

# Bluetooth for Raspberry Pi

Here's an example script that allows a Raspberry Pi and an Adafruit Feather ESP32 to communicate wirelessly via Bluetooth:

## On the Raspberry Pi:

1. First, you need to install the necessary Bluetooth packages on your Raspberry Pi. You can do this by running the following commands:

```
sudo apt-get update
sudo apt-get install bluetooth bluez blueman
```

3. Next, you need to pair your Raspberry Pi with the Adafruit Feather ESP32.

Enable the Bluetooth on the esp32 by running the following command on the esp32:

```
//This example code is in the Public Domain (or CC0 licensed, at your option.)
//By Evandro Copercini - 2018
//
//This example creates a bridge between Serial and Classical Bluetooth (SPP)
//and also demonstrate that SerialBT have the same functionalities of a normal Serial

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(9600);
  SerialBT.begin("ESP32test"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");
}

void loop() {
  if (Serial.available()) {
    SerialBT.write(Serial.read());
  }
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
}
```

Then, run the following command on your Raspberry Pi:

```
bluetoothctl
```

3. In the `bluetoothctl` console, enter the following commands to scan for nearby Bluetooth devices and pair with the Feather ESP32:

```
scan on
pair XX:XX:XX:XX:XX:XX
```

Note: to find the mac address for the esp32:

```
// Complete Instructions to Get and Change ESP MAC Address: https://RandomNerdTutorials.com/get-change-esp32-esp8266-mac-address-arduino/

#ifdef ESP32
#include <WiFi.h>
#else
#include <ESP8266WiFi.h>
#endif

void setup(){
  Serial.begin(115200);
  Serial.println();
  Serial.print("ESP Board MAC Address: ");
  Serial.println(WiFi.macAddress());
}

void loop(){
}
```

1. Once your Raspberry Pi is paired with the Feather ESP32, you can use the `rfcomm` command to create a virtual serial port that will allow you to communicate with the Feather ESP32 over Bluetooth. Run the following command:

```
sudo rfcomm bind /dev/rfcomm0 XX:XX:XX:XX:XX:XX
```

(Note: replace `XX:XX:XX:XX:XX:XX` with the MAC address of your Feather ESP32.)

### On the Adafruit Feather ESP32:

1. First, you need to enable Bluetooth on your Feather ESP32. You can do this by including the following lines of code in your Arduino sketch:

```
#include <BluetoothSerial.h>

BluetoothSerial SerialBT;

void setup() {
  SerialBT.begin("FeatherESP32");
}

void loop() {
  SerialBT.println("Hello, Raspberry Pi!");
  delay(1000); // Wait 1 second before sending the next message
}
```

This code initializes the `BluetoothSerial` library, which allows the ESP32 to communicate with the Raspberry Pi over Bluetooth. It then enters the `loop()` function, which sends a message over Bluetooth once per second using the `SerialBT.println()` function. You can adjust the message content and frequency by modifying the string argument and the delay time, respectively.

### On the Raspberry Pi:

Use any programming language that supports serial communication to read the messages sent by the ESP32 over Bluetooth. Here's an example Python script that reads messages from the ESP32 and prints them to the console:

```
import serial

ser = serial.Serial('/dev/rfcomm0', baudrate=9600)

while True:
```

```
message = ser.readline()
print(message.decode().strip())
```

This code initializes a serial connection to the virtual serial port created by the `rfcomm` command on the Raspberry Pi, and enters an infinite loop that reads messages from the ESP32 using the `ser.readline()` function. It then decodes the message to a string and removes any trailing whitespace using the `decode()` and `strip()` functions, respectively. Finally, it prints the message to the console using the `print()` function.