

Instituto Tecnológico de Costa Rica  
Área Académica de Ingeniería en Computadores

Programa de Licenciatura de Ingeniería en Computadores



## **Trabajo Final de Graducación**

### **Manual de Uso de Interfaces**

Instrucciones de uso de diversas interfaces modulares

## **Desarrollo de un UAV modular de costo reducido con propósitos educativos y de investigación**

Elaborado por

Jose Ignacio Granados Marín

Ingeniería en Computadores

No. Carné: 2018319698

Supervisado por: MSc. Luis Alberto Chavarría Zamora

# Índice

<b>I</b>	<b>Cámara</b>	<b>4</b>
<b>1.</b>	<b>Interfaz MIPI CSI-2</b>	<b>4</b>
1.1.	Requerimientos . . . . .	4
1.2.	Conexión del módulo de cámara . . . . .	4
1.3.	Control del módulo de cámara a través de la línea de comandos . . .	5
1.3.1.	Captura de fotografía . . . . .	5
1.3.2.	Grabación de video . . . . .	5
1.4.	Control del módulo de cámara a través de código Python . . . . .	6
1.4.1.	Captura de fotografías fijas a través de código Python . . . . .	7
1.4.2.	Grabación de videos a través de código Python . . . . .	9
<b>2.</b>	<b>Interfaz USB</b>	<b>9</b>
2.1.	Requerimientos . . . . .	9
2.2.	Conexión del módulo de cámara . . . . .	10
2.3.	Control del módulo de cámara a través de la línea de comandos . . .	10
2.3.1.	Uso de la biblioteca fswebcam . . . . .	10
2.3.2.	Uso de la biblioteca ffmpeg . . . . .	11
2.4.	Control del módulo de cámara a través de código Python . . . . .	11
<b>3.</b>	<b>GigE</b>	<b>12</b>
3.1.	Requerimientos . . . . .	12
3.2.	Conexión del módulo de cámara . . . . .	12
3.3.	Control del módulo de cámara a través de la línea de comandos . . .	15
<b>II</b>	<b>GPS</b>	<b>16</b>
<b>4.</b>	<b>Interfaz GPIO</b>	<b>16</b>
4.1.	Requerimientos . . . . .	16
4.2.	Conexión del módulo GPS . . . . .	16

4.3. Control del módulo GPS a través de la línea de comandos . . . . .	19
4.4. Control del módulo GPS a través de código Python . . . . .	20
<b>5. Interfaz USB</b>	<b>21</b>
5.1. Requerimientos . . . . .	21
5.2. Conexión del módulo GPS . . . . .	21
5.3. Control del módulo GPS a través de la línea de comandos . . . . .	24
5.4. Control del módulo GPS a través de código Python . . . . .	24
 <b>III Micrófono</b>	 <b>25</b>
<b>6. Interfaz GPIO</b>	<b>25</b>
6.1. Requerimientos . . . . .	25
6.2. Conexión del micrófono . . . . .	25
6.3. Control del micrófono a través de código Python . . . . .	27
<b>7. Interfaz USB</b>	<b>28</b>
7.1. Requerimientos . . . . .	28
7.2. Conexión del micrófono . . . . .	28
7.3. Control del micrófono a través de la línea de comandos . . . . .	28
7.4. Control del micrófono a través de código Python . . . . .	29
 <b>IV Sensor IR</b>	 <b>31</b>
<b>8. Interfaz GPIO</b>	<b>32</b>
8.1. Requerimientos . . . . .	32
8.2. Conexión del sensor IR . . . . .	32
8.3. Control del sensor IR a través de código Python . . . . .	33
 <b>V Repositorio de Github</b>	 <b>34</b>

## Parte I

# Cámara

## 1. Interfaz MIPI CSI-2

### 1.1. Requerimientos

- Raspberry Pi con un puerto de módulo de cámara.
- Módulo de cámara Raspberry Pi.

### 1.2. Conexión del módulo de cámara

- Verifique que la Raspberry Pi esté apagada.
- Localice el puerto del módulo de cámara.
- Levante el clip de plástico del puerto.
- Inserte el cable de cinta del módulo de cámara.
- Empuje el clip de plástico de nuevo en su lugar.
- Encienda su Raspberry Pi.
- Habilite el módulo de cámara de la Raspberry Pi:

```
sudo raspi-config
```

Interface Options - Camera

- Reinicie su Raspberry Pi.

```
sudo reboot
```

## 1.3. Control del módulo de cámara a través de la línea de comandos

### 1.3.1. Captura de fotografía

- Abra una ventana de terminal.
- Escriba el siguiente comando para tomar una imagen:

```
raspistill -o image_MIPi_CSI2_CommandLine.jpg
```

Si desea modificar el tamaño y aspecto de la imagen, puede utilizar el comando “raspistill”, por medio del uso de “-h” y “-w” para modificar el alto y ancho respectivamente. Por ejemplo:

```
raspistill -o image_MIPi_CSI2_CommandLine_resize.jpg  
-w 352 -h 288
```

### 1.3.2. Grabación de video

- Escriba el siguiente comando para grabar un video y guardarlo en el escritorio:

```
raspivid -o video_MIPi_CSI2_CommandLine.h264
```

Para reproducir el archivo de video, haga doble clic en el icono del archivo en el escritorio para abrirlo en VLC Media Player.

