

# Development of a Three-Layer IoT System using Adaptive User Interface

Kirti Panwar Bhati

School of Electronics, Devi Ahilya University, Indore, India

[kirti.1809@gmail.com](mailto:kirti.1809@gmail.com)

**Abstract** – IoT is evolving at a very rapid pace, and the development of an IoT System requires taking into account at least a three-layer implementation. The top layer is the User Interface layer, which plays an important role in any Internet of Things (IoT) system. It allows the user to interact with the underlying system without bothering about the internal details. The user interface should be user-friendly and should be manageable by the layman as well. The proposed system allows user to use either the Line User Interface through Telegram App or GUI for interacting with the system. Raspberry Pi is used for the implementation of this system, that is a low-cost hardware with feature-rich applications development capabilities.

**Index Terms:** IoT, Interface, GUI, telegram, Python.

## I. INTRODUCTION

The adaptive user interface proposed here is a user interface for controlling the IoT Device. It consists of two interfaces. One is Line User Interface (LUI) using Telegram App and other is Graphical User Interface developed using Python. Controlling the IoT application should be user-friendly and not require knowledge of the internal details and the technical details of the system. Here, Tri-color LED used to demonstrate the verification of the adaptive User Interface. Raspberry Pi 3 development board is used for implementation of this system.

## II. ARCHITECTURE OF PROPOSED SYSTEM

Figure 1 shows the three-layer architecture of the proposed system. The Adaptive User Interface allows the use of any one user interface available out of two interfaces. Network layer is responsible for communication between the user and the hardware. The perception layer consists of IoT device with sensors and actuators. The network layer is responsible for communication between hardware and the user. Top layer is the Application Layer that directly interacts with the user. The application layer consists of two interfaces. One is the Telegram Application and the second is GUI designed using the Python Language using the tkinter module.

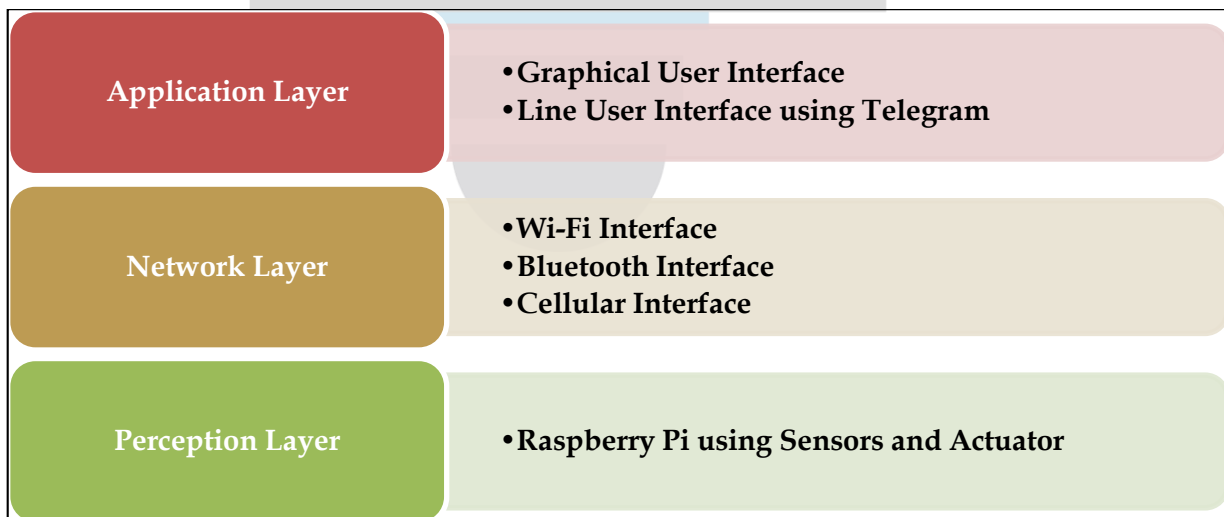


Figure 1: Three-layer architecture of proposed system

## III. METHODOLOGY

Figure 2 shows the process of developing the system. First the GUI is developed using the Python language using tkinter module. Then a Bot is created using Botfather in Telegram for implementation of our own Bot. The authentication key of this bot is used in Python script for secure communication between Telegram App and hardware. The LED used at perception layer for testing the system from both the Apps. System implementation is checked and if it works well then system can be deployed for user.

