

MAJOR PROJECT REPORT
ON
**APPLIANCE AUTOMATION
AND ACTUATORS
(BASED ON IOT AND PYTHON)**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE
AWARD OF THE DEGREE OF
**BACHELOR OF TECHNOLOGY
(Computer Science & Engineering)**

Under the guidance of

Ms.Rubinderjit Kaur

Submitted by

**PANKAJ BALOTRA
(1801271)**



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
SARDAR BEANT SINGH STATE UNIVERSITY,
GURDASPUR 2022**

CERTIFICATE

I hereby certify that the work which is being presented in the project entitled **“APPLIANCE AUTOMATION AND ACTUATORS (BASED ON IOT AND PYTHON)”** by **PANKAJ BALOTRA** in partial fulfilment for the award of degree of Bachelors of Technology in Computer Science and Engineering submitted in Department of Computer Science and Engineering, Sardar Beant Singh State University, Gurdaspur is an authentic record of our work under the supervision of **Ms. Rubinderjit Kaur**. The matter enclosed in this project report has not been submitted by me in any other university/institute for the award of B.Tech Degree.

PANKAJ BALOTRA
(1801271)

ABSTRACT

APPLIANCE AUTOMATION AND ACTUATORS project is based on arduino microcontroller programing and Python serial programing. The project is about connecting the user to access the appliances wirelessly ,more efficiently and shockfree way with bluetooth. Automation helps in connecting multiple devices at same time within a network. The arduino microcontroller has multiple functionality through which it can controller various electrical appliances as well as circuits. The bluetooth module used in the project is **HC-05** through which signal is send and recived by bluetooth with python serial program workingin the backend as a serial monitor for the microcontroller.The serial monitor gui is made with the **pyqt5 tool** to make an gui application for the front end user and easy to access.

ACKNOWLEDGEMENT

It is great pleasure to present this report on the project named “**APPLIANCE AUTOMATION AND ACTUATORS (BASED ON IOT AND PYTHON)**” undertaken by us as part of our batch curriculum. First and foremost, we would like to express our thanks towards our training guide **Ms. Rubinderjit Kaur** for placing complete faith and confidence in our ability to carry out this project and for providing us her time, inspiration, encouragement, guidance and commitments to help us to complete this project. The constant encouragement received from Head Department of Computer Science & Engineering, Sardar Beant Singh State University, Gurdaspur has been of great importance in carrying out the present project work. We express our sincere thanks to the people who helped us along the way in completing our project.

PANKAJ BALOTRA

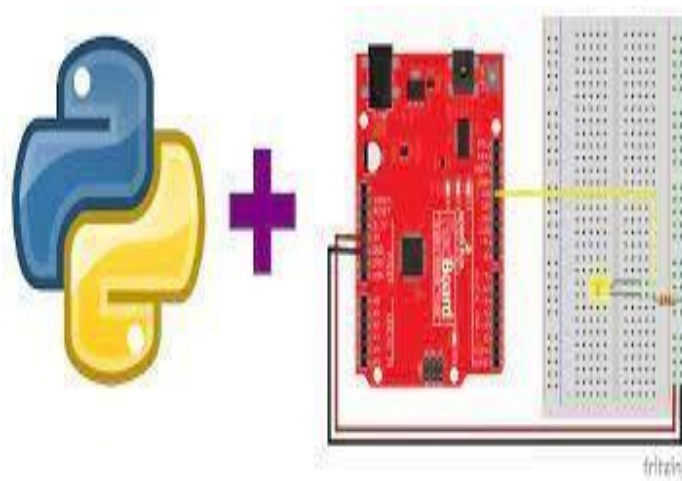
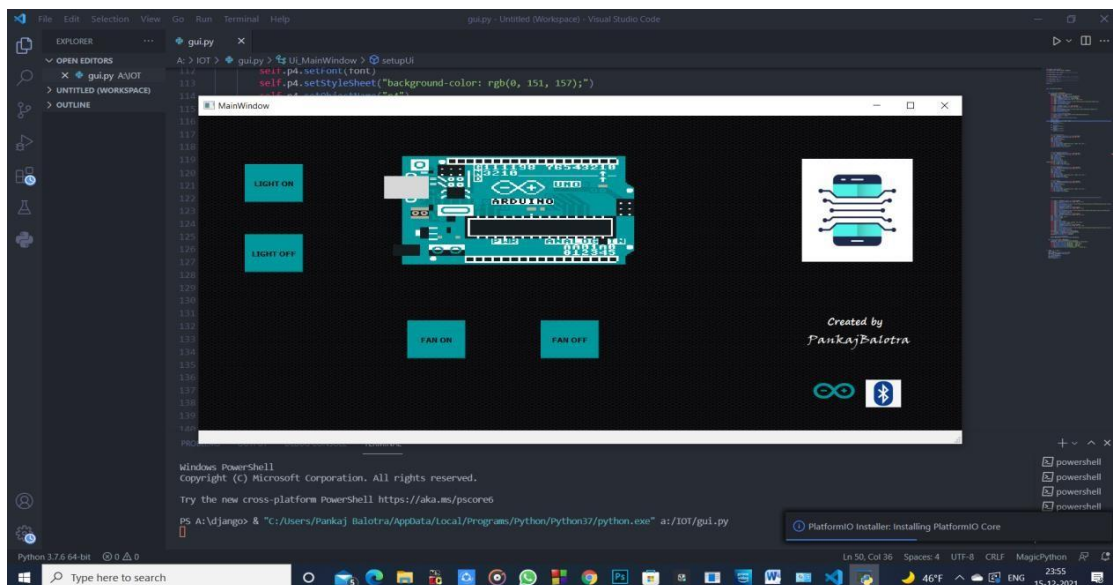
| | | |
|---|---|----------|
| 1 | CHAPTER-1 INTRODUCTION | Page No: |
| | 1.0- INTRODUCTION TO PROJECT | 6 |
| 2 | CHAPTER-2 2.0 –ARDIUNO MICROCONTROLLER AND HC-05 2.1 –PYTHON PYSERIAL LIBRARY AND SERIAL MONITOR 2.3 –PROGRAMMING OF ARDIUNO AND SOFTWARE USED | 7-12 |
| 3 | CHAPTER-3 3.0 -SYNOPSIS OF PROJECT-FRONTEND-BACKEND CODING | 13-19 |
| 4 | CHAPTER-4 4.0 –OUTPUT | 20-21 |
| 5 | CHAPTER-5 5.0- CONCLUSION | 22 |

CHAPTER-1

INTRODUCTION

1.0- INTRODUCTION TO PROJECT

APPLIANCE AUTOMATION project is based on arduino microcontroller programming and Python serial programming. The project is about connecting the user to access the appliances wirelessly, more efficiently and shockfree way with bluetooth. Automation helps in connecting multiple devices at same time within a network. The arduino microcontroller has multiple functionality through which it can controller various electrical appliances as well as circuits. The bluetooth module used in the project is **HC-05** through which signal is send and recived by bluetooth with python serial program working in the backend as a serial monitor for the microcontroller. The serial monitor gui is made with the **pyqt5** tool to make an gui application for the front end user and easy to access.