## 1.4. Control del módulo de cámara a través de código Python

- Abra una ventana de terminal.
- Instale Python 3.

```
python3 --version  
sudo apt-get update  
sudo apt-get install python3
```

- Instale la biblioteca picamera.

```
sudo apt-get install python-picamera python3-picamera
```

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como seeCamera.py
- Introduzca el siguiente código:

```
1 from picamera import PiCamera  
2 from time import sleep  
3  
4 camera = PiCamera()  
5  
6 camera.start_preview()  
7 sleep(5)  
8 camera.stop_preview()
```

- Guarde y ejecute el programa.

```
python3 seeCamera.py
```

La vista previa de la cámara sólo funcionará si la Raspberry Pi se encuentra conectada a un monitor.

En caso de que la fotografía se encuentre al revés, puede girarla 180 grados con el siguiente código:

```
1 from picamera import PiCamera
2 from time import sleep
3
4 camera = PiCamera()
5 camera.rotation = 180
6
7 camera.start_preview()
8 sleep(5)
9 camera.stop_preview()
```

#### 1.4.1. Captura de fotografías fijas a través de código Python

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como takePicture.py.
- Introduzca el siguiente código:

```

1 from picamera import PiCamera
2 from time import sleep
3
4 camera = PiCamera()
5
6 camera.start_preview()
7 sleep(5)
8 camera.capture("image_MIPi_CSI2_Python.jpg")
9 camera.stop_preview()

```

Es importante esperar, al menos, dos segundos antes de capturar una imagen, ya que esto le dará tiempo al sensor de la cámara para detectar los niveles de luz.

- Guarde y ejecute el programa.

```
python3 takePicture.py
```

En caso de que desee tomar múltiples fotografías, puede utilizar el siguiente código:

```

1 from picamera import PiCamera
2 from time import sleep
3
4 camera = PiCamera()
5
6 camera.start_preview()
7 for i in range(5):
8     sleep(5)
9     camera.capture("image%s.jpg" % i)
10 camera.stop_preview()

```



### 1.4.2. Grabación de videos a través de código Python

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como recordVideo.py.
- Introduzca el siguiente código:

```
1 from picamera import PiCamera
2 from time import sleep
3
4 camera = PiCamera()
5
6 camera.start_preview()
7 camera.start_recording("video_MIPi_CSI2_Python.h264")
8 sleep(5)
9 camera.stop_recording()
10 camera.stop_preview()
```

- Guarde y ejecute el programa.

```
python3 recordVideo.py
```

## 2. Interfaz USB

### 2.1. Requerimientos

- Raspberry Pi con un puerto USB.
- Módulo de cámara USB.

## 2.2. Conexión del módulo de cámara

- Verifique que la Raspberry Pi esté encendida.
- Localice el puerto USB.
- Conecte el módulo de cámara en el puerto USB.
- Verifique la Raspberry Pi reconoció correctamente el módulo de cámara a través del comando:

```
lsusb
```

## 2.3. Control del módulo de cámara a través de la línea de comandos

### 2.3.1. Uso de la biblioteca fswebcam

- Instale la biblioteca fswebcam.

```
sudo apt install fswebcam
```

- Abra una ventana de terminal.
- Escriba el siguiente comando para tomar fotografía con una resolución determinada y ocultar el banner estándar:

```
fswebcam -r 352x288 --no-banner -s brightness=75%  
image_USB_fswebcam.jpg
```

Para encontrar las resoluciones de cámara compatibles con la Raspberry Pi actual, puede utilizar el siguiente comando:

```
fswebcam v4l2-ctl --list-formats-ext
```

### 2.3.2. Uso de la biblioteca ffmpeg

- Instale la biblioteca ffmpeg.

```
sudo apt install ffmpeg
```

- Abra una ventana de terminal.
- Escriba el siguiente comando para tomar fotografía:

```
ffmpeg -video_size 352x288 -i /dev/video0 -frames 1  
image_USB_ffmpeg.jpg
```

## 2.4. Control del módulo de cámara a través de código Python

- Instale la biblioteca opencv.

```
sudo apt install python3-opencv
```

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como cameraUSB.py.
- Introduzca el siguiente código:

```
1 import cv2
2
3 cam = cv2.VideoCapture(0)
4
5 ret, image = cam.read()
6
7 cv2.imwrite("image_USB_Python.jpg", image)
8 cam.release()
9 cv2.destroyAllWindows()
```

- Guarde el programa.

```
python3 cameraUSB.py
```

## 3. GigE

### 3.1. Requerimientos

- Raspberry Pi con un puerto Ethernet.
- Módulo de cámara Ethernet.

