STM32F103C8T6 - STM32 Development Boards (Blue Pill)

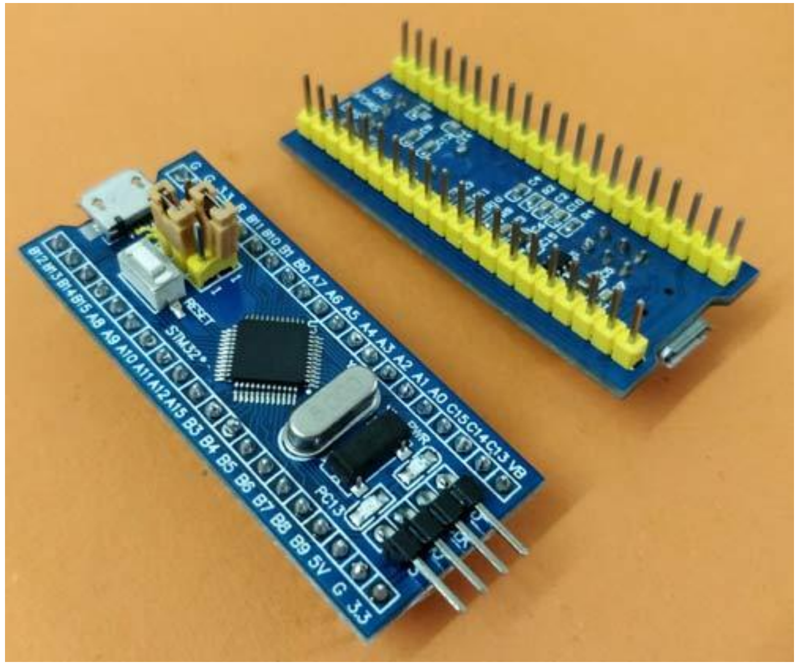
Dr. M. Premkumar

Associate Professor / EEE

Dayananda Sagar College of Engineering, Bengaluru

Introduction to the STM32 (Blue Pill) Boards

- The STM32 board - Blue Pill is a Development board for the ARM Cortex M3 Microcontroller.
- The microcontroller on top of it is the STM32F103C8T6 from STMicroelectronics.
- Apart from the Microcontroller, the board also holds two crystal oscillators, one is an 8 MHz crystal, and the other is a 32 kHz crystal, which can be used to drive the internal RTC (Real Time Clock).
- Because of this, the MCU can operate in deep sleep modes making it ideal for battery operated applications.
- Since the MCU works with 3.3V, the board also houses a 5V to 3.3V voltage regulator IC to power the MCU.
- Even though the MCU operates at 3.3V most of its GPIO pins are 5V tolerant.
- There are also two on-board LEDs, one (red colour) is used for power indication, and the other (green colour) is connected to the GPIO pin PC13.
- It also has two header pins which can be used to toggle the MCU boot mode between programming mode and operating mode.



Difference Between Blue Pill and Arduino Uno

Specifications	STM32F103C8	Arduino Uno
Number of cores	1	1
Architecture	32-bit	8-bit
CPU frequency	72 MHz	16 MHz
RAM	20 kB	2 kB
Flash	64 kB	32 kB
GPIOs	37	14
Buses	SPI, I2C, UART, CAN	SPI, I2C, UART

STM32F103C8T6 - Specifications

STM » stands for the manufacturers name STMicroelectronics
32 » stands for 32-bit ARM architecture
F103 » stands to indicate that the architecture ARM Cortex M3
C » 48-pin
8 » 64KB Flash memory
T » package type is LQFP
6 » operating temperature -40°C to +85°C

Architecture: 32-bit ARM Cortex M3

Operating Voltage: 2.7V to 3.6V

CPU Frequency: 72 MHz

Number of GPIO pins: 37

Number of PWM pins: 12

Analog input Pins: 10 (12-bit)

