

PyQt GUI for Temperature & Humidity Sensor

ECEA5347: Rapid Prototyping of Embedded Interface Designs

Glenn Frey Olamit

November 2, 2021

1. Implementation & Assumption Notes

I use Python, PyQt5 and Qt to implement this project as recommended in this course. I used the following commands to install Qt, PyQt5, and the necessary modules: `sudo apt-get install qt5-default pyqt5-dev pyqt5-dev-tools, sudo apt-get install qttools5-dev-tools, sudo apt-get install qtsql.`

I used Raspberry Pi model 3b+ for this project and installed the latest Raspbian OS for this embedded project as it fit the requirement. I installed Geany, a lightweight GUI text editor as resources must be conserved and MobaXterm to work remotely.

I used Sqlite3 for the following benefit: 1. SQLite has a relatively low overhead. 2. It is a self-contained system. No external dependencies are required to make it function. 3. No separate server process. SQLite won't chew up your Raspberry Pi's RAM and CPU when not being utilized. 4. Zero configuration is needed making it easy to use right out of the box.

I assume the values extracted from the psudosensor in Celsius and the Values I choose to display are in Celsius.

I assume in project requirement number three the temperature and humidity reading is also recorded to the database. I used one button to read both data and display the UI.

In project requirement five I used a table to display the ten values being recorded. A text to display the number of recordings and a progress bar to indicate the status and completion.

In project requirement seven I used QLabel to display alarm and status of temperature and humidity.

I have not done the optional task as I have a problem installing PyQtcharts. I guess it has something to do with the version incompatibility. Nevertheless, I have finished all the required tasks.

