How to communicate via a COM port using Qt

Dr Frazer Noble

Introduction

In this presentation, I will describe:

- How to control an Arduino's pin via a COM port using Qt.

Getting started

Create a new directory named "qt_python_serial" in `C:\Users\%USER%/`.

Note: Replace `%USER%` with your username.

Connect an Arduino to your computer.

Open the Start menu, type "Device Manager", and press `Enter`. This will display Windows' Device Manager.

Expand the "Ports" menu to view the Arduino's COM port.

Something similar to the following will be displayed:

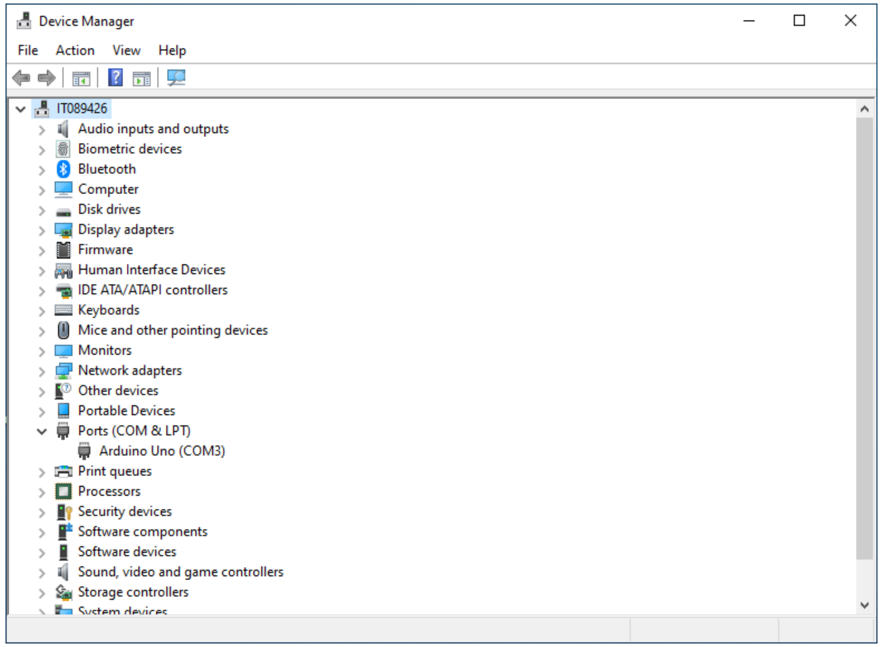


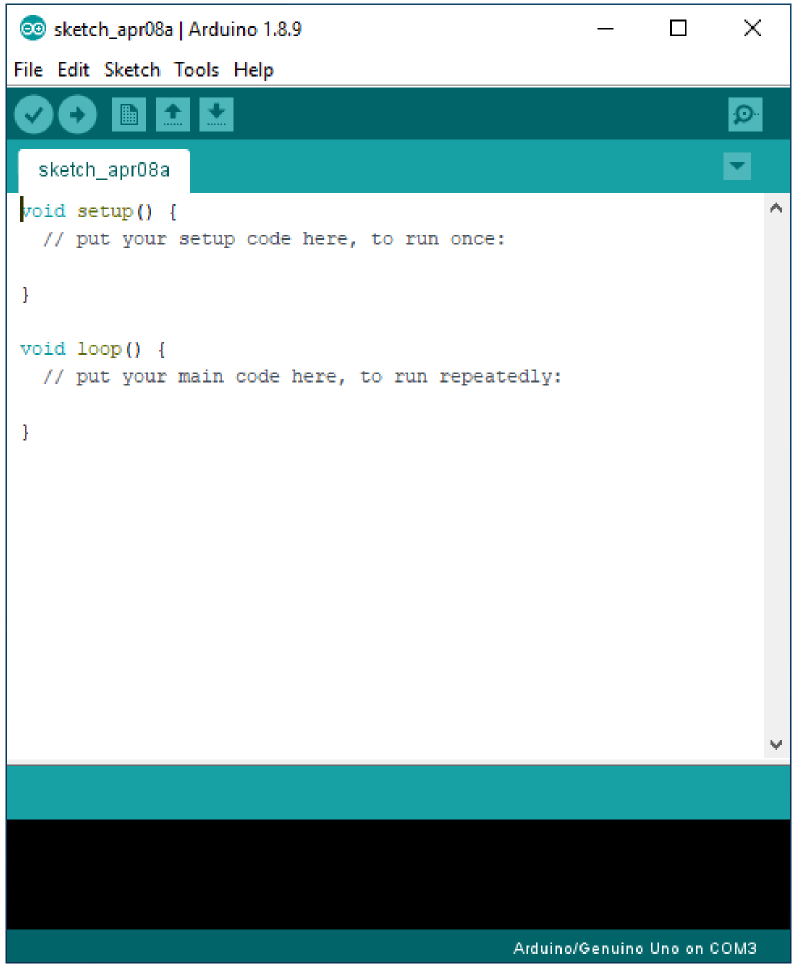
Figure: Windows' Device Manager. Here, we can see that an Arduino Uno is connected to the computer. It has enumerated as `COM3` .

Open the Arduino IDE.

Open the Start menu, type "Arduino", and press `Enter`. This will display the Arduino IDE.

Create a new sketch: Select "File > New" or press `Ctrl`+`N`.

Something similar to the following will be displayed:


sketch_apr08a | Arduino 1.8.9
File Edit Sketch Tools Help
sketch_apr08a
Arduino/Genuino Uno on COM3

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure: Arduino IDE.

Type the following C++ program into the file:

```
void setup()
{
  Serial.begin(9600);

  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
}
```

```
void loop()
{
  if (Serial.available() >= 2)
  {
    char buffer[3] {};

    Serial.readBytes(buffer, 3);

    int pin {atoi(buffer)};

    switch(pin)
    {
      case 13:
      {
        digitalWrite(13, !digitalRead(led));
        Serial.println(digitalRead(led));
        break;
      }
    }
  }
}
```

Save the sketch in `C:\Users\%USER%\qt_python_serial`: either:

1. Select "File > Save";

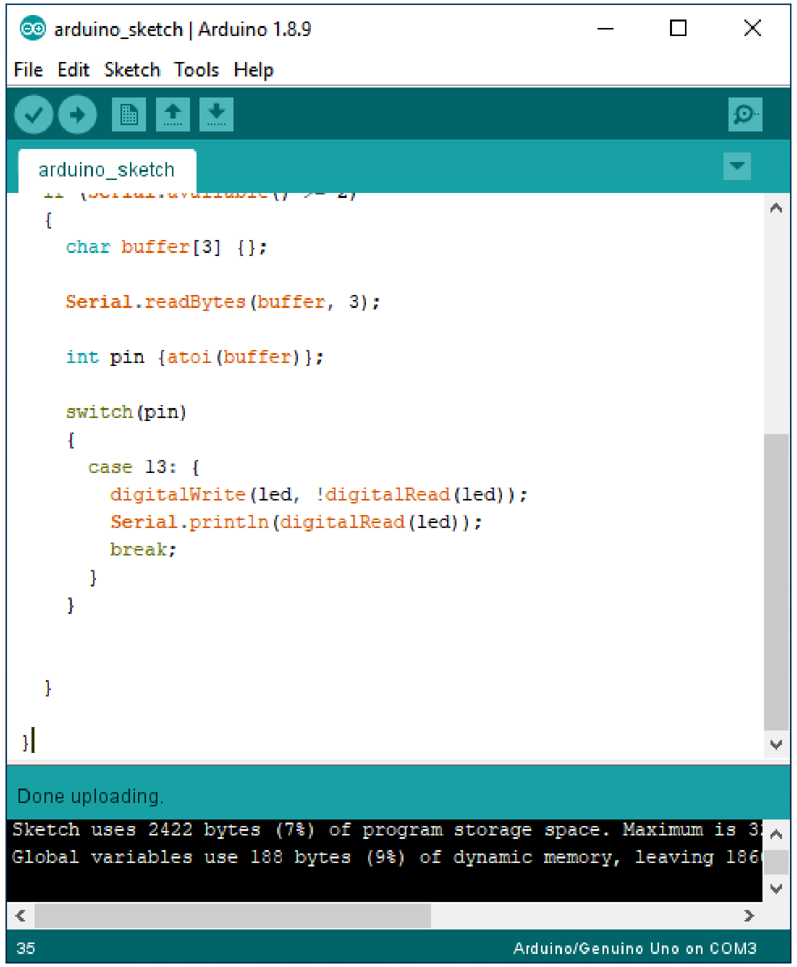
   or,
2. Press `Ctrl`+`S`, and navigate to `C:\Users\%USER%\qt_python_serial`.

Upload the sketch to the Arduino: either:

1. Select "Sketch > Upload";

   or,
2. Press the → button.

Something similar to the following will be displayed:


arduino_sketch | Arduino 1.8.9
File Edit Sketch Tools Help
arduino_sketch

```
  {
    char buffer[3] {};

    Serial.readBytes(buffer, 3);

    int pin {atoi(buffer)};

    switch(pin)
    {
      case 13: {
        digitalWrite(led, !digitalRead(led));
        Serial.println(digitalRead(led));
        break;
      }
    }


  }

}
```

```
Done uploading.
Sketch uses 2422 bytes (7%) of program storage space. Maximum is 3
Global variables use 188 bytes (9%) of dynamic memory, leaving 186
```

35
Arduino/Genuino Uno on COM3

Figure: Arduino IDE.

Open a new terminal.

Open the Start menu, type "Windows PowerShell", and then select the app.

Type the following command into the terminal and then press press `Enter`:

```
cd C:\Users\%USER%\qt_python_serial
```

This will change the current directory to `C:\Users\%USER%\qt_python_serial`.

Create a new virtual environment named "venv".

Type the following command into the terminal and then press `Enter`:

```
python -m venv venv
```

This will create the `venv` virtual environment in `qt_python_serial`.

Activate the virtual environment.

Type the following command into the terminal and then press `Enter`:

```
.\venv\Scripts\Activate.ps1
```

This will activate the `venv` virtual environment.

