

# Flashing STM32 using STLINK or RPI GPIO

Alok Ranjan Kesari and G. V. V. Sharma

## CONTENTS

<b>1</b>	<b>Components</b>	<b>1</b>
<b>2</b>	<b>Software Setup</b>	<b>1</b>
2.1	Install Necessary Packages	1
2.2	Installing Openocd and Programming Environment	1
2.3	Configure Openocd . . .	1
<b>3</b>	<b>Hardware Setup</b>	<b>2</b>
3.1	STLINK . . . . .	2
3.2	RPI GPIO . . . . .	2
<b>4</b>	<b>Make File and Flashing</b>	<b>2</b>

*Abstract*—This manual shows how to program an STM32 board using STLINK and Raspberry Pi. The procedure is the same for any Linux machine.

## 1 COMPONENTS

The necessary components for this manual are listed in Table I.

Component	Quantity
STM32F103C8T6	1
Raspberry Pi 3	1
STLINK V2	1
Female-Female Jumper Wires	5

TABLE I

The authors are with the Department of Electrical Engineering, IIT Hyderabad alok.kesari@yahoo.co.in, gadepall@iith.ac.in

## 2 SOFTWARE SETUP

Open a terminal and execute the following commands

```
cd ~
mkdir -p ~/sandbox
cd ~/sandbox
```

### 2.1 Install Necessary Packages

```
sudo apt-get install git autoconf libtool make
automake texinfo pkg-config libusb-1.0-0
libusb-1.0-0-dev gcc-arm-none-eabi
libnewlib-arm-none-eabi telnet
```

### 2.2 Installing Openocd and Programming Environment

```
git clone git://repo.or.cz/openocd.git
git clone https://github.com/gadepall/
STM32F103C8T6.git
```

### 2.3 Configure Openocd

```
cd openocd
./bootstrap
./configure --enable-sysfsgpio --enable-
bcm2835gpio
make
sudo make install
```

While using STLINK, **./configure** without the **--enable** switches is sufficient.

### 3 HARDWARE SETUP

#### 3.1 STLINK

Connect the STLINK to a USB port of the Raspberry Pi. The hardware connections between the STLINK and STM32 are available in Table II. See Fig. 1 as well for the *black pill* board. Fig. 2 shows the *blue pill* board.

STM32	STLINK
GND	GND
3.3V	3.3V
SWDIO	SWDIO
SWCLK	SWCLK

TABLE II: STLINK-STM32 connections

#### 3.2 RPI GPIO

On the RPi, type

```
pinout
```

This will give the GPIO pin configuration on the RPi. Now open

```
nano /usr/local/share/openocd/scripts/interface/
sysfsgpio-raspberrypi.cfg
```

and verify that the file contains the lines (at different locations)

```
# Each of the SWD lines need a gpio number
set: swclk swdio
# Header pin numbers: 23 22
sysfsgpio_swd_nums 11 25

# Header pin numbers: TRST -26, SRST -18
sysfsgpio_srst_num 24
reset_config srst_only srst_push_pull
```

Note that the above configuration was obtained for an **RPi Zero W**. It may be different for other RPis. Remove the # before the lines, if necessary. These lines provide information on the pin connections between the Rpi and STM32 as shown in Table III. The pin num-

Raspberry Pi	STM32
GND (Pin 6)	GND
3.3V (Pin 1)	3.3V
GPIO 25 (Pin 22)	SWDIO
GPIO 11 (Pin 23)	SWCLK
GPIO 24 (Pin 18)	RESET

TABLE III: Raspberry Pi and STM32 Connections

bers for RPi in the above table will be different for different RPis based on `sysfsgpio-raspberrypi.cfg`.

