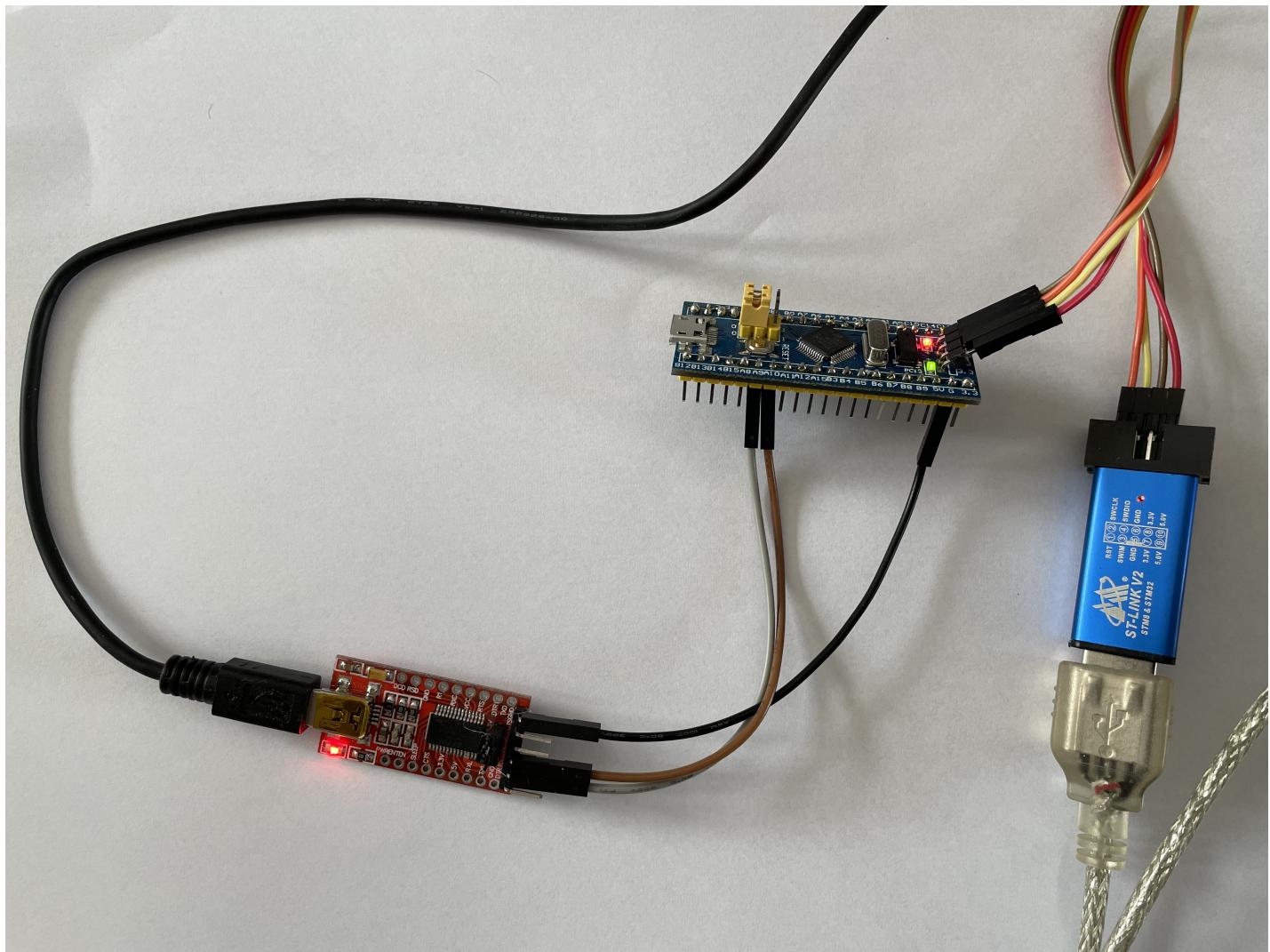


My "Blue Pill" Projects Test Setup

Description and Evaluation

by Dr. Markus Reinhardt
April 30, 2021



Contents

1. Project goal	4
2. Project Hardware setup	4
3. Project Software Setup (first sketch)	6
3.1. LCD control via I2C (second sketch)	7
A. Appendix A	9
A.1. Pin-out of the "Blue Pill" board	9
A.2. Helpful Links	9

List of Figures

1.	Hardware setup	4
2.	VSCode/PlatformIO (Arduino environment) IDE	6
3.	VSCode/PlatformIO IDE (second sketch)	7
4.	VSCode/PlatformIO IDE (second sketch)	8
5.	Pin-out of the "Blue Pill" board	9

List of Tables

1. Project goal

The project is created to test the HW / SW setup for projects based on the so-called "Blue Pill STM32 board".

