

[BLE] Configuración

- [Activación de Bluetooth](#)
- [Conexión de Bluetooth](#)
 - [Raspberry Server \(Servidor de Tierra\)](#)
 - [Raspberry Cliente \(Barco\)](#)
- [Comunicación de Bluetooth](#)
 - [Raspberry Servidor \(Servidor de tierra\)](#)
 - [Raspberry Cliente \(Barco\)](#)

Activación de Bluetooth

Configuración bluetooth

```
sudo apt update
sudo apt install -y bluetooth bluez bluez-tools
sudo systemctl start bluetooth
sudo systemctl enable bluetooth
```

Conexión de Bluetooth

Raspberry Server (Servidor de Tierra)

- Primero, se mira cual es la Mac de la raspberry que aloja el servidor de tierra:

```
[pi@raspberrypi:~ $ hciconfig
hci0:    Type: Primary   Bus: UART
        BD Address: B8:27:EB:7D:8C:DE  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING PSCAN
        RX bytes:78864 acl:148 sco:0 events:2614 errors:0
        TX bytes:45355 acl:147 sco:0 commands:648 errors:0
```

- Se realizan los siguientes comandos para poner el dispositivos bluetooth visible:

Configuración bluetooth

```
bluetoothctl
power on
discoverable on
pairable on
agent on
default-agent
```

- Muestra de confirmación de establecimiento de conexión con la raspberry del barco:

```
[CHG] Device E4:5F:01:E2:BB:FE Connected: yes
Request confirmation
[agent] Confirm passkey 655825 (yes/no): yes
[CHG] Device E4:5F:01:E2:BB:FE Connected: no
[CHG] Device E4:5F:01:E2:BB:FE Connected: yes
Authorize service
[agent] Authorize service 0000111f-0000-1000-8000-00805f9b34fb (yes/no): yes
Authorize service
```

- Confirmación de conexión con la raspberry Pi del barco:

```
[pi@raspberrypi:~ $ hcitool con
Connections:
> ACL E4:5F:01:E2:BB:FE handle 12 state 1 lm CENTRAL AUTH ENCRYPT
```

Raspberry Cliente (Barco)

- Primero, se mira cual es la Mac de la raspberry del barco:

```
pi@raspberrypi:~ $ hciconfig
hci0: Type: Primary Bus: UART
      BD Address: E4:5F:01:E2:BB:FE ACL MTU: 1021:8 SCO MTU: 64:1
      UP RUNNING PSCAN
      RX bytes:7120 acl:96 sco:0 events:462 errors:0
      TX bytes:70200 acl:91 sco:0 commands:405 errors:0
```

- Se realizan los siguientes comandos para iniciar el escaneo de dispositivos bluetooth:

Conexión bluetooth

```
bluetoothctl
power on
scan on
```

- Se espera a que aparezca el siguiente mensaje en la raspberry:

```
[NEW] Device B8:27:EB:7D:8C:DE raspberrypi
```

- Una vez encontrado el dispositivo al que nos queremos conectar realizamos los siguientes comandos (dentro de bluetoothctl):

Conexión bluetooth

```
pair B8:27:EB:7D:8C:DE
```

- Tras este comando nos pedirá confirmar una "passkey" para poder enlazar el dispositivo bluetooth:

```
[agent] Confirm passkey 655825 (yes/no): yes
[CHG] Device B8:27:EB:7D:8C:DE Bonded: yes
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE UUIDs: 0000180a-0000-1000-8000-00805f9b34fb
[CHG] Device B8:27:EB:7D:8C:DE ServicesResolved: yes
[CHG] Device B8:27:EB:7D:8C:DE Paired: yes
Pairing successful
```

- Confiamos en el dispositivo servidor:

Conexión bluetooth

```
trust B8:27:EB:7D:8C:DE
```

```
[bluetooth]# trust B8:27:EB:7D:8C:DE
[CHG] Device B8:27:EB:7D:8C:DE Trusted: yes
Changing B8:27:EB:7D:8C:DE trust succeeded
```

- Nos conectamos al servidor:

Conexión bluetooth

```
connect B8:27:EB:7D:8C:DE
```

```
[bluetooth]# connect B8:27:EB:7D:8C:DE
Attempting to connect to B8:27:EB:7D:8C:DE
[CHG] Device 6A:DC:8F:AE:4C:EC ManufacturerData Key: 0x004c
[CHG] Device 6A:DC:8F:AE:4C:EC ManufacturerData Value:
  0c 0e 00 fd fc 58 7e 13 47 b6 19 db aa 95 5b 5e .....X~.G.....[^
  10 06 47 1d 61 f8 d2 38 ..G.a..8
[CHG] Device 5C:97:C0:A5:8F:75 RSSI: -54
[CHG] Device 5C:97:C0:A5:8F:75 ManufacturerData Key: 0x004c
[CHG] Device 5C:97:C0:A5:8F:75 ManufacturerData Value:
  01 00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 .....
  00 .
[CHG] Device B8:27:EB:7D:8C:DE Connected: yes
```

- Confirmación de conexión con la raspberry Pi del servidor de tierra:

```
pi@raspberrypi:~ $ hcitool con
Connections:
< ACL B8:27:EB:7D:8C:DE handle 12 state 1 lm PERIPHERAL AUTH ENCRYPT
```

Comunicación de Bluetooth

Raspberry Servidor (Servidor de tierra)

- Realizamos una escucha activa mediante el comando:

Escucha activa

```
sudo rfcomm listen /dev/rfcomm0
```

```
[pi@raspberrypi:~ $ sudo rfcomm listen /dev/rfcomm0
Waiting for connection on channel 1
Connection from E4:5F:01:E2:BB:FE to /dev/rfcomm0
Press CTRL-C for hangup
Disconnected
```

- Muestra del mensaje enviado por el cliente (barco):

```
[pi@raspberrypi:~ $ cat /dev/rfcomm0
Hola desde el cliente
```

Raspberry Cliente (Barco)

- Creamos un documento Python para enviar un mensaje de prueba:

client_ble.py

```
import bluetooth, time

# Dirección MAC del servidor (Raspberry Pi A)
server_mac_address = "B8:27:EB:7D:8C:DE"
port = 1

# Crear un socket Bluetooth
client_sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)

try:
    # Conectar al servidor
    client_sock.connect((server_mac_address, port))
    print("Conectado a", server_mac_address)
    time.sleep(5)
    # Enviar datos
    message = "Hola desde el cliente \n"
    client_sock.send(message)
    print("Enviado:", message)
except OSError as e:
    print("Error:", e)
finally:
    # Cerrar conexión
    print("Cerrando conexión")
    client_sock.close()
```