```
#! /usr/bin/env python3

import sys

from PyQt5.QtWidgets import QApplication, QWidget

app = None

def main():

    global app
    app = QApplication(sys.argv)

    widget = QWidget()
    widget.setWindowTitle("Hello, World!")
    widget.show()

    print("Qt is now handling everything!")
    app.exec_()
    print("Qt is done.")

if __name__ == "__main__":
    main()
```

```
#! /usr/bin/env python3

import sys

from PyQt5.QtCore import Qt, QTimer, pyqtSignal
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QHBoxLayout, QPushButton

class ControlWidget(QWidget):
    """The ContralWidget has Start and Stop buttons to control the generation of signals."""

    # The valueUpdated signal is emitted periodically if the timer is active.
    valueUpdated = pyqtSignal(int)

    def __init__(self, *args, **kwargs):

        super().__init__(*args, **kwargs)

        self.setWindowTitle("Control Widget")
        self.setMinimumSize(400, 100)
        self.counter = 0
        self.timer = QTimer()

        layout = QHBoxLayout()

        self.start_button = QPushButton("Start")
        self.stop_button = QPushButton("Stop")
        self.stop_button.setEnabled(False)

        layout.addWidget(self.start_button)
        layout.addWidget(self.stop_button)

        self.setLayout(layout)

        self.start_button.clicked.connect(self.startButtonClicked)
        self.stop_button.clicked.connect(self.stopButtonClicked)
        self.timer.timeout.connect(self.timerTimeout)

    def startButtonClicked(self):
        print("start!")
        self.start_button.setEnabled(False)
        self.stop_button.setEnabled(True)
        self.timer.start(100)  # 10 Hz.

    def stopButtonClicked(self):
        print("stop!")
        self.start_button.setEnabled(True)
        self.stop_button.setEnabled(False)
        self.timer.stop()

    def timerTimeout(self):
        print("timer!")
        self.counter += 1
        self.valueUpdated.emit(self.counter)

class ViewWidget(QLabel):
    """The ViewWidget shows a number it receives as a signal in large letters."""

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.setWindowTitle("View Widget")
        self.setAlignment(Qt.AlignCenter)
        self.setText("(no value received yet)")
        font = self.font()
        font.setPointSize(48)
        self.setFont(font)

    def setValue(self, value):
        print("setValue!")
        self.setText("{}".format(value))

app = None

def main():

    global app
    app = QApplication(sys.argv)

    control_widget = ControlWidget()

    view_widget1 = ViewWidget()
    view_widget2 = ViewWidget()
    view_widget3 = ViewWidget()

    control_widget.valueUpdated.connect(view_widget1.setValue)
    control_widget.valueUpdated.connect(view_widget2.setValue)
    control_widget.valueUpdated.connect(view_widget3.setValue)

    control_widget.show()
    view_widget1.show()
    view_widget2.show()
    view_widget3.show()

    app.exec_()

if __name__ == "__main__":
    main()
```