Install bluez on the Raspberry Pi

Created by Tony DiCola

https://learn.adafruit.com/install-bluez-on-the-raspberry-pi
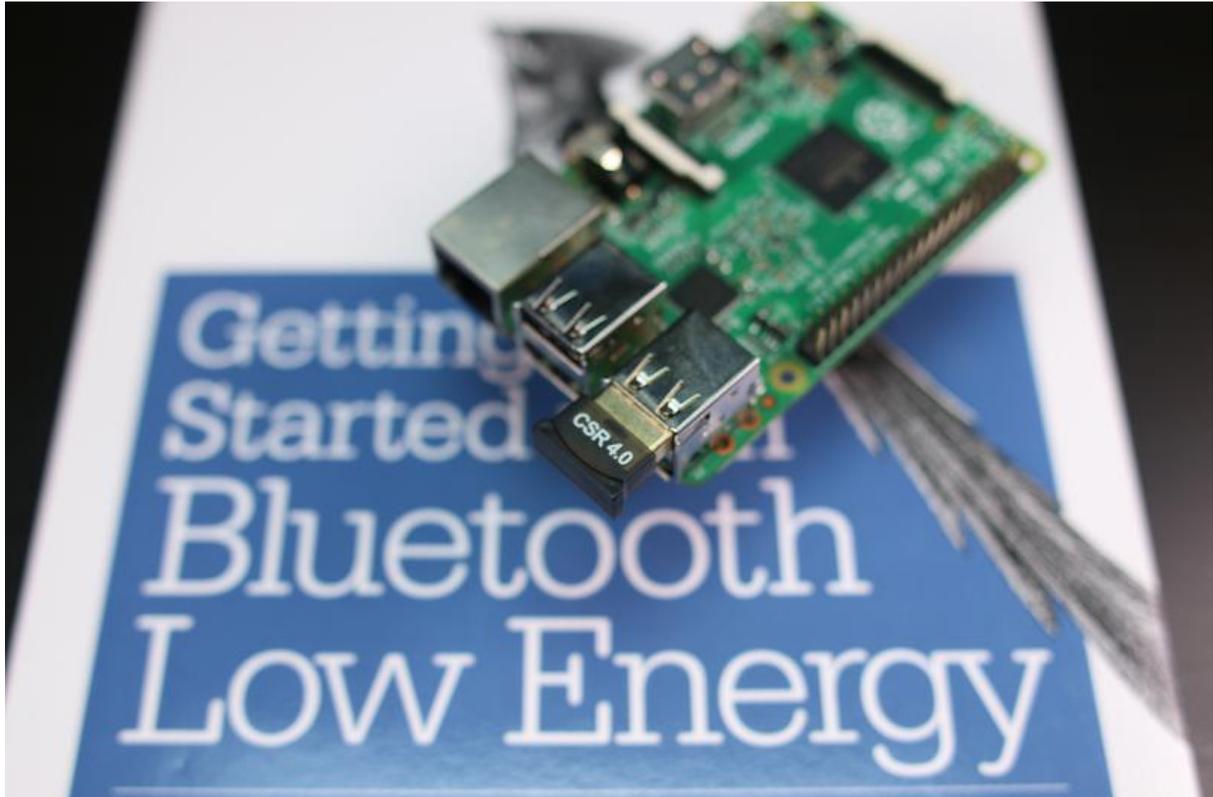
Last updated on 2022-12-01 02:42:18 PM EST

Table of Contents

Overview

> This Guide is obsolete. Modern versions of Raspberry Pi OS (formerly Raspbian) come with bluez already installed and enabled. The steps described in this Guide are no longer necessary.

This guide will walk through how to compile and install bluez () on the Raspberry Pi. Bluez is the Linux Bluetooth system and allows a Raspberry Pi to communicate with Bluetooth classic and Bluetooth low energy (LE) devices. Although bluez is quite full-featured it can be somewhat challenging to install and use. However this guide will show you step-by-step what you need to do to compile and install the latest version of bluez. Grab a Bluetooth USB adapter, like this handy Bluetooth 4.0 USB module (), and follow this guide to get setup using bluez in almost no time.

