

Vežbe 11 - Zadaci

Instalirati virtuelno okruženje:

sudo apt install python3-venv

Napraviti virtuelno okruženje:

python3 -m venv ime_okruzenja

Aktivirati virtuelno okruženje:

source ime_okruzenja/bin/activate

Instalirati PySide6 komandom:

pip3 install PySide6

Otvoriti folder u kom je napravljeno virtuelno okruženje u VScode. Prilikom realizacije .py fajla, VScode bi trebao prepoznati interpreter u donjem desnom uglu (Python) a pored se nalazi verzija pajtona gde u zagradi pored broja verzije mora pisati (*'ime_okruzenja':venv*). Ako to nije slučaj, kliknuti na verziju i iz padajućeg menija odabrati interpreter virtuelnog okruženja. Ako to nije moguće aktivirati ga manuelno kao što je objašnjeno u koracima iznad.

Dizajner bi trebao biti dostupan sa instalacijom pyside6. Pokreće se iz terminala komandom:

pyside6-designer

Za prevođenje ui dizajna u py fajl koristiti komandu: ***pyside6-uic ime_fajla.ui > ime_fajla.py***

Ako se pojavi greška:

qt.qpa.plugin: Could not load the Qt platform plugin 'xcb' in " even though it was found. This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb. Aborted (core dumped)

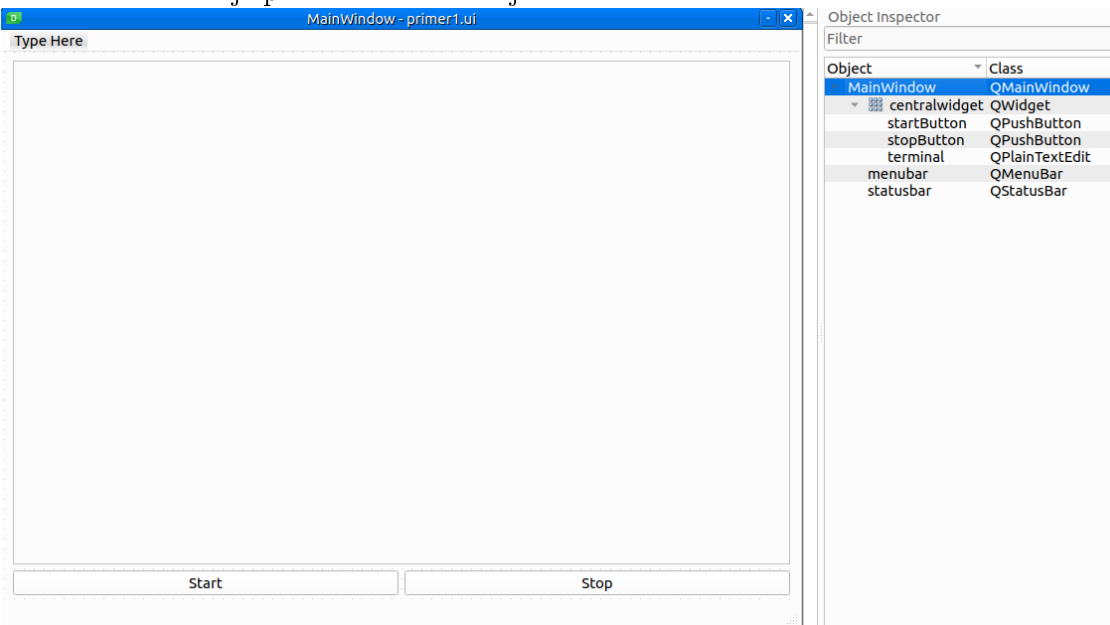
Rešenje:

sudo apt install '^libxcb.*-dev'

Zadatak 1. Realizovati aplikaciju koja prima podatke poslate preko serijskog porta i te podatke prikazuje u terminalu (QPlainTextEdit). Pokrenuti dizajner iz terminala komandom

pyside6-designer

U MainWidnow dodati QPlainTextEdit, nazvati ga terminal, dodati dva QPushButton-a i dati imena startButton i stopButton. Sve elemente postaviti da budu poravnati u GridLayout-u. Izgled bi trebao biti kao što je prikazan na sledećoj slici.