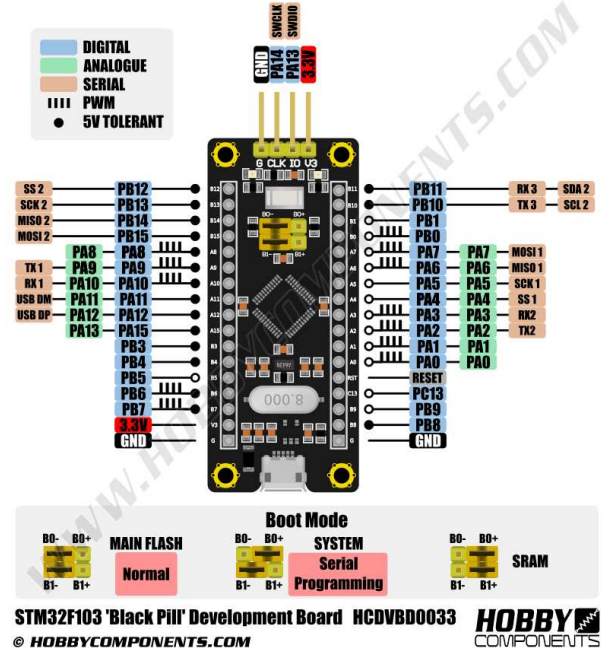


Fig. 1: STM32F103C8T6 Pin Configuration (Black Pill)

### 4 MAKE FILE AND FLASHING

1. Communicate with the STM32 board. While using STLINK,

```
cd ~/sandbox/openocd
sudo openocd -f /usr/local/share/openocd/
scripts/interface/stlink.cfg -f usr/local/
share/openocd/scripts/target/stm32f1x.
cfg
```

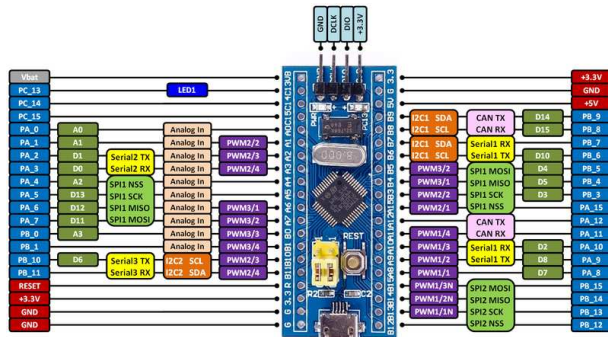


Fig. 2: STM32F103C8T6 Pin Configuration (Blue Pill)

```
OBJCOPY = /data/data/com.termux/files/home/gcc-arm-none-eabi-8-2019-q3-update/install-native/bin/arm-none-eabi-objcopy
OBJDUMP = /data/data/com.termux/files/home/gcc-arm-none-eabi-8-2019-q3-update/install-native/bin/arm-none-eabi-objdump
/data/data/com.termux/files/home/gcc-arm-none-eabi-8-2019-q3-update/install-native/bin/arm-none-eabi-size $(PRJ_NAME).elf
```

are suitably modified based on the path in your OS.

- Go to the telnet terminal

```
reset halt
flash write_image erase main.bin 0
x08000000
reset run
```

- Align the two pin caps beside the reset button. Press the reset button. You should see an LED blinking.
- Modify main.c in the STM32F103C8T6 directory and modify the code to keep the LED on. Flash it to the STM32 and verify.

If only RPI GPIO is used, then

```
cd ~/sandbox/openocd
cp ~/sandbox/STM32F103C8T6/refs/openocd.cfg ~/sandbox/openocd
sudo openocd -f /usr/local/share/openocd/scripts/interface/sysfsgpio-raspberrypi.cfg -c
"transport select swd" -c "adapter_khz 1000" -f /usr/local/share/openocd/scripts/target/stm32f1x.cfg
```

- Open a new terminal and type

```
telnet localhost 4444
```

This will establish a connection between the RPI and STM32

- Open another new terminal and type

```
cp ~/sandbox/STM32F103C8T6/examples/blink.c ~/sandbox/STM32F103C8T6/src/main.c
sudo make bin
cp main.bin cd ~/sandbox/openocd
```

Ensure that the following lines in the Makefile

```
CC= /data/data/com.termux/files/home/gcc-arm-none-eabi-8-2019-q3-update/install-native/bin/arm-none-eabi-gcc
```