CHAPTER-2

2.0 – ARDIUNO MICROCONTROLLER

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

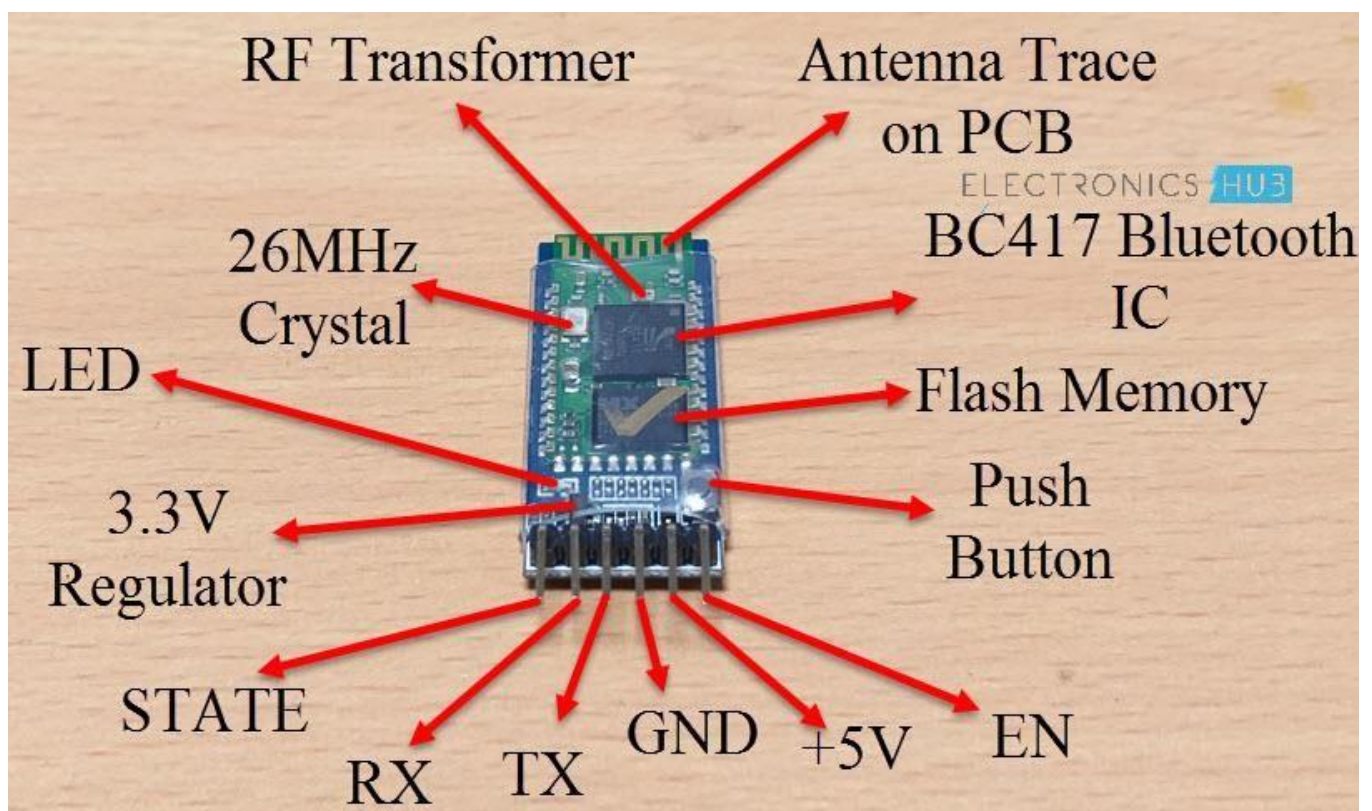
Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

HC-05-BLUETOOTH



HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup..... HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.

2.1 -PYTHON PYSERIAL LIBRARY

OVERVIEW-

This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend. It is released under a free software license, see LICENSE for more details. Copyright (C) 2001-2020 Chris Liechti Other pages (online) • project page on GitHub • Download Page with releases (PyPi) • This page, when viewed online is at <https://pyserial.readthedocs.io/en/latest/> or <http://pythonhosted.org/pyserial>

FEATURES-

- Same class based interface on all supported platforms.
- Access to the port settings through Python properties.
- Support for different byte sizes, stop bits, parity and flow control with RTS/CTS and/or Xon/Xoff.
- Working with or without receive timeout.
- File like API with “read” and “write” (“readline” etc. also supported).
- The files in this package are 100% pure Python.
- The port is set up for binary transmission. No NULL byte stripping, CR-LF translation etc. (which are many times enabled for POSIX.) This makes this module universally useful.
- Compatible with io library • RFC 2217 client (experimental), server provided in the examples

Requirements-

- Python 2.7 or Python 3.4 and newer
- If running on Windows: Windows 7 or newer
- If running on Jython: “Java Communications” (JavaComm) or compatible extension for Java For older installations (older Python versions or older operating systems), see older versions below.

Installation-

This installs a package that can be used from Python (import serial). To install for all users on the system, administrator rights (root) may be required.

