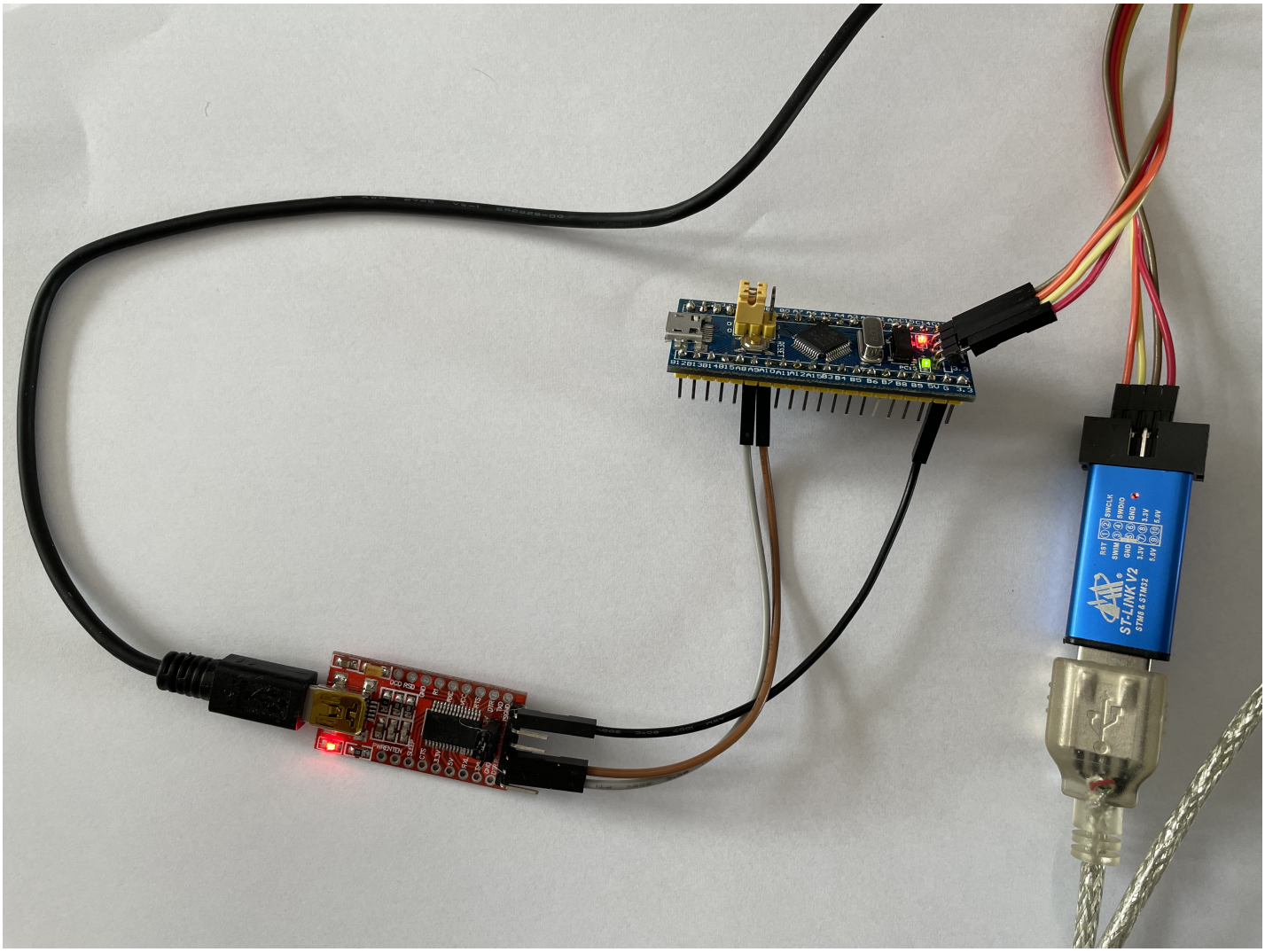
My "Blue Pill" Projects Test Setup

Description and Evaluation

by Dr. Markus Reinhardt

May 18, 2021



Contents

List of Figures

List of Tables

1. Project goals

The projects are created to test the basic HW / SW concepts for projects based on the so-called "Blue Pill STM32 board".
The used IDE 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.
In a third project two rotary encoders are used as input devices.
In a fourth project the ADC of the STM32 processor is tested with different resolutions.
In a fifth project the PWM generation is tested.
In a sixth project the usage of Flash storage for string and other constants is tested.
In a seventh project the usage of the FreeRTOS real time operating system is tested with a C based standard way of task definition.
In an eight project the usage of the FreeRTOS real time operating system is tested with a C++ based way of task definition.
The "Blue Pill" STM32 board has a6 STM32F103C8T6 processor with 20KB RAM and 64KB EEPROM running at 72MHz. See also "Blue Pill F103C8" in PlatformIO.
