suntis consequatur, cones magnitin nonecuptatet lab ipsum fugitia nos modit hil mod ut aut re, voluptaturi dolori te vel molorestiae nobitat.
Eperorum volut latur? Qui sequatiorro consecte velestrum facerferes aut laboremos mi, niet alitecus sam qui delest ut la as nullab ius endi conseque por aspernatem conet, omnis ad et que la nonsequate voluptatur?
Lor secabor a quiatiu mquatetur magnatquo mi, conseque erferro quis aceaquam que et laboris citatempero volupta nis dolore eos sed eatur? Olorehenim harum vendem quunt, con parum a cum sit vellandios volescipsunt ilitios moles et aut aspeditatia alit rerferum re illigni mendel min ped quam lam es incieni hicimet lautem rest ommolenis molorerferio vel inum iumquia nobisti andaepe riorepudant estotaqui ut idebis dolupta dolori diciis eaquis ra quidest, niminvelit licipiet eaquatur, ipsandam natibus expe audaecaecto blabore moluptas vellestis et mod moluptat dolla que volorpor alibus aturibus, nonsequi sapis etur?
Ut plia sitates cuptat por modis asinctae. Volore inte atessim agnimilis ariat labori quate sunt. Boratio et et restiam vit offici omnis at ut lis intiatus nis dicium fuga. Net ipis sim erferissus sitatiis exerat.
Occulpa riones molupta eume sequam que nem nonseribus dit dolupta tionsequid ut officim hit harchitibea doluptur?

Augsburg, May 2020 Jedzia Dax

This page intentionally left blank.

Notes

List of Software

Following is an overview of the software-projects provided by me.

This is a draft. This chapter is only relevant, when the “Projects” repository is published.

The Electronic/ARM Projects Directory Guide

This is a helping hand for easy finding what you want to, without getting too much confused. For details refer to the documentation (if some exists) in the referred sub-projects.

Note: Update this document with

```
E:\Projects\Elektronik\ARM>pandoc "!Contents.md" -s -o doc/Contents.icml
```

before updating it in `ARM\doc\Software-Intro.indd`.

Directory Structure

Abbrevations: - StdPeriph: Is build or based on the Standard Peripherals library. The following quote from ST Website means that if it uses StdPeriph, then it also uses CMSIS, because StdPeriph is based on that. > The STM32F0xx Standard Peripherals library is full CMSIS compliant.

eclipse\blinkyF103

Blue Pill blink. uses stm32f1-stdperiph. Contains “diag/Trace.h”, **debug-trace** output code usage. Current battlefield for the blink template testing, that is not working.

BlackPill\SW4Test

System Workbench for STM32 example Project. StdPeriph based Uses J-Link.

BlackPill\BlinkyCM3

Makefile based with libopencm3. - Git-Export: `E:\Projects\Elektronik\ARM\BlackPill\BlackPill-libopencm3-blinky-template.zip`. - Git initialzed template, before i added eclipse support `E:\Projects\Elektronik\ARM\BlackPill\BlackPill-libopencm3-blinky-template.rar`.

BlackPill\BlackPill-libopencm3-blinky-template

Based on *BlinkyCM3* (Makefile based with libopencm3) with Eclipse support. - Git initialzed template `E:\Projects\Elektronik\ARM\BlackPill\BlackPill-libopencm3-blinky-eclipse-template.rar`.

eclipse

My personal setup of eclipse notes. Version 2019-03

```
C:\Toolchain\eclipse\eclipse-cpp-2019-03-R-win32-x86_64\
```

run via

```
@echo off
REM set PATH=C:\Toolchain\ARM\gcc-arm-none-eabi-4_8-140708\bin;C:\Toolchain\mingw-w64\x86_64-w64-mingw32-8.3.0-crt-6.0.0-multilib\bin;C:\msys64\usr\bin;C:\Toolchain\OpenOCD\OpenOCD-20190210-0.10.0\bin;%PATH%
REM set PATH=D:\Users\Jedzia.pubsiX\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc\4.8.3-2014q1\bin;C:\Toolchain\mingw-w64\x86_64-w64-mingw32-8.3.0-crt-6.0.0-multilib\bin;C:\msys64\usr\bin;C:\Toolchain\OpenOCD\OpenOCD-20190210-0.10.0\bin;%PATH%
set PATH=C:\Toolchain\ARM\GNU MCU Eclipse\ARM Embedded GCC\8.2.1-1.4-20190214-0604\bin;C:\Toolchain\mingw-w64\x86_64-w64-mingw32-8.3.0-crt-6.0.0-multilib\bin;C:\msys64\usr\bin;C:\Toolchain\OpenOCD\OpenOCD-20190210-0.10.0\bin;%PATH%

REM set PATH=D:\Program Files (x86)\CMake\bin;%PATH%

REM set TOOL_PATH=/C/Toolchain/ARM/GNU Tools ARM Embedded/8 2018-q4-major/bin

REM Used by gdb to refer to your master C:\Users\<USER>\.gdbinit
set HOME=%USERPROFILE%

REM set CC=arm-none-eabi-gcc
REM set CFLAGS_arm-none-eabi=-EL -mips32 -msoft-float
REM set AR_arm-none-eabi=arm-none-eabi-ar

REM set MAKEFLAGS=-j8

REM call C:\D\ldc2vars64.bat
rem cd E:\Projects\D

start "" "C:\Toolchain\eclipse\eclipse-cpp-2019-03-R-win32-x86_64\eclipse.exe"
```

ToDo:

This is a draft. This chapter is only relevant, when the “Projects” repository is published.