USART Peripherals: 3

I2C Peripherals: 2

SPI Peripherals: 2

Can 2.0 Peripheral: 1

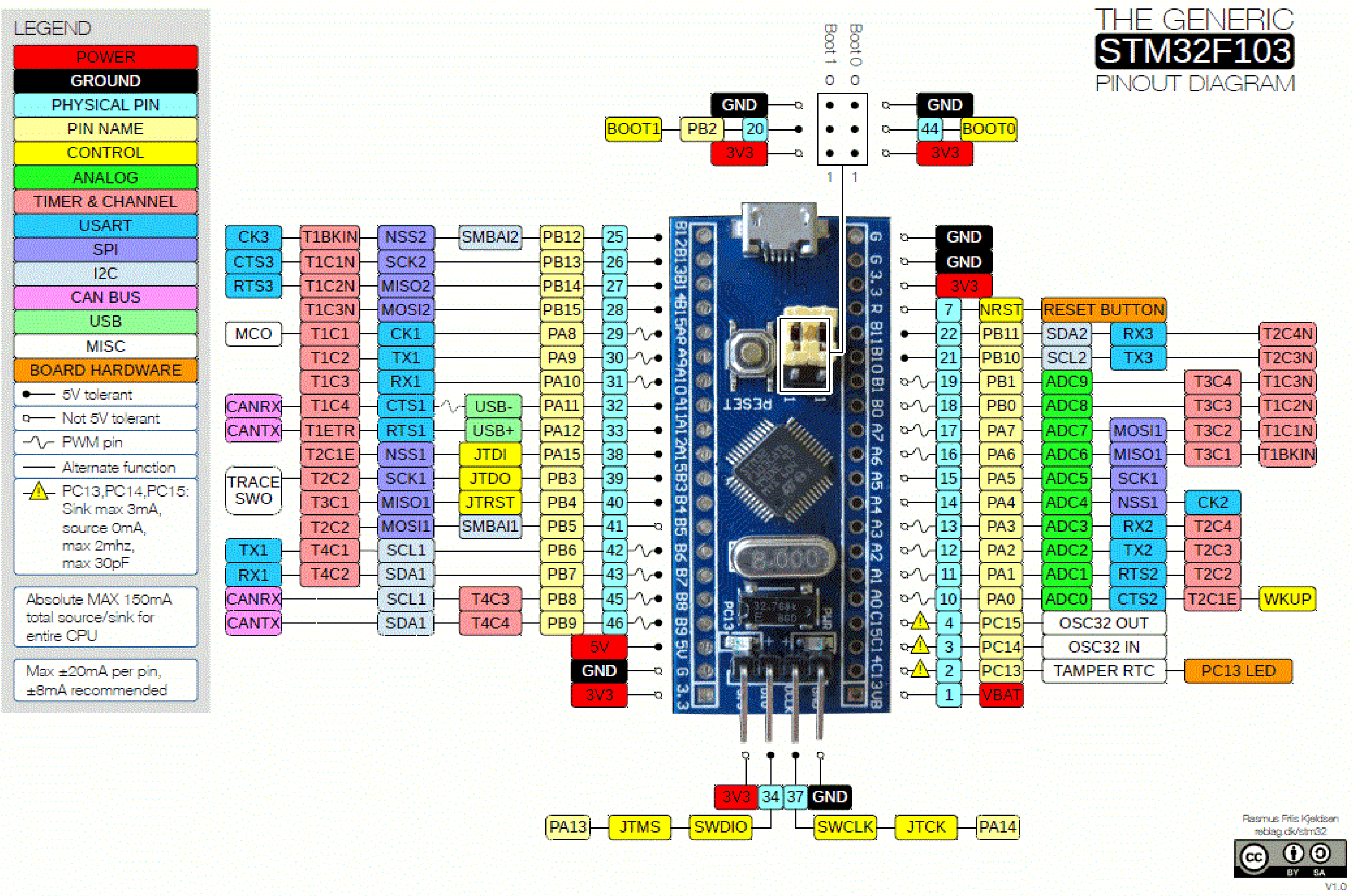
Timers: 3(16-bit), 1 (PWM)

Flash Memory: 64KB

RAM: 20kB

Pin Details

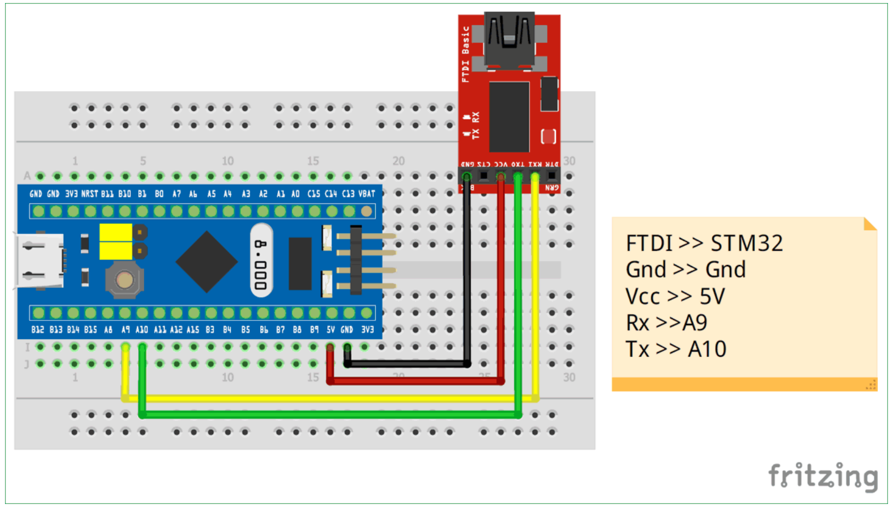
- Unlike Arduino boards, the STM32 development board has to be manually set to programming mode using the boot 1 and boot 0 jumper wires.
- The position of Boot 1 is normally not disturbed. But the boot 0 jumper has to be set as 3.3V for programming mode and set to ground for operating mode.



