



PRÁCTICA 4 “MANEJO REMOTO DE RASPBERRY PI MEDIANTE BOTS DE TELEGRAM”

Objetivo: Manejo y monitoreo remoto de Raspberry Pi y sus puertos GPIO mediante la programación y aplicación de Bots de Telegram

Introducción.

Una de las características más importantes de los sistemas embebidos en tiempo real es que brinden la posibilidad al usuario de obtener información del medio en el que se encuentra cuando éste lo requiera; el manejo remoto de los sistemas embebidos cobra gran relevancia cuando este se encuentra en un ambiente hostil para el ser humano o alejado a gran distancia de los usuarios para los que fue diseñado. Por lo anteriormente señalado, en esta práctica se pretende que la creación de Bots a través de la API de Telegram, permita a los usuarios y diseñadores de Sistemas embebidos ejecutar instrucciones en su tarjeta de desarrollo (para esta asignatura Raspberry Pi) mediante algún dispositivo inteligente con conexión a internet.

En un contexto general, podemos definir a un Bot como un programa autónomo e interactivo que son creados para realizar un propósito específico, el cual puede ser controlado por secuencias de texto o inteligencia artificial. En el caso particular de la API para desarrollar Bots de Telegram, es compatible para la mayoría de los lenguajes de programación de alto nivel, por lo que puede ser fácilmente programado en el lenguaje de programación predilecto; pero para la Raspberry Pi se utiliza con mayor frecuencia Python.

Cada Bot, es identificado mediante un token único que nos va a permitir comunicarnos con nuestra tarjeta de desarrollo mediante Telegram, por lo que, el desarrollo de esta práctica nos permitirá desarrollar sistemas embebidos que realicen diversas acciones mediante la utilización de hardware diverso y sensores de forma remota y la posibilidad de realizarlas a petición del usuario o de forma automática.

Duración.

1 semana



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
FUNDAMENTOS DE SISTEMAS EMBEBIDOS



Material.

- Raspberry Pi
- Dispositivo inteligente (celular, tablet, etc.)
- 1 LED
- 1 resistencia 330 Ω
- 2 cables jumpers macho-hembra
- 1 sensor ultrasónico (puede ser reemplazado por fotoresistencia, sensor PIR o inclusive push button)

Desarrollo.

Antes de iniciar con el desarrollo y la programación de Bots en la API de Telegram, es necesario realizar algunos pasos de instalación e importación de bibliotecas, tanto en la Raspberry Pi como en el dispositivo inteligente con el que se comunicará la Tarjeta de desarrollo y Telegram. Por lo que se deben realizar los pasos que se enlistan a continuación.

Instalación en Raspberry

- `sudo apt-get update && sudo apt-get upgrade -y`
- `sudo apt-get install python-setuptools`
- `sudo apt-get install python-pip`
- `sudo pip install pyTelegramBotAPI`
- `sudo apt-get install sysstat`
- `sudo chown -R pi:pi /home/pi/pyTelegramApi`
- `cd /home/pi/pyTelegramBotApi`
- `sudo python setup.py install`

Instalación en Dispositivo Inteligente

- Instalar Telegram
- Buscar @BotFather.
- Escribir el comando /newbot.
- Ingresar el alias o nombre mostrado que será asignado al bot.
- Ingresar un nombre de usuario para el bot.
- Guardar el token generado.
- Buscar el bot creado por el nombre de usuario.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
FUNDAMENTOS DE SISTEMAS EMBEBIDOS