This is a draft. This chapter is only relevant, when the “Projects” repository is published.

This is a draft. This chapter is only relevant, when the “Projects” repository is published.

Reference

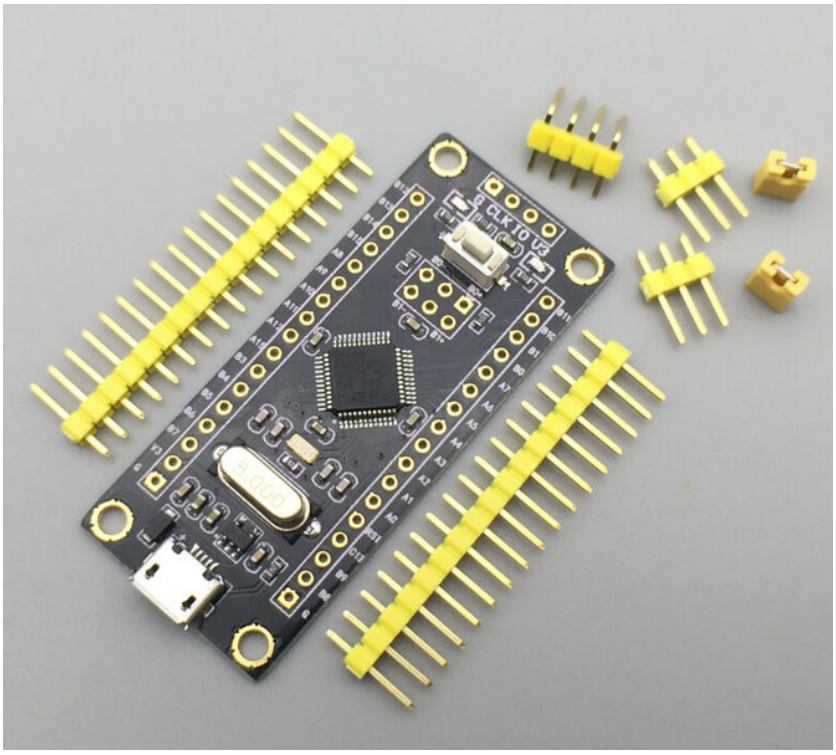
1. ***Arduino IDE***
   Download: http://arduino.cc/en/Main/Software.
   Boards Manager Package for STM32F1x series: http://dan.drown.org/stm32duino/package_STM32duino_index.json.
   Prefered Installation: https://wiki.stm32duino.com/index.php?title=Installation.
   Arduino STM32 files: https://github.com/rogerclarkmelbourne/Arduino_STM32/archive/master.zip.
2. ***ST-LINK/V2***
   https://www.st.com/en/development-tools/st-link-v2.html#tools-software
3. ***STM32duino-bootloader***
   https://github.com/rogerclarkmelbourne/STM32duino-bootloader
   With binaries at https://github.com/rogerclarkmelbourne/STM32duino-bootloader/tree/master/binaries.
   Black Pill Bootloader: https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/generic_boot20_pb12.bin?raw=true
   Maple Mini Bootloader: https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/maple_mini_boot20.bin?raw=true.

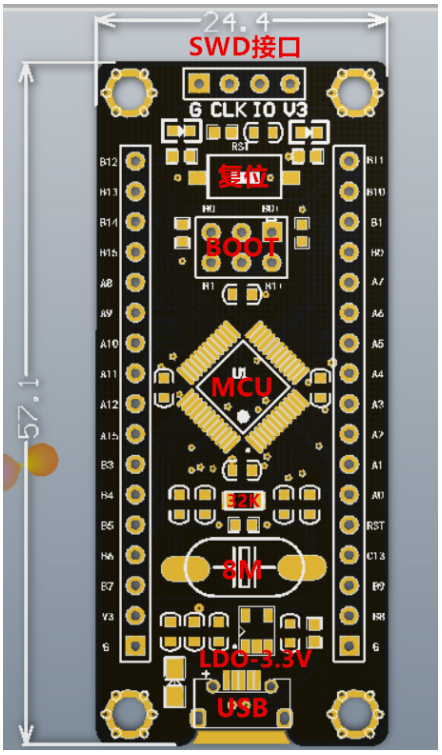
4. ***OpenBLT Bootloader***
   https://sourceforge.net/projects/openblt/
   https://github.com/feaser/openblt

This is a draft. This chapter is only relevant, when the "Projects" repository is published.

Black Pill, STM32F103 board

The Black Pill is the latest “pill” board. It is newer and less common than the Blue Pill and the Red Pill. It is available on AliExpress, eBay and Taobao.