The SW setup is based on VSCode with PlatformIO extension and STM32 board operated with the Arduino environment.

The HW setup allows to program the board with an ST-Link V2 adapter and also serial communications with a (Linux) PC via a USB/Serial adapter.

In a second HW setup the control of a LCD via the I2C interface is tested.

The "Blue Pill" STM32 board has a STM32F103C8T6 processor with 20KB RAM and 64KB EEPROM running at 72MHz. See also "Blue Pill F103C8" in PlatformIO.

2. Project Hardware setup

The project setup for the HW part is shown in figure 1.

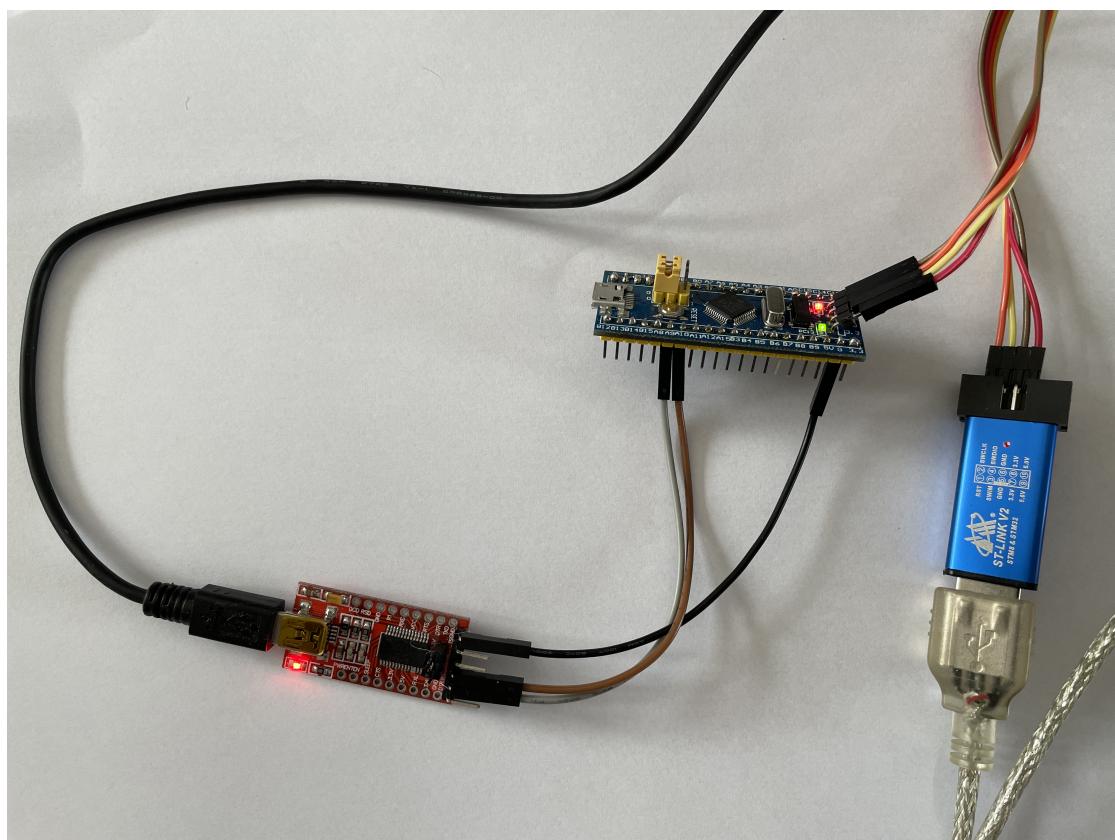


Figure 1: Hardware setup

Note: The jumper positions of the two yellow jumpers on the (blue colored) "Blue Pill"

board.