Note: To deactivate the `venv` virtual environment, type `deactivate` into the terminal and then press `Enter`.

Install `pyserial`.

Type the following command into the terminal and then press `Enter`:

```
python -m pip install pyserial
```

This will install the latest version of `pyserial` into the `venv` virtual environment.

Install `pyside6`.

Type the following command into the terminal and then press `Enter`:

```
python -m pip install pyside6
```

This will install the latest version of `pyside6` into the `venv` virtual environment.

Update `pip`.

Type the following command into the terminal and then press `Enter`:

```
python -m pip install --upgrade pip
```

This will update `pip` to the latest version.

Qt Creator

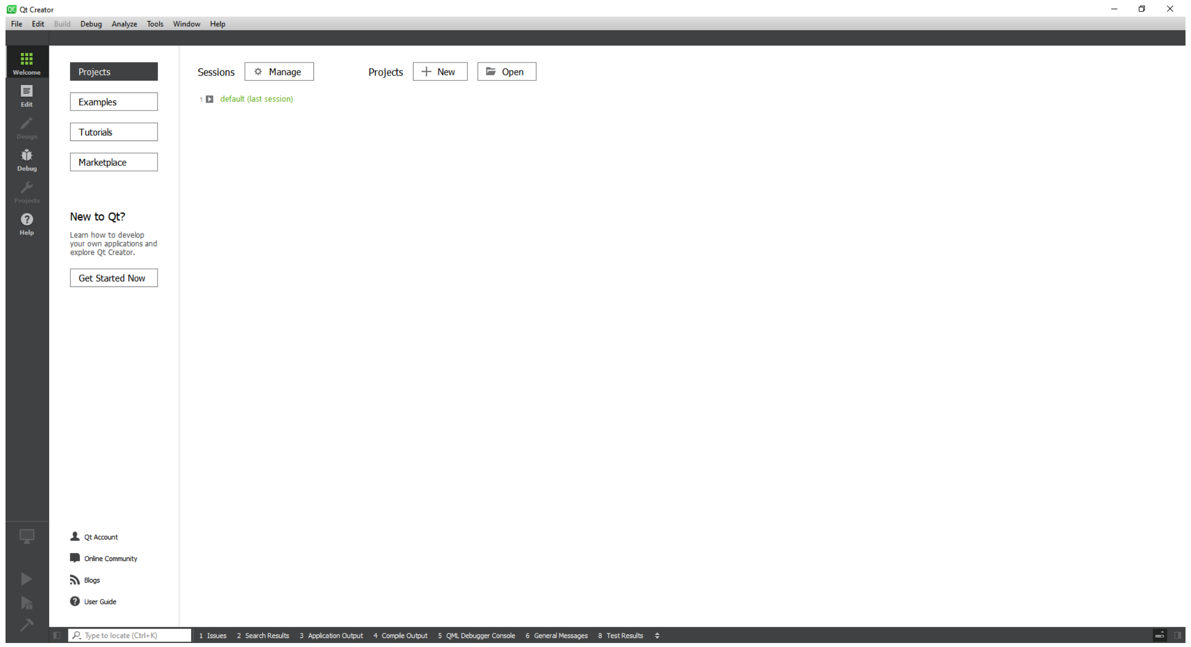
Open Qt Creator. Either:

1. Open the Start menu; type "Qt Creator"; and select the app;
   or,
2. Browse to `C:\Qt\Tools\QtCreator\bin` and double left click on `qtcreator.exe`.

This will open Qt Creator.

Note: `C:\Qt\` is Qt's default installation directory. If you have installed it elsewhere, you will need to browse to that directory instead.

Something similar to the following will be displayed:



Qt Creator
File Edit Build Debug Analyze Tools Window Help
Welcome
Edit
Design
Debug
Projects
Help
Projects
Examples
Tutorials
Marketplace
New to Qt?
Learn how to develop your own applications and explore Qt Creator.
Get Started Now
Sessions
Manage
Projects
New
Open
default (last session)
Qt Account
Online Community
Blogs
User Guide
Type to locate (Ctrl+K)
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

Figure: Qt Creator.

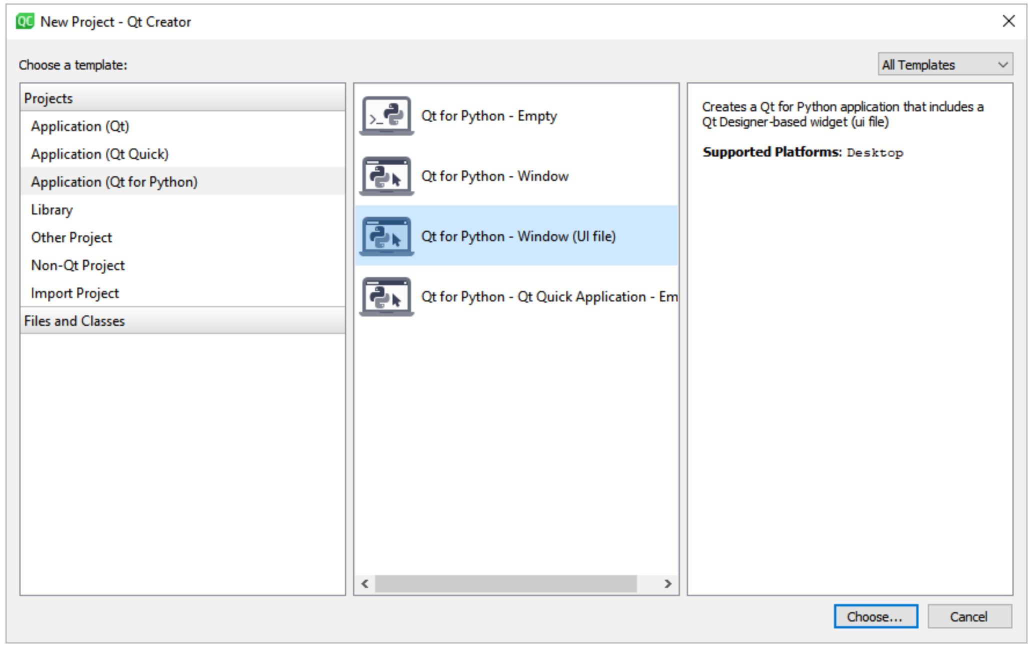
Either:

1. Left click on the "New Project" button;

   or,

2. Select `File > New Project`;

Something similar to the following will be displayed:



Creates a Qt for Python application that includes a Qt Designer-based widget (ui file)

Supported Platforms: `Desktop`

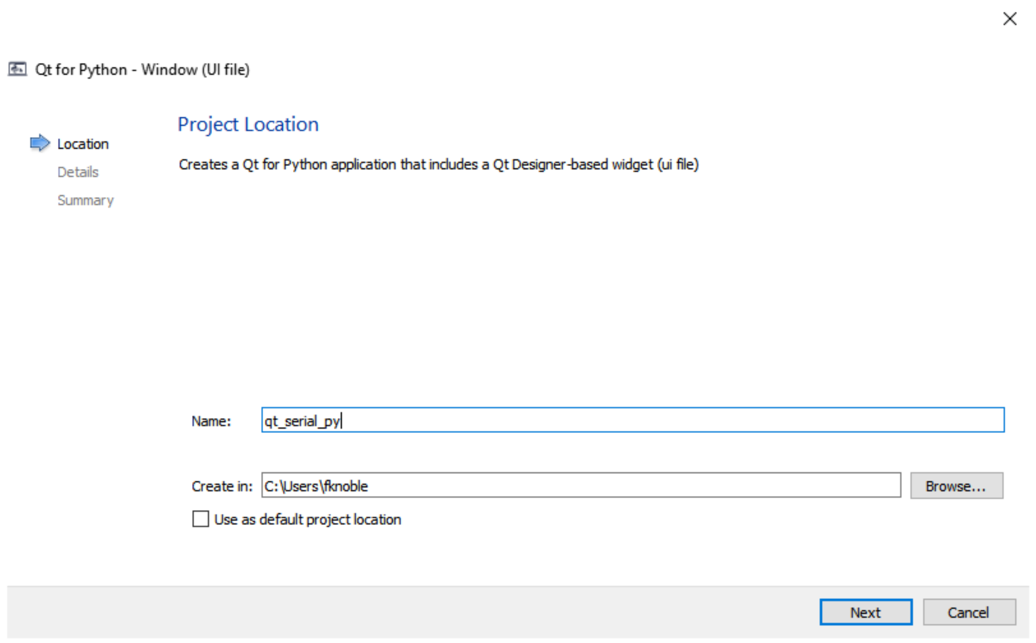
Figure: New project window.

Select "Application (Qt for Python)" from the "Project" menu.

Select "Qt for Python - Window (UI file)".

Left click on the "Choose..." button.

Something simiar to the following will be displayed:


Qt for Python - Window (UI file)

Location
Details
Summary

Project Location

Creates a Qt for Python application that includes a Qt Designer-based widget (ui file)

Name: qt_serial_py

Create in: C:\Users\fknoble

Browse...