### 3.2. Conexión del módulo de cámara

- Conecte la Raspberry Pi a la red más cercana.
- Deshabilite el Screen Blanking.

```
sudo raspi-config
```

Display Options - Screen Blanking

- Actualice el sistema.

```
sudo apt update && sudo apt dist-upgrade -y
```

- Instale omxplayer y dnsmasq.

```
sudo apt install omxplayer dnsmasq
```

- Desconecte la Raspberry Pi de la red más cercana.
- Configure la dirección IP estática en eth0.

```
sudo nano /etc/dhcpd.conf
```

Descomente las siguientes líneas:

```
interface eth0
static ip_address=192.168.0.10/24
```

Presione Ctrl + O para guardar.

Presione Ctrl + X para salir.

- Reinicie el servicio dhcpd.

```
sudo systemctl restart dhcpd
```

- Revise la dirección IP actual.

```
ip a
```

- Mueva la configuración actual de dnsmasq.

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

- Configure el dnsmasq.

```
sudo nano /etc/dnsmasq.conf
```

Agregue las siguientes líneas:

```
interface=eth0
dhcp-range=192.168.0.11,192.168.0.254,255.255.255.0,24h
address=/#/192.168.0.10
dhcp-leasefile=/var/lib/misc/dnsmasq.leases
```

Presione Ctrl + O para guardar.

Presione Ctrl + X para salir.

- Habilite e inicie el servicio dnsmasq.

```
sudo systemctl daemon-reload
sudo systemctl habilitar dnsmasq
sudo systemctl start dnsmasq
```

### 3.3. Control del módulo de cámara a través de la línea de comandos

- Supervise los DHCP Leases.

```
tail -f /var/lib/misc/dnsmasq.leases
```

Si no logra supervisar los DHCP Leases, reinicie el servicio dnsmasq.

```
sudo systemctl restart dnsmasq
```

- Localice el puerto Ethernet.
- Conecte el módulo de cámara en el puerto Ethernet.
- Copie la dirección mac y la dirección ip que logró supervisar.  
Presione Ctrl + C para salir.
- Agregue la entrada de host a dnsmasq.

```
sudo nano /etc/dnsmasq.conf
```

Agregue y modifique la siguiente línea con la dirección MAC y la dirección IP previamente copiadas.

```
dhcp-host=00:00:00:00:00:00,192.168.0.229
```

Presione Ctrl + O para guardar.

Presione Ctrl + X para salir.

- Reinicie el servicio dnsmasq.

```
sudo systemctl daemon-reload  
sudo systemctl restart dnsmasq
```

- Observe la cámara IP.

```
omxplayer "rtsp://username:password@ip_of_camera:554cam/  
realmonitor?channel=1&subtype=0"
```

Si desea, puede modificar el nombre de usuario (username) y la contraseña (password) a su gusto, antes de ejecutar el comando anterior.

## Parte II

# GPS

## 4. Interfaz GPIO

### 4.1. Requerimientos

- Raspberry Pi con un puerto GPIO.
- Módulo GPS.

### 4.2. Conexión del módulo GPS

- Localice los puertos VCC, TX y GND del módulo GPS.
- Localice el puerto GPIO de la Raspberry Pi.



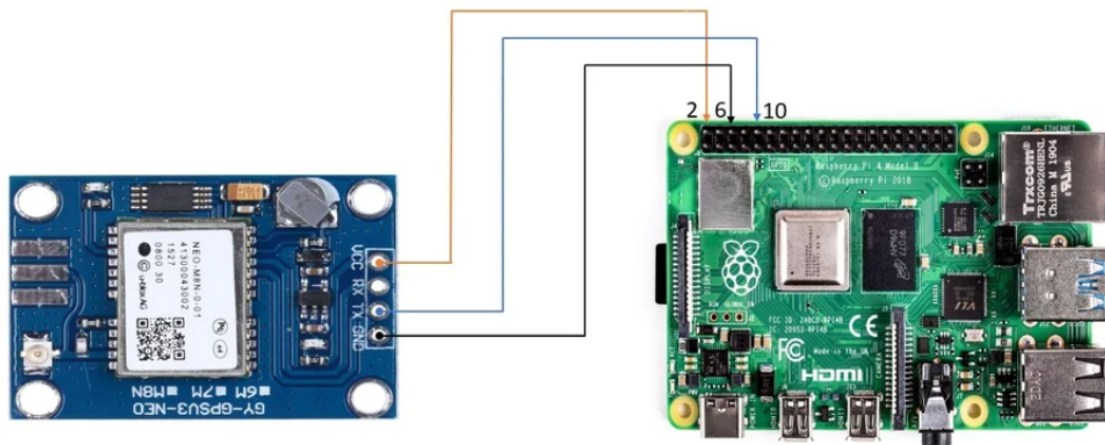


Figura 1: Conexión de la Raspberry Pi con el módulo GPS a través de la interfaz GPIO.

