

11

Blynk Application with Raspberry Pi

11.1 Introduction to Blynk

There are three major components in the platform:

Blynk App—Allow to create an interface by using various widgets.

Blynk Server—It is responsible for the communication between the smartphone and hardware. It is an open-source platform.

Blynk Library—Blynk can work with Arduino, Raspberry Pi, or on similar boards.

Blynk works with any model of Raspberry Pi through Wi-Fi/Ethernet.

11.1.1 Installing Blynk on Raspberry Pi

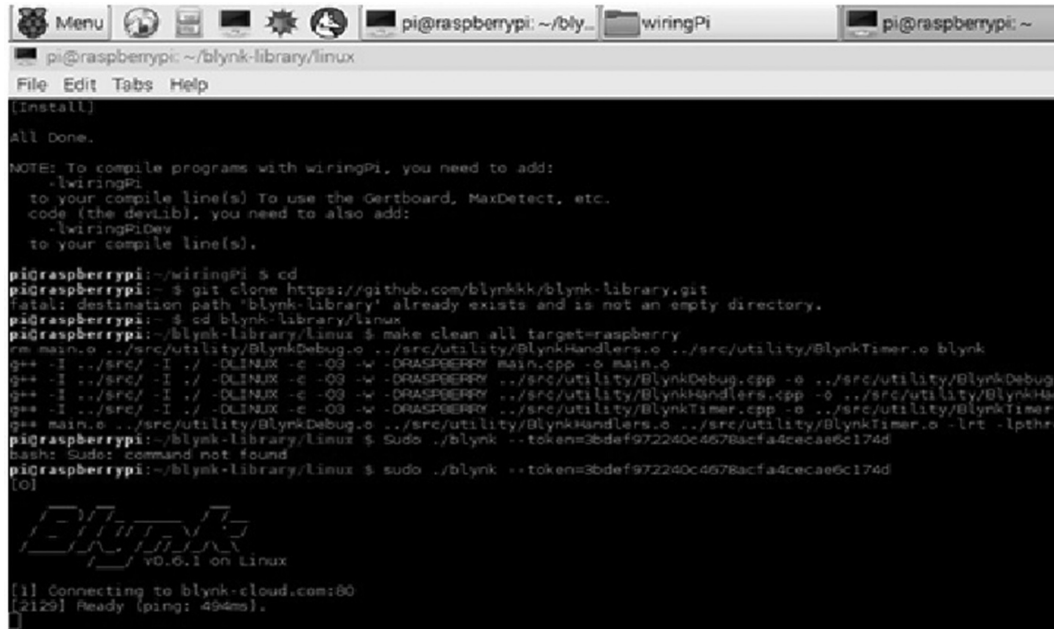
To install Blynk on Raspberry Pi, first install the Blynk libraries. It is required to have the latest version of Raspbian to install Blynk on Pi.

```
sudo apt-get update
sudo apt-get install git-core // If GIT is not installed, install it
git clone git://git.drogon.net/wiringPi // Install WiringPi using GIT
cdwiringPi // go to wiringPi and run the build.
./build
```

To install Blynk globally, run the commands:

```
git clone https://github.com/blynkkk/blynk-library.git
cd blynk-library/linux
make clean all target=raspberry
Sudo./blynk --token=token num // authentication token
```

Figure 11.1 shows the window after the execution of the commands.



```

pi@raspberrypi: ~/blynk-library/linux
File Edit Tabs Help
[install]
all Done.
NOTE: To compile programs with wiringPi, you need to add:
  -wiringPi
to your compile line(s) To use the Gertboard, MaxDetect, etc.
code (the devlib), you need to also add:
  -wiringPi.Dev
to your compile line(s).
pi@raspberrypi:~/wiringPi $ cd
pi@raspberrypi:~$ git clone https://github.com/blynkkk/blynk-library.git
fatal: destination path 'blynk-library' already exists and is not an empty directory.
pi@raspberrypi:~$ cd blynk-library/linux
pi@raspberrypi:~/blynk-library/linux$ make clean all target=raspberrypi
rm main.o ../src/utility/BlynkDebug.o ../src/utility/BlynkHandlers.o ../src/utility/BlynkTimer.o blynk
g++ -I ../src/ -I ../ -DLINUX -c -O3 -W -DRASPBERRY main.cpp -o main.o
g++ -I ../src/ -I ../ -DLINUX -c -O3 -W -DRASPBERRY ../src/utility/BlynkDebug.cpp -o ../src/utility/BlynkDebug.o
g++ -I ../src/ -I ../ -DLINUX -c -O3 -W -DRASPBERRY ../src/utility/BlynkHandlers.cpp -o ../src/utility/BlynkHandlers.o
g++ -I ../src/ -I ../ -DLINUX -c -O3 -W -DRASPBERRY ../src/utility/BlynkTimer.cpp -o ../src/utility/BlynkTimer.o
g++ main.o ../src/utility/BlynkDebug.o ../src/utility/BlynkHandlers.o ../src/utility/BlynkTimer.o -lrt -lpthread
bash: Sudo: command not found
pi@raspberrypi:~/blynk-library/linux$ sudo ./blynk --token=3bdef972240c4678acfa4cecae6c174d
[0]
[1] Connecting to blynk-cloud.com:80
[2129] Ready (ping: 494ms).
```