☐ Use as default project location

Next Cancel

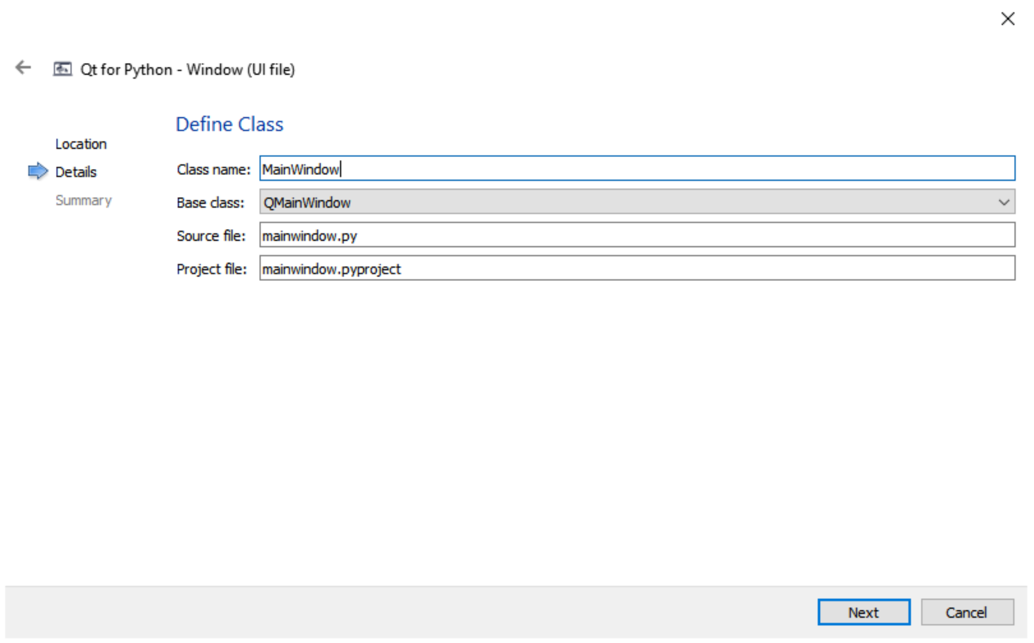
Figure: New project window.

Enter "qt_serial_py" for the project's name.

Save the project in `C:\Users\%USER%\qt_python_serial`.

Left click on the "Next" button.

Something similar to the following will be displayed:


Qt for Python - Window (UI file)

Location
Details
Summary

Define Class

Class name:	MainWindow
Base class:	QMainWindow
Source file:	mainwindow.py
Project file:	mainwindow.pyproject

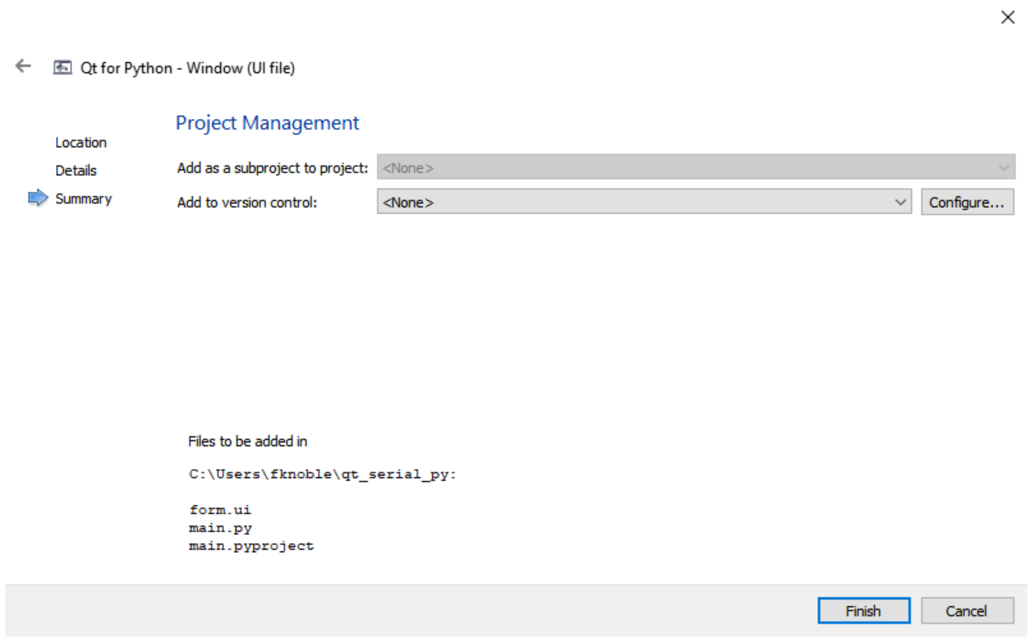
Figure: New project window.

Enter "MainWindow" as the class name.

Select `QMainWindow` for the base class.

Left click on the "Next" button.

Something similar to the following will be displayed:



Add as a subproject to project: <None>

Add to version control: <None>

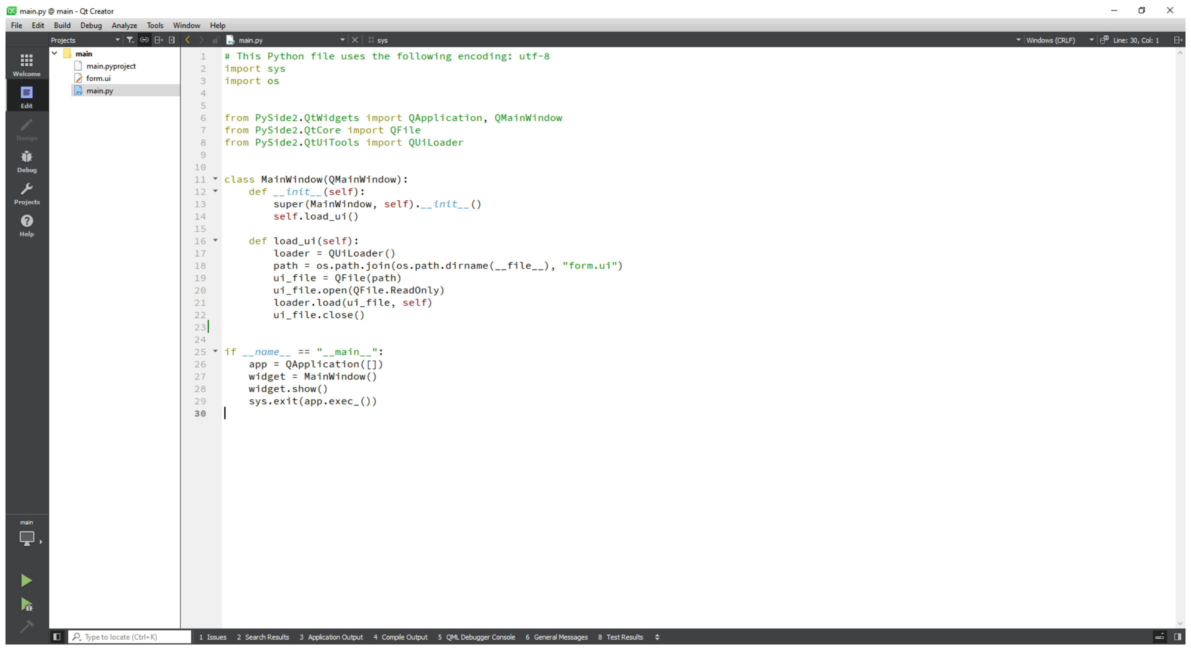
Files to be added in

```
C:\Users\fknoble\qt_serial_py:

form.ui
main.py
main.pyproject
```

Figure: New project window.

Left click on the "Finish" button.

Something similar to the following will be displayed:


main.py @ main - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
main
main.pyproject
form.ui
main.py
Welcome
Edit
Design
Debug
Projects
Help
Windows (CRLF)
Line: 30, Col: 1
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

```python
# This Python file uses the following encoding: utf-8
import sys
import os


from PySide2.QtWidgets import QApplication, QMainWindow
from PySide2.QtCore import QFile
from PySide2.QtUiTools import QUiLoader


class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.load_ui()

    def load_ui(self):
        loader = QUiLoader()
        path = os.path.join(os.path.dirname(__file__), "form.ui")
        ui_file = QFile(path)
        ui_file.open(QFile.ReadOnly)
        loader.load(ui_file, self)
        ui_file.close()


if __name__ == "__main__":
    app = QApplication([])
    widget = MainWindow()
    widget.show()
    sys.exit(app.exec_())
```

Figure: `main.py`.

The project consists of the following directories and files:

```
qt_serial_py/
    .gitignore
    form.ui
    main.py
    main.pyproject
    main.pyproject.user
```

`.gitignore`: A file used by Git to indicate files that should not be included in a repo'.

`form.ui`: A file used by Qt to record the widgets used in the GUI and their properties.

`main.py`: The program's source code.

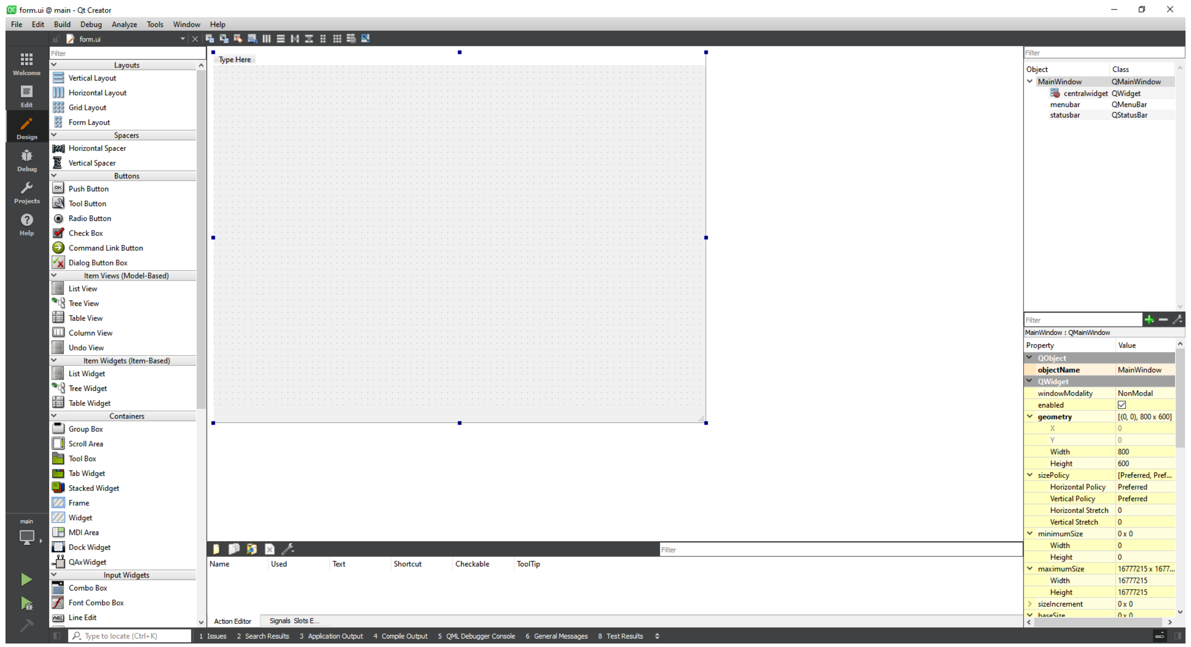
`main.pyproject`: A file used by Qt to record project settings.