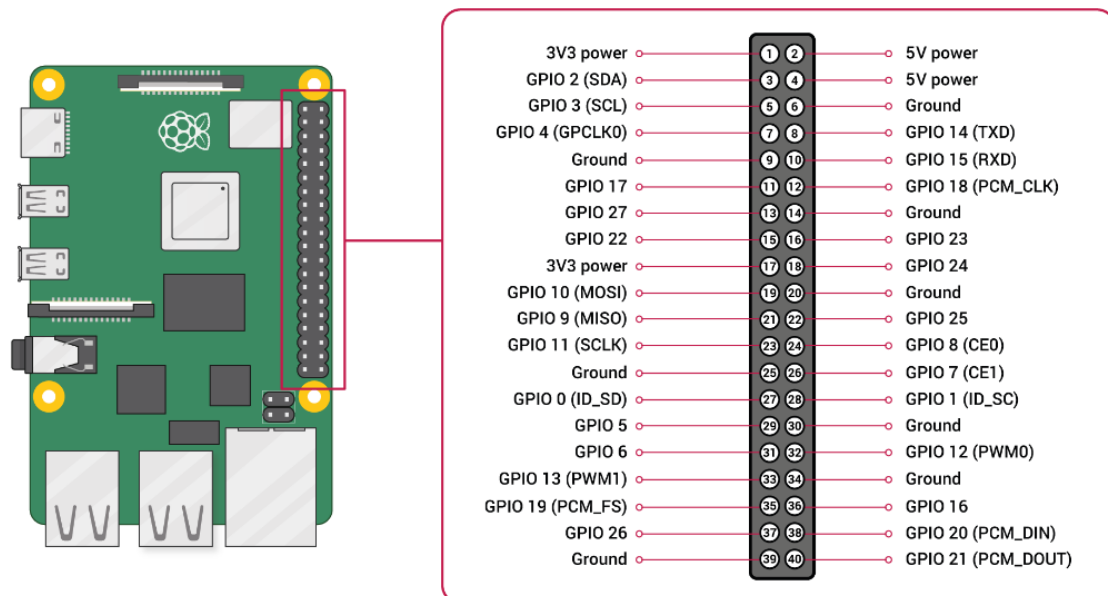


Figura 2: Diagrama de pines GPIO de la Raspberry Pi.

- Realice la siguiente conexión entre los diferentes dispositivos:
- Edite el archivo `/boot/config.txt`.

```
sudo nano /boot/config.txt
```

Agregue las siguientes líneas al final del archivo:

```
dtparam=spi=on
dtoverlay=pi3-disable-bt
core_freq=250
enable_uart=1
force_turbo=1
```

Presione Ctrl + O para guardar.

Presione Ctrl + X para salir.

- Realice una copia del archivo cmdline.txt, para usos futuros.

```
sudo cp /boot/cmdline.txt /boot/cmdline_backup.txt
```

- Edite el archivo cmdline.txt.

```
sudo nano /boot/cmdline.txt
```

Reemplace el contenido del archivo con la siguiente línea:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes
rootwait quiet splash plymouth.ignore-serial-consoles
```

Presione Ctrl + O para guardar.

Presione Ctrl + X para salir.

- Reinicie la Raspberry Pi.

```
sudo reboot
```

- Deshabilite el servicio Raspberry Pi Serial Getty.

Si en su salida, Serial0 está vinculado con ttyAMA0, ejecute los siguientes comandos:

```
sudo systemctl stop serial-getty@ttyAMA0.service
sudo systemctl disable serial-getty@ttyAMA0.service
```

Si en su salida, Serial0 está vinculado con ttys0, ejecute los siguientes comandos:

```
sudo systemctl stop serial-getty@ttys0.service
sudo systemctl disable serial-getty@ttys0.service
```

### 4.3. Control del módulo GPS a través de la línea de comandos

- Habilite el ttyso.

```
sudo systemctl enable serial-getty@ttys0.service
```

- Instale la biblioteca Minicom.

```
sudo apt-get install minicom
```

- Instale la biblioteca Pynmea2.

```
sudo pip install pynmea2
```

- Pruebe el módulo GPS.

```
sudo cat /dev/ttyAMA0
```

#### 4.4. Control del módulo GPS a través de código Python

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como gpsGPIO.py.
- Introduzca el siguiente código:

```
1 import serial
2 import pynmea2
3
4 while True:
5
6     port = "/dev/ttyAMA0"
7     ser = serial.Serial(port, baudrate = 9600, timeout = 0.5)
8     dataout = pynmea2.NMEAStreamReader()
9     newdata = ser.readline()
10
```

```

11         if newdata[0:6] == b"$GNRMC":
12
13             newmsg = pynmea2.parse(newdata.decode())
14             lat = newmsg.latitude
15             lng = newmsg.longitude
16             gps = "Latitude = " + str(lat) + "and Longitude = " +
↵             str(lng)
17
18             print(gps)

```

- Guarde y ejecute el programa.