Programming is done with the (metallic blue) ST-Link V2 module. There are four pins connected with the Blue Pill board. The connections are done as follows:

Blue Pill	Cable color	ST-Link V2
GND	Brown	GND (Pin 6)
3.3V	Red	3.3V (Pin 8)
CLK	Orange	SWCLK (Pin 2)
DIO	Yellow	SWDIO (Pin 4)

A USART serial interfacing between the Blue Pill board and the PC / IDE is realized with the (red) USB/Serial adapter board. Note the jumper position on the board is such that the 3.3V output voltage is provided (but not connected in this setup) as the "Blue Pill" processor is operated with 3.3V. In the software the Arduino Serial1 interface port is used. The connection between the USB/Serial adapter and the Blue Pill board is done as follows:

Blue Pill	Cable color	USB/Serial
GND	Black	GND
TX1 (Pin PA9)	Gray	RX
RX1 (Pin PA10)	Brown	TX

The USB/Serial adapter appears under /dev/ttyUSB0 in the Linux operating system. This port has to be selected in the IDE, when the Serial Monitor is activated.

3. Project Software Setup (first sketch)

The project setup for the IDE part with the first Arduino sketch is shown in figure 2. The picture shows the PlatformIO IDE within the VSCode editor and the main Arduino

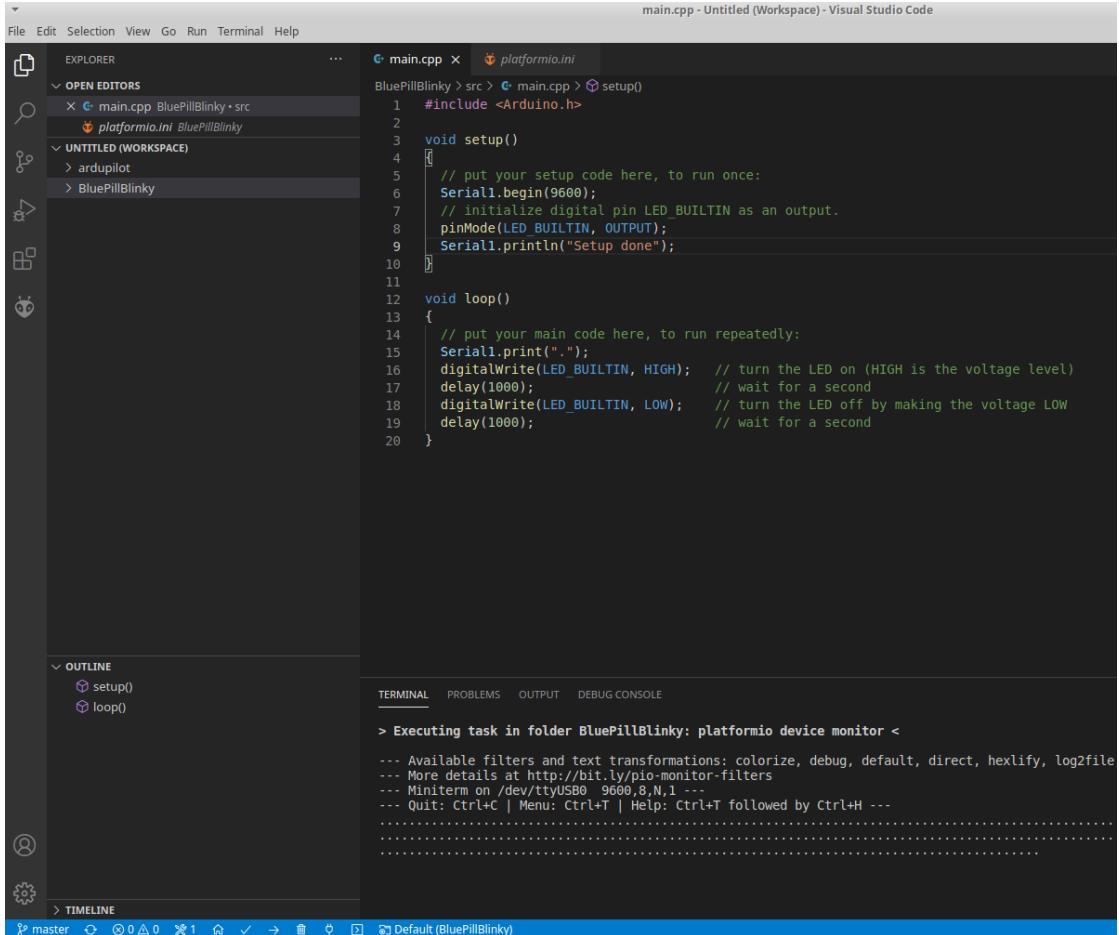


Figure 2: VSCode/PlatformIO (Arduino environment) IDE

sketch which implements the simple blinking of the on-board LED and the output of data via the Serial1 port.

