



**VILNIUS UNIVERSITY
ŠIAULIAI ACADEMY**

BACHELOR PROGRAMME SOFTWARE ENGINEERING

Programming of Embedded Systems

Laboratory work No. 4

Student : Hanna Asiadouskaya

Lecturer: dr. Nerijus Ramanauskas

Šiauliai, 2025

Aim of the laboratory work:

The aim of the practical work is to understand USB Virtual COM Port, that includes toggling an LED, setting PWM duty cycle for another LED, and checking the state of a switch.

Variant No 2.

No	LED A	LED B
2	LED1	LED 3

Code and Comments:

Importing modules for GPIO control, for serial communication and PWM control.

Creates a USB Virtual COM Port object for sending and receiving data over USB, then Green LED and Red LED as an output pin. Configure the switch pin as an input, so 0 – not pressed, and 1 – pressed.

```
1 from machine import Pin
2 from pyb import USB_VCP, Timer
3
4 vcp = USB_VCP()
5 green_LED = Pin("LED1", Pin.OUT)
6 red_LED = Pin ("LED3", Pin.OUT)
7 pin_SW = Pin("SW", Pin.IN, Pin.PULL_DOWN)
8
9     - - - - -
```

There we are configuring PWM on Red LED, for it configuring timer, on that create PWM channel 2, and to that assign red led pin.

```
8
9 width_per = 1
10 tim = Timer(1, freq=1000)
11 ch = tim.channel(2, Timer.PWM, pin=red_LED)
12
13     - - - - -
```

Now let's observe algorithms in infinity loop.

Set that it waits for 7 bytes of data from USB.

There handling receiving led command, so when it received it toggle green led. If the LED is ON, turn it OFF and vice versa.

```
14     cmd = vcp.recv(7, timeout=200)
15     if (cmd == b'led'):
16         vcp.send("Green led toggled\r\n", timeout=200)
17         if green_LED.value():
18             green_LED.low()
19         else:
20             green_LED.high()
```