```
python3 gpsGPIO.py
```

## 5. Interfaz USB

### 5.1. Requerimientos

- Raspberry Pi con un puerto GPIO.
- Módulo GPS USB.

### 5.2. Conexión del módulo GPS

- Diríjase a la carpeta dev.

```
cd /dev/
```

- Verifique las conexiones seriales utilizadas por la Raspberry Pi.

```
ls
```

- Localice el puerto USB.
- Conecte el módulo GPS en el puerto USB.
- Vuelva a verificar las conexiones seriales.

```
ls
```

Deberá aparecer una nueva conexión serial que represente el módulo GPS USB conectado previamente. Por lo general, suele aparecer con el nombre de `ttyACM0`.

- Instale el paquete `gpsd`.

```
sudo apt-get install gpsd gpsd-clients gpsd-tools
```

- Detenga el servicio serial `gpsd` asociado al módulo GPS USB.

```
sudo systemctl stop gpsd.socket  
sudo systemctl disable gpsd.socket
```

- Modifique el paquete `gpsd`.

```
sudo nano /lib/systemd/system/gpsd.socket
```

Comente la siguiente línea mediante la adición del símbolo numeral al inicio de la misma.

```
ListenStream:127.0.0.1:2947
```

Descomente la siguiente línea mediante la eliminación del símbolo numeral al inicio de la misma.

```
LListenStream:0.0.0.0:2947
```

Presione Ctrl + O para guardar.

Presione Ctrl + X para salir.

- Detenga todos los procesos `gpsd` que estén siendo ejecutados.

```
sudo killall gpsd
```

- Enlace el paquete `gpsd` con el puerto serial asociado al módulo GPS USB.

```
sudo gpsd /dev/ttyACM0 -F run/gpsd.sock
```

### 5.3. Control del módulo GPS a través de la línea de comandos

- Pruebe el módulo GPS con alguno de los siguientes comandos:

```
cgps -s  
gpsmon  
xgps
```

### 5.4. Control del módulo GPS a través de código Python

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como gpsUSB.py.
- Introduzca el siguiente código:

```
1 from gps import *  
2 import time  
3  
4 gpssd = gps(mode=WATCH_ENABLE|WATCH_NEWSTYLE)  
5 print("latitude\t longitude\t\t\t\t\t time utc\t\t\t\t\t altitude\t epv\t\t\t\t\t  
↪ \t\t\t\t\t speed\t\t\t\t\t climb")  
6  
7 while True:  
8  
9     report = gpssd.next() #  
10    if report["class"] == "TPV":  
11  
12  
13  
14
```



```

15         print(getattr(report,"lat",0.0),"\t",
               ↪ getattr(report,"lon",0.0),"\t",
16                   getattr(report,"time",""),"\t",
               ↪ getattr(report,"alt","nan"),"\t",
17                   getattr(report,"epv","nan"),"\t",
               ↪ getattr(report,"ept","nan"),"\t",
18                   getattr(report,"speed","nan"),"\t",
               ↪ getattr(report,"climb","nan"),"\t")
19
20     time.sleep(1)

```

- Guarde y ejecute el programa.