`main.pyproject.user`: A file used by Qt to record user-specific project settings.

`form.ui`

Left click on `form.ui` in the "Projects" menu.

Something similar to the following will be displayed:



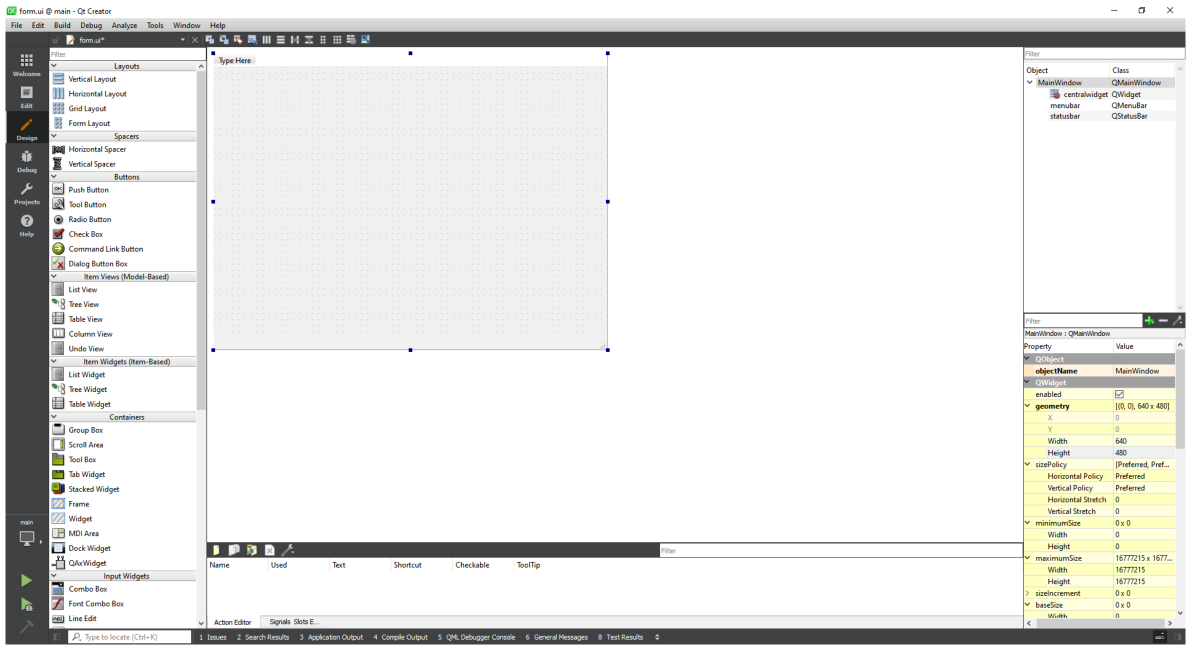
Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
menubar	QMenuBar
statusbar	QStatusBar

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
windowModality	NonModal
enabled	☑
geometry	[(0, 0), 800 x 600]
X	0
Y	0
Width	800
Height	600
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0

Figure: `form.ui`.

Resize `form.ui`'s form to 640 x 480 pixels.


form.ui @ main - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Type Here
Action Editor
Signals Slots E...

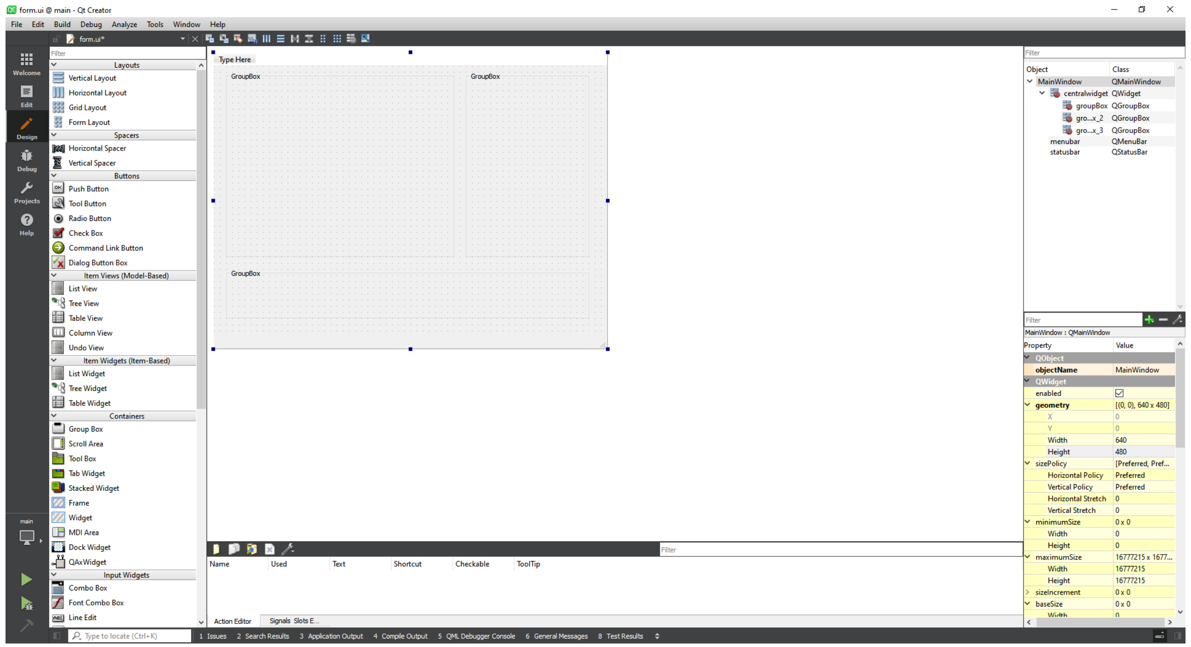
Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
menubar	QMenuBar
statusbar	QStatusBar

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	✓
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui`.

Drag 3 GroupBox widgets onto `form.ui`'s form as illustrated:



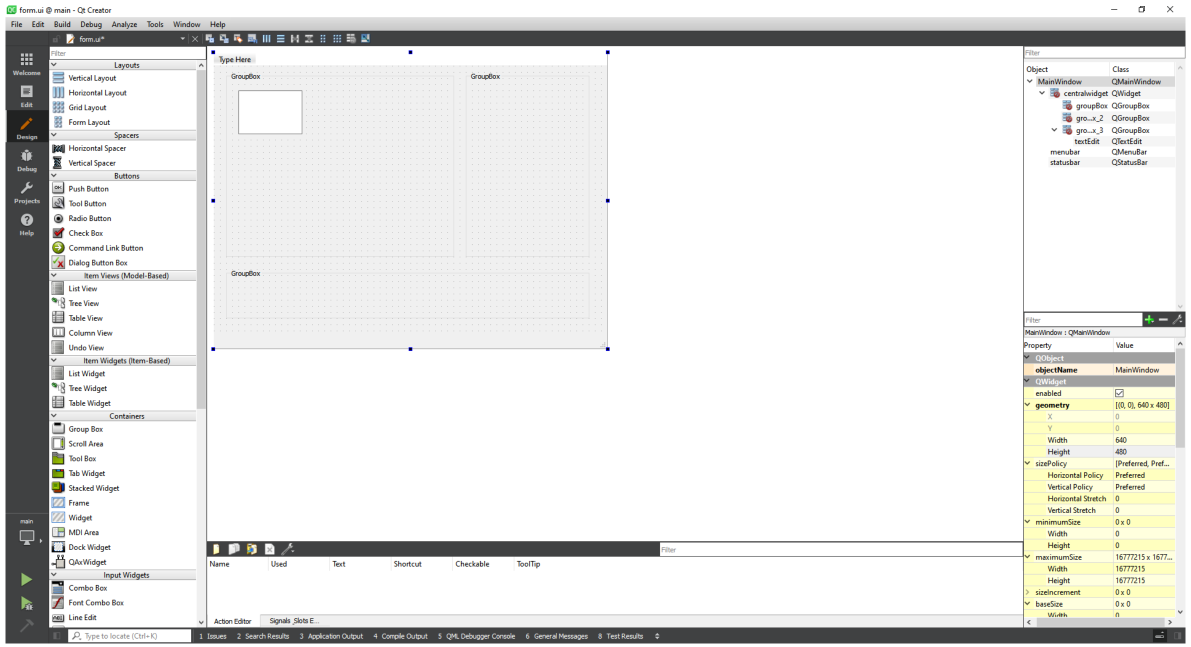
Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
groupBox	QGroupBox
gro...x_2	QGroupBox
gro...x_3	QGroupBox
menubar	QMenuBar
statusbar	QStatusBar

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	☑
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui`.