The picture also shows the output of dots (see the sketch code) via the USB/Serial adapter and the Serial1 Arduino port to the Arduino Serial Monitor displayed in the lower part of the IDE.

The development cycle is controlled by pressing the relevant buttons in the blue bottom line of the IDE (see the call-outs of the buttons when moving over them with the mouse pointer).

Important Note: If you have multiple projects within the Platform IDE, do not forget to select the right project in the bottom line (in the figure here: "Default (BluePill-Blinky)") before compiling.

Compilation is done by pressing the "Check" button.

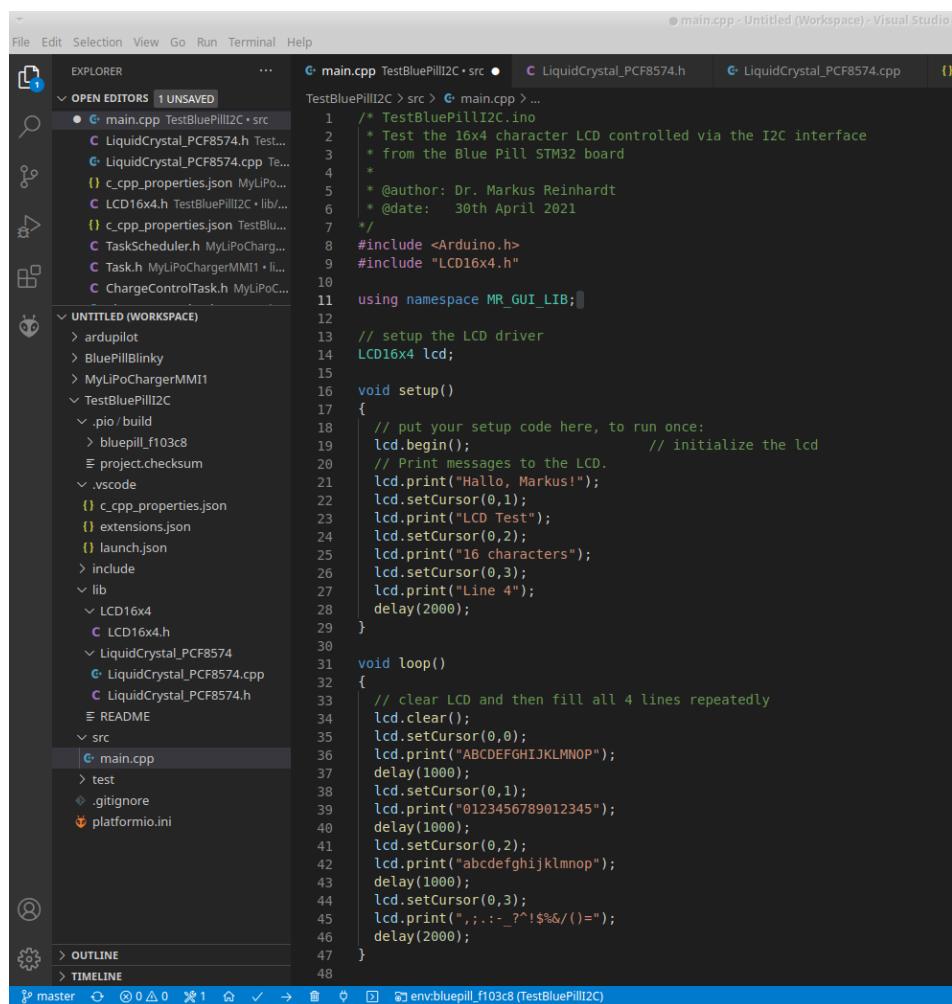
Program download is done by pressing the "Right Arrow" button.

Activation of the Serial Arduino Monitor is done with the "Plug" button.

To switch back to the PlatofromIO home screen is done with the "Home" button.

