VILNIUS UNIVERSITY
SIAULIAI ACADEMY

PROGRAMŲ SISTEMOS BACHELOR STUDY PROGRAMME

Software engineering

ANNA KUTOVA

Programming of Embedded Systems
Laboratory work No.4
Virtual COM Port (VCP)

Šiauliai, 2025

Laboratory Work Report

Table of Contents

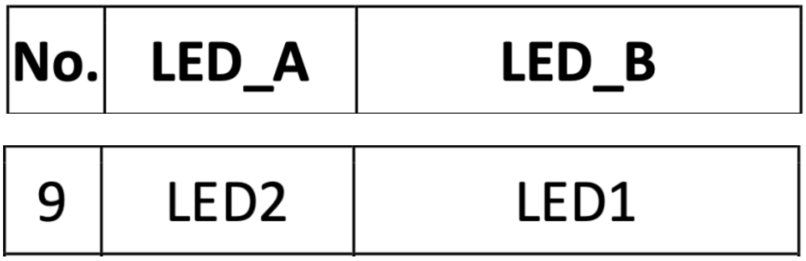
1. The Aim of the Laboratory Work

The purpose of this laboratory work is to study the Virtual COM Port (VCP) and implement data communication between a microcontroller and a PC via the USB-to-serial interface. The task involves developing a program that allowes:

- to turn ON / turn OFF LED_A
- to change the PWM signal duty cycle on LED_B (PWM frequency on your own choice)
- to receive the state of SW pin.

2. Variant No and Data

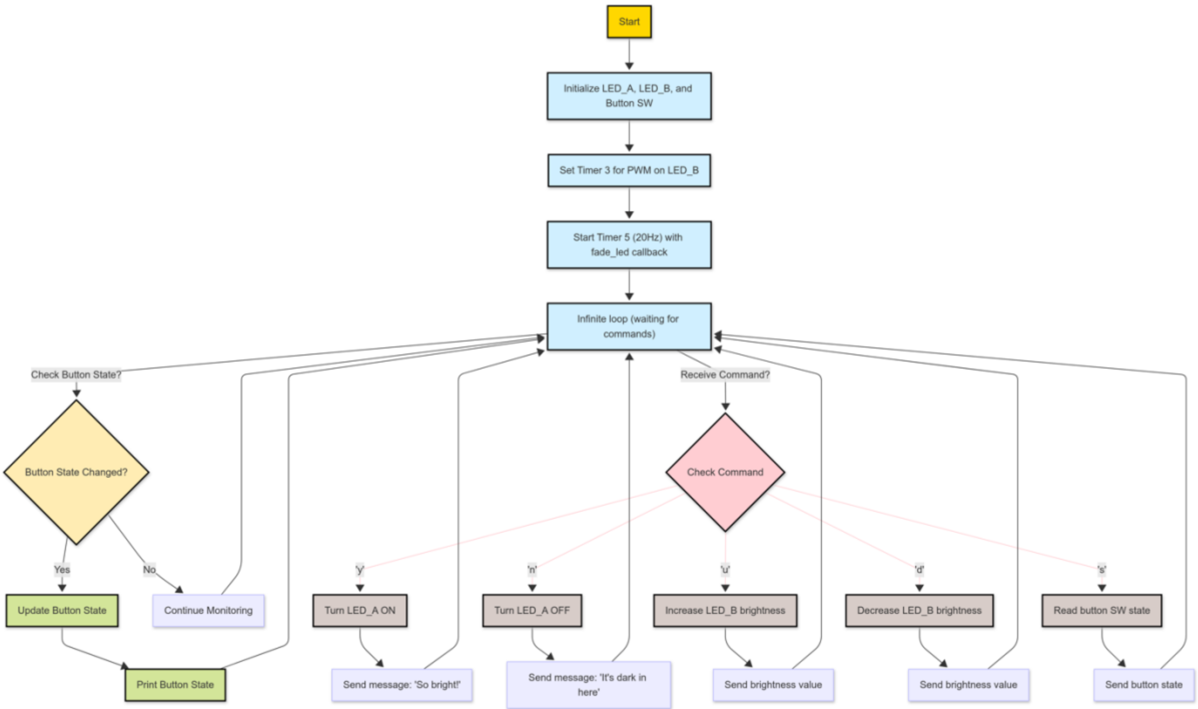
Variant No: 9



No.	LED_A	LED_B
9	LED2	LED1

Date: 04/03/2025

3. Program Algorithm



4. Program Body with Comments

```
from pyb import USB_VCP, Pin, Timer

# Initialize button SW
pin_SW = Pin("SW", Pin.IN, Pin.PULL_DOWN)

#-------- LED_A (on/off control)
vcp = USB_VCP()
blue_light = Pin("LED2", Pin.OUT)   # LED_A is LED2

#-------- LED_B (PWM control on LED1)
p = Pin("LED1")  # LED_B is LED1

# Set TIM3 frequency to 6000Hz
tim = Timer(3, freq=6000)

# Configure PWM on Channel 3
ch = tim.channel(3, Timer.PWM, pin=p)

# Initial PWM duty cycle
brightness = 0    # Start at 0%
step = 15          # Step for manual brightness change

def fade_led(timer):
    global brightness, step
    brightness += step
    if brightness >= 100 or brightness <= 0:
        step = -step  # Reverse direction
    ch.pulse_width_percent(brightness)

# Use timer interrupt to change brightness
```