How to Program STM32F103C8T6?

- All the existing methods to program an ARM chip can be used for the STM32 board as well.
- One famous and commonly used IDE is the Keil ARM MDK and apart from that we can also use IAR workbench, Atollic TrueStudio, MicroC Pro ARM, Crossworks ARM, Ride 7, PlatformIO+STM32 etc.
- But what made this board a lot popular is its ability to be programmed with the Arduino IDE.
- This way people can get started and build projects with STM32 in no-time since many will be familiar with Arduino IDE and its easy to use programming language and readily available libraries.
- STM32 – (BluePill) Development Board (**STM32F103C8T6**) - FTDI Programmer – Breadboard - Connecting wires

- The STM32 Blue Pill development board does not come with a bootloader to make it Arduino IDE compatible. However, this bootloader can be flashed into the STM32 board and then the micro-USB port can also be directly used to upload the programs.
- The Vcc pin of the FTDI board is connected to the STM32 5V pin of power the board and the ground is connected to the Ground of STM32.
- The Rx and Tx pin of the FTDI board is connected to the A9 and A10 pin of the STM32 respectively. Where the A9 is the Tx pin of STM32 MCU and the A10 is Rx pin.
- **Step 1:** If you have not yet installed the Arduino IDE, download and install it.
- **Step 2:** After Installing the Arduino IDE open and download the required packages for the STM32 board. This can be done by selecting File -> Preferences.
- **Step 3:** Clicking on Preferences will open the below shown dialog box. In the additional Boards Manager URL text box paste the below link.



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

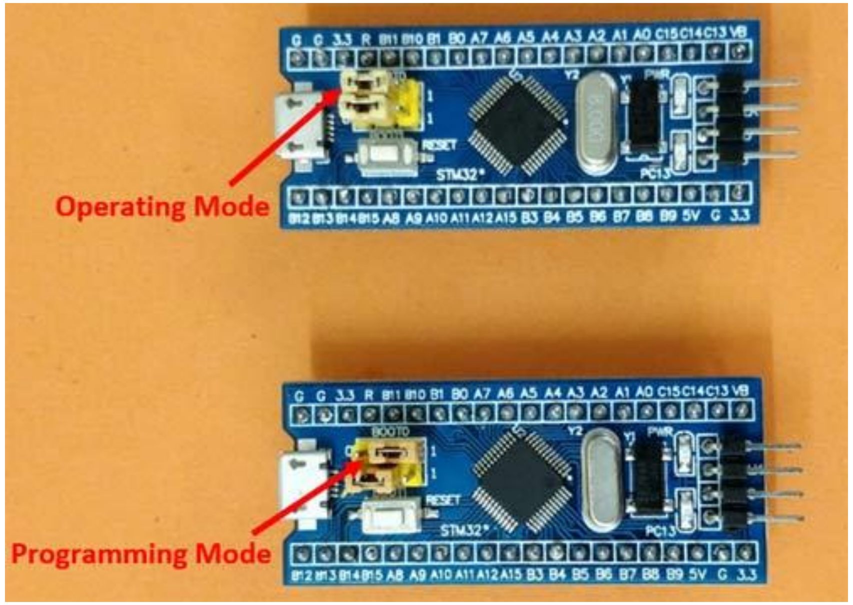
https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json

http://dan.drown.org/stm32duino/package_STM32duino_index.json

Step 4: Now go to Tool -> Boards -> Board Manager. This will open the Boards manager dialog box, search for “STM32” and install the package that appears.

Step 5: After the package, installation is completed. Go to Tools and scroll down to find the Generic STM32F103C series. Then make sure the variant is 64k Flash type, CPU speed is 72MHz and change the upload method to Serial.

