****

**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 2](#_Toc192090654)

[2. Variant No and Data 2](#_Toc192090655)

[3. Program Algorithm 3](#_Toc192090656)

[4. Program Body with Comments 3](#_Toc192090657)

[5. Program screenshot 5](#_Toc192090658)

[6. Conclusions 5](#_Toc192090659)

# **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.

1. **Variant No and Data**

**Variant No:** 9

**A white rectangular object with black lines

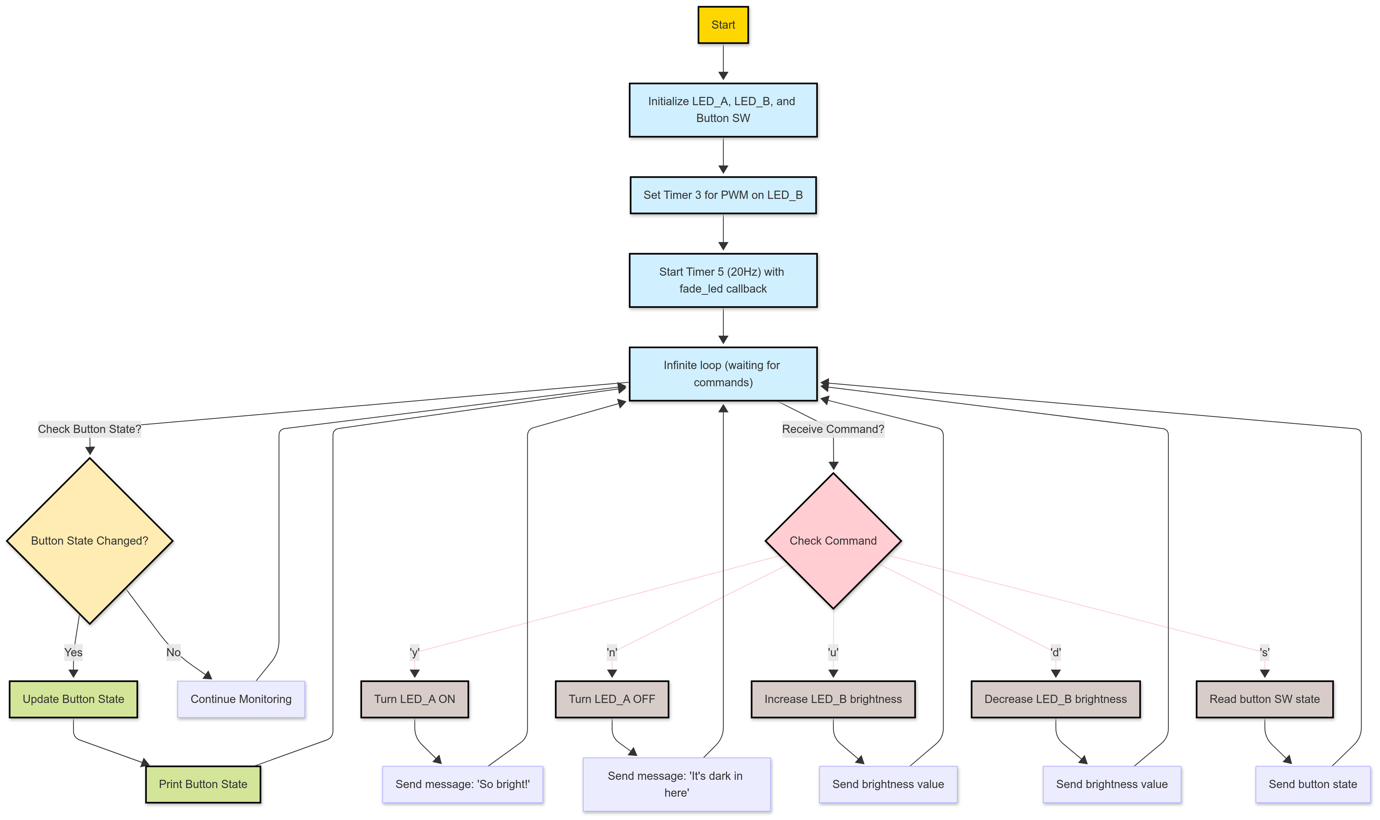
AI-generated content may be incorrect.**

A white rectangular object with black lines

AI-generated content may be incorrect.

**Date:** 04/03/2025

# **Program Algorithm**



# **Program Body with Comments**

1. **from** **pyb** **import** USB\_VCP, Pin, Timer
2. *# Initialize button SW*
3. pin\_SW = Pin("SW", Pin.IN, Pin.PULL\_DOWN)
4. *#-------- LED\_A (on/off control)*
5. vcp = USB\_VCP()
6. blue\_light = Pin("LED2", Pin.OUT) *# LED\_A is LED2*
7. *#-------- LED\_B (PWM control on LED1)*
8. p = Pin("LED1") *# LED\_B is LED1*
9. *# Set TIM3 frequency to 6000Hz*
10. tim = Timer(3, freq=6000)
11. *# Configure PWM on Channel 3*
12. ch = tim.channel(3, Timer.PWM, pin=p)
13. *# Initial PWM duty cycle*
14. brightness = 0 *# Start at 0%*
15. step = 15 *# Step for manual brightness change*
16. **def** fade\_led(timer):
17. **global** brightness, step
18. brightness += step
19. **if** brightness >= 100 **or** brightness <= 0:
20. step = -step *# Reverse direction*
21. ch.pulse\_width\_percent(brightness)
22. *# Use timer interrupt to change brightness*
23. fade\_timer = Timer(5)
24. fade\_timer.init(freq=20, callback=fade\_led) *# 20Hz → period 50ms*
25. *# Initialize the button state*
26. prev\_state = pin\_SW.value()
27. **while** True:
28. cmd = vcp.recv(1, timeout=5000) *# Expecting 1-character commands*
29. **if** cmd: *# Check if cmd is not None*
30. **if** cmd == b'y': *# Turn LED\_A ON*
31. blue\_light.high()
32. vcp.send("So bright! **\r\n**", timeout=5000)
33. **elif** cmd == b'n': *# Turn LED\_A OFF*
34. blue\_light.low()
35. vcp.send("It's dark in here( **\r\n**", timeout=5000)
36. **elif** cmd == b'u': *# Increase LED\_B brightness*
37. **if** brightness + step <= 100:
38. brightness += step
39. ch.pulse\_width\_percent(brightness)
40. vcp.send(f"Brightness: {brightness}%**\r\n**", timeout=5000)
41. **elif** cmd == b'd': *# Decrease LED\_B brightness*
42. **if** brightness - step >= 0:
43. brightness -= step
44. ch.pulse\_width\_percent(brightness)
45. vcp.send(f"Brightness: {brightness}%**\r\n**", timeout=5000)
46. **elif** cmd == b's': *# Get button state*
47. state = "PRESSED" **if** pin\_SW.value() **else** "RELEASED"
48. vcp.send(f"Button state: {state}**\r\n**", timeout=5000)
49. *# Detect button state change*
50. current\_state = pin\_SW.value()
51. **if** current\_state != prev\_state:
52. **if** current\_state:
53. **print**("BUTTON PRESSED")
54. **else**:
55. **print**("BUTTON RELEASED")
56. prev\_state = current\_state
57. **Program screenshot**A screenshot of a computer

    AI-generated content may be incorrect.

# **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.