2. Code

2.1 project.ui

```
breakline  
-----  
<?xml version="1.0" encoding="UTF-8"?>  
<ui version="4.0">  
<class>Form</class>  
<widget class="QWidget" name="Form">  
<property name="geometry">  
<rect>  
<x>0</x>  
<y>0</y>  
<width>782</width>  
<height>534</height>  
</rect>  
</property>  
<property name="windowTitle">  
<string>Form</string>  
</property>  
<widget class="QPushButton" name="button">  
<property name="geometry">  
<rect>  
<x>80</x>  
<y>260</y>  
<width>111</width>  
<height>31</height>  
</rect>  
</property>  
<property name="text">  
<string>Read Data</string>  
</property>  
</widget>  
<widget class="QLabel" name="label">  
<property name="geometry">  
<rect>  
<x>70</x>  
<y>150</y>  
<width>121</width>  
<height>21</height>  
</rect>  
</property>  
<property name="font">
```

```
<font>
<pointsize>14</pointsize>
</font>
</property>
<property name="text">
<string>Temperature</string>
</property>
</widget>
<widget class="QLabel" name="label_2">
<property name="geometry">
<rect>
<x>90</x>
<y>210</y>
<width>91</width>
<height>21</height>
</rect>
</property>
<property name="font">
<font>
<pointsize>14</pointsize>
</font>
</property>
<property name="text">
<string>Humidity</string>
</property>
</widget>
<widget class="QTextEdit" name="textEdittemp">
<property name="geometry">
<rect>
<x>30</x>
<y>180</y>
<width>211</width>
<height>21</height>
</rect>
</property>
</widget>
<widget class="QTextEdit" name="textEdithumidity">
<property name="geometry">
<rect>
<x>30</x>
<y>230</y>
<width>211</width>
<height>21</height>
</rect>
```

```
</property>
</widget>
<widget class="QPushButton" name="pushButton">
<property name="geometry">
<rect>
<x>80</x>
<y>492</y>
<width>101</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Read 10 Values</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_avetemp">
<property name="geometry">
<rect>
<x>200</x>
<y>20</y>
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Average Temperature</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_avehumidity">
<property name="geometry">
<rect>
<x>560</x>
<y>20</y>
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Average Humidity</string>
</property>
</widget>
<widget class="QTextEdit" name="textEdit_avetemp">
<property name="geometry">
<rect>
```

```
<x>40</x>
<y>20</y>
<width>151</width>
<height>31</height>
</rect>
</property>
</widget>
<widget class="QTextEdit" name="textEdit_avehumidity">
<property name="geometry">
<rect>
<x>400</x>
<y>20</y>
<width>151</width>
<height>31</height>
</rect>
</property>
</widget>
<widget class="QTextEdit" name="textEdit_mintemp">
<property name="geometry">
<rect>
<x>40</x>
<y>60</y>
<width>151</width>
<height>31</height>
</rect>
</property>
</widget>
<widget class="QTextEdit" name="textEdit_maxtemp">
<property name="geometry">
<rect>
<x>40</x>
<y>100</y>
<width>151</width>
<height>31</height>
</rect>
</property>
</widget>
<widget class="QTextEdit" name="textEdit_minhumidity">
<property name="geometry">
<rect>
<x>400</x>
<y>60</y>
<width>151</width>
<height>31</height>
```

```
</rect>
</property>
</widget>
<widget class="QTextEdit" name="textEdit_maxhumidity">
<property name="geometry">
<rect>
<x>400</x>
<y>100</y>
<width>151</width>
<height>31</height>
</rect>
</property>
</widget>
<widget class="QPushButton" name="pushButton_mintemp">
<property name="geometry">
<rect>
<x>200</x>
<y>60</y>
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Min Temperature</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_maxtemp">
<property name="geometry">
<rect>
<x>200</x>
<y>100</y>
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Max Temperature</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_minhumidity">
<property name="geometry">
<rect>
<x>560</x>
<y>60</y>
```

```
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Min Humidity</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_maxhumidity">
<property name="geometry">
<rect>
<x>560</x>
<y>100</y>
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Max Humidity</string>
</property>
</widget>
<widget class="QLabel" name="label_Result">
<property name="enabled">
<bool>true</bool>
</property>
<property name="geometry">
<rect>
<x>10</x>
<y>400</y>
<width>241</width>
<height>31</height>
</rect>
</property>
<property name="font">
<font>
<pointsize>14</pointsize>
</font>
</property>
<property name="text">
<string>TextLabel</string>
</property>
</widget>
<widget class="QProgressBar" name="progressBar">
<property name="enabled">
```



```
<widget class="QLabel" name="label_temp">
<property name="geometry">
<rect>
<x>10</x>
<y>300</y>
<width>251</width>
<height>41</height>
</rect>
</property>
<property name="text">
<string>TextLabel</string>
</property>
</widget>
<widget class="QLabel" name="label_humidity">
<property name="geometry">
<rect>
<x>10</x>
<y>350</y>
<width>251</width>
<height>41</height>
</rect>
</property>
<property name="text">
<string>TextLabel</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_Display">
<property name="geometry">
<rect>
<x>440</x>
<y>490</y>
<width>141</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Display last 10 Values</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_Exit">
<property name="geometry">
<rect>
<x>660</x>
<y>490</y>
```

```
<width>80</width>
<height>31</height>
</rect>
</property>
<property name="text">
<string>Exit</string>
</property>
</widget>
</widget>
<resources/>
<connections/>
</ui>
```

2.2 project.py

breaklines _____

```
import re
import sys
import sqlite3
import time
from PyQt5.QtSql import QSqlDatabase, QSqlQueryModel, QSqlQuery,
QSqlTableModel
from PyQt5.QtWidgets import QApplication, QWidget, QTextEdit,
QTableView, QLabel, QProgressBar, QTableWidgetItem, QHeaderView
from PyQt5 import uic
from psuedoSensor import PseudoSensor
from os.path import exists

connection = sqlite3.connect("project.db")
cursor = connection.cursor()
if not exists("project.db"):
    print("File projects.db does not exist.")
    sys.exit()
db = QSqlDatabase.addDatabase("QSQLITE")
db.setDatabaseName("project.db")
db.open()

class AppDemo(QWidget):
    def __init__(self):
        super().__init__()
        uic.loadUi('project.ui', self)
```

```

        self.texteditreadTemp = self.findChild(QTextEdit,
"textEdittemp")
        self.texteditreadHumidity = self.findChild(QTextEdit,
"textEdithumidity")
            self.button.clicked.connect(self.readData)
            self.texteditaveTemp = self.findChild(QTextEdit,
"textEdit_avetemp")
            self.pushButton_avetemp.clicked.connect(self.avetemp)
            self.texteditminTemp = self.findChild(QTextEdit,
"textEdit_mintemp")
            self.pushButton_mintemp.clicked.connect(self.mintemp)
            self.texteditmaxTemp = self.findChild(QTextEdit,
"textEdit_maxtemp")

self.pushButton_maxtemp.clicked.connect(self.maxtemp)
            self.texteditaveHumidity = self.findChild(QTextEdit,
"textEdit_avehumidity")

self.pushButton_avehumidity.clicked.connect(self.avehumidity)
            self.texteditminHumidity = self.findChild(QTextEdit,
"textEdit_minhumidity")

self.pushButton_minhumidity.clicked.connect(self.minhumidity)
            self.texteditmaxHumidity = self.findChild(QTextEdit,
"textEdit_maxhumidity")

self.pushButton_maxhumidity.clicked.connect(self.maxhumidity)
            self.Result = self.findChild(QLabel, "label_Result")
            self.pushButton.clicked.connect(self.recordTenValues)
            self.progressBar = self.findChild(QProgressBar,
"progressBar")
            self.progressBar.setMinimum(0)
            self.progressBar.setMaximum(10)
            self.pushButton.clicked.connect(self.recordTenValues)
            self.temprange = self.findChild(QLabel, "label_temp")
            self.humidityrange = self.findChild(QLabel,
"label_humidity")
            self.tableWidget = self.findChild(QTableWidget,
"tableWidget")
            self.tableWidget.setRowCount(10)
            self.tableWidget.setColumnCount(4)

self.pushButton_Display.clicked.connect(self.displayTable)