From PyPI

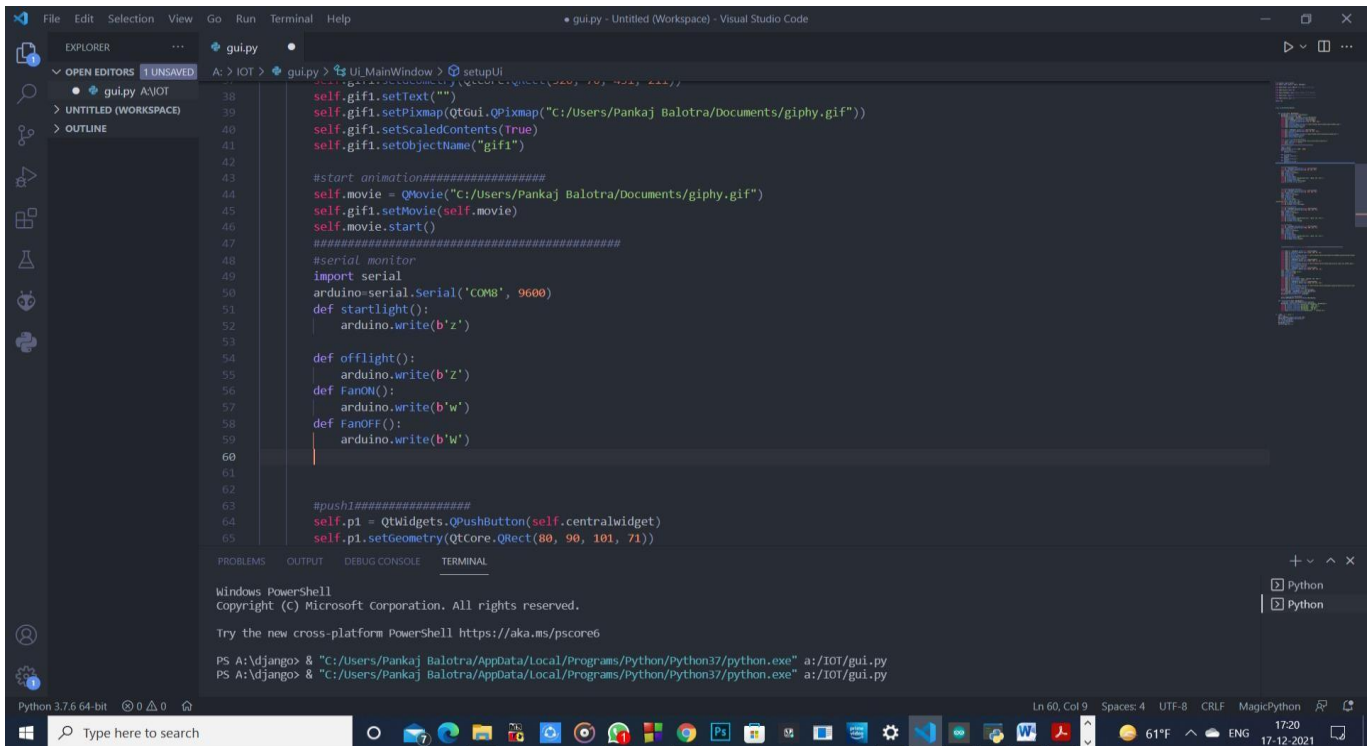
pySerial can be installed from PyPI:

python -m pip install pyserial

Using the python/python3 executable of the desired version (2.7/3.x). Developers also may be interested to get the source archive, because it contains examples, tests and the this documentation.

SERIAL MONITOR WITH PYTHON-

- The Serial Monitor is an essential tool when creating projects with Arduino. It can be used as a debugging tool, testing out concepts or to communicate directly with the Arduino board.
- The Serial Monitor tool is a really great tool for debugging and establishing communication between a computer and an Arduino. To use it is really easy, but there are some things that we need to do first.
- A major advantage with having the Serial Monitor integrated with the editor is the possibility to have multiple monitors open simultaneously. In the old editor, when changing the port/board, it changed across all windows. In addition, we were limited to one Serial Monitor window, an obstacle that the IDE 2.0 removes.



```
File Edit Selection View Go Run Terminal Help
gui.py - Untitled (Workspace) - Visual Studio Code

EXPLORER
OPEN EDITORS 1 UNSAVED
gui.py A:\IOT
UNTITLED (WORKSPACE)
OUTLINE

38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

self.gif1.setText("")
self.gif1.setPixmap(QtGui.QPixmap("C:/Users/Pankaj Balotra/Documents/gifpy.gif"))
self.gif1.setScaledContents(True)
self.gif1.setObjectName("gif1")

#start animation#####
self.movie = QMovie("C:/Users/Pankaj Balotra/Documents/gifpy.gif")
self.gif1.setMovie(self.movie)
self.movie.start()

#####
#serial_monitor
import serial
arduino=serial.Serial('COM8', 9600)
def startlight():
    arduino.write(b'z')

def offlight():
    arduino.write(b'Z')
def FanON():
    arduino.write(b'w')
def FanOFF():
    arduino.write(b'W')

#####
self.p1 = QtWidgets.QPushButton(self.centralwidget)
self.p1.setGeometry(QtCore.QRect(80, 90, 101, 71))

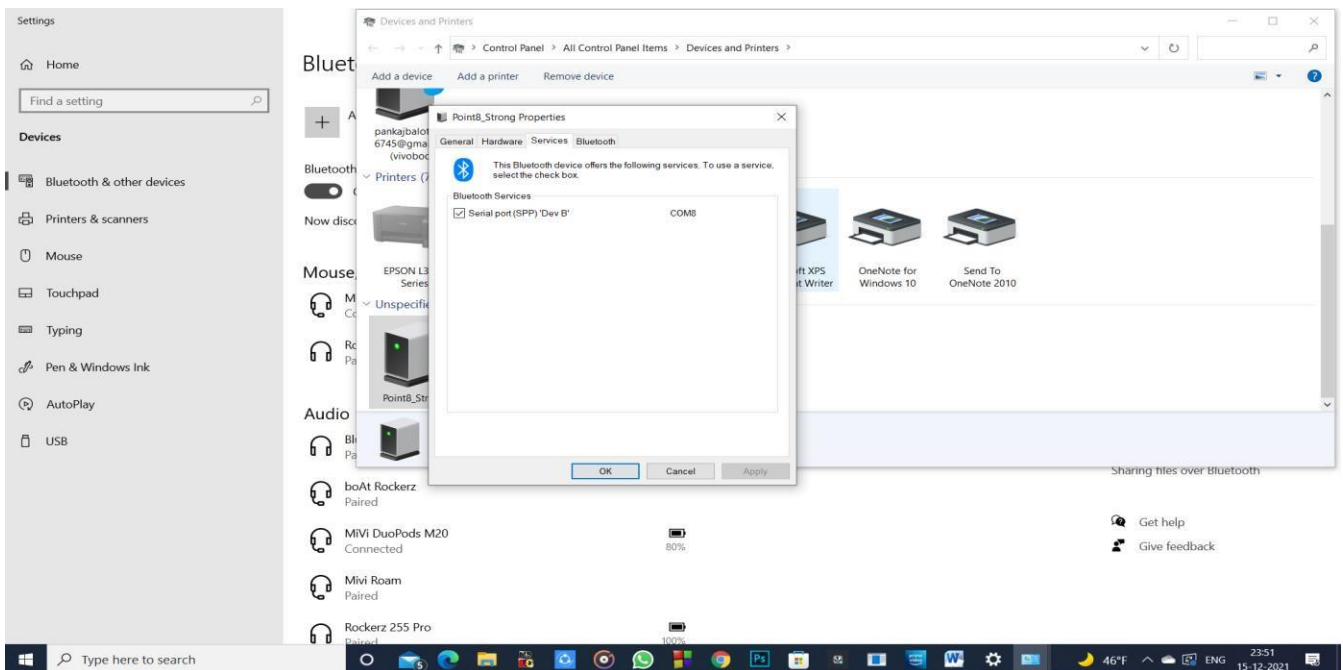
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS A:\djangoo & "C:/Users/Pankaj Balotra/AppData/Local/Programs/Python/Python37/python.exe" a:/IOT/gui.py
PS A:\djangoo & "C:/Users/Pankaj Balotra/AppData/Local/Programs/Python/Python37/python.exe" a:/IOT/gui.py

Python 3.7.6 64-bit
Type here to search
Ln 60, Col 9 Spaces: 4 UTF-8 CRLF MagicPython 17:20 17-12-2021
```