Drag 1 TextEdit widget onto `form.ui`'s form as illustrated:



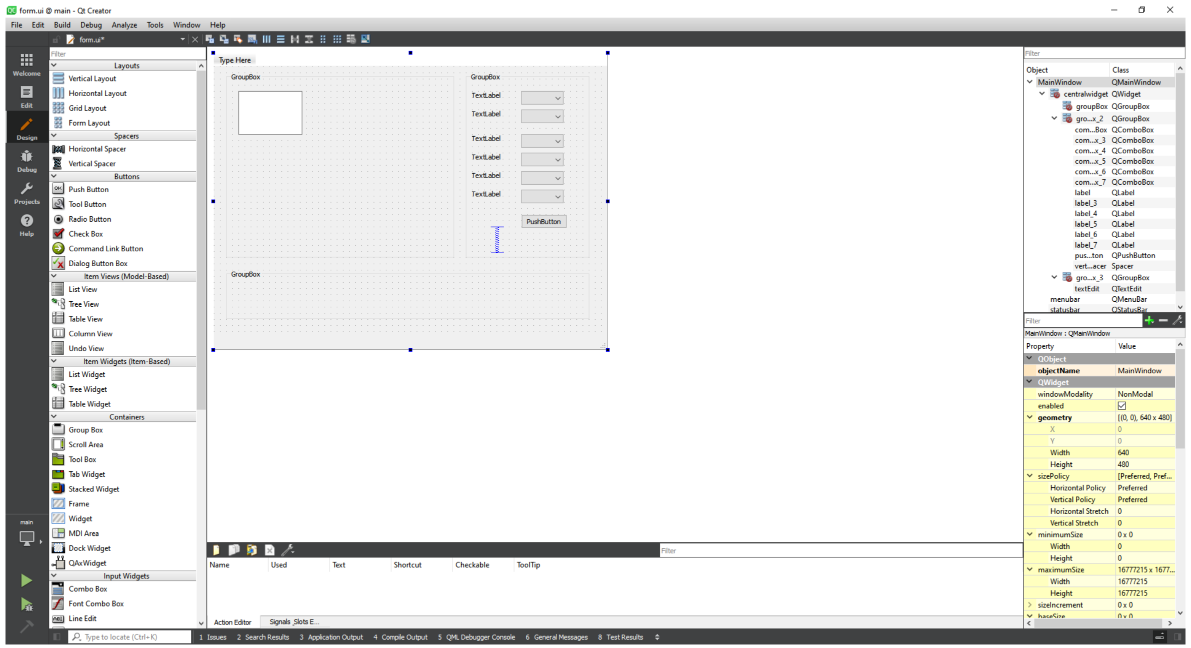
Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
groupBox	QGroupBox
gro...x_2	QGroupBox
gro...x_3	QGroupBox
textEdit	QTextEdit
menubar	QMenuBar
statusbar	QStatusBar

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	☑
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui`.

Drag 6 Label widgets, 6 ComboBox widgets, a PushButton widget, and a VerticalSpacer widget onto `form.ui` 's form as illustrated:



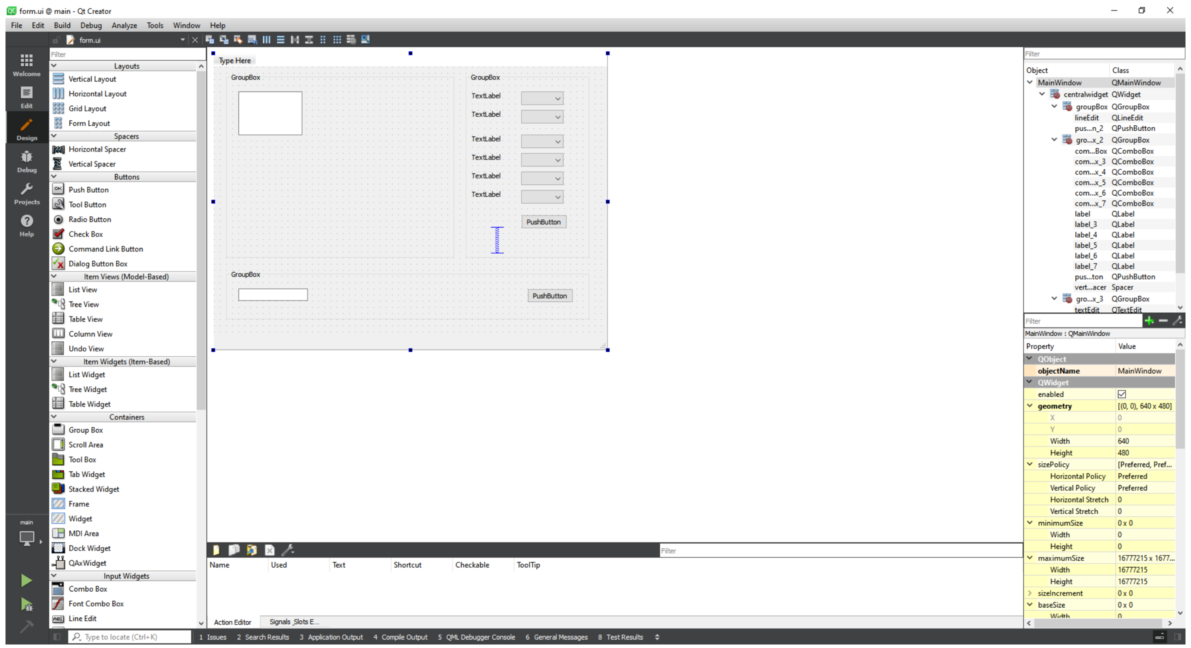
Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
groupBox	QGroupBox
gro...x_2	QGroupBox
com...Box	QComboBox
com...x_3	QComboBox
com...x_4	QComboBox
com...x_5	QComboBox
com...x_6	QComboBox
com...x_7	QComboBox
label	QLabel
label_3	QLabel
label_4	QLabel
label_5	QLabel
label_6	QLabel
label_7	QLabel
pus...ton	QPushButton
vert...acer	Spacer
gro...x_3	QGroupBox
textEdit	QTextEdit
menubar	QMenuBar
statusbar	QStatusBar

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
windowModality	NonModal
enabled	✓
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0

Figure: `form.ui` .

Drag 1 LineEdit widget, and 1 PushButton widget onto `form.ui` 's form as illustrated:



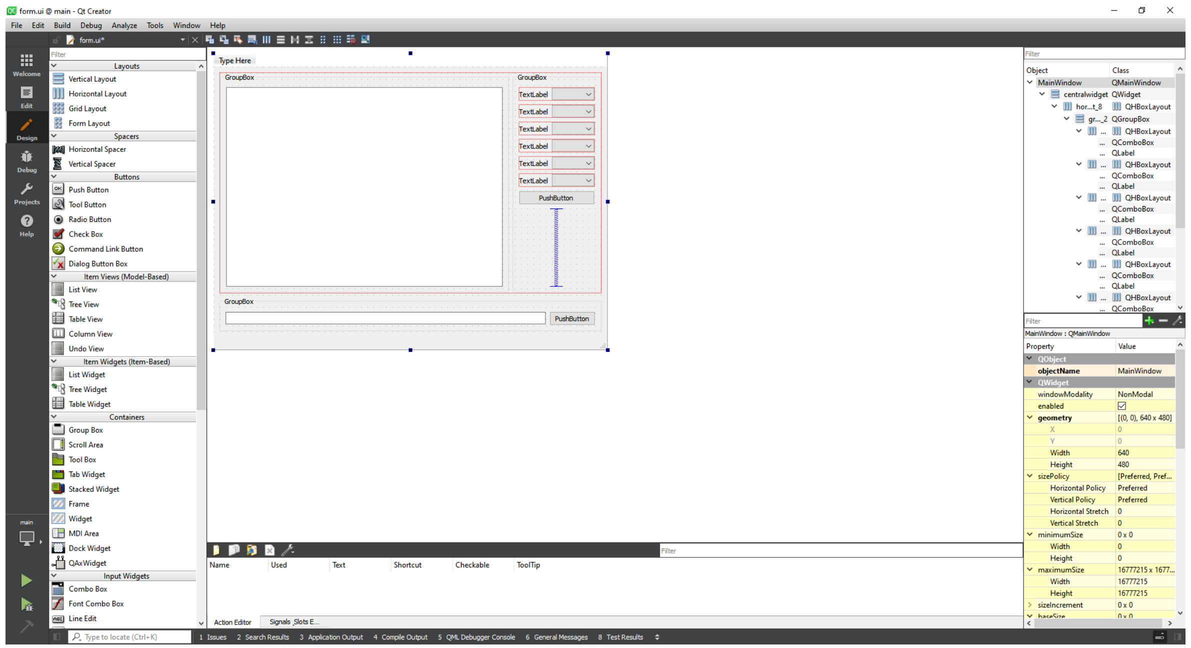
Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
groupBox	QGroupBox
lineEdit	QLineEdit
pus...n_2	QPushButton
gro...x_2	QGroupBox
com...Box	QComboBox
com...x_3	QComboBox
com...x_4	QComboBox
com...x_5	QComboBox
com...x_6	QComboBox
com...x_7	QComboBox
label	QLabel
label_3	QLabel
label_4	QLabel
label_5	QLabel
label_6	QLabel
label_7	QLabel
pus...ton	QPushButton
vert...acer	Spacer
gro...x_3	QGroupBox
textEdit	QTextEdit

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	✓
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui` .

Layout the widgets as illustrated:



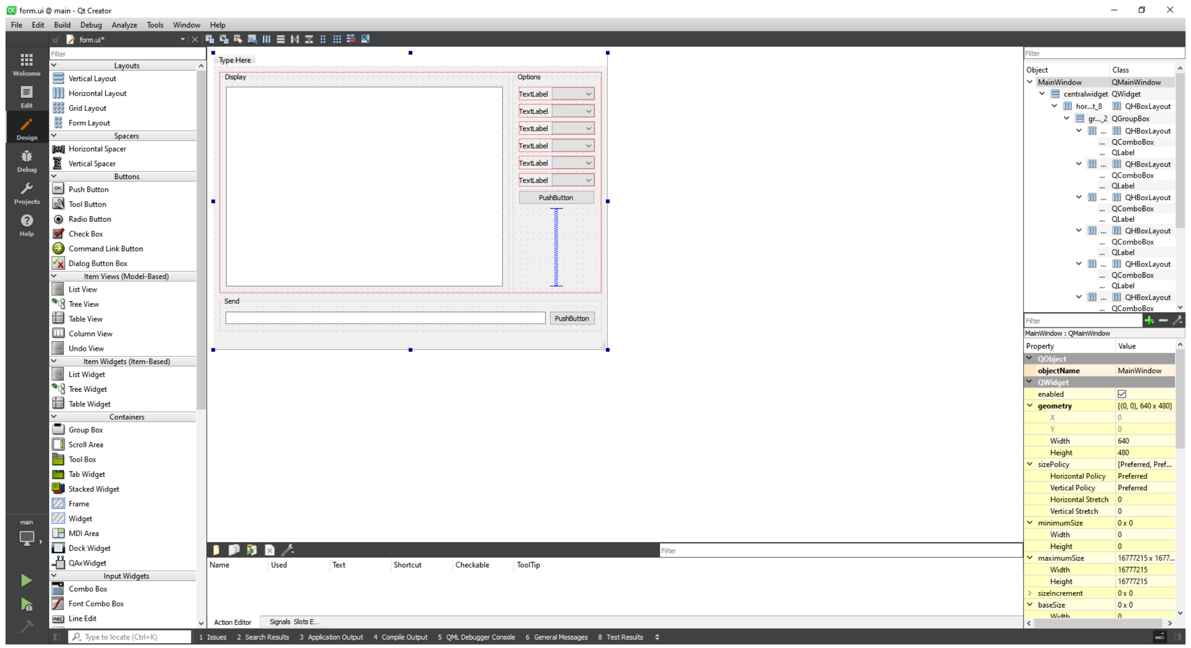
```
Object                  Class
MainWindow              QMainWindow
  centralwidget         QWidget
    hor...t_8           QHBoxLayout
      gr..._2           QGroupBox
        ...             QHBoxLayout
          ...           QComboBox
          ...           QLabel
        ...             QHBoxLayout
          ...           QComboBox
          ...           QLabel
        ...             QHBoxLayout
          ...           QComboBox
          ...           QLabel
        ...             QHBoxLayout
          ...           QComboBox
          ...           QLabel
        ...             QHBoxLayout
          ...           QComboBox
          ...           QLabel
        ...             QHBoxLayout
          ...           QComboBox
```

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
windowModality	NonModal
enabled	✓
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0

Figure: `form.ui`.