```

```

        self.pushButton_Exit.clicked.connect(self.exit)

def exit(self):
    sys.exit()

def readData(self):

    ps = PseudoSensor()
    humidity,temperature = ps.generate_values()
    query = "INSERT INTO weather (temperature, humidity,
date) VALUES ( %f, %f, datetime('now'))"
    value = (temperature, humidity)
    cursor.execute(query % value)
    connection.commit()
    T = str(temperature)
    H = str(humidity)
    self.texteditreadTemp.setPlainText(T)
    self.texteditreadHumidity.setPlainText(H)
    print("H ",humidity)
    print("T ",temperature)
    if humidity>30 and humidity<50:
        self.humidityrange.setText("Within Normal
Humidity Range")
        print("Within Normal Humidity Range")
    else:
        self.humidityrange.setText("Warning! Beyond
Normal Humidity Range")
        print("Warning!!! Beyond Normal Humidity
Range")
    if temperature>30 and temperature<40:
        self.temprange.setText("Within Normal
Temperature Range")
        print("Within Normal Temperature Range")
    else:
        self.temprange.setText("Warning! Beyond
Normal Temp Range")
        print("Warning!!! Beyond Normal Temperature
Range")

def displayTable(self):

    query = "SELECT * FROM (SELECT * FROM weather
ORDER BY id DESC LIMIT 10) ORDER BY id ASC"

```

```

        cursor.execute(query)
        result = cursor.fetchall()
        print(result)
        self.tableWidget.setRowCount(0)
        header = self.tableWidget.horizontalHeader()

header.setSectionResizeMode(QHeaderView.ResizeToContents)
header.setSectionResizeMode(0, QHeaderView.Stretch)
for row_number, row_data in enumerate(result):
    self.tableWidget.insertRow(row_number)
    for column_number, data in
enumerate(row_data):
            self.tableWidget.setItem(row_number,
column_number, QTableWidgetItem(str(data)))

```

```

def avetemp(self):
    query = "SELECT SUM(temperature) FROM (SELECT
temperature FROM weather ORDER BY id DESC LIMIT 10)"
    cursor.execute(query)
    connection.commit()
    SUM = cursor.fetchall()
    print(SUM)
    TOTAL = re.findall(r"[-+]?\d*\.\d+|\d+",str(SUM))
    print(type(TOTAL))
    print(TOTAL)
    total = float(TOTAL[0])
    print(type(total))
    print(total)
    average = total/10
    print("average ",average)
    self.texteditaveTemp.setPlainText(str(average))

```

```

def mintemp(self):
    query = "SELECT MIN(temperature) FROM weather"
    cursor.execute(query)
    connection.commit()
    MINIMUM = cursor.fetchall()
    print(MINIMUM)
    MIN = re.findall(r"[-+]?\d*\.\d+|\d+",str(MINIMUM))
    print(type(MIN))
    print(MIN)
    MIN = float(MIN[0])