All the programs described below developed in the PlatformIO IDE are saved in the overall project Gitea repository.

2. Project1: Hardware Setup

A simple HW setup is shown in figure 1.
Note the jumper positions of the two yellow jumpers on the (blue colored) "Blue Pill" board!

Programming is done with the (metallic blue colored) ST-Link V2 module. Also the 3.3V power supply for the "Blue Pill" board is delivered by this module. There are four pins of the module connected with the "Blue Pill" board. The connections are as follows (Table 1):

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)

Table 1: Connection "Blue Pill" to ST-Link V2

A USART interfacing between the Blue Pill board and the PC / IDE is realized with the (red colored) USB/Serial adapter board. Note the jumper position on the board is such

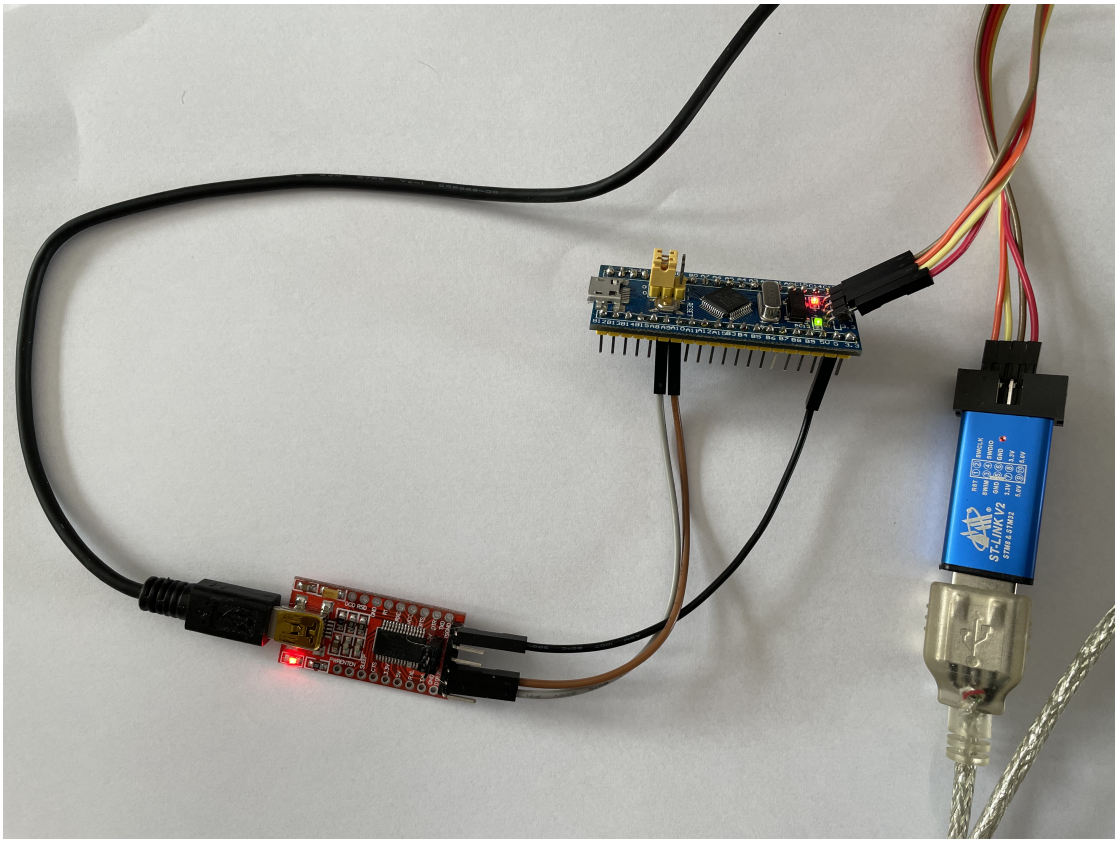


Figure 1: Project 1 hardware setup

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 (Table 2):