Change the GroupBox widgets' text as illustrated:



Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
hor...t_8	QHBoxLayout
gr..._2	QGroupBox
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox

MainWindow : QMainWindow

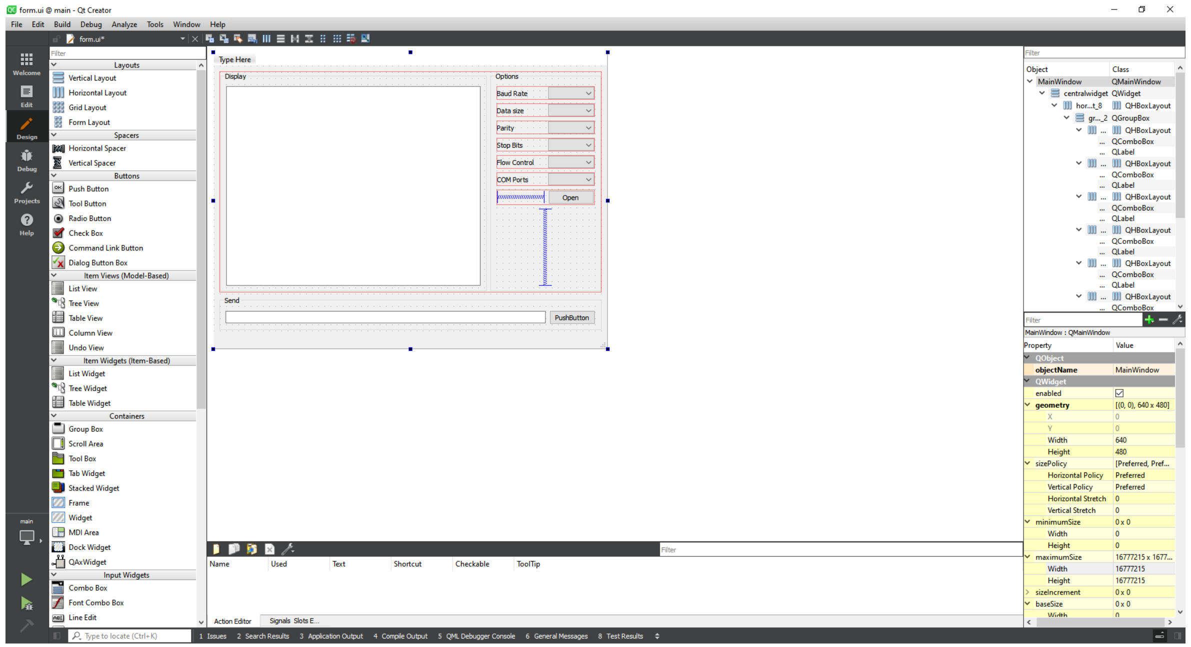
Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	✓
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui` .

Set the "Options" GroupBox's width to 180 px.

Change the TextEdit widget's name to "textEditDisplay".

Change the Label, ComboBox, and PushButton widgets' text as illustrated:



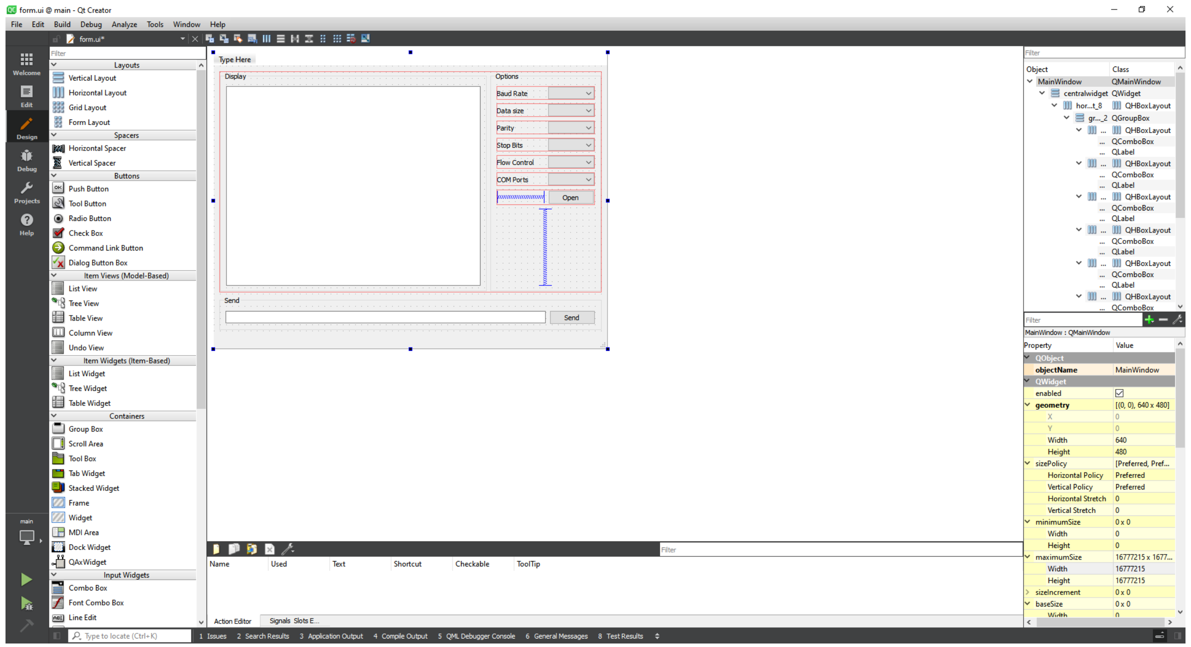
Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	✓
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui` .

Set the ComboBox widgets' width to 75 px. Change the objects' names to "comboBoxBaudRate", "comboBoxDataSize", ..., "comboBoxFlowControl".

Set the "Open" PushButton's width to 75 px. Change the object's name to "pushButtonOpen".

Change LineEdit and PushButton widgets' text as illustrated.



Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
hor...t_8	QHBoxLayout
gr..._2	QGroupBox
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox
...	QLabel
...	QHBoxLayout
...	QComboBox

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	☑
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Pref...
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 1677...
Width	16777215
Height	16777215
sizeIncrement	0 x 0
baseSize	0 x 0
Width	0

Figure: `form.ui`.

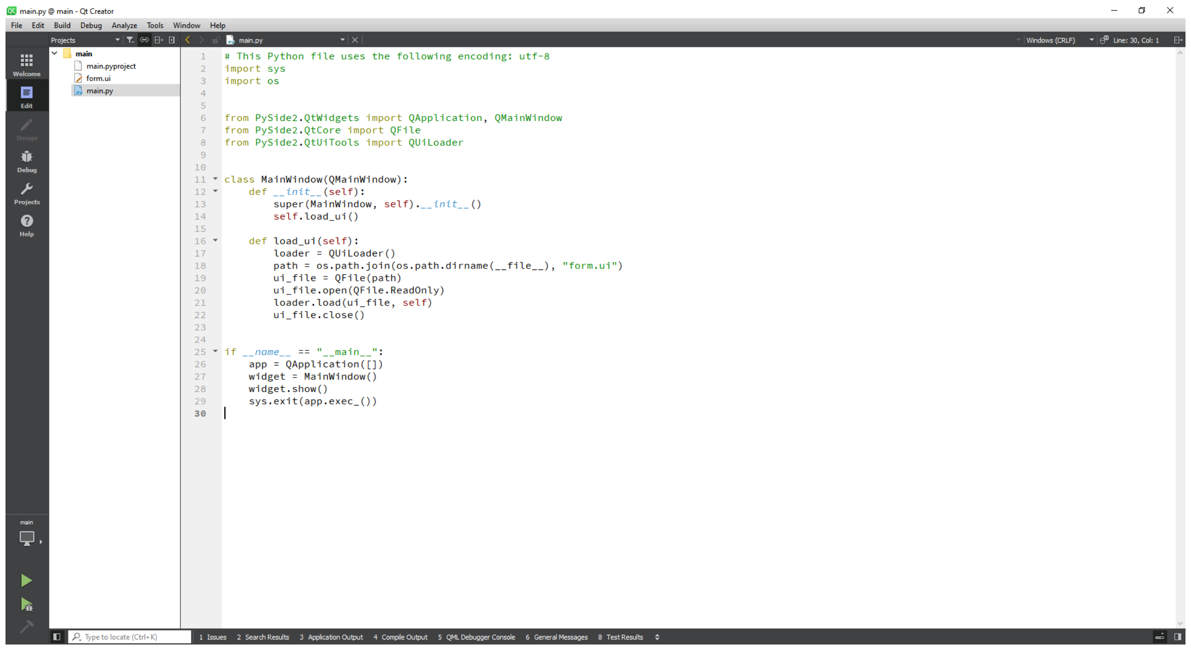
Set the "Send" PushButton's width to 75 px. Change the object's name to "pushButtonSend".