3.1. LCD control via I2C (second sketch)

The IDE with the second Arduino sketch is shown in figure 3. The sketch controls a



The screenshot shows the VSCode interface with the PlatformIO extension. The Explorer sidebar on the left lists project files and dependencies. The main editor window displays the `main.cpp` file for the `TestBluePillI2C` project. The code implements an LCD driver and a test loop to print various characters and strings to the LCD screen.

```

File Edit Selection View Go Run Terminal Help
main.cpp - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS 1 UNSAVED
main.cpp TestBluePillI2C + src
LiquidCrystal_PCF8574.h Test...
LiquidCrystal_PCF8574.cpp Te...
c_cpp_properties.json MyLiPo...
LCD16x4.h TestBluePillI2C + lib/...
c_cpp_properties.json TestBlu...
Taskscheduler.h MyLiPoCharg...
Task.h MyLiPoChargerMMI + lib...
ChargeControlTask.h MyLiPoC...
UNTITLED (WORKSPACE)
ardupilot
BluePillBlinky
MyLiPoChargerMMI
TestBluePillI2C
pio/build
bluepill_f103c8
project.checksum
vscode
c_cpp_properties.json
extensions.json
launch.json
include
lib
LCD16x4
LCD16x4.h
LiquidCrystal_PCF8574
LiquidCrystal_PCF8574.cpp
LiquidCrystal_PCF8574.h
README
src
main.cpp
test
.gitignore
platformio.ini
OUTLINE
TIMELINE
main.cpp - Untitled (Workspace) - Visual Studio Code
main.cpp TestBluePillI2C > src > main.cpp ...
LiquidCrystal_PCF8574.h
LiquidCrystal_PCF8574.cpp
/*
 * TestBluePillI2C.ino
 * Test the 16x4 character LCD controlled via the I2C interface
 * from the Blue Pill STM32 board
 *
 * @author: Dr. Markus Reinhardt
 * @date: 30th April 2021
 */
#include <Arduino.h>
#include "LCD16x4.h"
using namespace MR_GUI_LIB;
// setup the LCD driver
LCD16x4 lcd;
void setup()
{
    // put your setup code here, to run once:
    lcd.begin(); // initialize the lcd
    // Print messages to the LCD.
    lcd.print("Hallo, Markus!");
    lcd.setCursor(0,1);
    lcd.print("LCD Test");
    lcd.setCursor(0,2);
    lcd.print("16 characters");
    lcd.setCursor(0,3);
    lcd.print("Line 4");
    delay(2000);
}
void loop()
{
    // clear LCD and then fill all 4 lines repeatedly
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("ABCDEFGHIJKLMNP");
    delay(1000);
    lcd.setCursor(0,1);
    lcd.print("0123456789012345");
    delay(1000);
    lcd.setCursor(0,2);
    lcd.print("abcdefghijklmnp");
    delay(1000);
    lcd.setCursor(0,3);
    lcd.print(",;:_?^!$%&/()=");
    delay(2000);
}

```

Figure 3: VSCode/PlatformIO IDE (second sketch)

16x4 character LCD connected to the "Blue Pill" board via the I2C interface.

The HW setup for this test is shown in figure 4 The connection from the ST-Link V2