Level of support	Moderate
Bootloader	Flash with ***[3. STM32duino-bootloader]*** or ***[4. OpenBLT Bootloader]***
Flash	64KB
RAM	20KB
User LED(s)	PB12 (active LOW)
User button(s)	None
RTC Crystal	Yes
ST-Link header	Yes
Voltage Regulator	RT9193-33 (DE=A1D) - Max 300 mA
Manufacturer	HWA YEH
Price (shipped)	2.15 USD

Part	QTY	Location	Remarks

1. Black Pill Pin-Out

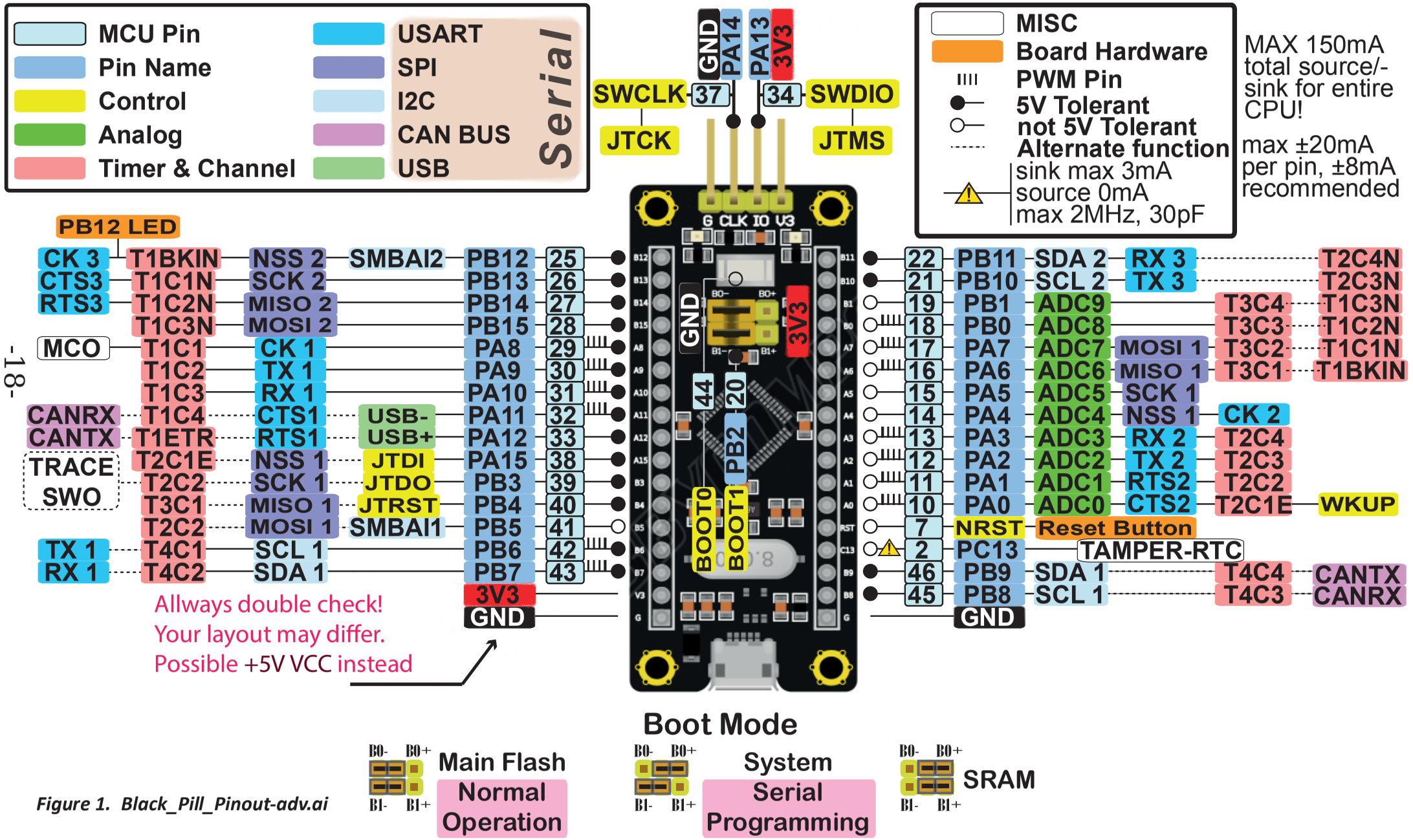
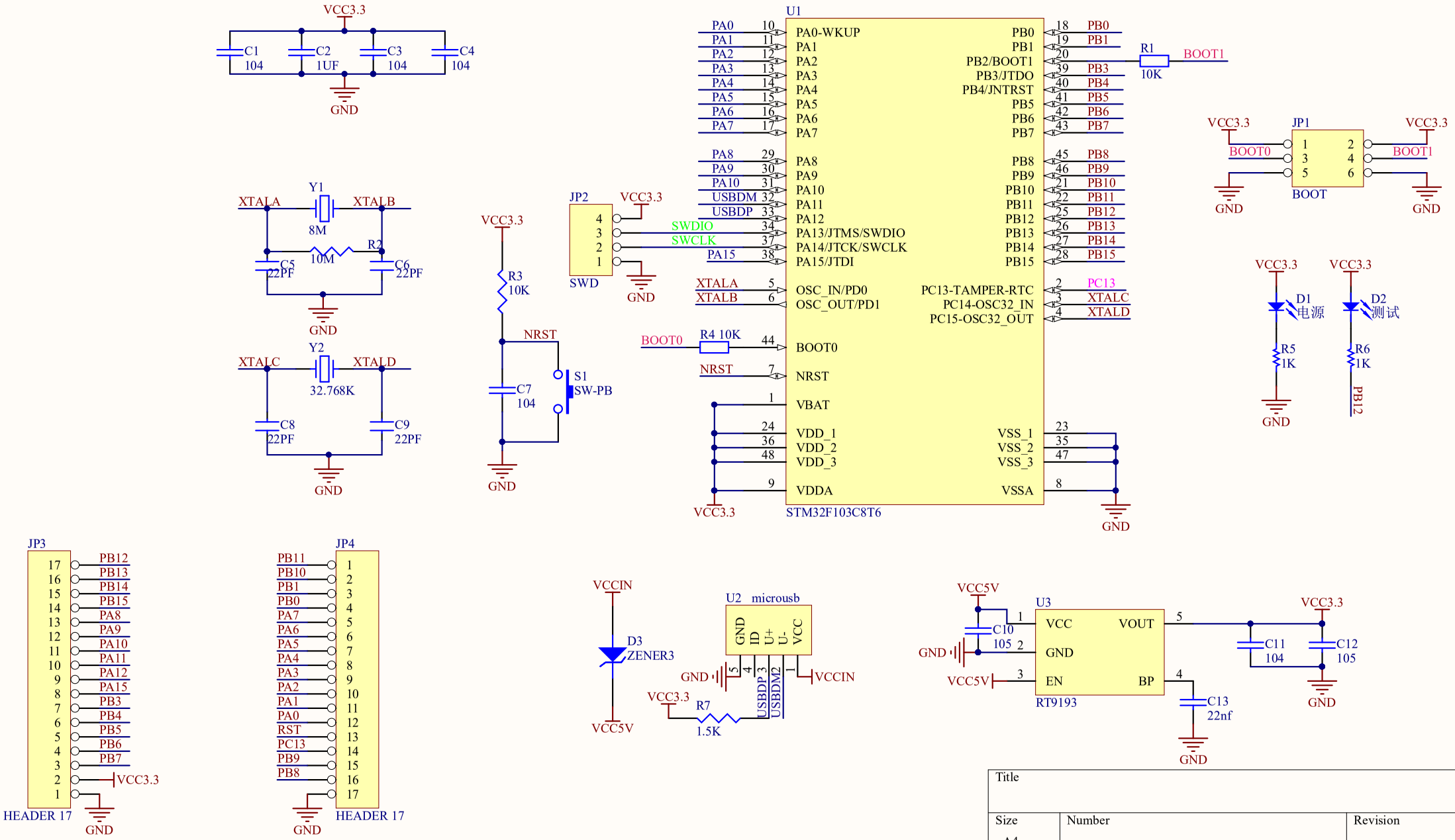


