KONGU ENGINEERING COLLEGE

PERUNDURAI, ERODE - 638060

DEPARTMENT OF ECE

REMOTE RASPBERRY PI CONTROL VIA TELEGRAM BOT USING PYTHON

MINIPROJECT REPORT

(Regulation 2020)

20ECE16 - EMBEDDED IOT

KONGU ENGINEERING COLLEGE

(Autonomous)
Perundurai, Erode – 638 060

Estd : 1984

Department of Electronics and Communication Engineering

20ECE16 - EMBEDDED IOT

Name: JOSIKA N T **Roll Number:** 20ECR065

Name: KARTEESWAR K P **Roll Number:** 20ECR071

Name: LATHIKA M V **Roll Number:** 20ECR088

Programme:B.E **Branch:**ECE **Semester:**VI **Section:B**

Certified that this is a bonafide record of work done by the above students for the ***20ECE16 - EMBEDDED IOT - MiniProject*** *titled* ***REMOTE RASPBERRY PI CONTROL VIA TELEGRAM BOT USING PYTHON*** *during the year 2022- 2023.*

Date of Submission: 22.05.2023 **Signature of In-charge**

ABSTRACT

This tutorial unveils a comprehensive guide for achieving remote control of a Raspberry Pi through the implementation of a Telegram Bot using Python. This innovative system requires only a stable internet connection, granting users the ability to seamlessly oversee and interact with their Raspberry Pi from any corner of the world. The integration of Telegram, a widely recognized open-source instant messaging platform, serves as the linchpin for enabling remote command execution, data exchange, and real-time communication within electronic projects. This report not only elucidates the design and methodology of the system but also emphasizes the transformative potential it carries. By merging the capabilities of the Raspberry Pi with the versatility of Telegram, users are introduced to a powerful tool with applications spanning the realms of Internet of Things (IoT), home automation, and industrial control. The integration of these technologies fundamentally reshapes the landscape of remote device management, opening the door to countless possibilities for innovation and practical application. In addition to addressing the technical aspects, this report explores the broader implications of this solution, offering a holistic perspective on its potential and practicality.

Table of Contents

Chapter – 1

1.1 INTRODUCTION

In today's dynamic world of electronic projects, the demand for effective remote device management has surged. As connectivity and automation take center stage, users are increasingly seeking solutions that enable them to control and interact with their electronic devices from anywhere. Traditional methods often fall short, thus ushering in a new era of alternative communication channels within the electronic landscape. Telegram, an open-source instant messaging service, has risen as a beacon of innovation in this realm. With its advanced automation capabilities and open API, it has become the preferred platform for those seeking to remotely control their electronic devices. This report serves as a comprehensive guide, shedding light on the path to harnessing Telegram's potential for remote Raspberry Pi control. Beyond the technical aspects, the report underscores the transformative potential of this integration. By merging the capabilities of the Raspberry Pi with Telegram's accessibility and versatility, users gain a powerful tool for IoT, home automation, and industrial control, effectively bridging the realm of imagination with practical reality on a global scale.

Chapter 2

DESIGN AND METHODOLOGY

2.1 CONNECTING THE RASPBERRY PI WITH A TELEGRAM BOT:

The fundamental objective of this section is to elucidate the intricate process of linking a Raspberry Pi to a Telegram Bot, creating a system that enables remote control and interaction with the Raspberry Pi from anywhere in the world. The core of this system hinges on the Python programming language, serving as the bridge between these two entities.