FIGURE 11.1

Raspberry Pi window.

11.2 Creating New Project with Blynk

Blynk application is available for Android and IOS both.

Steps to create Blynk project

1. Download Blynk for Android or IOS, as per the requirement, and create a new account (Figure 11.2).
2. Create a new project by providing a name, controller/processor, and internet connection to it (Figure 11.3). The background of the app can also be selected as dark or light.
3. The **Auth token** will be sent to the email address. This **Auth token** will be used for connecting Raspberry Pi to the new project widget.
4. Open the blank project and add widgets as required for the project (Figure 11.4). For a simple project of making a LED ON/OFF with button by clicking on it. Select a digital pin of Raspberry Pi to which LED is connected. Here GPIO21 is used to connect LED. Figure 11.5 shows the button for LED.
5. Add one more LED button connected to GPIO20. Now two LEDs can be controlled through the designed app (Figure 11.6).
6. Connect Raspberry Pi to the Blynk widget by running the **Auth token** command and check the workings.

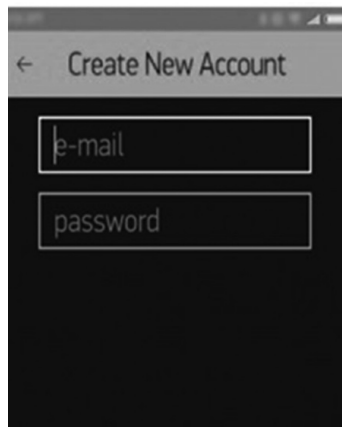


FIGURE 11.2
Creating a new account.

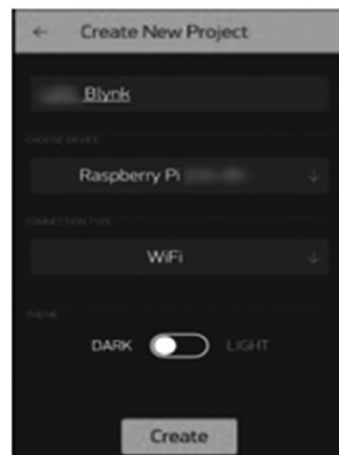


FIGURE 11.3
Create a new project.

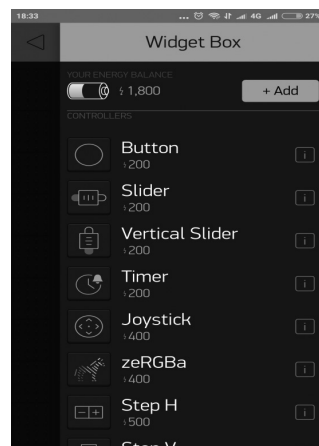


FIGURE 11.4
Widget box.

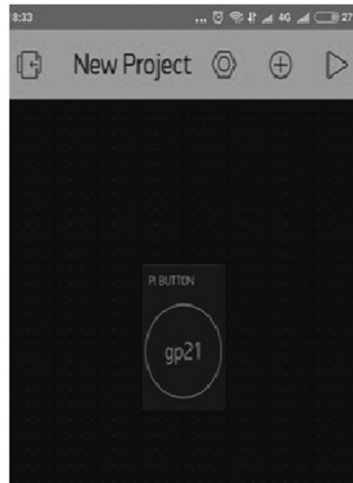


FIGURE 11.5
Button for LED.