ABOVE IS MY SERIAL MONITOR CODING TO COMMUNICATE WITH ARDIUNO AT 9600 FREQUENCY AND SPP = COM8.



THIS IS THE SPP(SERIAL PORT) OR ADDRESS OF THE HC-05 BLUETOOTH ON WHICH BLUETOOTH IS CONNECTED.

2.3 –PROGRAMMING OF ARDIUNO-



```
sketch_nov13b | Arduino 1.8.18 (Windows Store 1.8.55.0)
File Edit Sketch Tools Help

sketch_nov13b $

char inputByte;
void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {
  while(Serial.available() > 0) {
    inputByte = Serial.read();
    Serial.println(inputByte);
    if (inputByte == 'Z') {
      digitalWrite(13, HIGH);
    }
    else if (inputByte == 'z') {
      digitalWrite(13, LOW);
    }
    else if (inputByte == 'W') {
      digitalWrite(11, LOW);
    }
    else if (inputByte == 'w') {
      digitalWrite(11, HIGH);
    }
  }
}
```

Save Canceled.

Error downloading https://downloads.arduino.cc/packages/package_index.json

1 Arduino Uno on COM6

THE ABOVE IMAGE IS THE CODE WHICH IS UPLOADED INTO THE MICROCONTROLLER (ARDIUNO) THROUGH USB PORT.

ARDIUNO IDE SOTWEARE-

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, *avrdude* is used as the uploading tool to flash the user code onto official Arduino boards.

CHAPTER-3

3.0 - SYNOPSIS OF PROJECT-

OVERVIEW-

APPLIANCE AUTOMATION AND ACTUATORS project is based on arduino microcontroller programming and Python serial programming. The project is about connecting the user to access the appliances wirelessly, more efficiently and shockfree way with bluetooth. Automation helps in connecting multiple devices at same time within a network. The arduino microcontroller has multiple functionality through which it can control various electrical appliances as well as circuits. The bluetooth module used in the project is HC-05 through which signal is sent and received by bluetooth with python serial program working in the backend as a serial monitor for the microcontroller. The serial monitor gui is made with the pyqt5 tool to make an gui application for the front end user and easy to access.

1. GUI USING PYQT5 TOOL PYTHON –

PyQt5 is cross-platform GUI toolkit, a set of python bindings for Qt v5. One can develop an interactive desktop application with so much ease because of the tools and simplicity provided by this library. A GUI application consists of Front-end and Back-end.

2. INSTALLATION OF PYQT5-

First, we need to install PyQt5 library. For this, type the following command in the terminal or command prompt: **pip install pyqt5**-installation command.

3. CONVERTING .UI FILE INTO .PY FILE:

For this we have to go to sitepackages directory in terminal or command prompt and run the command as shown below. Getting the location of sitepackages is mentioned previously. >>> cd "C:\\Users\\.....\\Programs\\Python\\Python36-32\\lib\\site-packages"
[Location of sitepackages]
>>> pyuic5 "C:\\Users\\.....\\FILENAME.ui"[Exact location of .ui file] -o " C:\\Users\\.....\\FILENAME.py" [Location where want to put .py file]

2.0 – ARDUINO MICROCONTROLLER

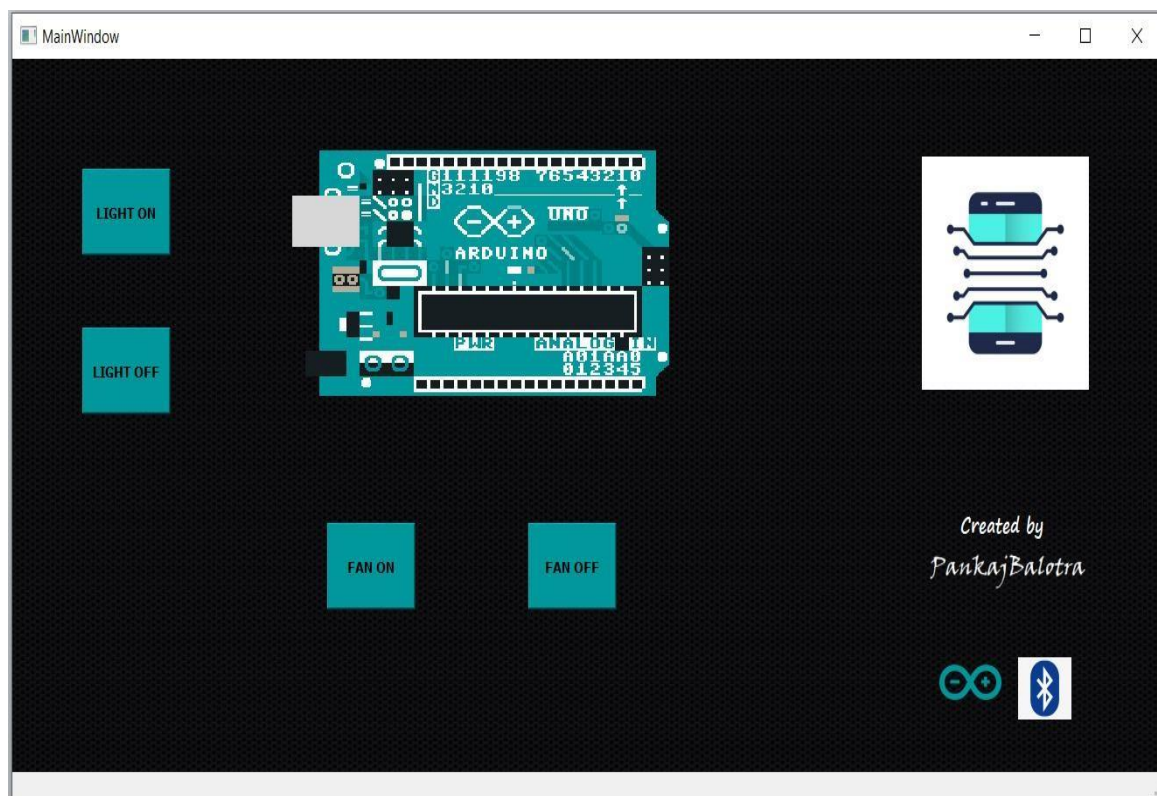
Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED,

publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

3.0 HC-05

HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup..... HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.

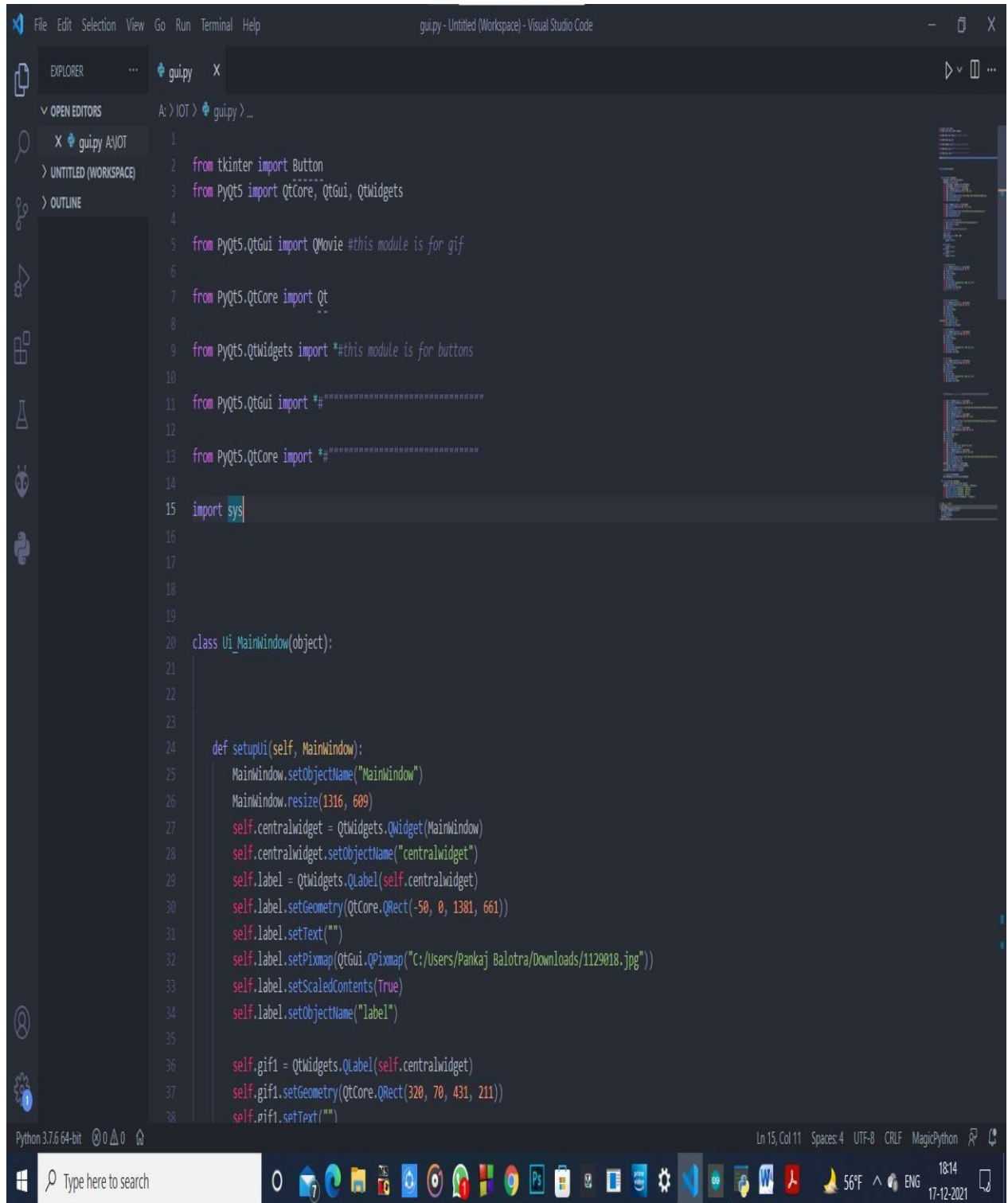
FRONT END OF PROJECT-



THIS IS MY GUI INTERFACE WHICH IS MADE BY PYQT5-PYTHON

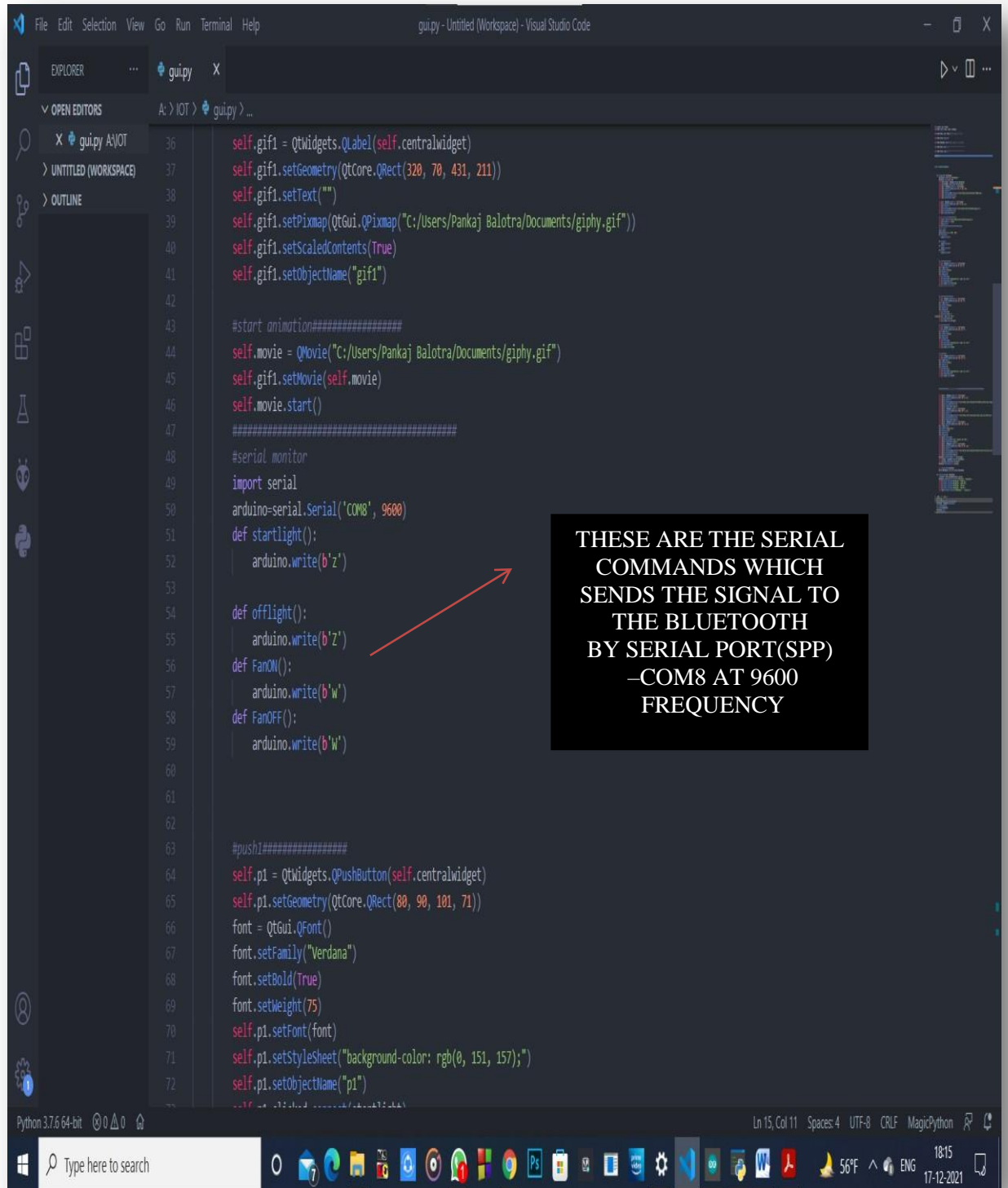
BACKEND-

PROGRAMMING OF THE PROJECT-



```
1 from tkinter import Button
2 from PyQt5 import QtCore, QtGui, QtWidgets
3
4 from PyQt5.QtGui import QMovie #this module is for gif
5
6 from PyQt5.QtCore import Qt
7
8 from PyQt5.QtWidgets import * #this module is for buttons
9
10
11 from PyQt5.QtGui import * #
12
13 from PyQt5.QtCore import * #
14
15 import sys
16
17
18
19
20 class Ui_MainWindow(object):
21
22
23
24     def setupUi(self, MainWindow):
25         MainWindow.setObjectName("MainWindow")
26         MainWindow.resize(1316, 609)
27         self.centralwidget = QtWidgets.QWidget(MainWindow)
28         self.centralwidget.setObjectName("centralwidget")
29         self.label = QtWidgets.QLabel(self.centralwidget)
30         self.label.setGeometry(QtCore.QRect(-50, 0, 1381, 661))
31         self.label.setText("")
32         self.label.setPixmap(QtGui.QPixmap("C:/Users/Pankaj Balotra/Downloads/1129018.jpg"))
33         self.label.setScaledContents(True)
34         self.label.setObjectName("label")
35
36         self.gif1 = QtWidgets.QLabel(self.centralwidget)
37         self.gif1.setGeometry(QtCore.QRect(320, 70, 431, 211))
38         self.gif1.setText("")
```