```

```

        print(type(MIN))
        print(MIN)
        print("minimum ",MIN)
        self.texteditminTemp.setPlainText(str(MIN))

    def maxtemp(self):
        query = "SELECT MAX(temperature) FROM weather"
        cursor.execute(query)
        connection.commit()
        MAXIMUM = cursor.fetchall()
        print(MAXIMUM)
        MAX = re.findall(r"[-]?\d*\.\d+|\d+",str(MAXIMUM))
        print(type(MAX))
        print(MAX)
        MAX = float(MAX[0])
        print(type(MAX))
        print(MAX)
        print("maximum ",MAX)
        self.texteditmaxTemp.setPlainText(str(MAX))

    def avehumidity(self):
        query = "SELECT SUM(humidity) FROM (SELECT
humidity FROM weather ORDER BY id DESC LIMIT 10)"
        cursor.execute(query)
        connection.commit()
        SUM = cursor.fetchall()
        print(SUM)
        TOTAL = re.findall(r"[-]?\d*\.\d+|\d+",str(SUM))
        print(type(TOTAL))
        print(TOTAL)
        total = float(TOTAL[0])
        print(type(total))
        print(total)
        average = total/10
        print("average ",average)
        self.texteditaveHumidity.setPlainText(str(average))

    def minhumidity(self):
        query = "SELECT MIN(humidity) FROM weather"
        cursor.execute(query)
        connection.commit()
        MINIMUM = cursor.fetchall()
        print(MINIMUM)
        MIN = re.findall(r"[-]?\d*\.\d+|\d+",str(MINIMUM))

```

```

        print(type(MIN))
        print(MIN)
        MIN = float(MIN[0])
        print(type(MIN))
        print(MIN)
        print("minimum ",MIN)
        self.texteditminHumidity.setPlainText(str(MIN))

def maxhumidity(self):
    query = "SELECT MAX(humidity) FROM weather"
    cursor.execute(query)
    connection.commit()
    MAXIMUM = cursor.fetchall()
    print(MAXIMUM)
    MAX = re.findall(r"[-+]?[0-9]*\.[0-9]+",str(MAXIMUM))
    print(type(MAX))
    print(MAX)
    MAX = float(MAX[0])
    print(type(MAX))
    print(MAX)
    print("maximum ",MAX)
    self.texteditmaxHumidity.setPlainText(str(MAX))

def recordTenValues(self):
    for i in range(10):
        ps = PseudoSensor()
        humidity,temperature = ps.generate_values()
        query = "INSERT INTO weather (temperature,
humidity, date) VALUES ( %f, %f, datetime('now'))"
        value = (temperature, humidity)
        cursor.execute(query % value)
        connection.commit()
        print("H ",humidity)
        print("T ",temperature)
        model = QSqlTableModel(None, db)
        model.setTable("weather")
        model.select()
        print("recording ",i+1)
        string =" Values Recorded"
        recording = str(i+1)+string
        self.Result.setText(recording)
        self.progressBar.setValue(i+1)
        if humidity>30 and humidity<50:

```

```

        self.humidityrange.setText("Within
Normal Humidity Range")
        print("Within Normal Humidity Range")
    else:
        self.humidityrange.setText("Warning!
Beyond Normal Humidity Range")
        print("Warning!!! Beyond Normal
Humidity Range")
        if temperature>30 and temperature<40:
            self.temprange.setText("Within Normal
Temperature Range")
            print("Within Normal Temperature
Range")
        else:
            self.temprange.setText("Warning!
Beyond Normal Temp Range")
            print("Warning!!! Beyond Normal
Temperature Range")
        time.sleep(60)