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

Table 2: Connection "Blue Pill" to USB/Serial adapter

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 1: PlatformIO IDE

The IDE for project 1 with the first Arduino sketch is shown in figure 2. The picture shows the PlatformIO IDE within the VSCode editor and the main Arduino 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 interface 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 PlatfromIO home screen is done with the "Home" button.
The project's software directory is `BluePillBlinky`.

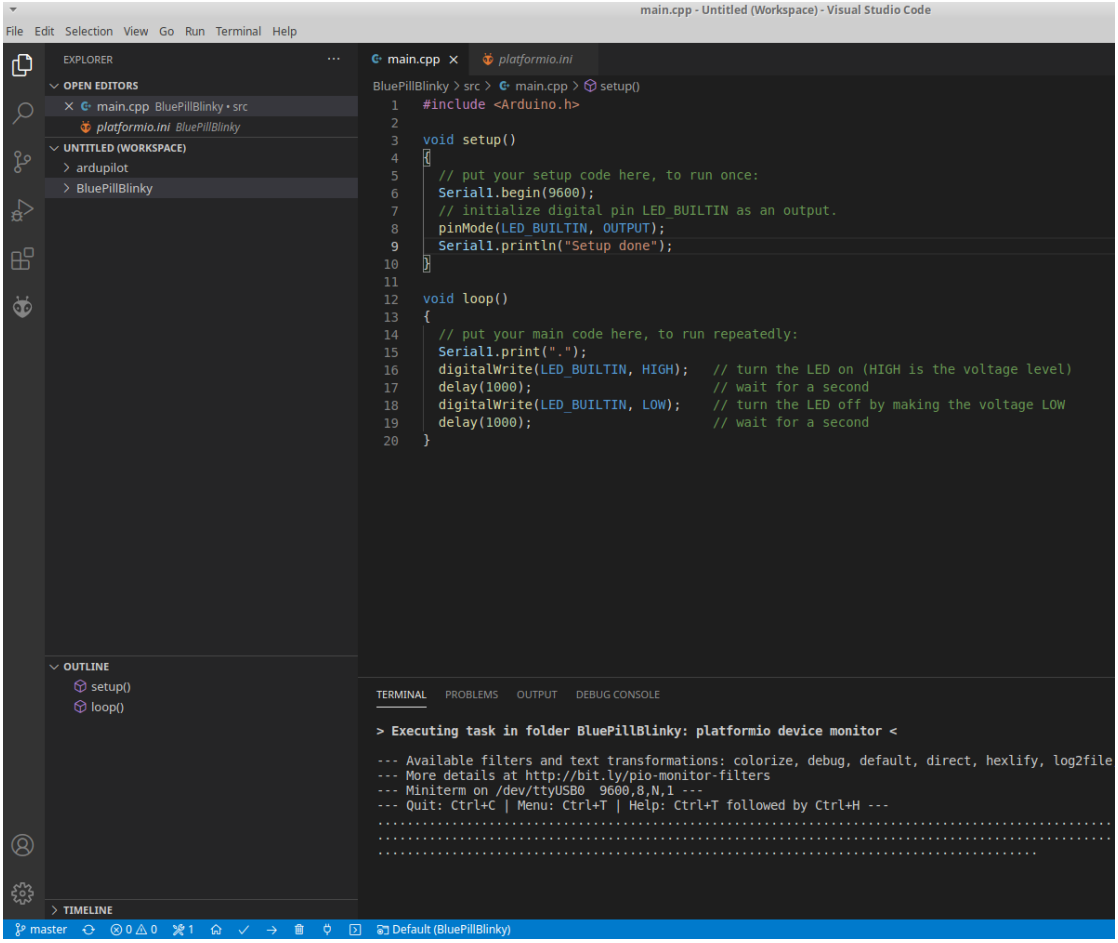


