

Homework

1 Brief

Repository **gigatik** is an STM32CubeIDE Version: 1.14.0 project. There is no STM32CubeMx HAL used for the implementation.

The project is written for STM32F103RB MCU on the Nucleo board. The system clock was set to 8 MHz. The project toggles two GPIO pins for a specific amount of time, set by messages received from UART. In software, the GPIO pins are used for toggling led0 and led1. There are two general-purpose timers (TIM3 and TIM4) that are responsible for toggling the pins for the correct amount of time. A ring buffer has been implemented for receiving UART messages.

2 Build options

The project needs to be built with STM32CubeIDE (see Figure 1).

There is a build option for UART baud rate and GPIO pin selection.

The led0 and led1, both can be assigned to one of GPIOB0, GPIOB1, GPIOB2, and GPIOB5.

Choose values 0, 1, 2, 5 for LED_0_GPIOB_PIN or LED_1_GPIOB_PIN.

To change the UART baud rate change the UART_BAUD_RATE.

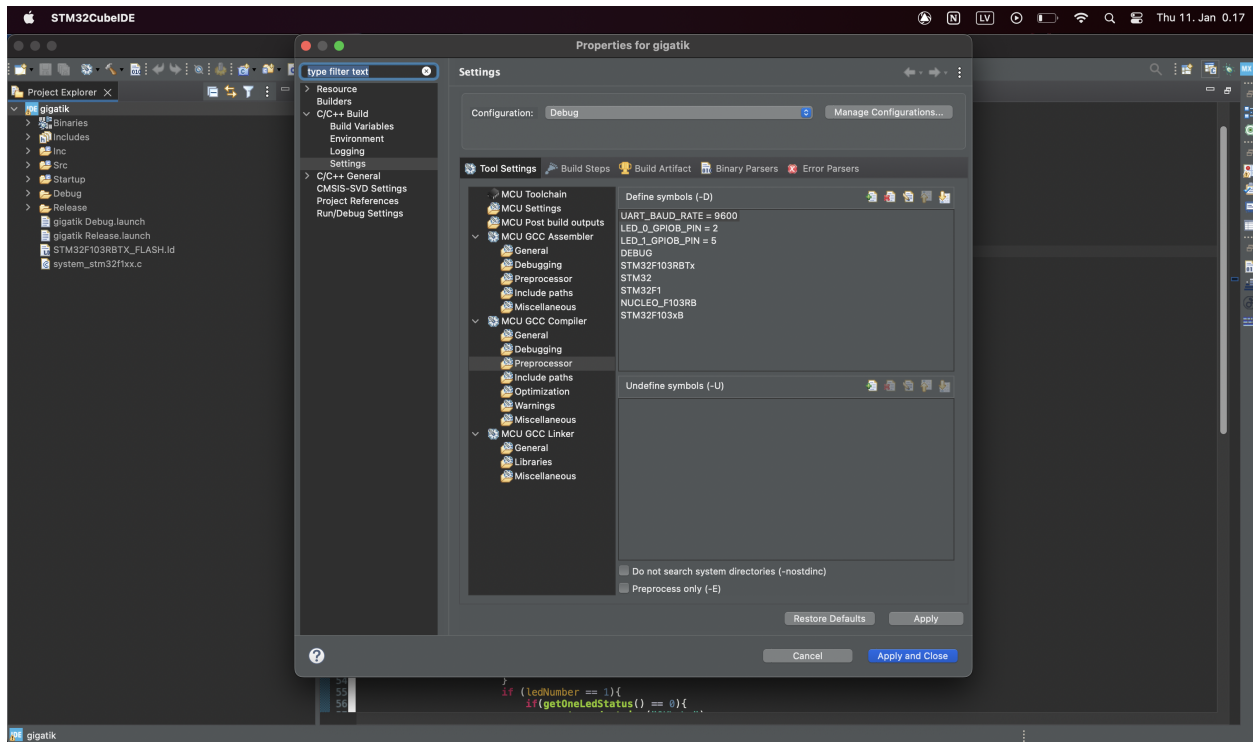


Figure 1: Build options in STM32CubeIDE

3 Accuracy

For the input command "set-led 0,50\r" measurements are shown in Figure 2. The measurements show 40 μ s of error, but I assume that it is even smaller. Measurements could be more accurate by increasing the sample rate of the USB oscilloscope.