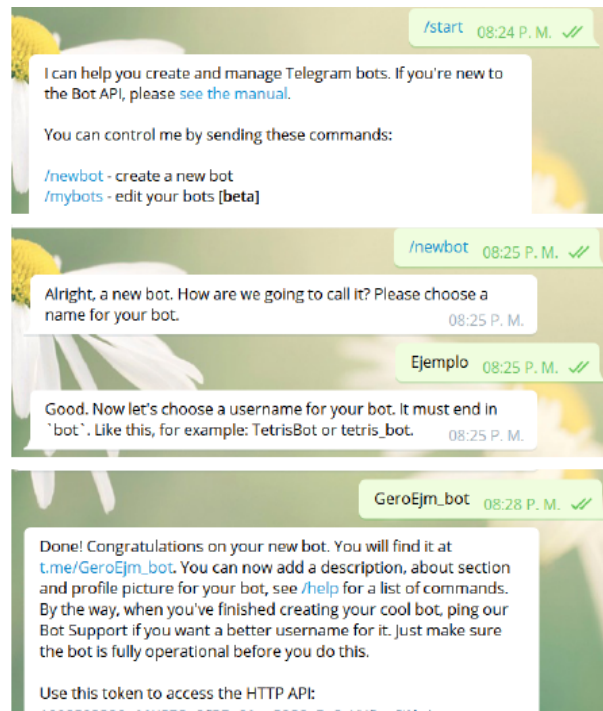


Figura 1. Captura de pantalla de la creación de un nuevo Bot en Telegram

Adicionalmente a los pasos anteriores, es necesario realizar una configuración utilizando una terminal en la Raspberry Pi, para poder introducir el token único creado por nuestro @BotFather en Telegram y que fue asignado al Bot creado para esta práctica. Cabe recordar que, sin este paso, nuestra Raspberry y nuestro Dispositivo inteligente no podrán comunicarse, por lo que se deben seguir las instrucciones que se indican a continuación:

Configuración en Raspberry

- Escribir el comando `cd /home/pi/pyTelegramBotApi/examples` para ingresar al directorio que contiene ejemplos de la biblioteca descargada
- Abrir el archivo `echo_bot.py`.
- Sustituir el valor de la variable `API_TOKEN` con el Token generado al crear el bot en el chat de BotFather dentro de la aplicación de Telegram.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
FUNDAMENTOS DE SISTEMAS EMBEBIDOS



Una vez realizado lo anterior, contamos con lo necesario para iniciar la programación de nuestros Bots mediante la API de Telegram. Para verificar que se ha establecido correctamente la conexión entre Telegram y nuestra Raspberry Pi, realizaremos la siguiente actividad:

Comprobación de Conectividad

- Abrir el archivo echo_bot.py.
- Modificar los valores de retorno de los comandos /help y /start que se encuentran dentro del archivo
- Ejecutar el programa con el comando python echo_bot.py .
- Escribir los comandos /help y /start en la aplicación de Telegram, dentro del chat con el bot, para que este devuelva el mensaje escrito en el punto anterior.

Nota: Si concluyó satisfactoriamente la actividad anterior, verifica con tu profesor el correcto funcionamiento.

```
echo_bot.py - Mousepad
Archivo  Editar  Búsqueda  Ver  Documento  Ayuda

#!/usr/bin/python

# This is a simple echo bot using the decorator mechanism.
# It echoes any incoming text messages.

import telebot

API_TOKEN = '1874926335:AAF15_lwZM2c6HZzX95qZymw6PF8cyuMPB4'

bot = telebot.TeleBot(API_TOKEN)

# Handle '/start' and '/help'
@bot.message_handler(commands=['help', 'start'])
def send_welcome(message):
    bot.reply_to(message, """\
Hi there, I am EchoBot.
I am here to echo your kind words back to you. Just say anything nice and I'll say the exact same!""")

# Handle all other messages with content_type 'text' (content_types defaults to ['text'])
@bot.message_handler(func=lambda message: True)
def echo_message(message):
    bot.reply_to(message, message.text)

bot.polling()
```

Figura 2. Ejemplo de archivo echo_bot.py modificado con API_TOKEN generado y mensaje de retorno comando /help y /start

Una vez verificada la conectividad entre nuestra tarjeta de desarrollo y Telegram, podemos realizar algunos ejemplos de Bots para ejecutar acciones básicas de forma remota en nuestra Raspberry Pi. Para ello te sugerimos realizar el siguiente ejemplo utilizando tu archivo echo_bot.py.