As a companion to this guide, check out the following video () which dives into Bluetooth low energy and installing and using bluez on the Pi:

In addition you might want to read this Introduction to Bluetooth Low Energy () guide for more information on BLE.

You will also want to be familiar with the basics of using a Raspberry Pi, like loading an operating system on a microSD card and connecting to a command terminal on the Pi. Check out the learn Raspberry Pi series () for more information on the basics.

Installation

> This Guide is obsolete. Modern versions of Raspberry Pi OS (formerly Raspbian) come with bluez already installed and enabled. The steps described in this Guide are no longer necessary.

Follow the instructions below to download, compile, install, and configure bluez on the Raspberry Pi. Before you get started you'll need to make sure your Raspberry Pi has access to the internet, either through a wired or wireless connection. In addition you'll want to be familiar with accessing a command terminal on the Pi, like with the SSH tool ().

You will also need to make sure your Raspberry Pi is running the latest version of the Raspbian Jessie () operating system (either the full or lite version). These instructions

won't work with the older Wheezy release--make sure you're running Jessie and its systemd service host.

Download Source

To install bluez () you'll need to download and compile the latest version of its source code. You could install bluez from a prebuilt package in the Raspbian repository how ever the version in the repository almost certainly is out of date. If you're using Bluetooth low energy features you really want to be running the absolute latest version of bluez to get the latest bug fixes and features. Compiling bluez from source will ensure you have the latest and greatest release.

Visit the bluez download page () and copy the link to the latest source release (under the User Space BlueZ Package section). For example at the time of this guide's writing the latest version of bluez is 5.37 and can be downloaded from:

```
http://www.kernel.org/pub/linux/bluetooth/bluez-5.37.tar.xz
```

Connect to a terminal on your Pi and execute the following command to download and open the bluez source archive. Remember, find the latest version of bluez from its homepage and use that download URL--the instructions below show using version 5.37 of bluez.

```
cd ~
wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.37.tar.xz
tar xvf bluez-5.37.tar.xz
```

You might need to change the tar command above to specify the filename for the version of bluez you're using.

After the command runs you should see it unzip the bluez source into a new folder, like bluez-5.37. Now change into that directory to continue with the installtion by running (note change the cd command to use the directory for the version of bluez you downloaded):

```
cd bluez-5.37
```

Install Dependencies

Now you'll need to install a few dependencies that the bluez library uses. Run the following commands to install these dependencies:

```
sudo apt-get update
sudo apt-get install -y libusb-dev libdbus-1-dev libglib2.0-dev libudev-dev libical-
dev libreadline-dev
```
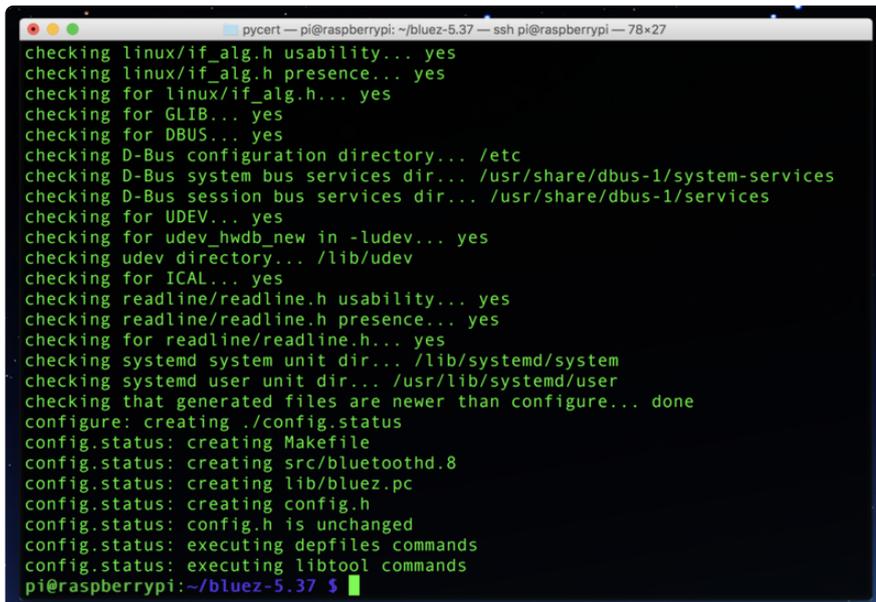
After the commands run you should see them install a few dependencies and finish without any error messages.