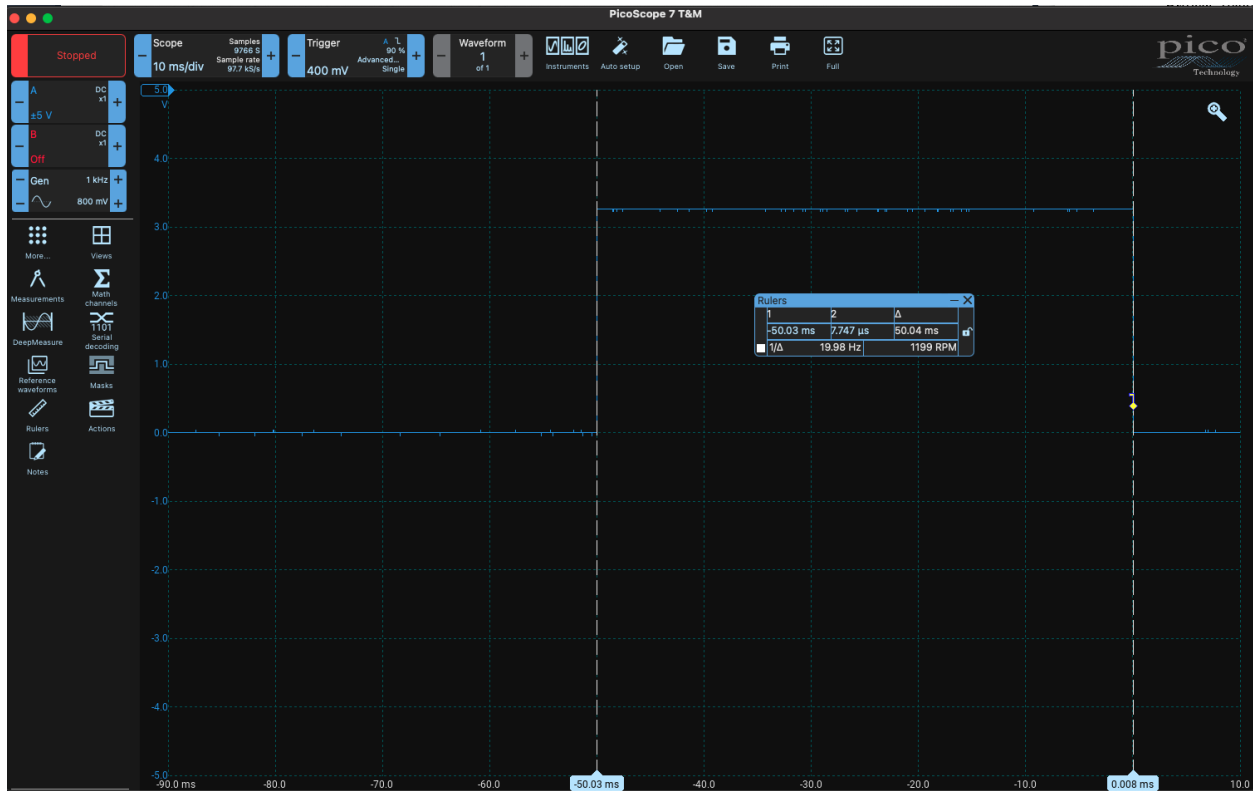


Figure 2: Measured led1 on-time on GPIOB2 for UART message "set-led 0,50\r"

4 Experience and possible improvements

Previously I had an experience with using STM32CubeMx HAL libraries. This is my first experience with making a whole project with bare metal programming. I choose to use STM32F103 with a Nucleo board for a better debugging experience. Initially, I used Atmega328p MCU, but the 8-bit restriction was quite frustrating.

The whole project does not show my C++ programming skills. I aimed to write a working program in C and then rewrite it in C++. C implementation was the initial goal because of many available examples on the Internet that could be easily adjustable for the exercise. I ran out of time and did not manage to write in C++. But it works in C for now.

The "echo" command works well, however, I did not measure the UART response time speed.

I implemented two timers for UART messages that toggle LEDs. This allows for simultaneously toggling both LEDs. But it does not register a UART message for the same LED, while it is toggled on. This could be overcome by implementing an instruction buffer, which stores instruction in memory. Instruction would be taken from the buffer when the LED shuts off. Like task scheduling.