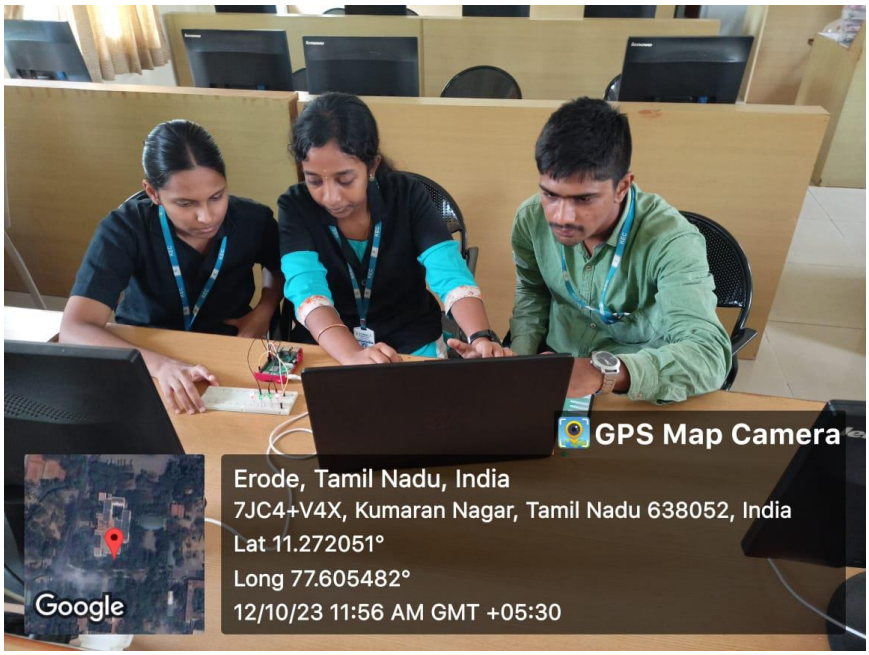

Fig: 01 CONNECTING THE RASPBERRY PI WITH A TELEGRAM BOT (2:1)

- **Python Script as Intermediary:** A crucial Python script acts as an intermediary, ensuring seamless data and command flow between the Raspberry Pi and Telegram Bot.
- **Interfacing with Telegram API:** The Python script connects to the Telegram API, facilitating communication between the Raspberry Pi and Telegram Bot.
- **Command Dispatch via Telegram:** Users send commands via text messages or images through the Telegram app, which the Python script routes to the Raspberry Pi.

- **Continuous Message Monitoring:** The Raspberry Pi maintains constant vigilance, monitoring incoming messages from the Telegram Bot in real-time.
- **Command Interpretation and Execution:** Upon receiving a command, the Python script interprets and executes corresponding actions, such as controlling GPIO pins and sensors.

2.2 FLOWCHART:

This visual representation provides a step-by-step diagram of how communication flows between the Telegram Bot, the Raspberry Pi, and connected devices. It simplifies the understanding of how messages are received, processed, and executed, ultimately enabling remote control and automation. The flowchart is a valuable tool for users to grasp the system's workings at a glance, making it particularly useful for those who prefer visual aids to comprehend technical processes.

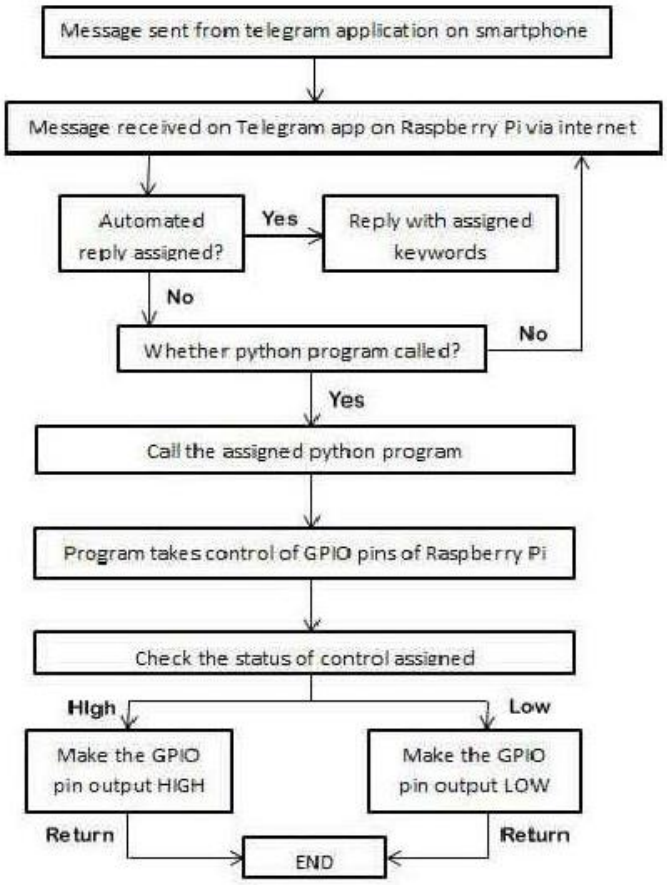

Fig: 02 FLOWCHART (2:2)

2.3 SETTING UP A TELEGRAM BOT:

- Create a Telegram account or use an existing one.
- Open Telegram and search for the "BotFather" account to create a new bot.
- Follow the instructions to set a name and username for your bot and receive the API token.
- Configure the bot's profile with a profile picture and description.
- Set up privacy settings for the bot, such as controlling who can start a conversation with it.
- Customize the bot's responses and commands using the BotFather's commands.
- Test the bot by sending messages and commands to ensure it responds correctly.