Figure 1. Black_Pill_Pinout-adv.ai

2. Black Pill Schematic



Title		
Size A4	Number	Revision
Date: 2016/9/20	Sheet of	
File: F:\项目文档\..\STM32F103C8T6核心板D	Drawn By:	

1	2	3	4

Figure 2. Black_Pill_Schematic.pdf

Comment	Description	Designator	Footprint	LibRef	Quantity
104	[NoValue], [NoValue], [NoValue], [NoValue], Capacitor	C1, C3, C4, C7, C11	0603C	CAP	5
1UF		C2	0603C	CAP	1
22PF	SMD-CAP	C5, C6, C8, C9	0603C	CAP-0.1UF	4
105	Capacitor	C10, C12	0603C	CAP	2
22nf	Capacitor	C13	0603C	CAP	1
电源		D1	0603D	LED	1
测试		D2	0603D	LED	1
ZENER3	Zener Diode	D3	0805D	ZENER3	1
BOOT		JP1	HDR2X3	HEADER 3X2	1
SWD	4 Pin Header	JP2	HDR1X4	4 HEADER	1
HEADER 17		JP3, JP4		HEADER 17	2
Res2	Resistor	R1, R4	0603B	Res2	2
10M		R2	0603B	RES1	1
10K		R3	0603B	RES1	1
1K	Resistor	R5, R6	0603B	Res1	2
1.5K		R7	0603B	RES1	1
SW-PB		S1	贴片按键	SW-PB	1
STM32F103C8T6	STM32 ARM-based 32-bit MCU with 64 Kbytes Flash, 48-pin LQFP, Industrial Temperature	U1	LQFP48_N	STM32F103C8T6	1
microusb		U2	MICRO-USB	microusb	1
RT9193		U3	SOT23-5	RT9193	1
8M	晶振	Y1	49SMD	XT	1
32.768K	晶振	Y2	32.768K晶振	XT	1

3. Getting Started

via https://wiki.stm32duino.com/index.php?title=Black_Pill using Arduino IDE. For installing and setup refer to ***[1. Arduino IDE]***. Note: The Boards-Manager method is deprecated and not supported in forum. Use the dedicated download package if you want support for other devices.

- Buy a Black Pill
- Burn the bootloader
- Under '**`Tools→Board`**' select "Generic STM32F103 series"
- Under '**`Tools→Upload`** method' select "STM32duino bootloader"
- Select the correct COM port
- Open up the blink sketch, change the pin number to PB12
- Hit upload

It is also possible to upload sketches via UART and JTAG, though this will overwrite the bootloader.

3.1. Schematic and Design

Like the other pill boards, the design is closed source and the designer is unknown.

However, a File:Black Pill Schematic.pdf is available for this board (see Figure 2 on page 19).

3.2. Comparison with the Blue Pill

The Black Pill is similar to the more popular Blue Pill.

