

src/Uebungen/plf01_ledstrip_pub_template.py

```
1 #Exercise Title:    "LED-Strip"
2 #Name:              ?
3 #Group:             ?
4 #Class:             ?
5 #Date:              ?
6
7 import rclpy
8 import rclpy.logging
9 from rclpy.qos import QoSProfile
10
11 import os
12 import select
13 import sys
14 import time
15
16 import termios
17 import tty
18
19 # Fragen:
20 #-----
21 # Frage 1 - Wie kannst du die ROS_DOMAIN_ID in der CLI auf 0 setzen?
22 # Antwort 1: export ROS_DOMAIN_ID = 0
23 # Frage 2 - Wofür wird die ROS_DOMAIN_ID verwendet, und was passiert, wenn du sie
24 # auf 0 setzt?
25 # Antwort 2: Um eine adresse im Netzwerk direkt anzusprechen. Wenn sie 0 ist,
26 # wird die standartadresse verwendet
27 # Frage 3 - Warum haben Anwender, die eine virtuelle Maschine nutzen, Probleme
28 # damit, ROS-Nachrichten von Remote-Nodes anzuzeigen?
29 # Antwort 3: Weil die Virtuelle maschine ein eigenes netzwerk hat(ist mit "lan"
30 # mit dem computer verbunden)
31 # Frage 4 - Wie kannst du über die CLI herausfinden, welche ROS-Nachrichten
32 # gerade verfügbar sind?
33 # Antwort 4: ros2 topic list
34 # Frage 5 - Wie lautet der CLI-Befehl, um die USER-LED zu schalten?
35 # Antwort 5: ros2 topic pub /user std_msgs/String "data: engsan21"
36 # Frage 6 - Wie lautet der CLI-Befehl, um die TEAM-LED zu schalten?
37 # Antwort 6: ros2 topic pub /team4 std_msgs/UInt8MultiArray "data:
38 # [4,255,255,255,255]"
39 # Frage 7 - Was ist ein PWM-Signal, wofür wird es in unserem Beispiel verwendet,
40 # und wie bzw. von wem wird es erzeugt?
41 # Antwort 7: Ein rechtecksignal, dessen 1 und 0 zeit moduliert werden können. Die
42 # periodendauer bleibt dabei gleich.
43 # Um die LEDS anzusteuern. Das Signal kann vonn der LED
44 # "verarbeitet" werden
45 # Es wird von einem Quartz auf einem raspberry erzeugt
46 # Frage 8 - Was musst du tun, um deinen ROS-Node über einen ROS-Befehl in der CLI
47 # starten zu können? Wie könnte das Programm alternativ gestartet werden?
48 # Antwort 8: ros2 run turtlesim turtlesim_node. Es sollte möglich sein, das
49 # programm über die taskleiste zu starten
50 # Frage 9 - Welche LED-Farbe hat deine individuelle LED?
51 # Antwort 9: Grün
```

```
41 # Frage 10 - Welchen Datentyp müssen die ROS-Nachrichten mit den Topics "user"
    und "team" haben?
42 # Antwort 10: user: string(kürzel) team: UInt8MultiArray(liste aus mehreren
    bytes)
43
44
45 # Programmieraufgabe 1:
46 # -----
47 # Programmiere ein Programm zur Steuerung von RGBW-LEDs und speichere diese in
    dein robu Paket. Ein Subscriber-Node wurde bereits programmiert
48 # und auf dem Raspberry gestartet
49 # (siehe: https://github.com/mlieschnegg/robu\_rpi\_ws/blob/main/src/robu\_rpi\_examples/robu\_rpi\_examples/ledstrip\_sub.py, von LI/SH).
50 # Die Topics dieser ROS-Nachrichten lauten "user" sowie "team1" bis "team6".
51 #
52 # Am LED-Streifen sind folgende LEDs reserviert:
53 #   - Eine individuelle LED (Topic "user") für dich.
54 #   - Fünf LEDs für deine ROBU-Gruppe (Topics "team1" bis "team6").
55 #
56 # Implementiere:
57 #   - Je einen Publisher für deine individuelle LED sowie die LEDs deines Teams.
58 #
59 # Anforderungen:
60 #   - Wenn du die Taste 'u' drückst, soll die individuelle LED ein- bzw.
    ausgeschaltet werden.
61 #   - Wenn du die Taste 't' drückst, sollen die LEDs deines Teams mit einer
    bestimmten Farbe (freie Wahl!) konfiguriert werden.
62 #
63 # Hinweis:
64 #   - Der grundlegende Aufbau deines Programms entspricht dem Programm aus Übung
    6.
65 #   - Studiere den Code des Subscribers und finde heraus welchen Datentyp und
    Nachrichteninhalt
66 #     du beim Publisher zum Schalten der LEDs verwenden musst.
67
68 #Abgabe
69 #-----
70 #Drucke diese Dokument aus (doppelseitig) und gib es bei LI/SH ab
71 from std_msgs.msg import String, UInt8MultiArray
72
73 def get_key():
74     old_settings = termios.tcgetattr(sys.stdin)
75     ts = time.time()
76     key = ''
77     try:
78         tty.setraw(sys.stdin.fileno())
79         while True:
80             rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
81             if rlist:
82                 key += os.read(sys.stdin.fileno(), 1).decode("utf-8")
83             else:
84                 break
85     return key
```

```
86     finally:
87         termios.tcsetattr(sys.stdin, termios.TCSADRAIN, old_settings)
88
89
90 def main():
91     rclpy.init()
92
93     qos = QoSProfile(depth=10)
94
95     node = rclpy.create_node('ledstrip')
96     pubu = node.create_publisher(String, 'user', qos)
97     pubt = node.create_publisher(UInt8MultiArray, 'team4', qos)
98
99     teammsg = UInt8MultiArray()
100    usermsg = String()
101    usermsg.data = 'engsan21'
102    i = 0
103    a = 0
104
105    try:
106        while(1):
107            key = get_key()
108            if key != '':
109                str = "String: " + key.replace(chr(0x1B), '^') + ", Code:"
110                for c in key:
111                    str += " %d" % (ord(c))
112                print(str)
113                if key == 'u':
114                    pubu.publish(usermsg)
115                elif key == 't':
116                    if(i==0):
117                        teammsg.data = [4,255,255,255,255]
118                        pubt.publish(teammsg)
119                        i=1
120                    if(i==1):
121                        teammsg.data = [4,0,0,0,0]
122                        pubt.publish(teammsg)
123                        i=0
124
125    except Exception as e:
126        print(e)
127    rclpy.shutdown()
128
129
130 if __name__ == '__main__':
131     main()
```