Figure 2: VSCode/PlatformIO (Arduino environment) IDE

4. Project 2: LCD control via I2C

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

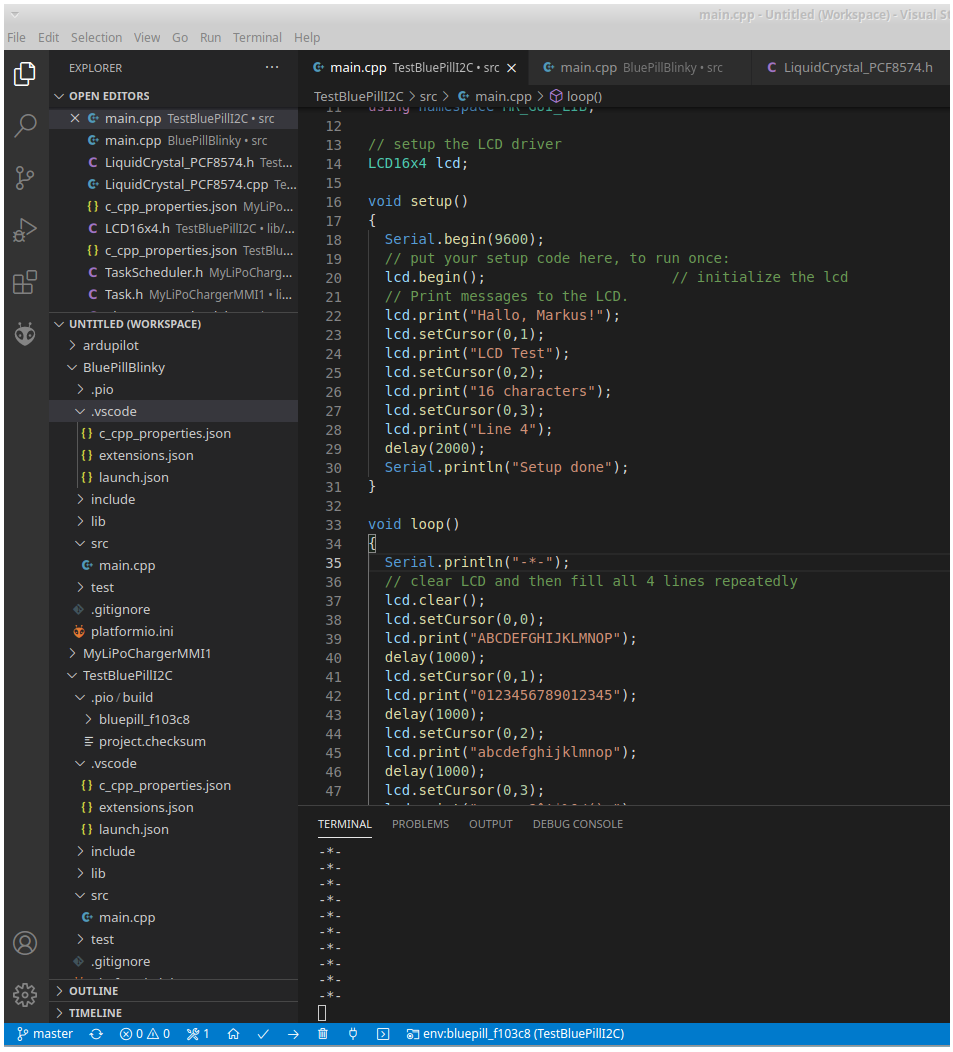


Figure 3: VSCode/PlatformIO IDE

16x4 character LCD connected to the "Blue Pill" board via the I2C interface.
In this sketch the Serial Arduino interface is used for test messages to the Serial Monitor of the IDE. It is connected to pins PA9 (TX1) and PA10 (RX1) and via the USB/Serial adapter to the PC. The IDE shows in the lower part the Serial Monitor and the messages received from the "Blue Pill" board.