3.3. Similarities

- STM32F103C8T6 32 bit ARM M3 microcontroller
- 4-pin ST-Link header
- Micro-USB
- No USB reset circuitry (reset over USB is still possible, though less reliable than the Maple Mini)

3.4. Differences

- Different pinout
- User LED on PB12 vs PC13 on a Blue pill
- Black pill does not have a VIN pin, it can only be powered from USB or a 3.3 V source.

3.5. Advantages of the Black Pill

- All components are on one side of the board
- Stronger Micro-USB connector
- Mounting holes
- Ships with the correct pull-up resistor on PA12 for USB port

3.6. Advantages of the Blue Pill

- More widely available, and more widely used
- Slightly cheaper
- Vbat pin can be connected to a battery for the RTC. The black pill has Vbat hardwired to 3.3V.
- Shorter and narrower
- Two additional IO pins: PC14 and PC15

4. Burning the bootloader

Most STM32F103 boards do not come with a USB bootloader installed. The main exception is the Maple mini. The STM32F4 series MCU has a built in Serial and USB (DFU) bootloader, so there is no need to install an additional bootloader.

On the STM32F103 there are 2 ways to flash the bootloader onto the board

1. Using the native bootloader (built into the STM32F103), by connecting to USART1 using a USB to serial TTL adapter.
2. Using a SWD programmer tool like the ST-Link or Black Magic Probe.

Currently it is not possible to burn the bootloader from within the Arduino IDE.

The source for the DFU-Bootloader used is the *STM32duino-bootloader* provided by Roger Clark ***[3. STM32duino-bootloader]***. An alternative Bootloader to be installed is the ***[4. OpenBLT Bootloader]***. When using *OpenBLT* you compile the bootloader first and have to substitute the binary file in the instructions for the one you created. For in-depth instructions refer to https://www.feaser.com/openblt/doku.php?id=manual:demos.

4.1. Required Hardware

It is possible to flash the bootloader using any of the following hardware:

- A USB to TTL serial adapter
  - A CH340G, CP2102, or FT232RL board can be bought for less than a dollar
    Note: Some versions of Linux (Ubuntu 14.04 and older) have a CH340G driver with no parity. You will need to update to a modern kernel or recompile the driver from source to allow uploading using this adapter. See section 4.3 on page 22 for instructions.
  - An Arduino Nano or Uno can act as a USB to serial adapter
  - The adapter would preferably be 3.3 volt, but some STM32 chips have 5V tolerant USART pins.
- A JTAG adapter
  - ST-Link section 4.2 on page 22
  - Black Magic Probe
    Most STM32 boards can be reprogrammed as a BMP

If you don't own any of the above hardware, it might be easier to start out with a Maple Mini, which comes with a USB bootloader pre-installed.

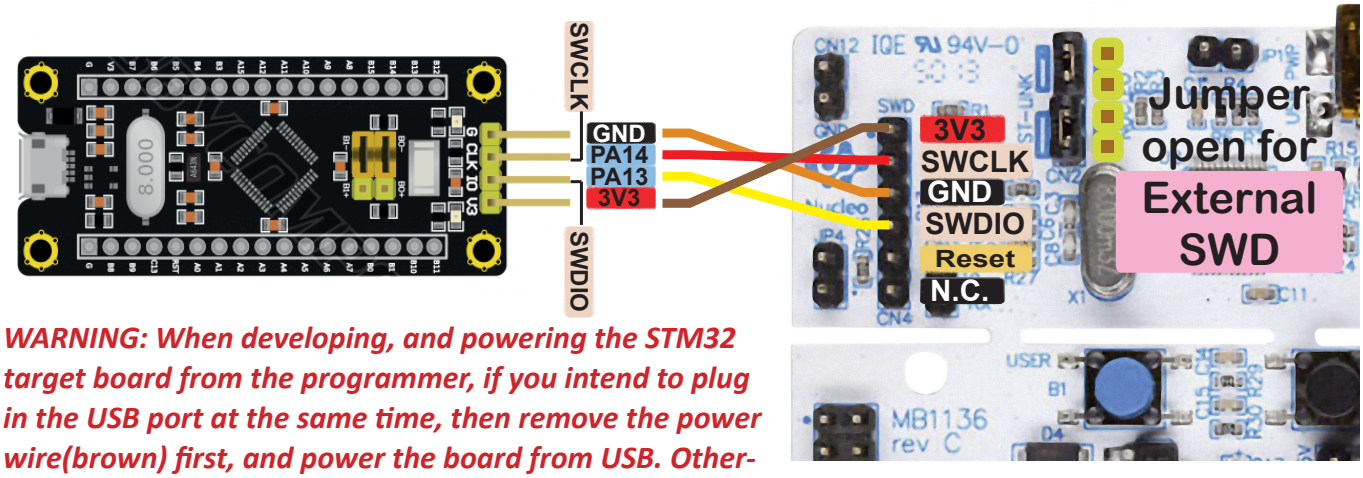
See below and/or Uploading a sketch for more on the serial and ST-Link upload methods.

4.2. Flashing the bootloader onto the Black Pill via ST-Link Interface

Obtain the software from ***[2. ST-LINK/V2]***, install driver and *STM32 ST-LINK* utility.

We need 3 wires for basic ST-Link functionality, the actual colours of the wires are not important.