Change the LineEdit's name to "lineEditSend".

`main.py`

Left click on `main.py` in the "Projects" menu.

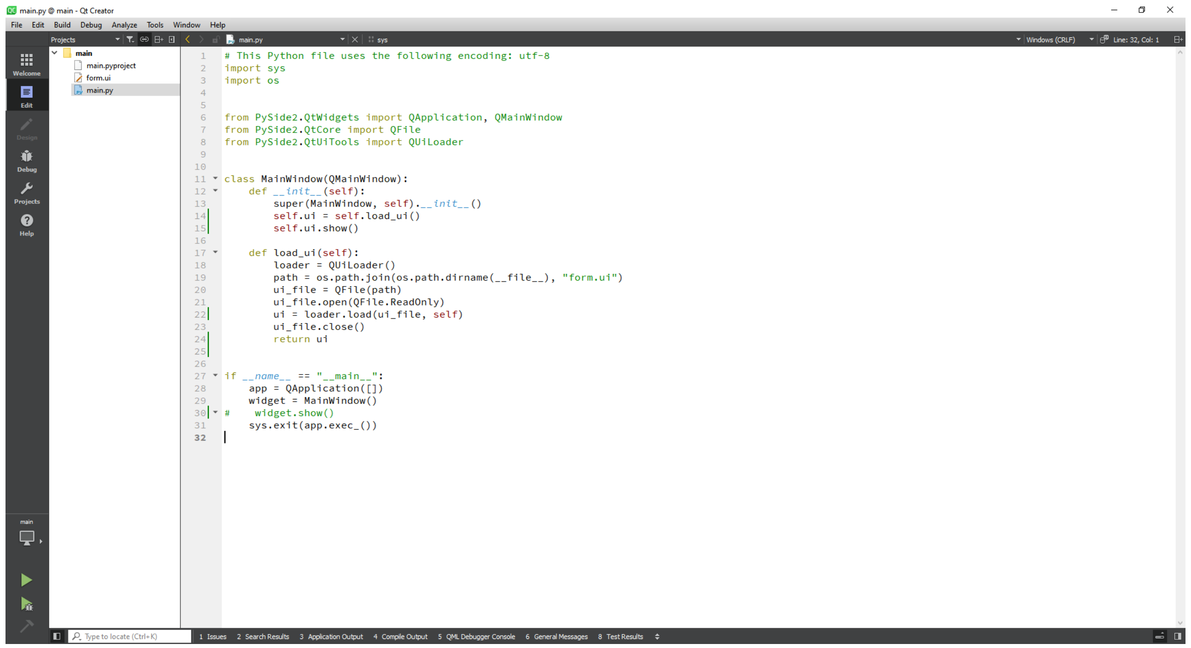
Something similar to the following will be displayed:



```python
# This Python file uses the following encoding: utf-8
import sys
import os


from PySide2.QtWidgets import QApplication, QMainWindow
from PySide2.QtCore import QFile
from PySide2.QtUiTools import QUiLoader


class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.load_ui()

    def load_ui(self):
        loader = QUiLoader()
        path = os.path.join(os.path.dirname(__file__), "form.ui")
        ui_file = QFile(path)
        ui_file.open(QFile.ReadOnly)
        loader.load(ui_file, self)
        ui_file.close()


if __name__ == "__main__":
    app = QApplication([])
    widget = MainWindow()
    widget.show()
    sys.exit(app.exec_())
```

Figure: `main.py`.

Edit `main.py` as illustrated:



```python
# This Python file uses the following encoding: utf-8
import sys
import os


from PySide2.QtWidgets import QApplication, QMainWindow
from PySide2.QtCore import QFile
from PySide2.QtUiTools import QUiLoader


class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.ui = self.load_ui()
        self.ui.show()

    def load_ui(self):
        loader = QUiLoader()
        path = os.path.join(os.path.dirname(__file__), "form.ui")
        ui_file = QFile(path)
        ui_file.open(QFile.ReadOnly)
        ui = loader.load(ui_file, self)
        ui_file.close()
        return ui


if __name__ == "__main__":
    app = QApplication([])
    widget = MainWindow()
#    widget.show()
    sys.exit(app.exec_())

```

Figure: `main.py`.

Change references to `PySide2` to `PySide6`.

Either:

1. Left click on the "Run" button;

   or,

2. Select "Build > Run".

Something similar to the following will be displayed:

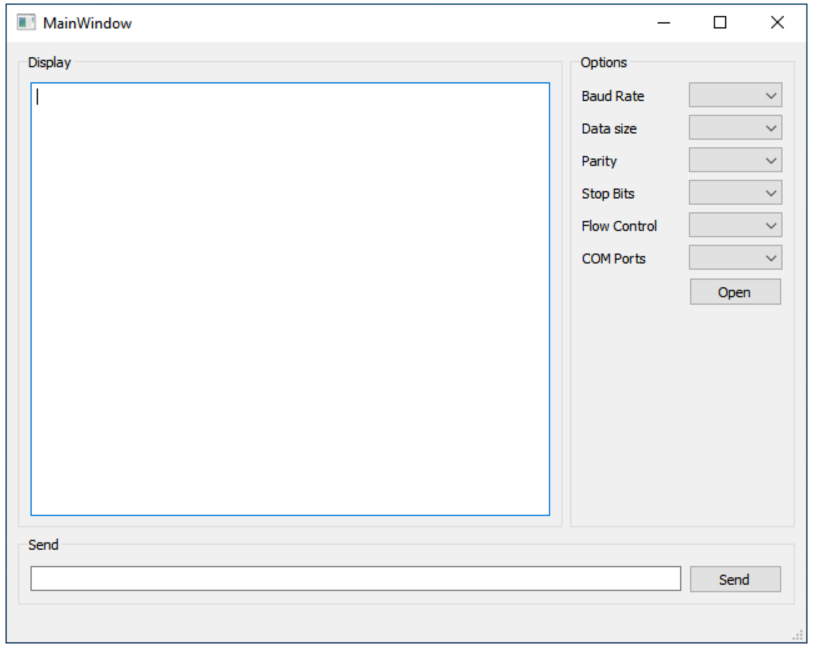


Figure: `main.py` 's GUI.

Left click on the "X" button to close the GUI.

Import the following additional modules from `PySide6.QtCore`:

- `Signal`, and
- `Slot`.

Add the following Python code to `main.py`:

```
import serial
import serial.tools.list_ports
```

This will import the `serial` and `serial.tools.list_ports` modules.

Add the following Python code to `MainWindow`'s `__init__()` member function:

```
self.ser = serial.Serial()

self.baudRate = "9600"
self.dataSize = "8"
self.parity = "N"
self.stopBits = "1"
self.flowControl = "False"
self.COMPort = "COM3"
self.timeout = 0.5
```

This will create an instance of the `Serial` class and data members, which are assigned common default values.

Add the following Python code to `MainWindow`'s `__init__()` member function:

```
[self.ui.comboBoxBaudRate.addItem(str(i)) for i in self.ser.BAUDRATES]
self.ui.comboBoxBaudRate.setCurrentText(self.baudRate)
[self.ui.comboBoxDataSize.addItem(str(i)) for i in self.ser.BYTESIZES]
self.ui.comboBoxDataSize.setCurrentText(self.dataSize)
[self.ui.comboBoxParity.addItem(str(i)) for i in self.ser.PARITIES]
self.ui.comboBoxParity.setCurrentText(self.parity)
[self.ui.comboBoxStopBits.addItem(str(i)) for i in self.ser.STOPBITS]
self.ui.comboBoxStopBits.setCurrentText(self.stopBits)
[self.ui.comboBoxFlowControl.addItem(str(i)) for i in [True, False]]
self.ui.comboBoxFlowControl.setCurrentText(self.flowControl)
[self.ui.comboBoxCOMPorts.addItem(str(i)) for i in serial.tools.list_ports.comports()]
self.ui.comboBoxCOMPorts.setCurrentIndex(0)
```

This will populate all the ComboBox widgets with items defined in the `Serial` class's corresponding `enum`s and set the current item to the corresponding data member.

Add the following Python code to `MainWindow`'s `__init__()` member function:

```
self.ui.comboBoxBaudRate.currentTextChanged.connect(self.baudRateChanged)
self.ui.comboBoxDataSize.currentTextChanged.connect(self.dataSizeChanged)
self.ui.comboBoxStopBits.currentTextChanged.connect(self.stopBitsChanged)
self.ui.comboBoxParity.currentTextChanged.connect(self.parityChanged)
self.ui.comboBoxFlowControl.currentTextChanged.connect(self.flowControlChanged)
self.ui.comboBoxCOMPorts.currentTextChanged.connect(self.comPortsChanged)
self.ui.pushButtonOpenPort.clicked.connect(self.open)
self.ui.pushButtonSend.clicked.connect(self.send)
self.ui.lineEditSend.returnPressed.connect(self.send)

self.ui.actionQuit.triggered.connect(self.quit)
```

This will connect each ComboBox widgets' `currentTextChanged` signal to the corresponding slot.

Add the following Python code to `main.py`'s `MainWindow` class:

```
@Slot()
def baudRateChanged(self, text):
    self.baudRate = text
    return

@Slot()
def dataSizeChanged(self, text):
    self.dataSize = text
    return

@Slot()
def stopBitsChanged(self, text):
    self.stopBits = text
    return
```

This will define slots `baudRateChanged()`, `dataSizeChanged()`, and `stopBitsChanged()`, which will assign each ComboBox widget's text to the corresponding data members when signals connected to them are emitted.

Add the following Python code to `main.py` 's `MainWindow` class:

```python
@Slot()
def parityChanged(self, text):
    self.parity = text
    return

@Slot()
def flowControlChanged(self, text):
    self.flowControl = text
    return

@Slot()
def comPortsChanged(self, text):
    self.COMPorts = text
    return
```

This will define slots `parityChanged()` , `flowControlChanged()` , and `comPortsChanged()` , which will assign each ComboBox widget's text to the corresponding data members when signals connected to them are emitted.