The HW setup for this test is shown in figure 4 The connection from the ST-Link V2 adapter to the "Blue Pill" board is now using the 5V pins according to the following

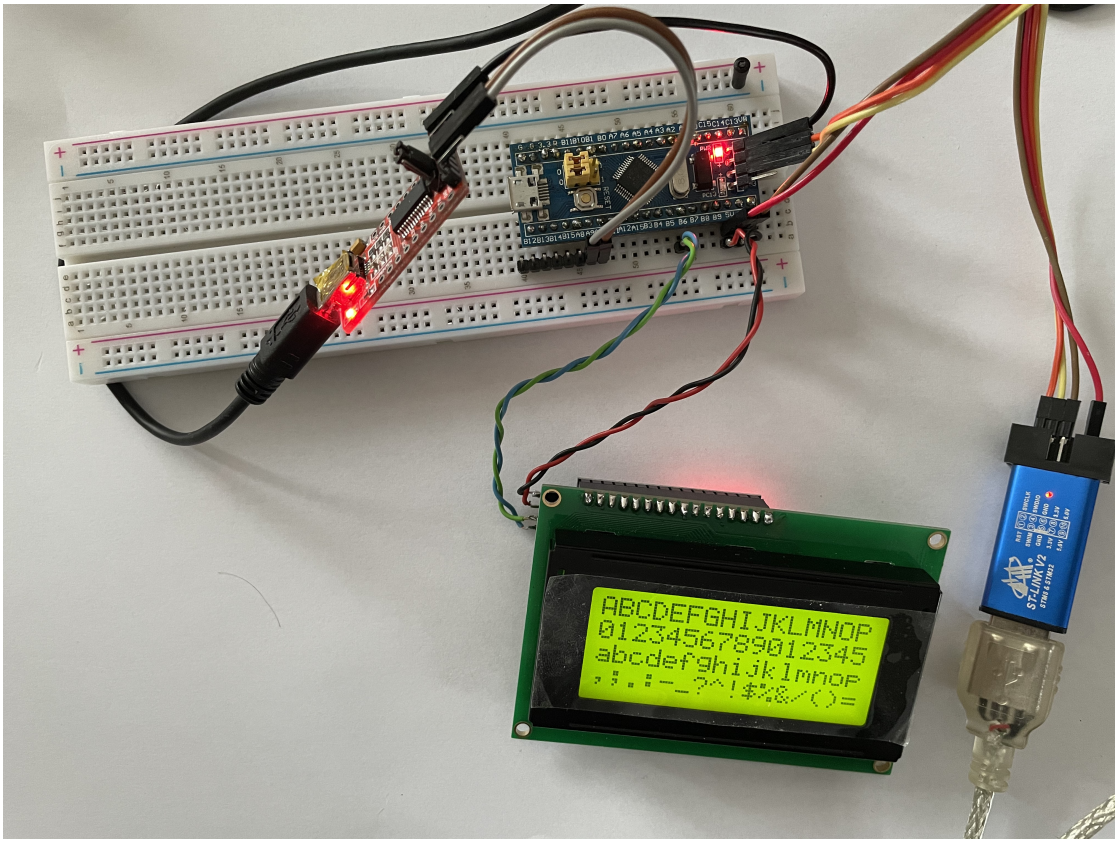


Figure 4: Project 2 HW setup with I2C-LCD

table (Table 3). 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. The connection between

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)

Table 3: Connection "Blue Pill" to USB/Serial adapter

the "Blue Pill" board and the I2C adapter of the LCD is done here as follows (Table 4): The connection between the "Blue Pill" board and the USB/Serial adapter is equal to

Blue Pill	**Cable color**	**I2C LCD**
GND	Black	GND
5V	Red	5V
SCL1 (Pin PB6)	Blue	SCL
SDA1 (Pin PB7)	Green	SDA

Table 4: Connection "Blue Pill" to I2C adapter of the LCD

setup 1 (Table 2).
The project's software directory is `TestBluePillI2C`.