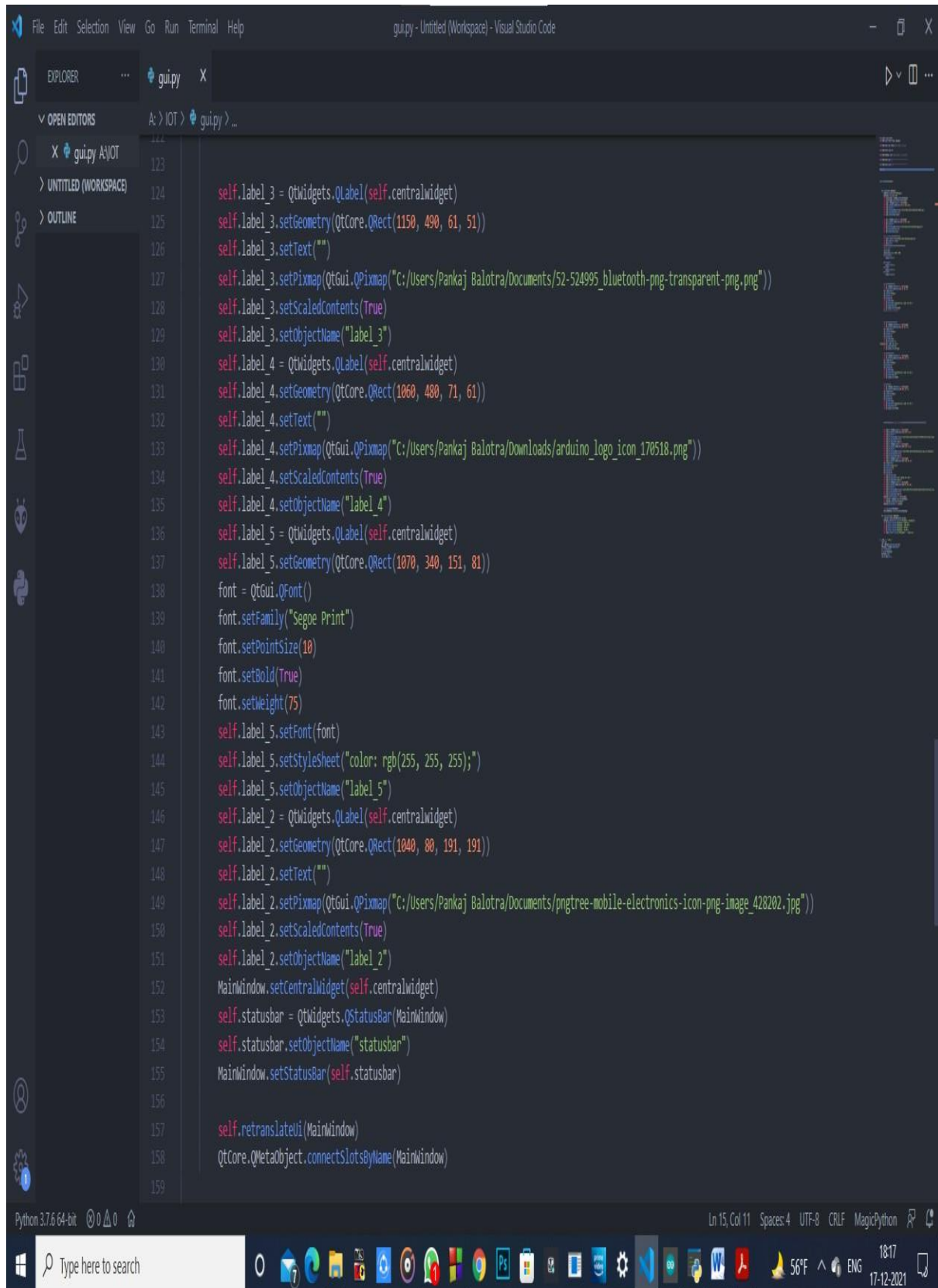
THESE ARE THE SCREEN SHOTS OF MY PROJECT BASED PYTHON CODING

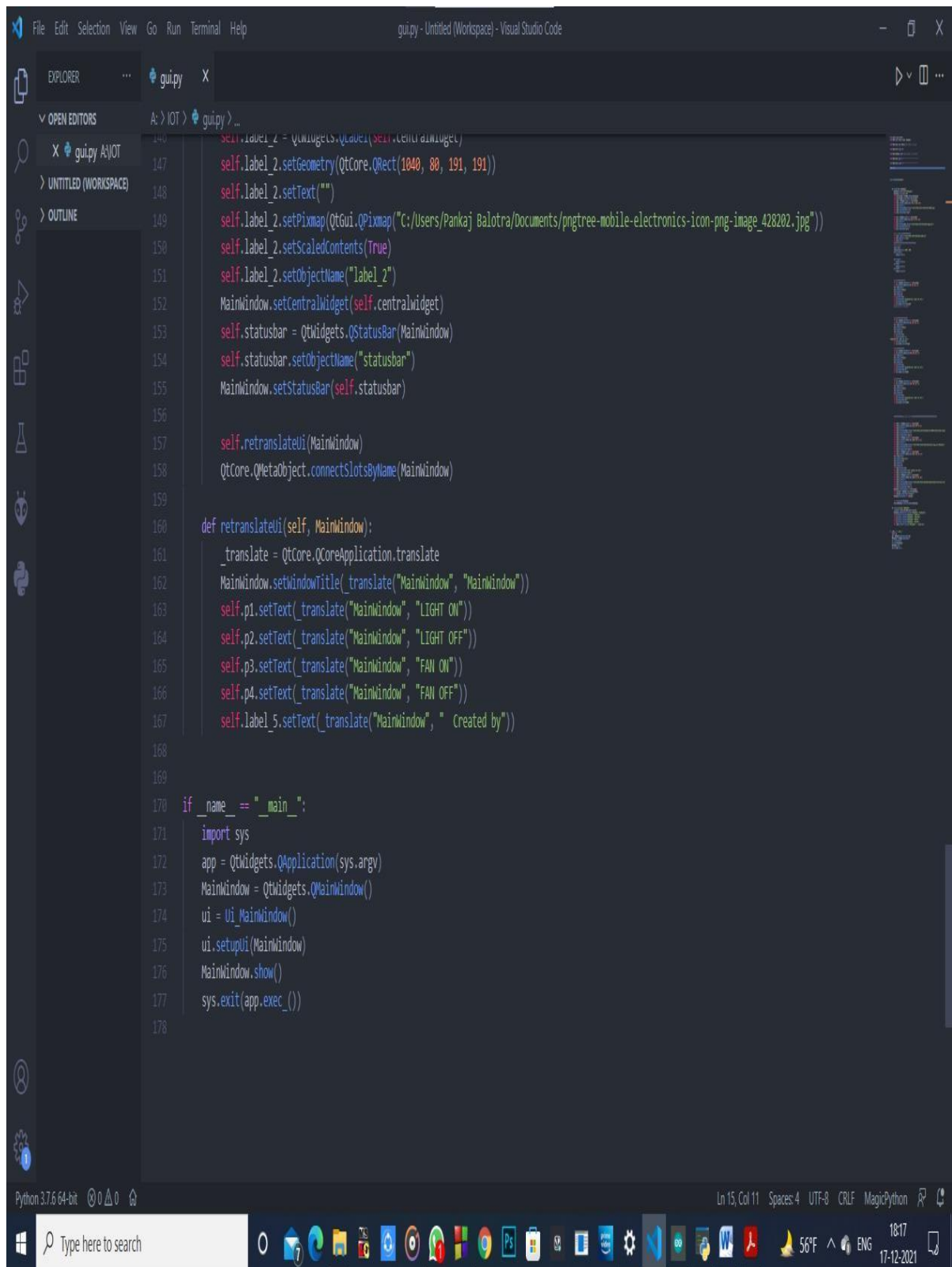


The image shows a Visual Studio Code editor window with a Python file named `gui.py`. The code is written in Python 3.7.6 and uses the Qt framework to create a GUI with three push buttons. The buttons are named `p1`, `p2`, and `p3`. Each button is created using `QtWidgets.QPushButton` and is configured with a specific geometry, font, and background color. The buttons are connected to functions `startlight` and `offlight` via the `clicked` signal.

```
63 #push1#####
64 self.p1 = QtWidgets.QPushButton(self.centralwidget)
65 self.p1.setGeometry(QtCore.QRect(80, 90, 101, 71))
66 font = QtGui.QFont()
67 font.setFamily("Verdana")
68 font.setBold(True)
69 font.setWeight(75)
70 self.p1.setFont(font)
71 self.p1.setStyleSheet("background-color: rgb(0, 151, 157);")
72 self.p1.setObjectName("p1")
73 self.p1.clicked.connect(startlight)
74 #adding function to the button
75
76
77
78
79 #push2#####
80 self.p2 = QtWidgets.QPushButton(self.centralwidget)
81 self.p2.setGeometry(QtCore.QRect(80, 220, 101, 71))
82 font = QtGui.QFont()
83 font.setFamily("Verdana")
84 font.setBold(True)
85 font.setWeight(75)
86 self.p2.setFont(font)
87 self.p2.setStyleSheet("\n"
88 "background-color: rgb(0, 151, 157);")
89 self.p2.setObjectName("p2")
90 self.p2.clicked.connect(offlight)
91
92 #push3#####
93 self.p3 = QtWidgets.QPushButton(self.centralwidget)
94 self.p3.setGeometry(QtCore.QRect(360, 380, 101, 71))
95 font = QtGui.QFont()
96 font.setFamily("Verdana")
97 font.setBold(True)
98 font.setWeight(75)
99 self.p3.setFont(font)
```