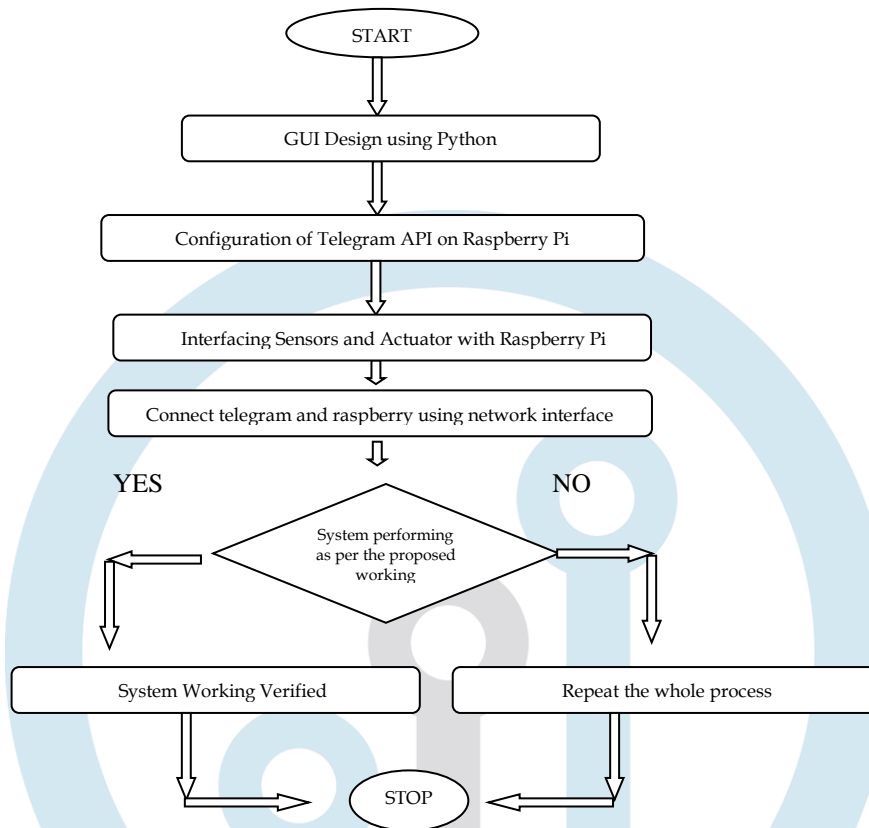


Figure 2: Flow diagram of the implementation of system

### Components Used for Implementation of System

- a. Component used for implementation of proposed system can be divided into the following category as given in Figure 3. Python is used for development of code and for designing the GUI. Telegram Android Mobile Application serving as one of the user interface for controlling IoT System. Raspberry Pi is used for development of IoT System and interface with Sensors and Actuators.

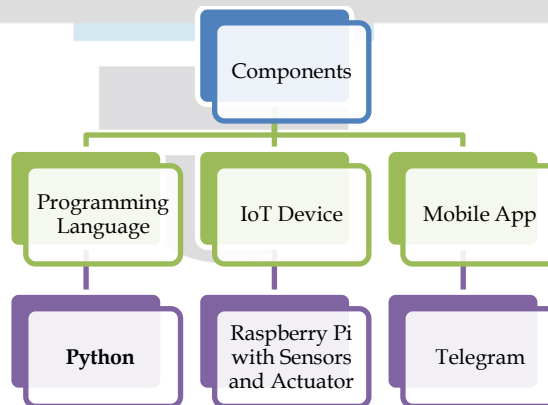


Figure 3: Components for Implementation of System

## IV. IMPLEMENTATION AND RESULT

The implemented system is working as expected. One of the interfaces implemented using a Graphical User Interface for controlling the system is shown in Figure 4. This interface is designed using Python. The main module for implementation is tkinter.

