**Connect a Raspberry Pi to a WiFi hotspot via Bluetooth**

What will we do in this tutorial?

Have you ever wondered how you can bring the power of Rasbperry Pi to your loved ones without them having to put their hands in it? Have you ever faced the need to connect a Raspberry Pi outside of your home to a WiFi network, without being able to modify the wpa\_ compliant file as most headless methods ask you to do? Have you ever wondered how smart switches can connect to your WiFi just through an app on your smartphone?

In this tutorial, we will answer each of these questions by setting up a method of connecting a Raspberry Pi to a WiFi hotspot via a Bluetooth connection.

In this tutorial, we will first lay down the technical bases that we will need here. We will then code a script to deploy on our Raspberry Pi. Finally, we will take a look at an example of an Android/iOS application written in Flutter, allowing us to connect via Bluetooth to the Raspberry Pi and to transmit it the WiFi credentials.

What will we need?

Of course, in order to follow this tutorial, we will need a Raspberry Pi or a similar device compatible with Bluetooth. Several options are available to you:

* Here are all the Raspberry Pi versions that have built-in Bluetooth and Wi-Fi technology :
  + Raspberry Pi 3 and 3B +
  + Raspberry Pi Zero W
  + Raspberry Pi 4
* You can use a [USB Bluetooth dongle](https://www.amazon.com/Plugable-Bluetooth-Adapter-Raspberry-Compatible/dp/B009ZIILLI/)
* You can use a HC-05 or HC-06 Bluetooth module with a little electronics

In my case, I will use a Raspberry Pi Zero WH.

We will also need a smartphone, either Android or iOS as the application we are going to develop will be written in Flutter, a cross-platform coding language.

How Bluetooth works?

*Bluetooth®* technology is the wireless communications technology for developers which allows devices to communicate with each other without the need for a central device like a router or access point.

A Bluetooth*®* device contains a table of data called an Attribute Table which can be accessed by other connected devices in various possible ways. It contains something like a series of records of various types. The main types are called [Services](https://www.bluetooth.com/specifications/gatt/services?utm_campaign=developer&utm_source=internal&utm_medium=blog&utm_content=a-developers-guide-to-bluetooth), [Characteristics](https://www.bluetooth.com/specifications/gatt/characteristics?utm_campaign=developer&utm_source=internal&utm_medium=blog&utm_content=a-developers-guide-to-bluetooth), and [Descriptors](https://www.bluetooth.com/specifications/gatt/descriptors?utm_campaign=developer&utm_source=internal&utm_medium=blog&utm_content=a-developers-guide-to-bluetooth).

Services, Characteristics and Descriptors are all types of Attribute and are organized in a hierarchy with Services at the top and Descriptors at the bottom. Services contain one or more Characteristics. All attributes have a type which is identified by a [UUID (Universally Unique Identifier)](https://www.bluetooth.com/specifications/assigned-numbers/16-bit-UUIDs-for-Members?utm_campaign=developer&utm_source=internal&utm_medium=blog&utm_content=a-developers-guide-to-bluetooth). Some Attributes are defined by the Bluetooth SIG, the technical standards body for Bluetooth and these have UUIDs which are 16 bits in length. Some Attributes are custom designed for a particular device by the product team and these have 128 bit UUIDs.

A Service is a container for logically related Bluetooth*®* data items. Those data items are in fact called Characteristics. Often a Service represents a particular feature (e.g. a hardware feature) of a device like the buttons or a particular sensor. Characteristics are items of data which relate to a particular internal state of the device or perhaps some state of the environment which the device can measure using a sensor. They have a type, a value, some properties and some permissions (READ, WRITE or NOTIFY).

To learn more about how Bluetooth works, I encourage you to read this [guide](https://www.bluetooth.com/blog/a-developers-guide-to-bluetooth/).

How are we going to connect our Raspberry Pi to WiFi via Bluetooth?

**Part 1 – Raspberry Pi side script**

**Part 2 – Flutter application**

<https://pub.dev/packages/flutter_blue>

<https://pub.dev/packages/flutter_blue/example>

<https://github.com/pauldemarco/flutter_blue/tree/master/example/lib>