- Programmer -> Target
- Ground (orange) -> GND
- SWDIO (yellow) -> PA13 - SWDIO
- SWCLK (red) -> PA14 - SWCLK



WARNING: When developing, and powering the STM32 target board from the programmer, if you intend to plug in the USB port at the same time, then remove the power wire(brown) first, and power the board from USB. Otherwise you may see some magic smoke.

To flash the Bootloader you can use the GUI-Utility, **`File→Open File(CTRL-O)`** `generic_boot20_pb12.bin` and transfer it to the device with **`Target→Program & Verify(CTRL-P)`**.

Or use the command-line interface tool:

```
ST-LINK_CLI.exe -c SWD -P generic_boot20_pb12.bin 0x8000000 -Rst -Run
```

under Linux:

```
./stlink/st-flash write generic_boot20_pb12.bin 0x8000000
./upload-reset
```

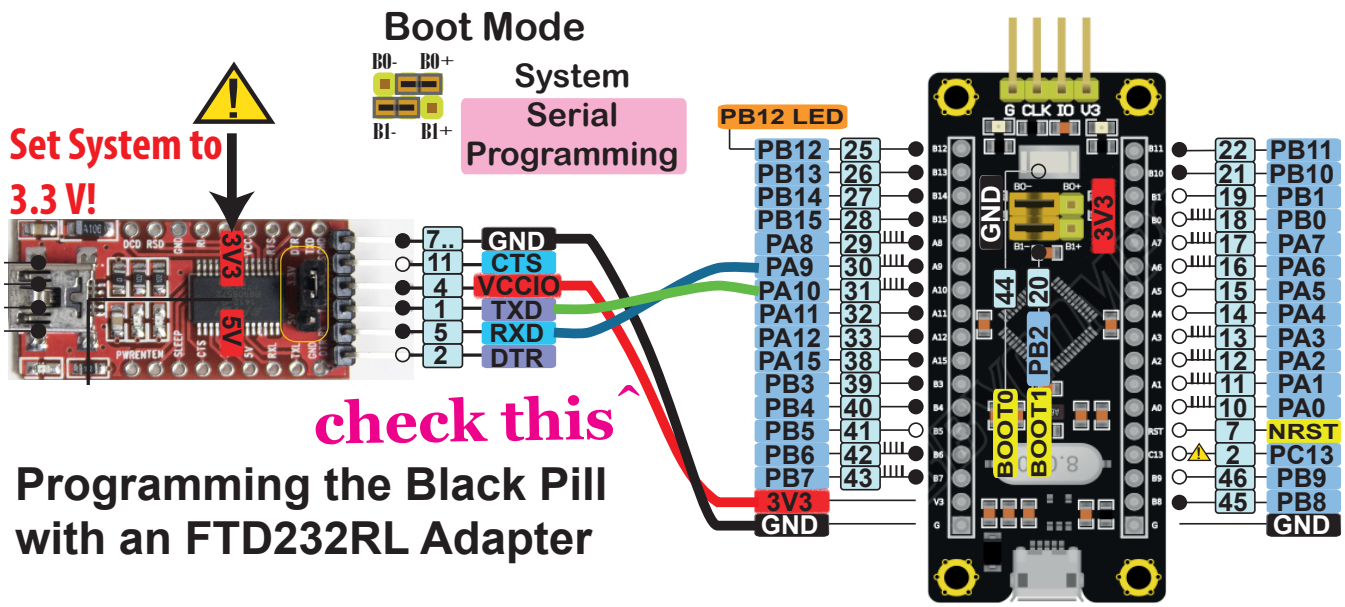
4.3. Flashing the bootloader onto the Black Pill via USB to Serial converter using UART1