```
python3 gpsUSB.py
```

## Parte III

# Micrófono

## 6. Interfaz GPIO

### 6.1. Requerimientos

- Raspberry Pi con un puerto GPIO e interfaz I2S.
- Micrófono con puerto GPIO.

### 6.2. Conexión del micrófono

- Localice los puertos VCC, GND y OUT del micrófono.
- Localice el puerto GPIO de la Raspberry Pi.

- Realice la siguiente conexión entre los diferentes dispositivos:

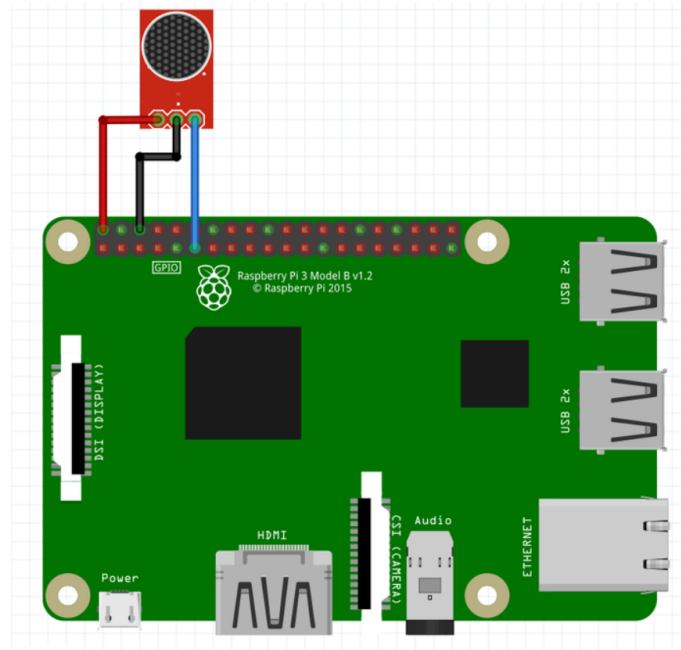


Figura 3: Conexión de la Raspberry Pi con el micrófono a través de la interfaz GPIO.

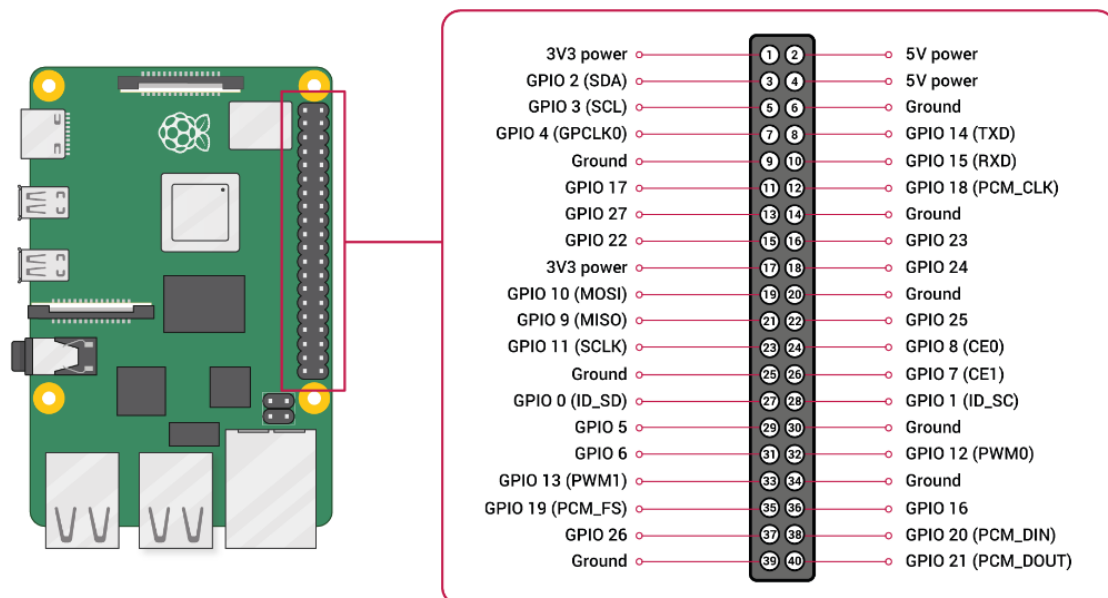


Figura 4: Diagrama de pines GPIO de la Raspberry Pi.

### 6.3. Control del micrófono a través de código Python

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como microphoneGPIO.py.
- Introduzca el siguiente código:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 channel = 17
5 GPIO.setmode(GPIO.BCM)
6 GPIO.setup(channel, GPIO.IN)
7
8 def callback(channel):
9     if GPIO.input(channel):
10         print("Sound Detected!")
11     else:
12         print("Sound Detected!")
13
14 GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)
15 GPIO.add_event_callback(channel, callback)
16
17 while True:
18     time.sleep(1)
```

- Guarde y ejecute el programa.

```
python3 microphoneGPIO.py
```

## 7. Interfaz USB

### 7.1. Requerimientos

- Raspberry Pi con un puerto USB.
- Micrófono USB.

### 7.2. Conexión del micrófono

- Localice el puerto USB.
- Conecte el micrófono en el puerto USB.
- Verifique que la Raspberry Pi reconoció correctamente el módulo de cámara a través del comando:

```
lsusb
```

### 7.3. Control del micrófono a través de la línea de comandos

- Verifique la tarjeta y anote el número del micrófono.

```
arecord -l
```

- Grabe un archivo wav.

```
arecord -D plughw:1 -r 48000 -t wav -V stereo  
audio_USB_CommandLine.wav
```

Modifique el número `plughw:`, de tal manera que coincida con el número de tarjeta anotado anteriormente.

- Acceda al archivo previamente grabado.

```
vlc test.wav
```

## 7.4. Control del micrófono a través de código Python

- Instale los paquetes libportaudio0, libportaudio2, libportaudiocpp0 y portaudio19-dev.

```
sudo apt-get install libportaudio0 libportaudio2  
libportaudiocpp0 portaudio19-dev
```

- Instale la biblioteca pyaudio.

```
sudo pip3 install pyaudio
```

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como microphoneUSBidentfier.py.
- Introduzca el siguiente código:

```
1 import pyaudio  
2  
3 p = pyaudio.PyAudio()  
4  
5 for ii in range(p.get_device_count()):  
6     print(p.get_device_info_by_index(ii).get("name"))
```

- Guarde y ejecute el programa.

```
python3 microphoneUSBidentifier.py
```

El código anterior le permitirá conocer el índice del micrófono USB conectado, ya que dicho número será necesario para poder realizar grabaciones con la biblioteca de pyaudio.