2.4 CONNECTING THE RASPBERRY PI TO THE INTERNET:

- Ensure that your Raspberry Pi is connected to the internet via Wi-Fi or Ethernet.
- Verify the connection by pinging a web address to confirm internet access.
- Set up a static IP address for your Raspberry Pi to ensure consistent connectivity.
- Implement firewall rules and security measures to protect your Raspberry Pi when connected to the internet.

2.5 PYTHON LIBRARIES USED:

- Python is the primary programming language used for this project.
- Essential libraries include:
- Telepot: for interacting with the Telegram Bot API.
- RPi.GPIO: for controlling the GPIO pins of the Raspberry Pi.
- Additional libraries may be used for specific project requirements, such as image processing or data logging.

2.6 SENDING MESSAGES AND IMAGES:

- The Python script facilitates sending and receiving messages and images between the Telegram Bot and the Raspberry Pi.
- Users can send specific commands to trigger actions on the Raspberry Pi.
- Implement error handling in the script to manage cases where messages or commands are not properly received or executed.

- Ensure that images are properly formatted and processed for transmission through Telegram.
- Consider adding functionalities for sending and receiving data in various formats, depending on project requirements.

2.7 SECURITY CONSIDERATIONS:

- Security is a top priority during system implementation.
- Protect the Telegram Bot API token by storing it securely and not exposing it in the script's code.
- Implement user authentication for command access to prevent unauthorized control.
- Limit access to critical functions to trusted users or specific user groups.
- Regularly update your Raspberry Pi's operating system and software components to patch security vulnerabilities.
- Implement encryption and secure communication protocols to protect data transmitted between the Raspberry Pi and the Telegram Bot.

Chapter 3

RESULT OBTAINED AND ANALYSIS

3.1 CONNECTION OF LEDS AND PYTHON CODE DEVELOPMENT:

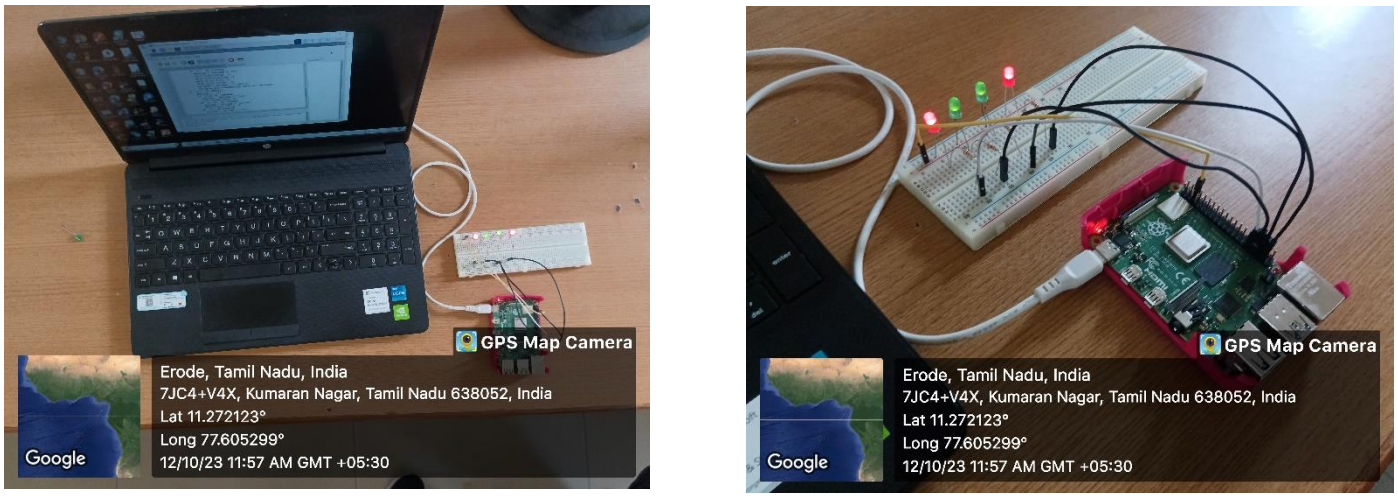

Fig:02 CONNECTION OF LEDS AND PYTHON CODE DEVELOPMENT

In this segment, we provide a detailed account of the system's implementation. This includes the physical connection of LEDs to their respective pins on the Raspberry Pi. Additionally, we cover the development of Python code in the Thonny compiler to control these LEDs. Below is a snippet of Python code utilized for this purpose.