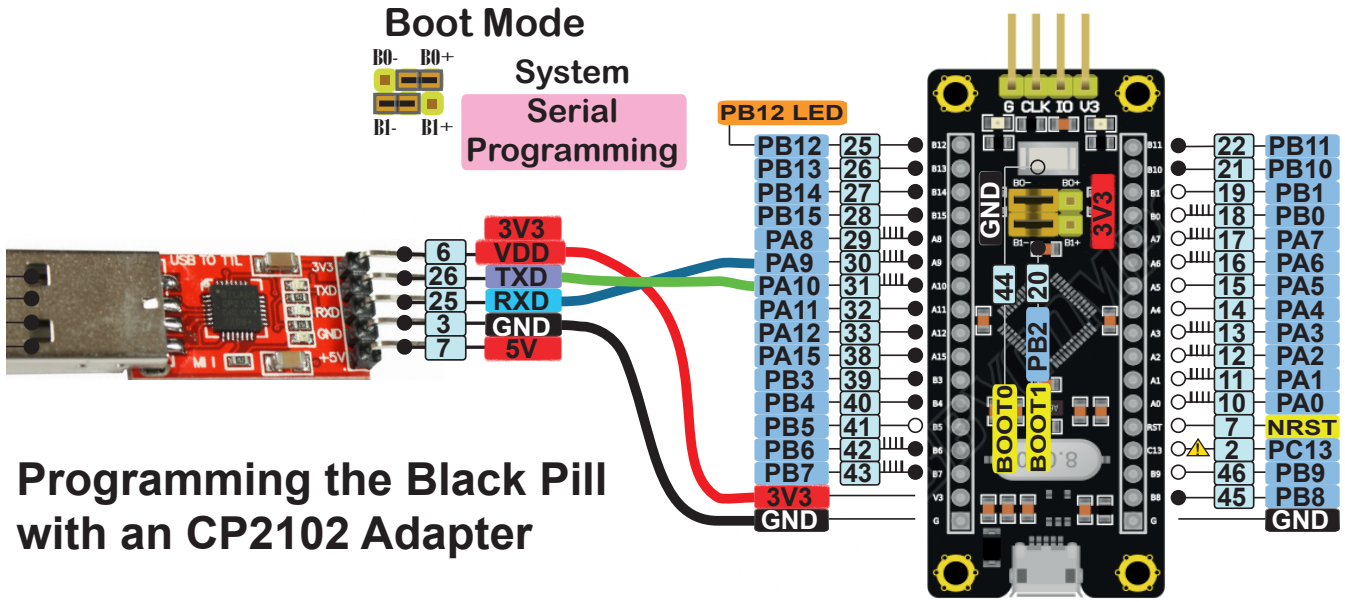
Windows

The Black Pill does not come with any USB bootloader. This process should be roughly the same for all F103 boards.

1. Download the correct bootloader binary, in this case generic_boot20_pb12.bin [https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/generic_boot20_pb12.bin?raw=true]
2. Set the 'boot 0' pin/jumper high, and 'boot 1' low
   - B0+ to center pin
   - B1- to center pin

3. Connect the board to the PC using a USB to serial converter:

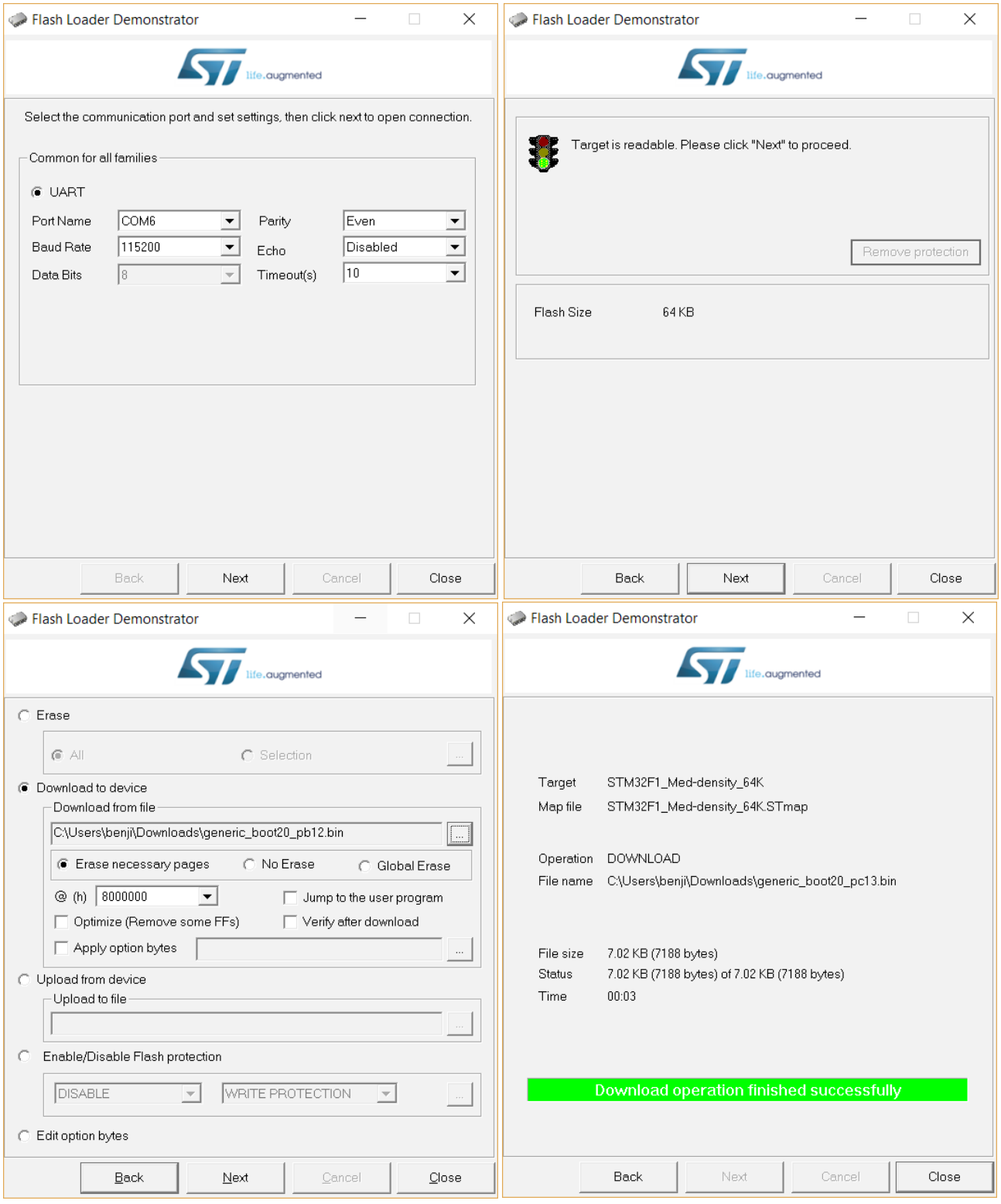




- RX to PA9
- TX to PA10
- GND to G on the ST-Link connector
- 3.3V to V3 on the ST-Link connector
- Ensure the USB to serial converter is in 3.3 V mode