Add the following Python code to `main.py`'s `MainWindow` class:

```python
@Slot()
def open(self):
    if not self.ser.is_open:
        self.ser.baudrate = int(self.baudRate)
        self.ser.bytesize = int(self.dataSize)
        self.ser.parity = self.parity
        self.ser.stopbits = int(self.stopBits)
        if (self.flowControl == "True"):
            self.ser.set_input_flow_control()
            self.ser.set_output_flow_control()
        self.ser.port = self.COMPort
        self.ser.timeout = self.timeout
        self.ser.open()
        self.ui.pushButtonOpenPort.setText("Close")
    else:
        self.ser.close()
        self.ui.pushButtonOpenPort.setText("Open")
    return
```

This will define a slot `open()`, which will open the serial port when a signal connected to it is emitted.

Add the following Python code to `main.py`'s `MainWindow` class:

```python
@Slot()
def send(self):
    if self.ser.is_open:
        data = "{}".format(self.ui.lineEditSend.text())
        bytesWritten = self.ser.write(data.encode("utf-8"))
        self.ui.textEditDisplay.append(data)

        data = self.ser.readline()
        self.ui.textEditDisplay.append(data.decode("utf-8"))
    return
```

This will define a slot `send()`, which will write `lineEditSend`'s text to the serial port when a signal connected to it is emitted.

Add the following Python code to `main.py` 's `MainWindow` class:

```
@Slot()
def quit(self):
    QApplication.quit()
    return
```

This will define a slot named `quit()`, which will close the application when a signal connected to it is emitted.

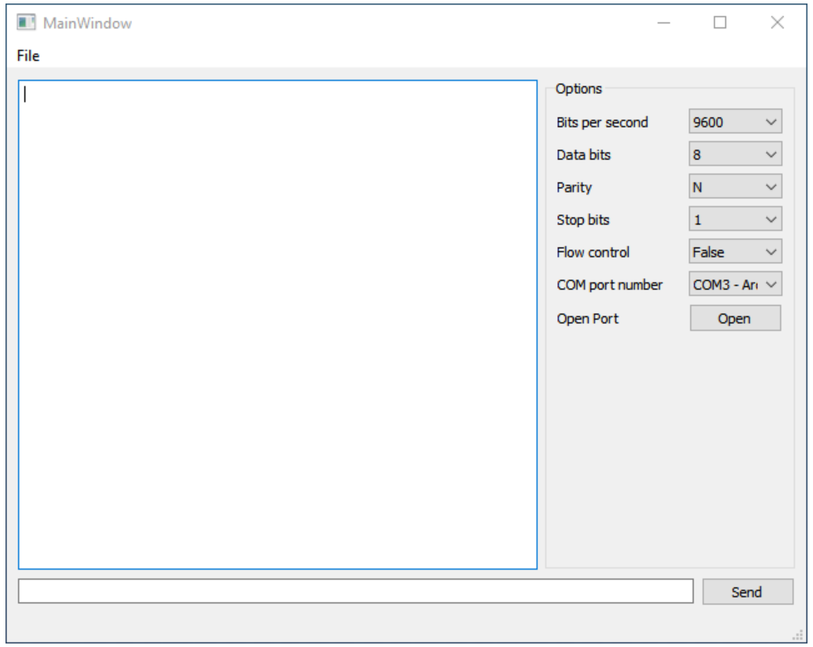
Either:

1. Left click on the "Run" button;

   or,

2. Select "Build > Run".

Something similar to the following will be displayed:



Bits per second 9600
Data bits 8
Parity N
Stop bits 1
Flow control False
COM port number COM3 - Ar

Figure: `main.py` 's GUI.

Left click on the "X" button or select "File > Quit" to close the GUI.

Testing

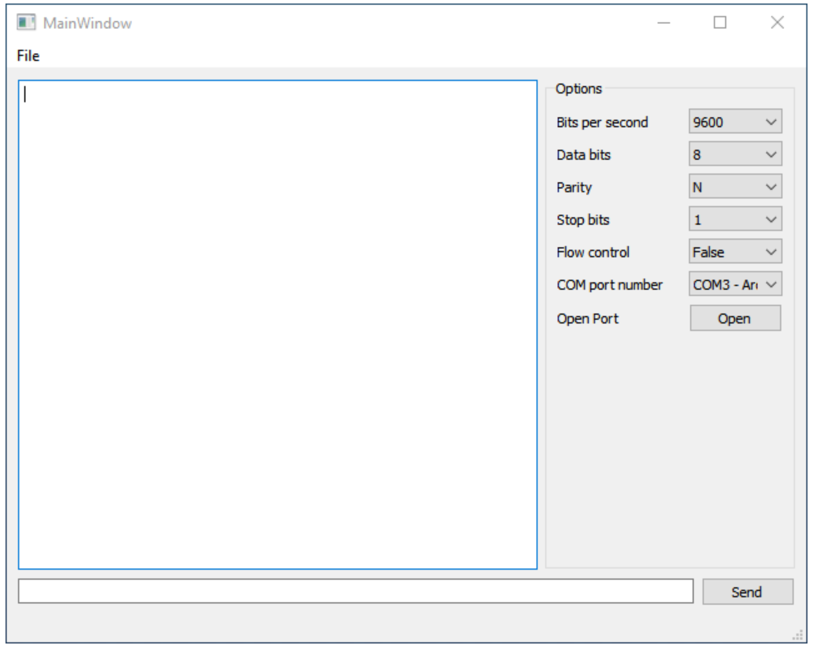
Either:

1. Left click on the "Run" button;

   or,

2. Select "Build > Run".

Something similar to the following will be displayed:



Bits per second 9600
Data bits 8
Parity N
Stop bits 1
Flow control False
COM port number COM3 - Ar
Open Port

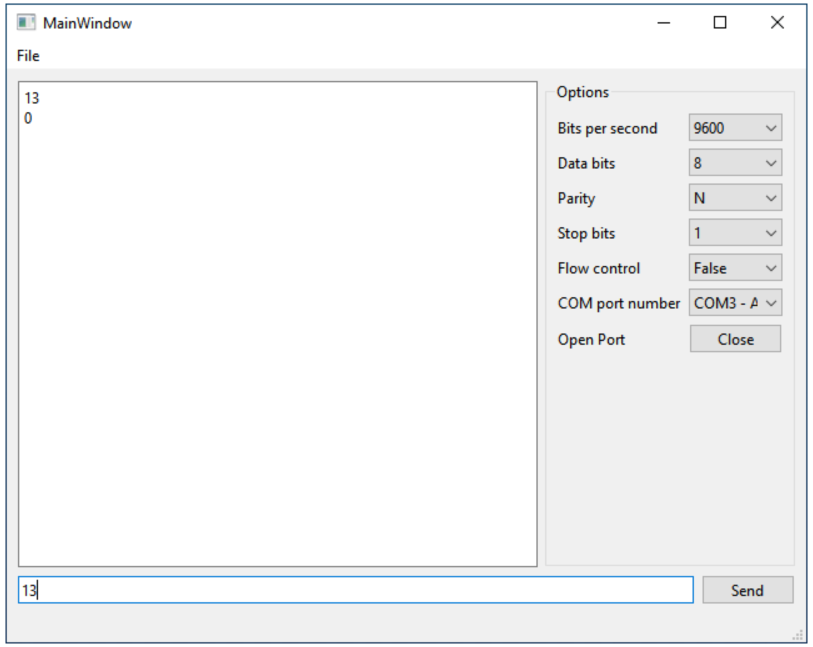
Figure: (Left) `main.py` 's GUI; and (Right) the Arduino plugged into the computer. Here, we can see that the LED connected to Pin 13 is ON.

Ensure that the default port settings (9600, 8, N, 1, False, COM3) are selected.

Left click on `main.py`'s "Open" button.

Type `13` into `main.py`'s input and then left click on `main.py`'s "Send" button.

Something similar to the following will be displayed:



```
MainWindow
File

13
0

Options
Bits per second   9600
Data bits         8
Parity            N
Stop bits         1
Flow control      False
COM port number   COM3 - A
Open Port         Close

13                Send
```

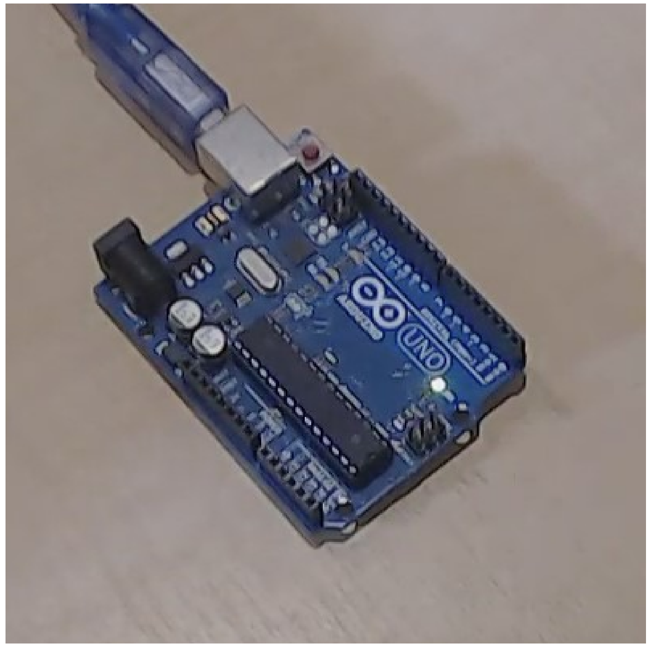


Figure: (Left) `main.py`'s GUI; and (Right) the Arduino plugged into the computer. Here, we can see that the LED connected to Pin 13 is OFF.

The LED has been turned off via a COM port using the GUI!

Conclusion

In this presentation, I have described:

- How to control an Arduino's pin via a COM port using Qt.

References

1. https://www.arduino.cc/
2. https://doc.qt.io/qtforpython/index.html.
3. https://pythonhosted.org/pyserial/index.html