```

```

if __name__ == '__main__':
    app = QApplication(sys.argv)

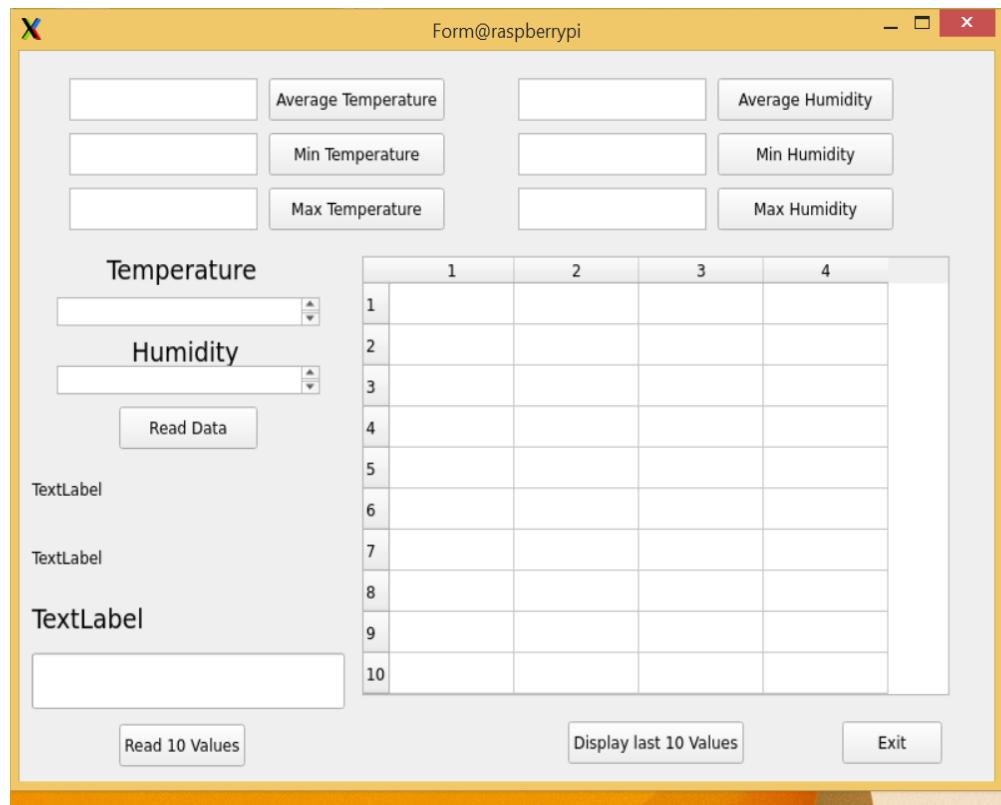
    demo = AppDemo()
    demo.show()

try:
    sys.exit(app.exec())
except SystemExit:
    print('Closing Window...')

```

3. UI Images

3.1 The UI at startup



3.2 The UI after its first single data point reading

Form@raspberrypi

Average Temperature	Average Humidity
Min Temperature	Min Humidity
Max Temperature	Max Humidity

Temperature
-12.795466263408322

Humidity
4.668805901959898

Read Data

Warning! Beyond Normal Temp Range

Warning! Beyond Normal Humidity Range

TextLabel

Read 10 Values Display last 10 Values Exit

	1	2	3	4
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

3.3 The UI after it has calculated a 10 point average

Form@raspberrypi

-14.5469377	Average Temperature	4.7133112	Average Humidity
-18.938107	Min Temperature	0.665084	Min Humidity
-10.97884	Max Temperature	8.500105	Max Humidity

Temperature

Humidity

Read Data

Warning! Beyond Normal Temp Range

Warning! Beyond Normal Humidity Range

10 Values Recorded

100%

	1	2	3	4
1	274	-16.33167	7.444409	2021-12-30 10:32:55
2	275	-18.42437	3.88381	2021-12-30 10:33:06
3	276	-10.97884	0.730348	2021-12-30 10:33:09
4	277	-13.125479	4.586641	2021-12-30 10:33:12
5	278	-12.712847	6.645478	2021-12-30 10:33:15
6	279	-16.810585	4.49394	2021-12-30 10:33:18
7	280	-15.121619	6.663496	2021-12-30 10:33:21
8	281	-14.540271	5.593991	2021-12-30 10:33:24
9	282	-15.656305	6.25407	2021-12-30 10:33:27
10	283	-11.767391	0.836929	2021-12-30 10:33:32

Read 10 Values

Display last 10 Values

Exit

4. The UI after it has seen either a temperature or humidity alarm (or both)

Form@raspberrypi

<input type="text"/>	Average Temperature	<input type="text"/>	Average Humidity
<input type="text"/>	Min Temperature	<input type="text"/>	Min Humidity
<input type="text"/>	Max Temperature	<input type="text"/>	Max Humidity

Temperature

Humidity

Warning! Beyond Normal Temp Range

Warning! Beyond Normal Humidity Range

6 Values Recorded

60%

	1	2	3	4
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				