4. Reset the board, the user LED should now be off
5. Download and install Flash Loader Demonstrator from here: [http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/flasher-stm32.html]

6. Use Flash Loader Demonstrator to flash the bootloader:



- Select the correct COM port, yours will be different from the screenshots

7. For normal use, set both boot pins low
   - B0- to center, pin B1 stays the same, B1- to center.

Linux

Steps 1-4 and 7 are the same for both Windows and Linux

For steps 5-6, use the command from the Maple Mini Linux procedure section 4.4 on page 25, changing the paths and filenames as necessary.

```
cd ~/.arduino15/packages/stm32duino/tools/stm32tools/1.0.9/linux/stm32flash
./stm32flash -w ~/Downloads/generic_boot20_pb12.bin -v -g 0x0 /dev/ttyUSB0
```

4.4. Flashing the bootloader - Maple Mini

Download the bootloader binary from here: [https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/maple_mini_boot20.bin?raw=true]

1. Connect Serial1 to your computer using a USB/TTL converter.
2. Connect the adapter TXD to RX1 and RXD to TX1.
3. A video of the connection procedure is here: [https://www.youtube.com/watch?v=G_RF0a0hrak]
4. Jumper the Boot1/PB2/D2 pin to ground.
5. Power the board from the 3.3v pin on the USB adapter. The on board USB connector should be disconnected.
6. Note: Some USB/TTL adapters (notably CP2102) have the RX and TX pins reversed, so RX connects to RX1 and TX to TX1.
7. Press and hold Button1 then press reset.

Windows: Download the STM Demonstrator GUI for windows from here [http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/flasher-stm32.html] to upload the bootloader.

After flashing is complete, disconnect the USB/TTL adapter from your computer and connect a USB cable directly to the Maple Mini. You should be able to upload using **`Tools→Bootloader Version→Bootloader 2.0 (20K Ram 120K Flash)`**.

Linux: The jumpers and wiring are the same as for Windows. Immediately after connecting the USB/TTL converter, open a terminal window and enter dmesg. Note the port assigned to the adapter. If this is the only USB serial device, it will likely be /dev/ttyUSB0. You can give the full path to the flash utility, but it's probably easier to go to the folder and run the program from there. Assuming the maple bootloader bin is in your ~/Download folder and your serial port is on /dev/ttyUSB0, run these commands:

```
cd ~/.arduino15/packages/stm32duino/tools/stm32tools/1.0.9/linux/stm32flash
./stm32flash -w ~/Downloads/maple_rev5_boot20.bin -v -g 0x0 /dev/ttyUSB0
```

The console should show that the board is detected and that the bootloader was written successfully.

```
stm32flash Arduino_STM32_0.9

http://github.com/rogerclarkmelbourne/arduino_stm32

Using Parser : Raw BINARY
Interface serial_posix: 57600 8E1
Version      : 0x22
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0410 (Medium-density)
- RAM        : 20KiB  (512b reserved by bootloader)
- Flash      : 128KiB (sector size: 4x1024)
- Option RAM : 16b
- System RAM : 2KiB
Write to memory
Erasing memory
Wrote and verified address 0x08001bc0 (100.00%) Done.

Starting execution at address 0x08000000... done.
```

Now connect your USB cable to the board and run dmesg. You should see the Maple pid/vid:

```
[Sat Oct 22 12:20:45 2016] usb 3-3: New USB device found, idVendor=1eaf, idProduct=0003
[Sat Oct 22 12:20:45 2016] usb 3-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[Sat Oct 22 12:20:45 2016] usb 3-3: Product: Maple 003
[Sat Oct 22 12:20:45 2016] usb 3-3: Manufacturer: LeafLabs
[Sat Oct 22 12:20:45 2016] usb 3-3: SerialNumber: LLM 003
```

The above procedures works for any STM32 board. You will need to download the correct bootloader binary to match your board and change the jumpers to Boot0=1 and Boot1=0 for the flash procedure. Once the flash procedure is complete, set both Boot jumpers to 0.

5. Notes

Reference

1. ***Arduino IDE***
   Download: http://arduino.cc/en/Main/Software.
   Boards Manager Package for STM32F1x series: http://dan.drown.org/stm32duino/package_STM32duino_index.json.
   Prefered Installation: https://wiki.stm32duino.com/index.php?title=Installation.
   Arduino STM32 files: https://github.com/rogerclarkmelbourne/Arduino_STM32/archive/master.zip.
2. ***ST-LINK/V2***
   https://www.st.com/en/development-tools/st-link-v2.html#tools-software
3. ***STM32duino-bootloader***
   https://github.com/rogerclarkmelbourne/STM32duino-bootloader
   With binaries at https://github.com/rogerclarkmelbourne/STM32duino-bootloader/tree/master/binaries.
   Black Pill Bootloader: https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/generic_boot20_pb12.bin?raw=true
   Maple Mini Bootloader: https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/maple_mini_boot20.bin?raw=true.

4. ***OpenBLT Bootloader***
   https://sourceforge.net/projects/openblt/
   https://github.com/feaser/openblt