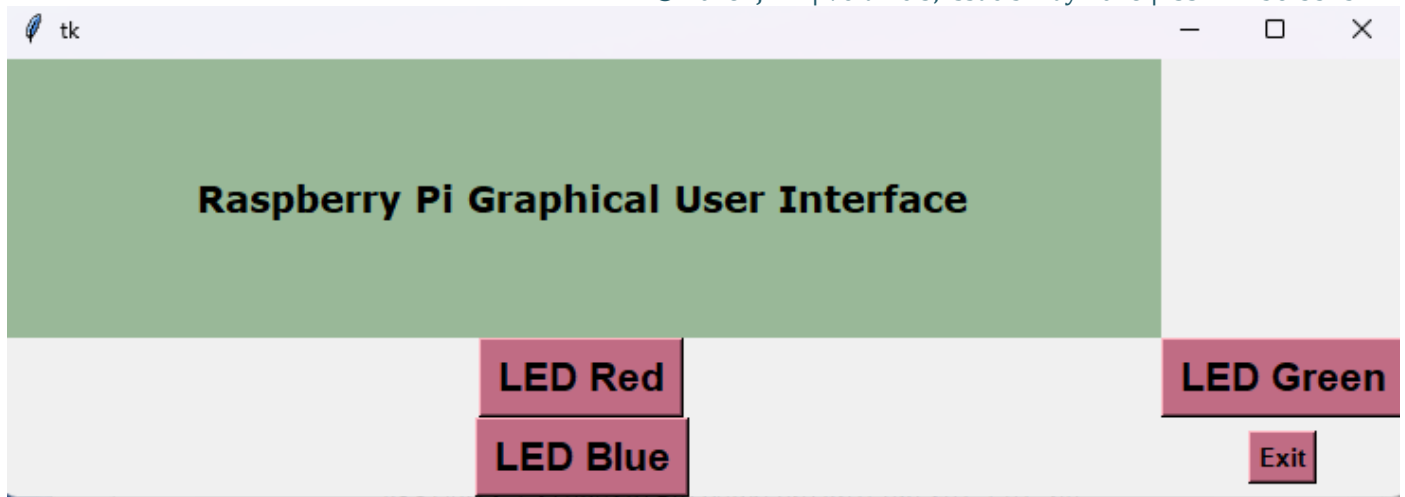


Figure 4: GUI Designed using tkinter module

The second interface is shown in Figure 5. It is a Telegram Bot named Kirti P B Bot. It is created with the help of the Botfather Bot. When a command is sent using this bot, it is received by the Raspberry Pi and an action is taken by the hardware.



Figure 5: Telegram Bot Kirti P B Configured using BotFather Bot

### Code Snippet

#### a. For Creating the Object of Bot for connecting with the Telegram App

```
bot = telepot.Bot('528229712:AAHIkghb8wxLPJjnB_KwN61j3q-V5s3PJrI')
```

#### b. For Setting the Pin of Raspberry Board

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(15, GPIO.OUT)
```

### c. For GUI Development.