5. Project 3: Test of Rotary Encoders

Figure 5 shows the test of two rotary encoders at the "Blue Pill" board. The connection

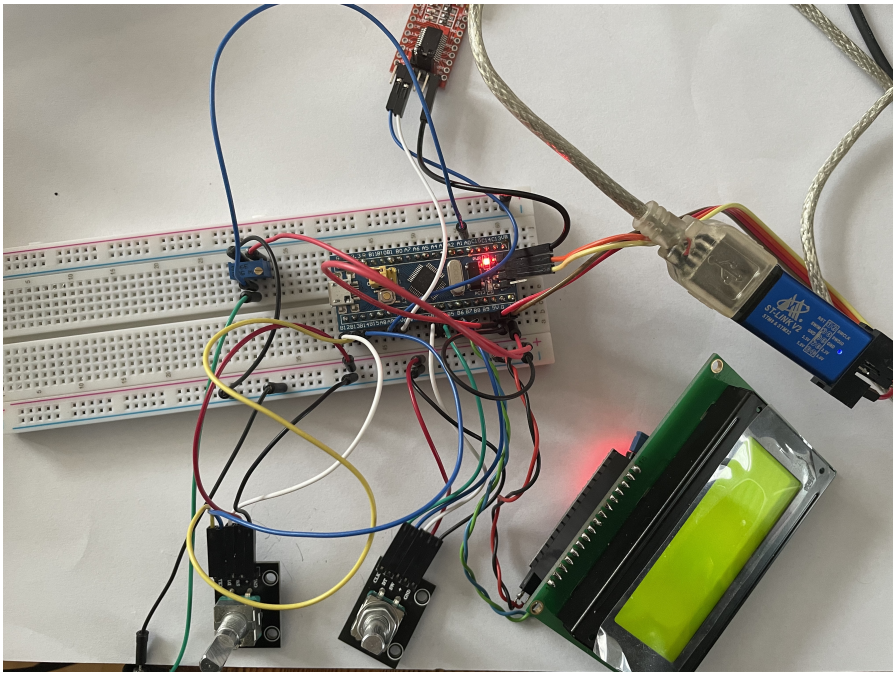


Figure 5: Rotary encoder test at the "Blue Pill" board

of the rotary encoders is done according to the following tables 5, 6):

Blue Pill	**Cable color**	**rotary encoder 1**
GND	Black	GND
5V	Red	5V
Pin PB12	White	SW
Pin PB13	Blue	DT
Pin PB14	Yellow	CLK

Table 5: Connection "Blue Pill" to rotary encoder 1

A key press or a turn of a rotary encoder creates a HW interrupt on the processor. The specific interrupt service routines evaluate the key hits and determine the turn direction. The project's software directory is `TestBluePillRotaryEncoders`.

Blue Pill	**Cable color**	**rotary encoder 2**
GND	Black	GND
5V	Red	5V
Pin PB3	White	SW
Pin PB4	Blue	DT
Pin PB5	Green	CLK

Table 6: Connection "Blue Pill" to rotary encoder 2

6. Project 4: Test of the ADC

The ADC of the "Blue Pill processor in the Arduino environment can be programmed to use for the resolution a different number of bits. Figure 6 shows the ADC characteristic of the "Blue Pill" processor for the 10bit and 12bit cases. The figures show in red color