- **Step 6:** Now, connect your FTDI board to the computer and check to which COM port the FTDI board is connected to using device manager. Then, select the same port number in Tools->Port
- **Step 7:** After all the changes are made, check the bottom right corner of the Arduino IDE and you should notice the following setting being set.
- The STM32 Board has to be put into programming mode before uploading any program, to do this put the boot 0 jumper in the lower position. The Jumper position for Programming mode and Operating mode are shown below.
- Now, to upload the program make sure the boot 0 jumper is in programming mode and then press the Reset button.
- As soon as you press the reset button the board will enter into programming mode and the green LED will be turned off, this indicates that the board is ready for upload.
- Press the upload button on the Arduino IDE and your program should get compiled and uploaded.



After uploading the program, the boot 0 should be changed back to operating mode so that next time when the Board is powered the uploaded program starts to get executed automatically.

Programming STM32F103C8 Board using USB Port

- In order to program the Board we utilized a FTDI programmer module and also had to toggle the boot 0 jumper between and 1 position while uploading and testing a code, which is surely a daunting task.
- Also the mini-USB port on the Development board was left totally unused. The reason for doing that is, when the STM32 development board is purchased it does not come with an Arduino ready boot loader and hence the board will not be discovered by your computer when connected through the USB.
- To program the STM32 Blue Pill board directly through USB port we need to first flash the Maple boot loader into the MCU.
- To do this we need to use a Serial FTDI board. This board is connected to the Rx and Tx pin of the STM32 as discussed earlier.
- The Vcc pin of the FTDI board is connected to the STM32 5V pin of power the board. The ground is connected to the Ground of STM32. The Rx and Tx pin of the FTDI board is connected to the A9 and A10 pin of the STM32 respectively. Where the A9 is the Tx pin of STM32 MCU and the A10 is Rx pin.

Uploading the Maple Boot loader to STM32 Development board

Step 1: We have to download the boot loader program file binaries (bin file) form the GitHub page. There are many versions of bin file, for the Blue Pill board use the below link and click on the download button to download the bin file.

https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/generic_boot20_pc13.bin

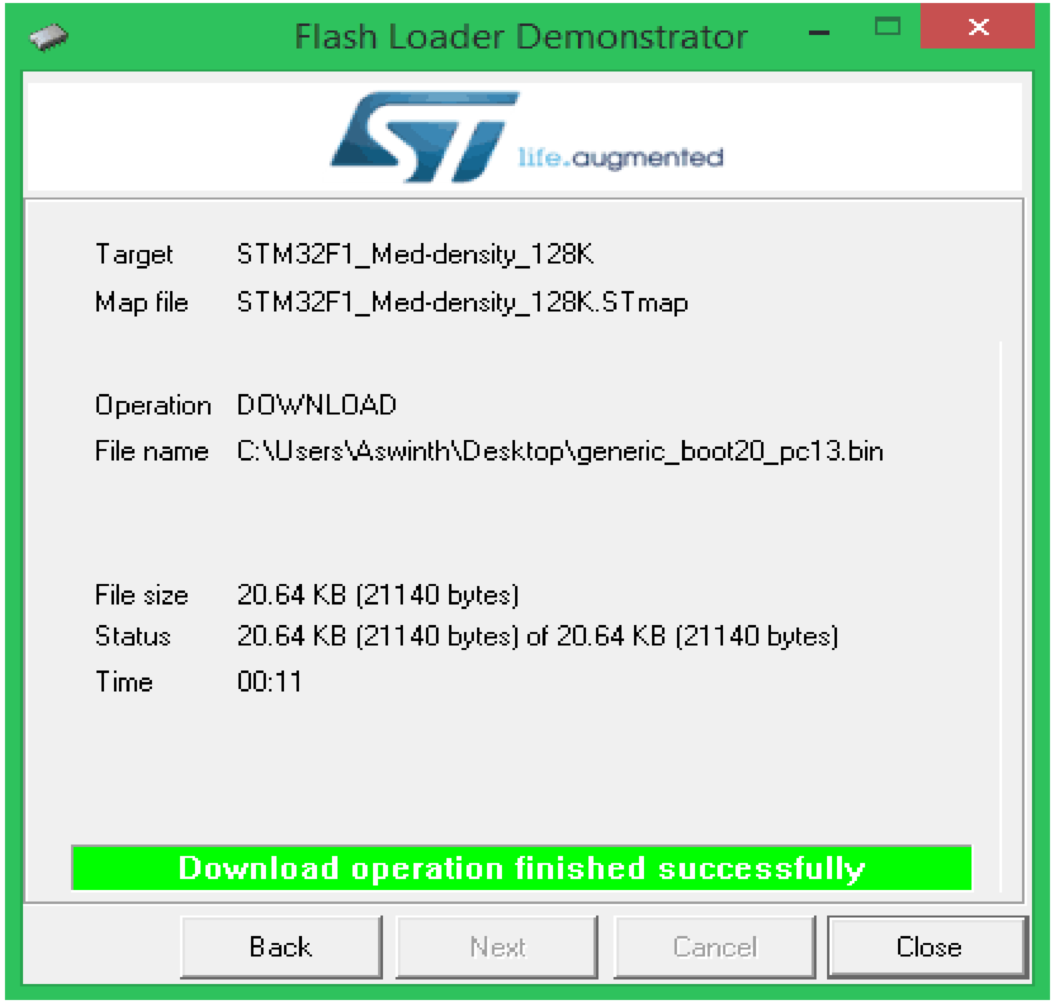
Step 2: Next we have to download and install the STM Flash loader software to flash the downloaded bin file into STM32. Click the below link to get into the ST website and scroll to the bottom and click on get software.