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como microphoneUSB.py.
- Introduzca el siguiente código:

```
1 import pyaudio
2 import wave
3
4 form1 = pyaudio.paInt16
5 chans = 1
6 sampRate = 44100
7 chunk = 4096
8 recordSecs = 11
9 devIndex = 1
10 wavOutputFilename = "audio_USB_Python.wav"
11
12 audio = pyaudio.PyAudio()
13
14 stream = audio.open(format = form1, rate = sampRate, channels =
    ↪  chans,
15                       input_device_index = devIndex, input = True,
16                       frames_per_buffer = chunk)
17
18 print("recording")
```

```

19
20 frames = []
21
22 for ii in range(0,int((sampRate / chunk) * recordSecs)):
23     data = stream.read(chunk)
24     frames.append(data)
25
26 print("finished recording")
27
28 stream.stop_stream()
29 stream.close()
30 audio.terminate()
31
32 wavefile = wave.open(wavOutputFilename,"wb")
33 wavefile.setnchannels(chans)
34 wavefile.setsampwidth(audio.get_sample_size(form1))
35 wavefile.setframerate(sampRate)
36 wavefile.writeframes(b''.join(frames))
37 wavefile.close()

```

- Guarde y ejecute el programa.

```
python3 microphoneUSB.py
```

Recuerde modificar el valor de la variable devIndex con el valor anotado en el paso anterior, para que el código pueda reconocer el micrófono conectado.

## Parte IV

# Sensor IR

## 8. Interfaz GPIO

### 8.1. Requerimientos

- Raspberry Pi con un puerto GPIO.
- Sensor IR con puerto GPIO.

### 8.2. Conexión del sensor IR

- Localice los puertos VCC, OUT y GND del sensor IR.
- Localice el puerto GPIO de la Raspberry Pi.
- Realice la siguiente conexión entre los diferentes dispositivos:

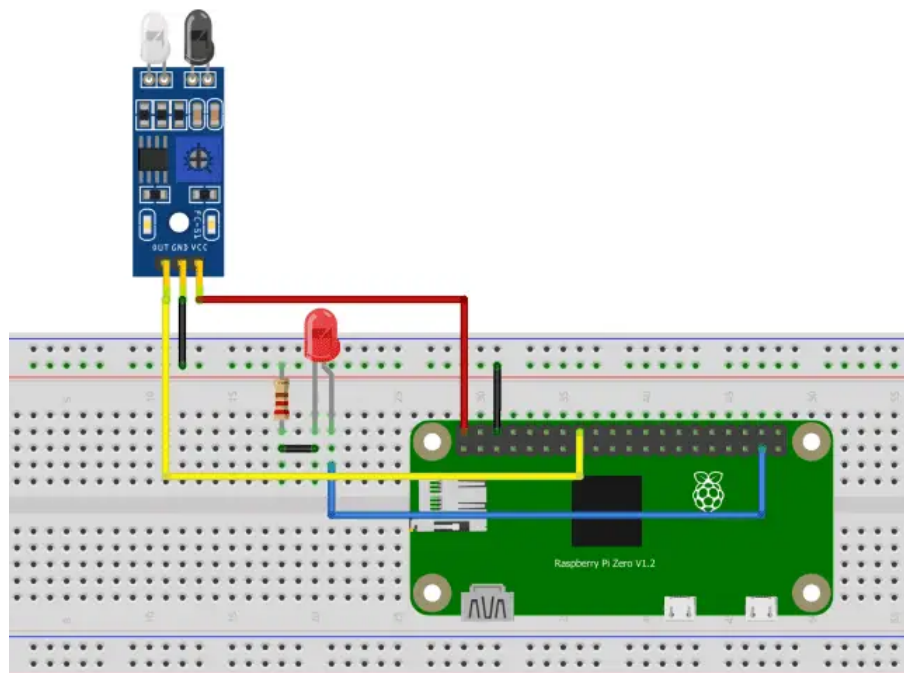


Figura 5: Conexión de la Raspberry Pi con el sensor IR a través de la interfaz GPIO.



