Word Count: 1037

Video, Remote GPIO Control over Bluetooth

In this video, we are going to learn how to control the GPIO pins of the Raspberry Pi 4 wirelessly using Bluetooth from an android smartphone, and later you will be given an activity to build a remote-control robot using the blue dot library.

We are going to use a nifty Bluetooth library called Blue Dot, to establish remote control between the raspberry pi and the smartphone. This library also provides functions to control one Raspberry PI from another Raspberry Pi. The Blue dot library uses a client-server model to establish communication. The BlueDot class inside the library creates a Bluetooth server, on the server device, that is the Raspberry Pi, and the Blue Dot application installed on the client device will connect to this server via Bluetooth Communication. The transmission of data from client to server or server to the client is a simple stream of data. No acknowledgments or data is sent in response to commands when using the Blue dot library.

Now, lets install the Bluedot library. Open the terminal and first type

sudo apt install python3-dbus

This module is a prerequisite for working with the bluedot library. This library provides an abstraction layer for the bluedot library to do inter-process communication for the Bluetooth interface.

Now enter the following to install bluedot

sudo pip3 install bluedot

If your Raspbian OS version, already has bluedot, it is always a good thing to check for any updates and upgrade it using the following command

sudo pip3 install bluedot - - upgrade

The next step is to Pair your raspberry pi to the android phone. Go to your android device’s settings, and inside the Bluetooth configuration menu, make your device discoverable and then enable the Bluetooth. Now go to your Raspberry Pi Desktop and click the Bluetooth icon and select “Make discoverable”, then click “add device”. Your phone will appear in the list, select it and enter a pin code for pairing. Now on the android phone, enter the same pin code when prompted and touch “Ok”.

You have now successfully paired you phone with the Pi. Now go to the play store in your android phone and search for the Blue dot app. Install the one made by martin, and open the app. It will ask for permission to use Bluetooth. Please accept it. Now you can see our Pi on the list of connected devices. Select it, and you can see a blue dot.

The Blue Dot is a joystick as well as a button. You can tell if the dot was pressed in the middle, on the top, bottom, left or right. The app can send the information of the exact relative position of touch within the dot to your Pi via the bluetooth. We are going to leverage this data incoming to implement wireless projects using the Raspberry Pi.

First, let’s control an LED using Bluetooth. Wire up a Red LED on the breadboard, with a current limiting resistor, and connect it to the GPIO Pin 17 as we did in the last section. Now, in the Thonny IDE, open the Bluetooth LED.py file from the repository and run the code. Now, on your Blue Dot app if you press the blue dot, the LED will become ON. It will be ON till you release the blue dot in the app. In the code, we have imported the BlueDot Class from the Bluedot module with the first command. This class consists of all the methods also known as APIs, that allow us to establish and communicate with the client device. I highly recommend you to go through the API documentation, to know about all the API’s offered by the bluedot library. The link is provided in the resources section. As we are working with the LED on the GPIO pin 17, we need to import the LED class from the gpiozero library using the following line of code. We cannot directly access the methods of a class. We need to create an instance of both class and assign it to variable which can later be used to access the method. An object named bd is created as an instance of the BlueDot class, with no arguments. This means that the Bluetooth communication will be established with the default configuration. Check out the class definition of BlueDot in the API Documentation, to learn to do custom Bluetooth configuration. We have created an object with the LED class by defining the GPIO pin as 17, which is where the Red LED is connected to. Then an infinite loop is created inside which, we use the wait\_for\_press method and the wait \_for \_release method to sense the button presses wirelessly and take appropriate action. Here the action is to make the LED on or off. These methods have conditional branching implemented inside themselves, thus the user doesn’t have to create an if else statement. But if you want more control over what is happening internally you will need to learn and implement these methods from the API documentation.

Now, I want you to build a two-wheel drive remote controlled robot using the Robot and the BlueDot Library. You will need to wire up two motors to the motor driver module, and control the bot based on the relative position of touch detected in the blue dot on the android app. I will give you a clue. You need to use the methods in the BlueDotPostion API to detect the position of the touch. If you go through the documentation, you can see that they have even proven methods to detect top press, bottom press, left press, and right press on the blue dot app. I think I have given too much away. Anyways, try to implement this and mount your Project on a chassis, and test out your remote-controlled car. I have provided the complete working code in the resources section if you are stuck with this activity.

Summary

In this video, we have covered the following topics

* Setting up the Pi with the Blue Dot Library
* Implementing GPIO Control over Bluetooth
* Remote Controlled Robot Project

In the next video, we will learn to control the GPIO of the Pi over the Internet.