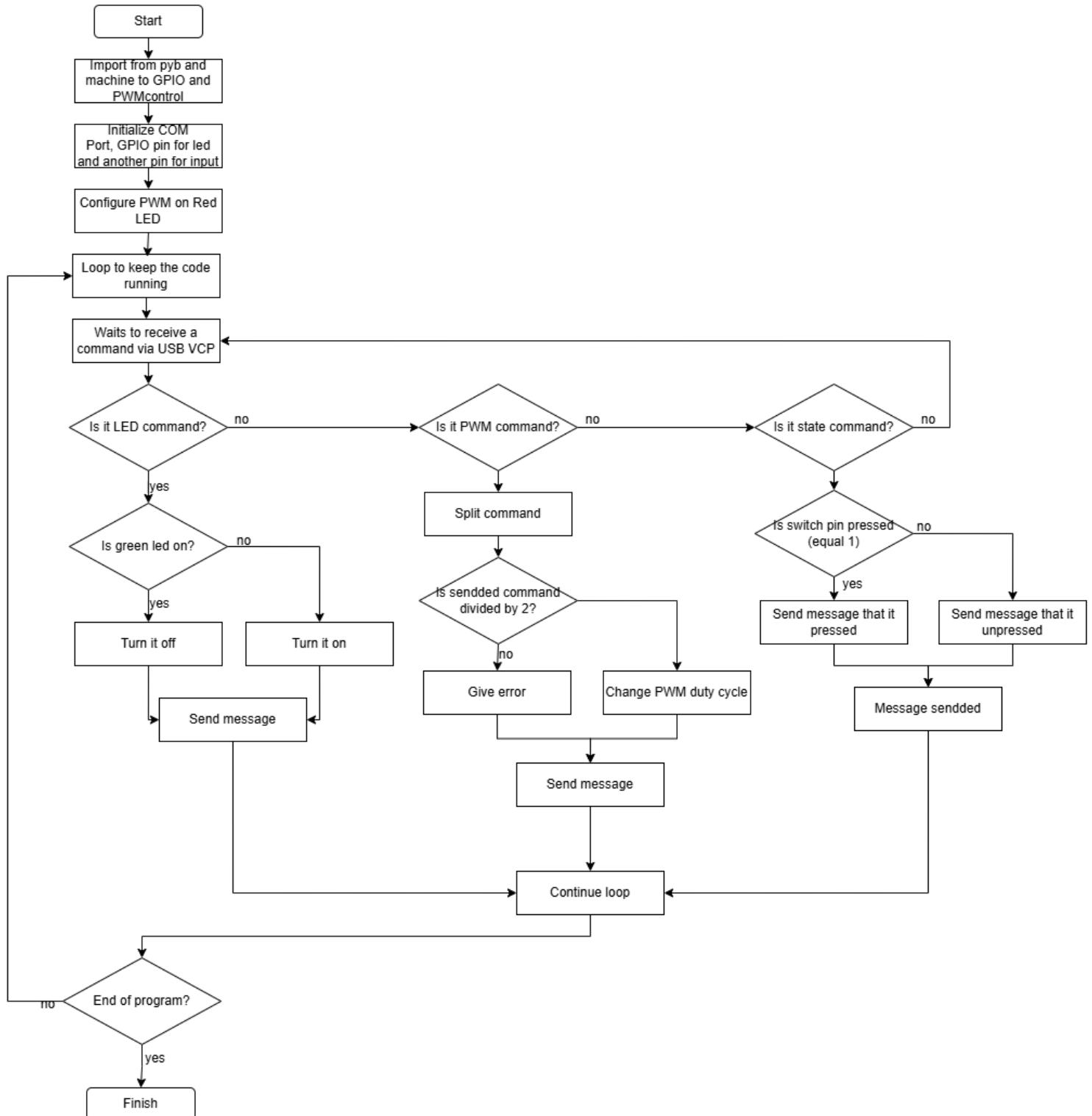
Next is checking if pwm command received, and handling the input so we can get the pwm and argument both in the same line. Also here handling if received more arguments then 2. And if there is no error, adjusts the red LED's brightness by.

```
21     elif ('pwm' in cmd):
22         divided = str(cmd).split(' ')
23         if len(divided) > 2:
24             vcp.send(f"Error, bad argument\r\n", timeout=200)
25         else:
26             value = int(divided[1].strip(''))
27             ch.pulse_width_percent(value)
28             vcp.send(f"Red led pulse changed\r\n", timeout=200)
```

And here part to check switch status (0 and 1)

```
29     elif (cmd == b'state'):
30         if pin_SW.value():
31             vcp.send(f"SW is pressed\r\n", timeout=200)
32         else:
33             vcp.send(f"SW is unpressed\r\n", timeout=200)
```

Algorithm:



Conclusion:

In this study, we successfully practiced programming on the STM32 Nucleo F756ZG microcontroller, explored data communication between microcontroller and PC COM port, configure PWM duty cycle for another LED, and checked the state of a switch.

Code as a text:

```
from machine import Pin
from pyb import USB_VCP, Timer

vcp = USB_VCP()
green_LED = Pin("LED1", Pin.OUT)
red_LED = Pin ("LED3", Pin.OUT)
pin_SW = Pin("SW", Pin.IN, Pin.PULL_DOWN)

width_per = 1
tim = Timer(1, freq=1000)
ch = tim.channel(2, Timer.PWM, pin=red_LED)

while True:
    cmd = vcp.recv(7, timeout=200)
    if (cmd == b'led'):
        vcp.send("Green led toggled\r\n", timeout=200)
        if green_LED.value():
            green_LED.low()
        else:
            green_LED.high()
    elif ('pwm' in cmd):
        divided = str(cmd).split(' ')
        if len(divided) > 2:
            vcp.send(f"Error, bad argument\r\n", timeout=200)
        else:
            value = int(divided[1].strip(""))
            ch.pulse_width_percent(value)
```

```
    vcp.send(f"Red led pulse changed\r\n", timeout=200)

elif (cmd == b'state'):
    if pin_SW.value():
        vcp.send(f"SW is pressed\r\n", timeout=200)
    else:
        vcp.send(f"SW is unpressed\r\n", timeout=200)
```