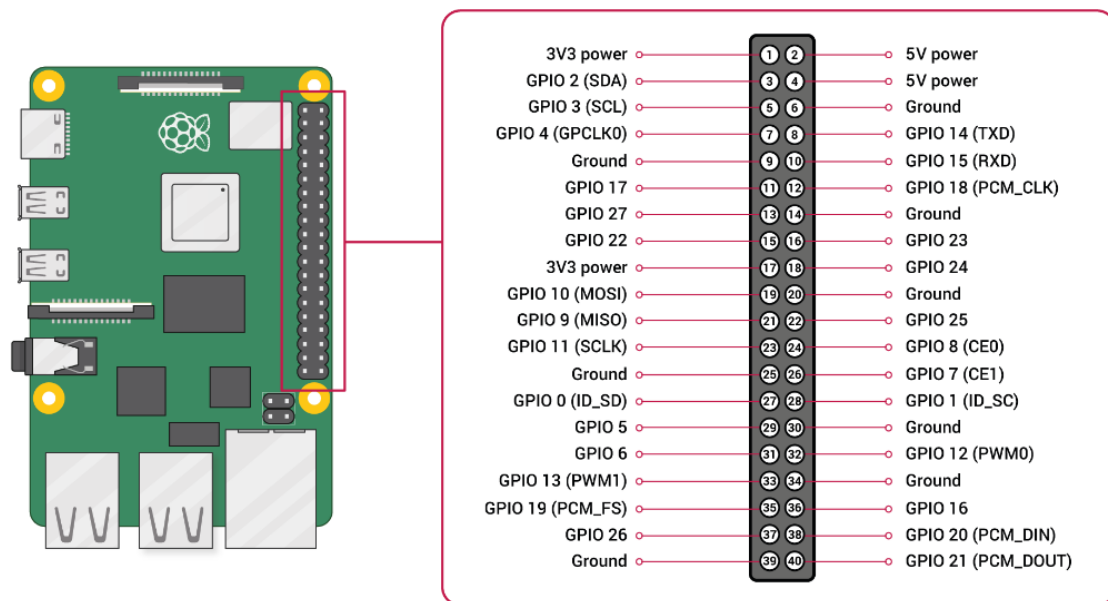


Figura 6: Diagrama de pines GPIO de la Raspberry Pi.

### 8.3. Control del sensor IR a través de código Python

- Abra un editor de Python 3.
- Cree un nuevo archivo y guárdelo como infraredGPIO.py.
- Introduzca el siguiente código:

```

1 import RPi.GPIO as GPIO
2 import time
3
4 sensorPin = 23
5 ledPin = 26
6
7 GPIO.setwarnings(False)
8 GPIO.setmode(GPIO.BCM)
9 GPIO.setup(sensorPin, GPIO.IN)
10 GPIO.setup(ledPin, GPIO.OUT)
11
12

```

```

13 try:
14
15     while True:
16
17         if GPIO.input(sensorPin):
18             GPIO.output(ledPin, False)
19             while GPIO.input(sensorPin):
20                 time.sleep(0.2)
21
22         else:
23             GPIO.output(ledPin, True)
24             print("Object Detected")
25
26 except KeyboardInterrupt:
27     GPIO.cleanup()

```

- Guarde y ejecute el programa.

```
python3 infraredGPIO.py
```

## Parte V

# Repositorio de Github

<https://github.com/NachoGranados/FinalGraduationProject.LowCostModularUAV>

## Referencias

- [1] Getting started with the Camera Module (no date) Projects.raspberrypi.org. Available at: <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/0> (Accessed: 16 August 2023).
- [2] Using USB webcams (no date) The Raspberry Pi Guide. Available at: <http://raspberrypi-guide.github.io/electronics/using-usb-webcams> (Accessed: 17 August 2023).
- [3] Nair, A. (2020) Raspberry pi tutorial on using a USB camera to display and record videos with python, Medium. Available at: <https://medium.com/prope1land/raspberry-pi-tutorial-on-using-a-usb-camera-to-display-and-record-videos-with-python-a41c6938f89f> (Accessed: 17 August 2023).
- [4] Sheaffer, R. (2021) Monitoring an IP camera using a raspberry pi (no internet), RickMakes. Available at: <https://www.rickmakes.com/monitoring-an-ip-camera-using-a-raspberry-pi-no-internet/> (Accessed: 17 August 2023).
- [5] Interfacing GPS module with Raspberry Pi (2020) Instructables. Available at: <https://www.instructables.com/Interfacing-GPS-Module-With-Raspberry-Pi/> (Accessed: 18 August 2023).
- [6] Raspberry pi documentation (no date) Raspberry Pi hardware. Available at: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (Accessed: 18 August 2023).
- [7] GPS (raspberry pi) (2018) Instructables. Available at: <https://www.instructables.com/GPS-Raspberry-Pi/> (Accessed: 18 August 2023).
- [8] Sound sensor (raspberry pi) (2019) Instructables. Available at: <https://www.instructables.com/Sound-Sensor-Raspberry-Pi/> (Accessed: 04 September 2023).

- [9] Zamir, Z. (2022) How to Use a USB Microphone with Raspberry Pi, How to use a USB microphone with Raspberry Pi. Available at: <https://linuxhint.com/use-usb-microphone-raspberry-pi/> (Accessed: 22 August 2023).
- [10] Hrisko, J. (2022) Recording audio on the Raspberry Pi with python and a USB microphone, Maker Portal. Available at: <https://makersportal.com/blog/2018/8/23/recording-audio-on-the-raspberry-pi-with-python-and-a-usb-microphone> (Accessed: 22 August 2023).
- [11] Using Infrared (IR) sensor with Raspberry Pi (2023) donskeytech.com. Available at: <https://www.donskeytech.com/using-infrared-ir-sensor-with-raspberry-pi/> (Accessed: 22 August 2023).