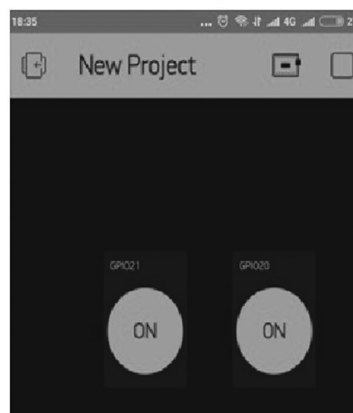


FIGURE 11.6
Project to control two LEDs.

11.3 Home Appliance Control with Blynk App

The Blynk application can be used to create an interface to control home appliances for smart control over Internet of Things (IoT). The system comprises of a Raspberry Pi, a power supply, two relay boards (if two appliances needs to be controlled), a transistor 2N2222 (as a switch), and a power supply. Connect the components as shown in [Figure 11.7](#) and follow the steps mentioned in [Section 11.2](#) to control two LEDs, as the same project will work to control two relays.

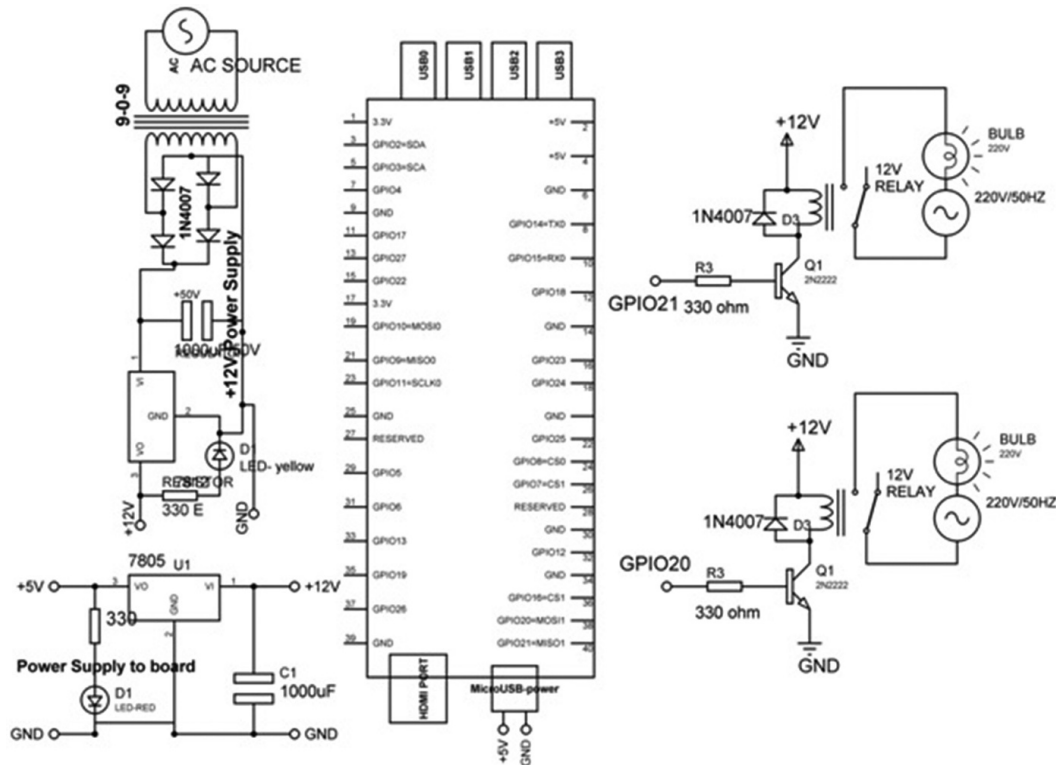


FIGURE 11.7
Circuit diagram for home appliance control.

Connections:

- Connect the base of the first transistor 2N2222 to GPIO21 through a resistor and emitter to the ground.
- Connect the base of the second transistor 2N2222 to GPIO20 through a resistor and emitter to the ground.
- Connect the collector of transistor to “L2” of relay.
- Connect the positive terminal of +12V battery to “L1” of relay.
- Connect a diode 1N4007 across “L1” and “L2.”
- Connect one terminal of AC to “common” of the relay and the other to one of the terminal of AC to the load (bulb).
- Connect the other terminal of the AC load to the “NO” terminal of the relay.