I started to develop on STM32CubeIDE. In the end, I cannot build the project from the command line with different build options in the command line. Currently, one should still use STM32CubeIDE to build and choose different build options. The project could be managed, for example, in Visual Studio Code with a separate Makefile, which would run execute also different build options.

There is some future work to do, but I enjoyed the task!

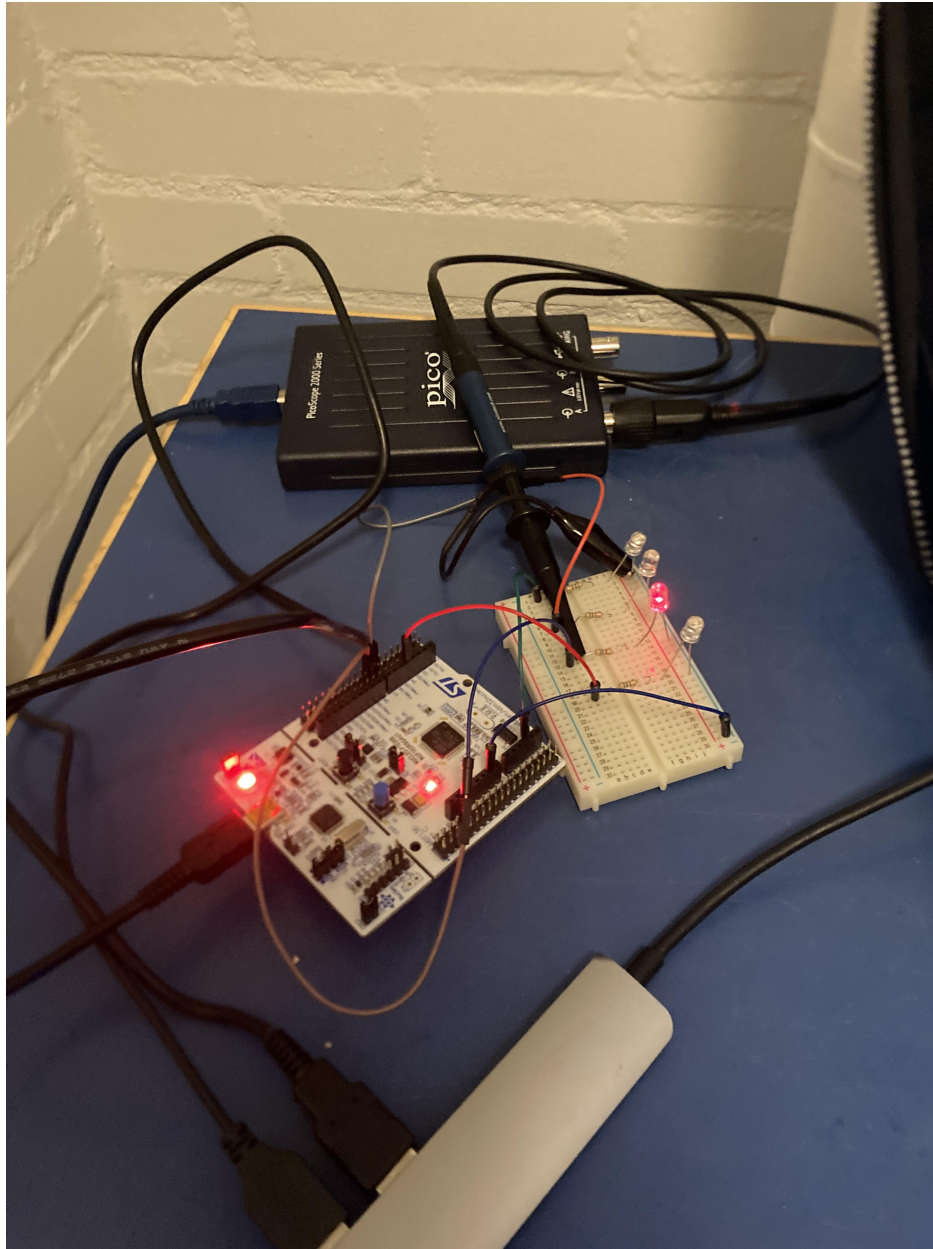


Figure 3: Hardware setup