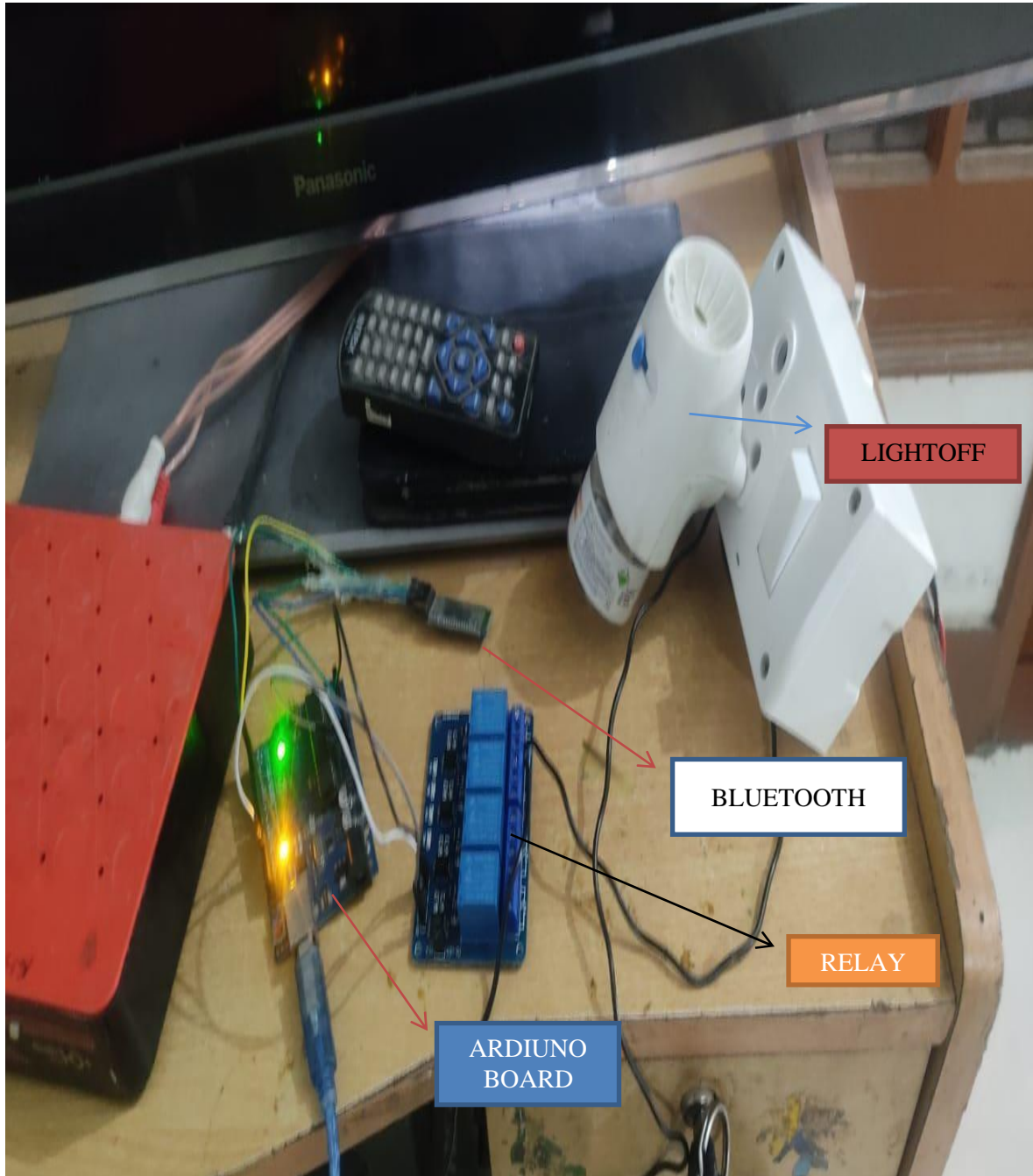
PUSH BUTTONS FOR THE GUI





CHAPTER-4

4.0 –OUTPUTS OF MY PROJECT



THIS IMAGE SHOWS THE INPUT IS OFF FROM THE BLUETOOTH DEVICE WHILE PRESSING LIGHT OFF ON THE DESKTOP APPLICATION



THIS IMAGE SHOWS THE INPUT IS ON FROM THE BLUETOOTH DEVICE WHILE PRESSING LIGHT ON USING THE DESKTOP APPLICATION

CHAPTER-5

5.0- CONCLUSION-

THE PROJECT IS ABOUT AUTOMATION OF EACH AND EVERY APPLIANCE IN OUR HOUSES OR AT ANY PLACE OF WORK ,SO THAT A SINGLE DEVICE CAN CONTROL THE WHOLE ENVIRONMENT EACH AND EVERY THING WHICH RUN ON ELECTRICITY- BULB, FAN, WATER-PUMPS MOTORS, TV,AND ANY THING AROUND US \.

KEY FEATURES-

- **AVOID ELECTRIC SHOCKS.**
- **TURN ON-OFF ANY APPLIANCE WIRELESSLY AND MORE SAFELY.**
- **CAN BE HELP FULL FOR THOSE HUMANS WHO ARE PHYSICALLY CHALLENGED.**
- **THIS PROJECT SHOWS THE CO-RELATION B/W PYTHON PROGRAMMING AND IOT.**
- **THESE TYPE OF PROJECT CAN BE HELPFULL IN BUILDING MORE SKILLS OTHER THAN A SINGLE BRANCH FOCUSED PROJECTS.**
- **THESE TYPE OF PROJECT INCLUDES SOFTWARE KNOWLEDGE AS WELL AS HARDWARE KNOWLEDGE.**