https://www.st.com/en/development-tools/flasher-stm32.html#getsoftware-scroll

Step 3: To download the software you have to enter your E-mail address and the download link will be sent to your E-mail. Then follow the link back to the website and click on get software again and you download will begin.

Step 4: Once downloaded install the software, make sure your STM32 board is connect to your computer though FTDI board and then launch the software. The software will automatically detect the COM port if not use the Device manager and make sure you select the correct COM port number. Leave the rest of the setting to its default.

- **Step 5:** Click on the Next button twice and the software will again automatically detect the board details. The board we are using is STM32F1 with 64K flash memory.
- **Step 6:** In the next step, select Download to device and browse to the location where we downloaded our bin file in step 1 and select it. Click on next.
- **Step 7:** The software will download some required files as shown below and will then begin the process of flashing.
- **Step 8:** Once the flashing is completes successfully, we will get the below screen. Click on close and exit the application. We have flashed the STM32 board with Arduino boot loader successfully. Now we have to prepare the Arduino IDE and install the drivers before we can program the STM32 board.
- After the Arduino package is installed, navigate to C:\Program Files (x86)\Arduino\hardware\Arduino_STM32-master\drivers\win where you will find install_drivers.bat and install_STM_COM_drivers.bat.



Target STM32F1_Med-density_128K
Map file STM32F1_Med-density_128K.STmap

Operation DOWNLOAD
File name C:\Users\Aswinth\Desktop\generic_boot20_pc13.bin

File size 20.64 KB (21140 bytes)
Status 20.64 KB (21140 bytes) of 20.64 KB (21140 bytes)
Time 00:11

Download operation finished successfully

Click on both the bat files and install the drivers.

Some Other Tips

- Download and install the version 1.8.5 of the Arduino IDE from here.
- Run the IDE, and on the Tools menu, select the Boards manager, and install the Arduino SAM boards (Cortex-M3) from the list of available boards. You must do this step, it installs the arm-none-eabi-g++ toolchain!
- Download zip file containing the STM32 files from here.
- Unzip to create the Arduino_STM32 folder.

Windows

- Copy the Arduino_STM32 folder to My Documents/Arduino/hardware (Note: if the hardware folder doesn't exist you will need to create it).
- If using Maple or Maple mini, you need to install drivers for the Serial and DFU (upload devices). Please open a cmd window (run as administrator), navigate to the folder: /drivers/win/ and run: install_drivers.bat. Note: This doesn't actually install drivers. Windows comes pre-installed with a compatible Serial USB driver and a DFU (upload) driver. However the built in drivers need to be associated with the USB ID of the Maple serial and DFU devices. The batch file and wdi-simple.exe do the clever stuff to convince Windows 7 or newer, that it should use its drivers with the Maple serial and DFU devices.
- Re-start the Arduino IDE, and select the appropriate board from the "Tools" -> "Board menu", and select the appropriate Com port for your Maple mini or serial upload device. Note: If you do not see a Maple Serial com device, this is probably because the Maple mini has not been loaded with the blink sketch. So upload a the Maple mini blink sketch from examples\Digital\Blink and the Maple serial device should now be available on the Port menu.

Link: https://github.com/rogerclarkmelbourne/Arduino_STM32/wiki/Installation