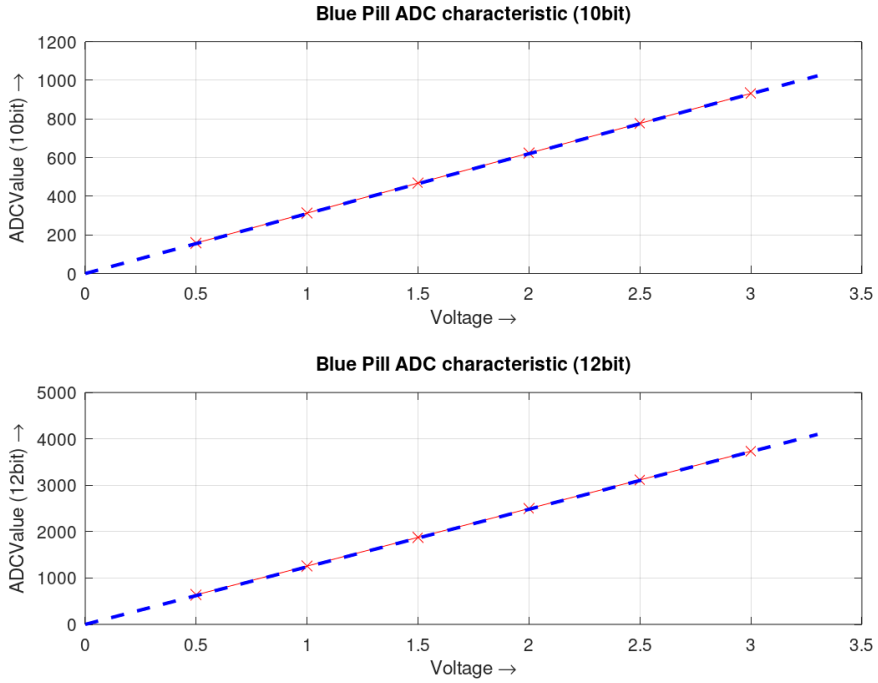


Figure 6: ADC characteristic of the "Blue Pill" processor

the measured ADC characteristic and in blue the ideal linear ADC characteristics. The ADC reference voltage (V_{ref}) is equal to the power supply voltage of the processor of 3.3V. The ADC value is given the as follows:

For the 10bit ADC resolution case:

$$\text{ADCValue} = \text{voltage (at ADC pin Ax)} \;/\; V_{ref} \cdot 1024$$

For the 12bit ADC resolution case:

$$\text{ADCValue} = \text{voltage (at ADC pin Ax)} \;/\; V_{ref} \cdot 4096$$

The project's software directory is `TestBluePillADC`.

7. Project 5: Test of PWM

The STM32 processor of the "Blue Pill" board has multiple PWM capable pins. Figure 7 shows the generated PWM signal of the "Blue Pill" processor on the screen of a hand-held oscilloscope.

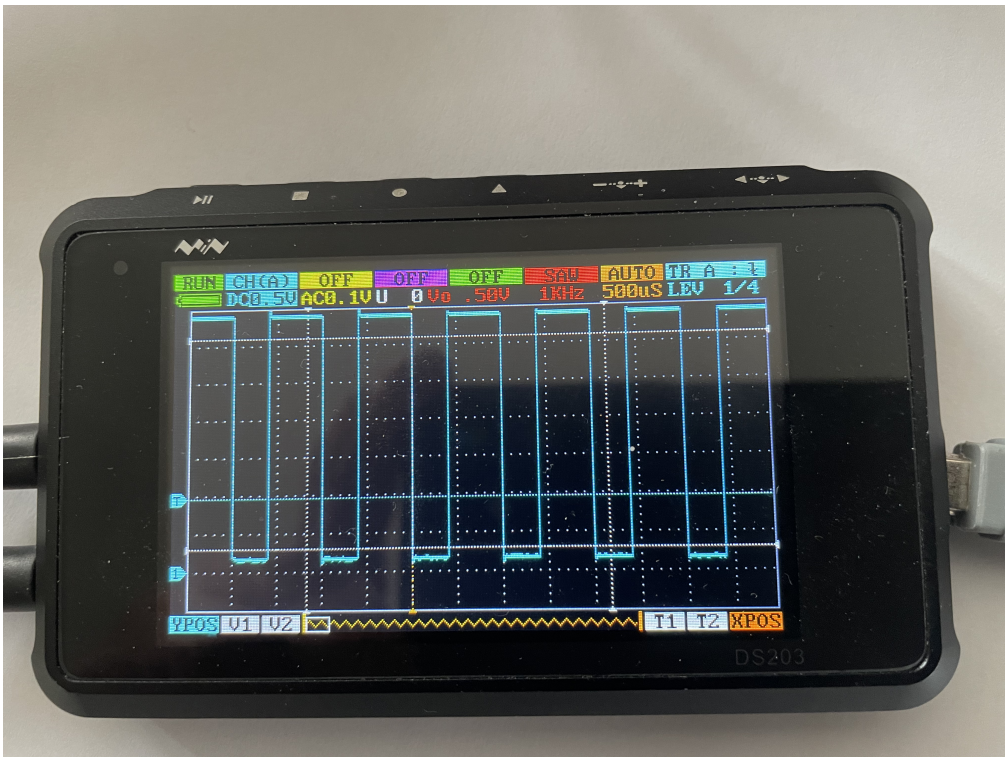


Figure 7: PWM signal of the "Blue Pill" processor

The project's software directory is `TestBluePillPWM`.