```
label_1 = Label(root, text="Raspberry Pi Graphical User Interface", font="Verdana 14 bold", fg="#000", bg="#99B898")
exitButton= Button(root, text="Exit", background= "#C06C84", command=btnExit, font= "Arial 10 bold")
ledButtonR= Button(root, text="LED Red",background= "#C06C84", command=btnClicked, font= "Arial 16 bold")
ledButtonG= Button(root, text="LED Green",background= "#C06C84", command=btnClicked, font= "Arial 16 bold")
ledButtonB= Button(root, text="LED Blue",background= "#C06C84", command=btnClicked, font= "Arial 16 bold")
```

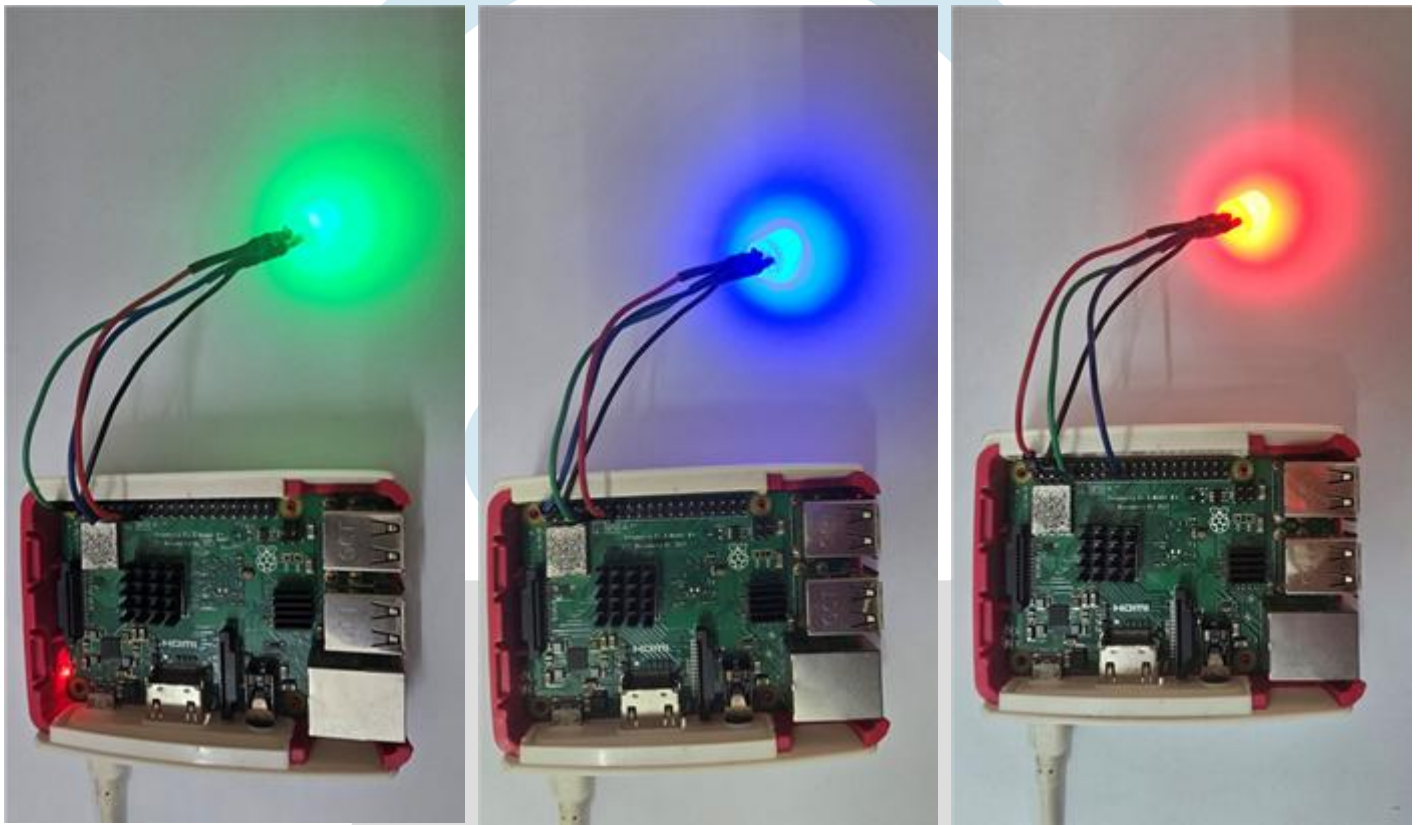


Figure 6: Controlling the LED using Telegram App and GUI

### CONCLUSION

IoT is evolving at a rapid pace, and to increase its penetration in every aspect of human life, an IoT system should be developed in a user-friendly way. User Interface plays an important role in interacting with the user and hides the technical underlying details. The developed system is a prototype for an adaptive user interface for interacting with the user. This system can be scaled and can be replicated for a similar system.

### REFERENCES

- [1] H. Saribekyan and A. Margvelashvili, "Security Analysis of Telegram 6.857 Final Project," 2017. Available: <https://courses.csail.mit.edu/6.857/2017/project/19.pdf>
- [2] A. Dargahi Nobari, N. Reshadatmand, and M. Neshati, "Analysis of Telegram, An Instant Messaging Service," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Nov. 2017, doi: <https://doi.org/10.1145/3132847.3133132>.
- [3] M. Brambilla, E. Umuhzoza, and R. Acerbis, "Model-driven development of user interfaces for IoT systems via domain-specific components and patterns," *Journal of Internet Services and Applications*, vol. 8, no. 1, Sep. 2017, doi: <https://doi.org/10.1186/s13174-017-0064-1>.
- [4] K. Srinath, "Python -The Fastest Growing Programming Language," 2017. Available: <https://www.irjet.net/archives/V4/i12/IRJET-V4I1266.pdf>
- [5] M. Summerfield, *Rapid GUI Programming with Python and Qt*. Prentice Hall, 2015.
- [6] D. Kuhlman, *A Python Book*. Platypus Global Media, 2011.