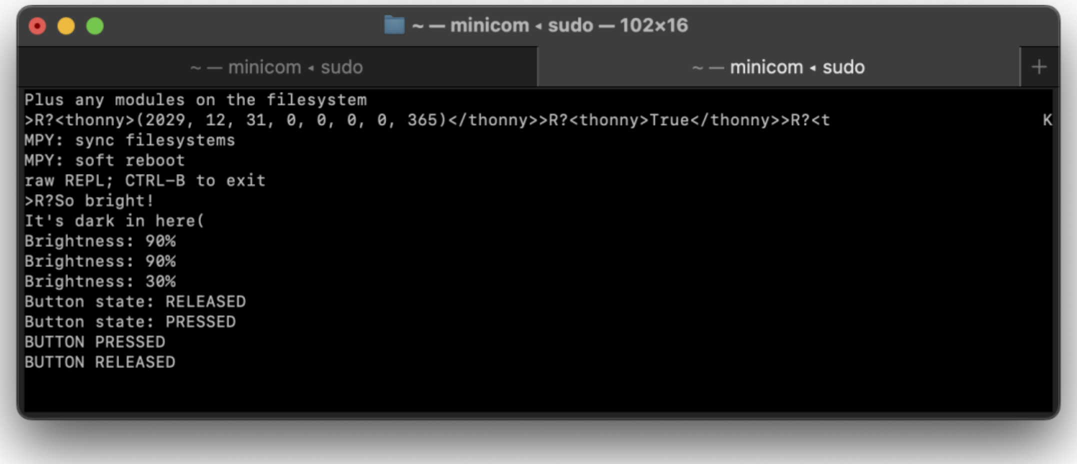
```python
fade_timer = Timer(5)
fade_timer.init(freq=20, callback=fade_led)  # 20Hz → period 50ms

# Initialize the button state
prev_state = pin_SW.value()

while True:
    cmd = vcp.recv(1, timeout=5000)  # Expecting 1-character commands
    if cmd:  # Check if cmd is not None
        if cmd == b'y':  # Turn LED_A ON
            blue_light.high()
            vcp.send("So bright! \r\n", timeout=5000)
        elif cmd == b'n':  # Turn LED_A OFF
            blue_light.low()
            vcp.send("It's dark in here( \r\n", timeout=5000)
        elif cmd == b'u':  # Increase LED_B brightness
            if brightness + step <= 100:
                brightness += step
            ch.pulse_width_percent(brightness)
            vcp.send(f"Brightness: {brightness}%\r\n", timeout=5000)
        elif cmd == b'd':  # Decrease LED_B brightness
            if brightness - step >= 0:
                brightness -= step
            ch.pulse_width_percent(brightness)
            vcp.send(f"Brightness: {brightness}%\r\n", timeout=5000)
        elif cmd == b's':  # Get button state
            state = "PRESSED" if pin_SW.value() else "RELEASED"
            vcp.send(f"Button state: {state}\r\n", timeout=5000)

    # Detect button state change
    current_state = pin_SW.value()
    if current_state != prev_state:
        if current_state:
            print("BUTTON PRESSED")
        else:
            print("BUTTON RELEASED")
        prev_state = current_state
```

5. Program screenshot



```
Plus any modules on the filesystem
>R?<thonny>(2029, 12, 31, 0, 0, 0, 0, 365)</thonny>>R?<thonny>True</thonny>>R?<t                  K
MPY: sync filesystems
MPY: soft reboot
raw REPL; CTRL-B to exit
>R?So bright!
It's dark in here(
Brightness: 90%
Brightness: 90%
Brightness: 30%
Button state: RELEASED
Button state: PRESSED
BUTTON PRESSED
BUTTON RELEASED
```

6. Conclusions

- Virtual COM Port (VCP) allows communication between a microcontroller and a PC using a USB connection.
- The program receives commands from the PC to control LED_A and adjust the brightness of LED_B using PWM.
- Messages are sent back to the PC to confirm actions, such as turning LED_A on/off.
- The microcontroller detects button state changes and prints the updated state.
- Using PWM and timers, the LED brightness smoothly changes without continuous manual control.
- This lab demonstrates how VCP can be used for real-time control and feedback in embedded systems.