Cabe destacar que para poder encender y apagar un LED de forma directa en la Raspberry Pi, se utilizaría un código en Python como el siguiente:



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
FUNDAMENTOS DE SISTEMAS EMBEBIDOS



```
from gpiozero import LED
from time import sleep
led = LED(17)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Partiendo de lo anterior, podemos destacar algunas líneas importantes en el código siguiente, mismo que deberás colocar en tu archivo `echo_bot.py` y posteriormente ejecutarlo.

Encender y Apagar un LED usando Telegram

```
import os
from gpiozero import LED ←
import telebot

led= LED(17) ←

API_TOKEN= 'debe colocar el token que se generó para su bot'

bot=telebot.TeleBot(API_TOKEN)

@bot.message_handler(commands=['apaga'])
def turn_off(message):
    led.off() ←
    bot.reply_to(message, """\
se apago\
""")

@bot.message_handler(commands=['enciende'])
def turn_on(message):
    led.on() ←
    bot.reply_to(message, """\
se encendio\
""")
```

Con el ejemplo anterior, se pretende que logre identificar el código necesario para la creación de un Bot que ejecute comandos básicos de la librería `gpiozero` para Raspberry Pi.

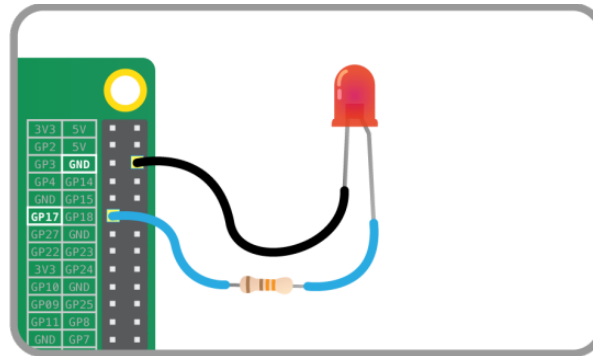


Figura 3. Ejemplo de conexión de LED para ejercicio de encendido y pagado del mismo

Ejercicio Propuesto.

- Diseñar una alarma de proximidad o movimiento utilizando un sensor ultrasónico, sensor PIR, fotoresistencia o el dispositivo que tenga disponible, que sea capaz de enviar una alerta al usuario mediante chat Bot de Telegram.

Sugerencias.

- Recuerde importar las librerías necesarias; para el caso de este ejercicio, se requiere importar la librería `DistanceSensor` y `LED` de `gpiozero` si utiliza un sensor ultrasónico; si utiliza la fotoresistencia requiere de importar la librería `LightSensor`.

Evaluación

Se comprobará que el alumno adquirió los conocimientos esperados mediante la siguiente lista de verificación.

- ✓ El alumno logró enviar un mensaje al usuario mediante la modificación de la función `/help` y `/start` del archivo `echo_bot.py`
- ✓ El alumno logró encender y apagar un LED mediante instrucciones en chat Bot de Telegram.
- ✓ El alumno logró implementar la alarma de proximidad, ya que, al activarse el sensor ultrasónico, se envía un mensaje de alerta al usuario mediante chat Bot de Telegram.
- ✓ El alumno comenta las líneas de código relevantes para cada ejercicio.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
FUNDAMENTOS DE SISTEMAS EMBEBIDOS



Bibliografía

García, Manuel J., Bot de Telegram que ejecute aventuras conversacionales, Universidad de Jaén, Recuperado el 24 de agosto de 2021 de: http://tauja.ujaen.es/bitstream/10953.1/5944/1/TFG_Garcia-Quesada_Manuel-Jesus.pdf

¿Qué es un robot? Definición de robot. Cloudflare. Recuperado el 28 de agosto de 2021 de: <https://www.cloudflare.com/es-la/learning/bots/what-is-a-bot/>

Lorenzo, P.(2016). Controla tu Raspberry Pi mediante Telegram. fwhibit. Recuperado el 28 de agosto de 2021 de: <https://fwhibit.es/controla-tu-raspberry-pi-mediante-telegram>

Nutall, B. (2017). Gpiozero Documentation. Raspberry Pi Foundation, Dave Jones. Recuperado el 29 de agosto del 2021 de <https://gpiozero.readthedocs.io/en/stable/>