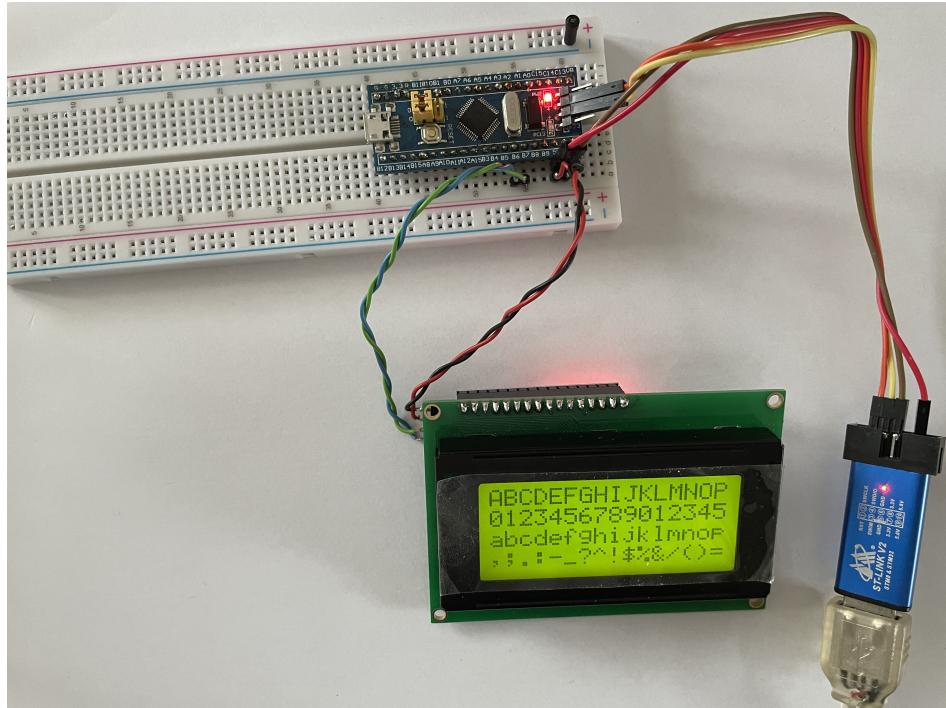


Figure 4: VSCode/PlatformIO IDE (second sketch)

adapter to the "Blue Pill" board is now using the 5V pins according to the following table. The 3.3V for the processor is now provided by the on-board fixed power regulator. The 5V power supply is also required for the LCD.

Blue Pill	Cable color	ST-Link V2
GND	Brown	GND (Pin 6)
5V	Red	5V (Pin 10)
CLK	Orange	SWCLK (Pin 2)
DIO	Yellow	SWDIO (Pin 4)

A. Appendix A

A.1. Pin-out of the "Blue Pill" board

Figure 5 shows the pin-out of the "Blue Pill" board.

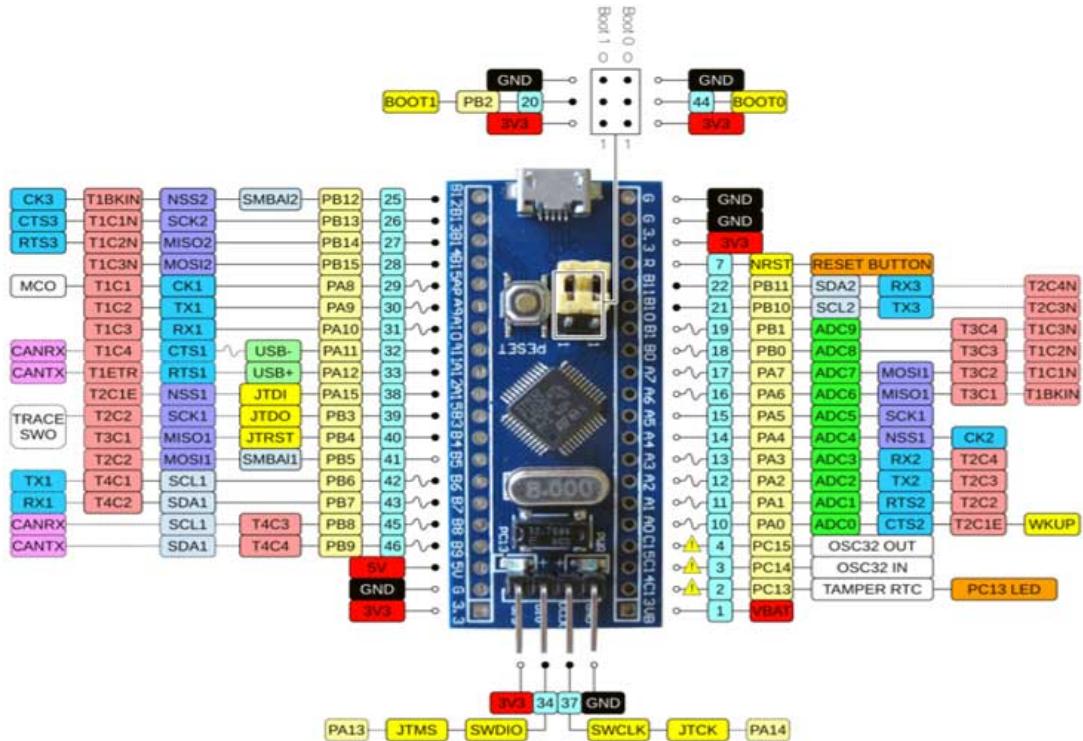


Figure 5: Pin-out of the "Blue Pill" board

A.2. Helpful Links

PlatformIO IDE

”Blue Pill F103C8” in PlatformIO

Installing PlatformIO and creating a sample program for STM32 Blue Pill

Arduino Getting Started and Tutorials