Universidad Politécnica Salesiana
Ingeniería en Electrónica y Automatización.

Integrantes Chipugsi Anthony
Oibe Cinthya

Práctica: #3
Tema: Clases en Raspy.
Fecha: 05/11/2025

05-11-25

Práctica #3
Trabajo en Clase Herencia
Fecha: 10/11/2025

Trabajo Robot herencia + Raspy Pi
Fecha: 13/11/2025

RECUPERACIÓN
Trabajo API con Telegram (Led ON - Led OFF)
Fecha: 21/11/2025

PRACTICA #5

RECUPERACIÓN
Trabajo CLASS ROBOT + API + ARCHIVO + TELEGRAM PRÁCTICA #6
Fecha: 21/11/2025

RECUPERACIÓN
Robot LED + BUTTON + DHT11
Fecha: 21/11/2025.

Fecha: 19/10/2025

Control de Raspberry PI con Telegram

Práctica 5

Anthony Fabricio Chipugsi Pilicita
achipugsip@est.ups.edu.ec
Cinthya Aracelly Orbe Muñoz
corbem@est.ups.edu.ec

I. Introducción

El control remoto de dispositivos físicos mediante aplicaciones de mensajería es una funcionalidad clave en proyectos de Internet de las Cosas (IoT). Telegram, gracias a su API abierta, permite la creación de bots capaces de recibir comandos y enviar respuestas en tiempo real [1]. En esta práctica se desarrolló un bot en Telegram para controlar un LED y obtener lecturas de temperatura y humedad desde un sensor DHT11 conectado a una Raspberry Pi. El objetivo fue demostrar la integración entre hardware, software y servicios en la nube, estableciendo las bases para aplicaciones de domótica y sistemas inteligentes.

II. Metodología

- **Configuración del entorno**

  - Se creó un bot en Telegram mediante **@BotFather**, obteniendo el token de autenticación [1].
  - Se instalaron las librerías necesarias en la Raspberry Pi:

    pip install telepot Adafruit_DHT

- **Conexión del hardware**

  - Se conectó un LED al pin **GPIO 17 (BCM)** con resistencia limitadora.
  - Se conectó el sensor **DHT11** al pin **GPIO 4 (BCM)** para la lectura de temperatura y humedad [3].

- **Programación en Python**
  - Se utilizó la librería **telepot** para recibir comandos desde Telegram.
  - Se implementaron los comandos 'ON' y 'OFF' para controlar el LED.
  - Se añadió el comando 'status' para enviar datos de temperatura y humedad.
  - El bot se ejecuta en segundo plano mediante MessageLoop, y se mantiene activo con un bucle **while.**

```
 GNU nano 8.4
import time, datetime
import RPi.GPIO as GPIO
import telepot
from telepot.loop import MessageLoop

led = 18
now = datetime.datetime.now()
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

 #LED
GPIO.setup(led, GPIO.OUT)
GPIO.output(led, 0) #Off initially

def action(msg):
        chat_id = msg['chat']['id']
        command = msg['text']
        command=command.upper()
        print ('Received: %s' % command)

        if 'ON' in command:
                message = "Turned on "

                if 'LED' in command:
                        message = message + "led"
                        GPIO.output(led, 1)
                        telegram_bot.sendMessage (chat_id, message)

        if 'OFF' in command:
                message = "Turned off "

                if 'LED' in command:
                        message = message + "led "
                        GPIO.output(led, 0)

                telegram_bot.sendMessage (chat_id, message)

#telegram_bot = telepot.Bot('8361411568:AAFnAMhY1-_ese9QSv_7UiLEfv9EVkSa2R0')
telegram_bot = telepot.Bot('8209470010:AAGeoz03Pd_Mvep5hpvG9N2B2yWAArUnCsQ')
print (telegram_bot.getMe())

MessageLoop(telegram_bot, action).run_as_thread()

print ('Up and Running....')

while 1:

        time.sleep(10)
```

Fig. 1. Código implementado

III. Resultados

- El bot respondió correctamente a los comandos enviados desde Telegram.
- El LED se encendió y apagó según las instrucciones 'ON' y 'OFF'.
- El sistema se mantuvo activo en segundo plano, permitiendo interacción continua.
- Se observó que el uso de telepot permite una estructura más directa que telebot, aunque con menor soporte para comandos decorados.

Fecha: 19/10/2025

Fig. 2. Implementacion de telegram con Raspberry

IV. Discusión

La práctica demostró la viabilidad de integrar la Raspberry Pi con servicios de mensajería en la nube, utilizando la API de Telegram como interfaz de control [1]. El uso de la librería telepot permitió una implementación sencilla y funcional del bot, aunque con menor flexibilidad que otras librerías como telebot [2]. La estructura basada en MessageLoop facilita la ejecución en segundo plano, ideal para sistemas embebidos. La integración con el sensor DHT11, aunque no incluida en este fragmento, puede extenderse fácilmente utilizando Adafruit_DHT.read_retry() [3].

V. Conclusiones

- Se logró implementar un bot en Telegram capaz de controlar un LED desde un dispositivo móvil.
- La práctica permitió comprender cómo integrar hardware con servicios en la nube mediante Python.
- El uso de telepot resultó efectivo para proyectos simples, aunque se recomienda telebot para estructuras más complejas.
- La combinación de Raspberry Pi y Telegram constituye una solución práctica y escalable para proyectos IoT.

VI. Referencias

[1] Telegram, “Bots: An introduction for developers,” Telegram Documentation, 2025. [Online]. Available: https://core.telegram.org/bots

[2] Telepot, “Python Telegram Bot API,” GitHub Repository, 2025. [Online]. Available: https://github.com/nickoala/telepot

[3] Adafruit, “DHT Humidity Sensing on Raspberry Pi,” Adafruit Learning System, 2025. [Online]. Available: https://learn.adafruit.com/dht

VII. Anexos

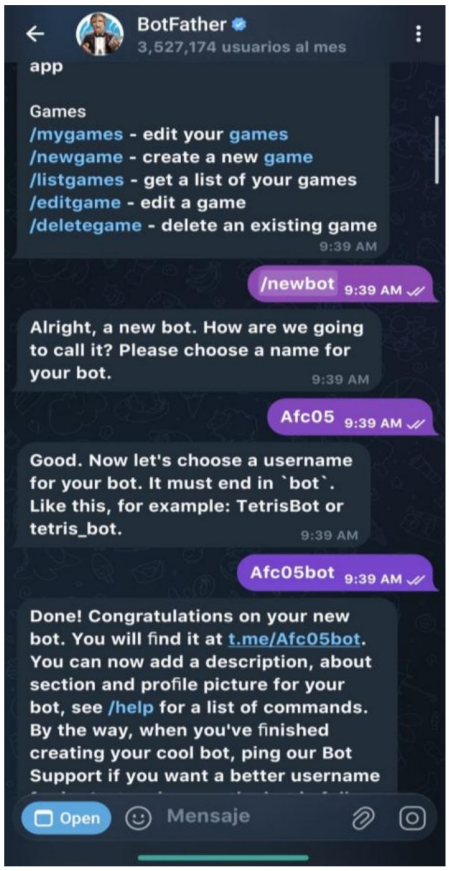

Fig. 3. Creacion del bot con @BotFather