3.2 PYTHON CODE:

```
import time, datetime
import RPi.GPIO as GPIO
import telepot
from telepot.loop import MessageLoop
green = 6
yellow = 13
red = 19
blue = 26
now = datetime.datetime.now()
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setwarnings(False)
##LED Blue
GPIO.setup(blue, GPIO.OUT)
GPIO.output(blue, 0) #Off initially
#LED Yellow
GPIO.setup(yellow, GPIO.OUT)
GPIO.output(yellow, 0) #Off initially
#LED Red
GPIO.setup(red, GPIO.OUT)
GPIO.output(red, 0) #Off initially
#LED green
GPIO.setup(green, GPIO.OUT)
GPIO.output(green, 0) #Off initially
def action(msg):
  chat_id = msg['chat']['id']
  command = msg['text']
  print ('Received: %s' % command)
  if 'on' in command:
    message = "on"
  if 'blue' in command:
    message = message + "blue "
    GPIO.output(blue, 1)
  if 'yellow' in command:
    message = message + "yellow "
    GPIO.output(yellow, 1)
  if 'red' in command:
    message = message + "red "
    GPIO.output(red, 1)
  if 'green' in command:
    message = message + "green "
    GPIO.output(green, 1)
  if 'all' in command:
    message = message + "all "
    GPIO.output(blue, 1)
    GPIO.output(yellow, 1)
    GPIO.output(red, 1)
    GPIO.output(green, 1)
```

```
  message = message + "light(s)"
  telegram_bot.sendMessage (chat_id, message)
 if 'off' in command:
  message = "off "
 if 'blue' in command:
  message = message + "blue "
  GPIO.output(blue, 0)
 if 'yellow' in command:
  message = message + "yellow "
  GPIO.output(yellow, 0)
 if 'red' in command:
  message = message + "red "
  GPIO.output(red, 0)
 if 'green' in command:
  message = message + "green "
  GPIO.output(green, 0)
 if 'all' in command:
  message = message + "all "
  GPIO.output(blue, 0)
  GPIO.output(yellow, 0)
  GPIO.output(red, 0)
  GPIO.output(green, 0)
 message = message + "light(s)"
 telegram_bot.sendMessage (chat_id, message)
 telegram_bot = telepot.Bot('6213189098:AAHrmPwIDNlF7UIojjcoe3KAezziVNCXUAo')
 print (telegram_bot.getMe())
 MessageLoop(telegram_bot, action).run_as_thread()
 print ('Up and Running....')
 while 1:
       time.sleep(10)
```

3.3 FUNCTIONALITY, RELIABILITY, AND SECURITY:

- This part offers insights into the functionality, reliability, and security aspects of the remote Raspberry Pi control system. It examines how well the system performs its intended functions, its ability to operate consistently over time, and the measures taken to ensure the security of the system.

3.4 USER EXPERIENCES AND CHALLENGES:

- We discuss user experiences and the challenges encountered during the implementation. This provides a practical view of real-world usage and insights into user interactions with the system.

3.5 Performance Analysis:

- An in-depth analysis evaluates the system's performance, assessing its responsiveness and effectiveness in real-time control. We measure how well the system performs its tasks and responds to user commands.

3.6 Potential Applications:

- This section explores potential applications of the system in various use cases, such as IoT, home automation, and industrial control. It identifies areas where the system can be beneficial and highlights its adaptability.

3.7 Evaluations and Success:

- Evaluations will highlight the success of the implementation against the project's objectives and expectations, offering a well-rounded perspective on the system's functionality and its ability to meet predefined goals.

Chapter 4
CONCLUSION

In conclusion, this comprehensive tutorial offers a transformative solution that empowers users to remotely control their Raspberry Pi devices from anywhere around the globe. It underscores the versatility and accessibility of Telegram for a wide range of applications, including IoT, home automation, and industrial control. The integration of the Raspberry Pi and Telegram has opened up previously unimaginable possibilities in the realm of remote device management and automation. This report concludes by emphasizing the system's global reach and by discussing its potential for future developments and applications, inspiring users to continue exploring and innovating in this dynamic field, where technology knows no bounds, and the power of remote control is at their fingertips.