Sačuvati templejt kao primer1.ui, u folderu gde se želi realizovati aplikacija. Ponovo otvoriti terminal i izvršiti konverziju u python kod pozivom komande

pyside6-uic primer1.ui -o primer1_uic.py

Kao izlaz dobija se novi python fajl *primer1_uic.py* koji će biti učitani u glavni program.

```
1     import sys
2     from primer1_ui import Ui_MainWindow
3     from PySide6.QtWidgets import QApplication, QMainWindow
4     from PySide6 import QtSerialPort, QtCore
5
6     class MainWindow(QMainWindow):
7         def __init__(self):
8             super().__init__()
9             self.ui = Ui_MainWindow()
10            self.ui.setupUi(self)
11
12            self.ui.startButton.clicked.connect(self.start_com)
13            self.ui.stopButton.clicked.connect(self.stop_com)
14            self.serial = QtSerialPort.QSerialPort('/dev/ttyUSB0',
15                                                    baudRate=QtSerialPort.QSerialPort.BaudRate.Baud115200,
16                                                    readyRead=self.receive)
17
18
```

```

19     def receive(self):
20         while self.serial.canReadLine():
21             text = self.serial.readLine().data().decode()
22             text = text.rstrip('\n')
23             self.ui.terminal.appendPlainText(text)
24
25     def start_com(self):
26         if not self.serial.isOpen():
27             self.ui.startButton.setDisabled(True)
28             self.ui.stopButton.setDisabled(False)
29             self.serial.open(QtCore.QIODevice.OpenModeFlag.ReadWrite)
30
31     def stop_com(self):
32         if self.serial.isOpen():
33             self.serial.close()
34             self.ui.startButton.setDisabled(False)
35             self.ui.stopButton.setDisabled(True)
36
37     if __name__ == '__main__':
38         app = QApplication(sys.argv)
39         window = MainWindow()
40         window.show()
41         sys.exit(app.exec())

```

Zadatak 2. Na identičnom izledu aplikacije realizovati identičan primer, ali sada koristiti biblioteku serial.

```

1  import sys
2  from PySide6.QtCore import QThread, Signal, Slot
3  from PySide6.QtWidgets import QMainWindow, QApplication
4  from primer1_ui import Ui_MainWindow
5  import serial
6
7  class MainWindow(QMainWindow):
8      def __init__(self):
9          super().__init__()
10         self.ui = Ui_MainWindow()
11         self.ui.setupUi(self)
12         self.ui.startButton.clicked.connect(self.start_com)
13         self.ui.stopButton.clicked.connect(self.stop_com)
14         self.ui.stopButton.setDisabled(True)
15
16         @Slot()
17         def start_com(self):
18             self.ui.startButton.setDisabled(True)
19             self.ui.stopButton.setDisabled(False)
20
21             self.th = Worker()

```

```

22     self.th.signalData.connect(self.print_data)
23     self.th.signalFinished.connect(self.th.deleteLater)
24     self.th.start()
25
26     @Slot()
27     def stop_com(self):
28         self.ui.startButton.setDisabled(False)
29         self.ui.stopButton.setDisabled(True)
30         self.th.serial.close()
31
32     @Slot()
33     def print_data(self, text):
34         self.ui.terminal.appendPlainText(text)
35
36 class Worker(QThread):
37     signalData = Signal(str)
38     signalFinished = Signal()
39
40     def __init__(self):
41         super().__init__()
42
43     def run(self):
44         text = ''
45         self.serial = serial.Serial('/dev/ttyUSB0', 115200)
46         while self.serial.is_open:
47             try:
48                 text = self.serial.readline().decode()
49                 text = text.rstrip('\n')
50                 self.signalData.emit(text)
51             except:
52                 print(text)
53         self.signalFinished.emit()
54
55 if __name__ == '__main__':
56     app = QApplication(sys.argv)
57     window = MainWindow()
58     window.show()
59     sys.exit(app.exec())

```