8. Project 6: Test of Flash Storage

The STM32 processor of the "Blue Pill" board has no EEPROM like AVR devices, but parts of the program flash memory can be reused to emulate EEPROM storage. The

project tested writing and retrieving standard and custom data types to and from the flash memory. It used the library from here Github Library FlashStorage_STM32.
The project's software directory is `TestBluePillFlashStorage`.

9. Project 7: Test of FreeRTOS (first example)

Installation of STM32FreeRTOS for the PlatformIO environment is described here:
https://platformio.org/lib/show/2093/STM32duino%20FreeRTOS/installation
Example sketches using STM32FreeRTOS are here:
https://github.com/stm32duino/STM32FreeRTOS/tree/master/examples

A first example project in PlatformIO that uses the STM32FreeRTOS real time operating system (RTOS) is created. It has two tasks. The first task is blinking the on-board LED, the second task is reading and digitizing via the ADC the voltage on port A0 and writing the result to the serial port.
The project's software directory is `TestFreeRTOS1`.

10. Project 8: Test of FreeRTOS C++ wrapper

A first example project in PlatformIO that uses the STM32FreeRTOS with a C++ wrapper for the task definition is created. It has two tasks. The first task is blinking the on-board LED, the second task is reading and digitizing via the ADC the voltage on port A0 and writing the result to the serial port.
The project's software directory is `TestFreeRTOSCPP1`.

A. Appendix A

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

Figure 8 shows the pin-out of the "Blue Pill" board.

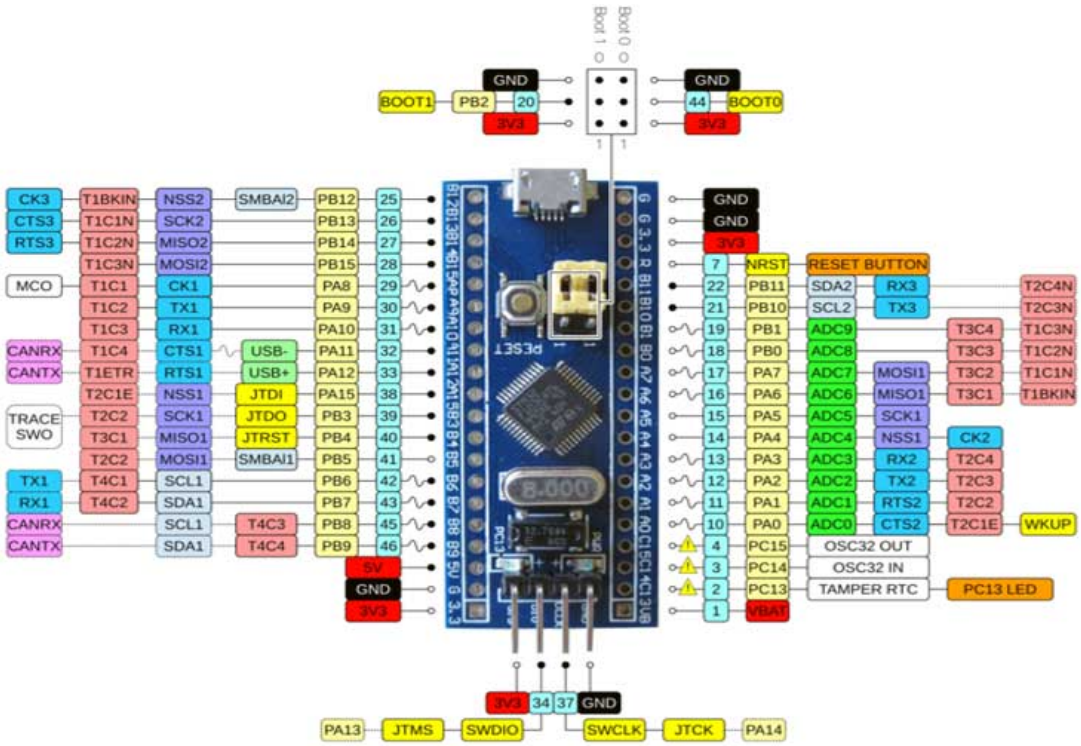


Figure 8: Pin-out of the "Blue Pill" board

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