Compile & Install bluez

To compile bluez you'll use the standard configure, make, and sudo make install commands as most other Linux software. From inside the bluez source directory run the configure script as follows:

```
./configure --enable-library
```

You should see the script finish without any error messages like below:

If you do see an error message it likely means you need a dependent library installed. Go back and check you installed all of the libraries mentioned in the install dependencies section above. Also check what library the configure script detected as missing and see if you can install it from the Raspbian repository (most libraries are

named 'libX-dev' where X is the name of the library, like GLib or readline). You can rerun the configure script until it finishes successfully.

Once the configure script has successfully run you're ready to compile the bluez code. This compilation will take about 10-20 minutes on a Raspberry Pi 2 (and slightly longer on a Pi 1 or Zero). Start the compile by running:

```
make
```

You should see output as the make command compiles different parts of bluez. Once it's finished you should see no error messages like the following:
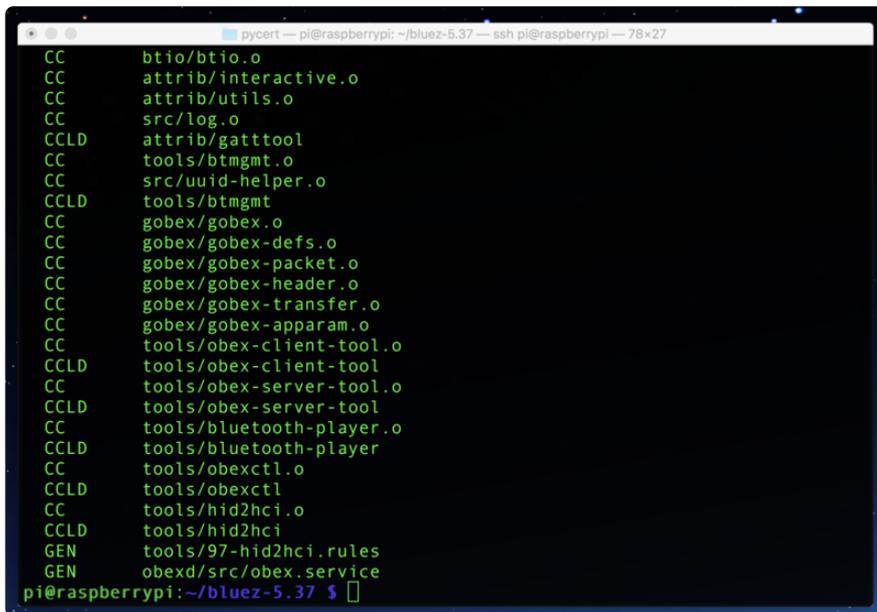
If you do see errors check that the configure script ran without any errors, and make sure all the dependencies are installed and try again.

After bluez has been compiled it can be installed by running the following command:

```
sudo make install
```

You should see the command finish with no errors like the following:

That's all there is to installing bluez from source! Now you'll learn a few tips about configuring bluez to run on the Pi.

Setup bluez Service

After installing bluez from source there's one more step to enable the bluetoothd service in bluez. This service talks to a part of bluez in the kernel and exposes bluetooth devices to user programs. The bluez service must be running for its tools and libraries to work!

With Raspbian Jessie the bluez service is run with systemd. Systemd is a process that controls other processes on the Pi, like the bluez service. First you can check that the bluez service is installed and in a good state by running the following command:

```
systemctl status bluetooth
```

You'll likely see the bluez service is loaded but not active, like:

```
● bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; disabled)
   Active: inactive (dead)
     Docs: man:bluetoothd(8)
```

You can manually start the bluez service by running the following command:

```
sudo systemctl start bluetooth
```

After running the above, run the status command again and you should see the service is now active:

```
● bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; disabled)
   Active: active (running) since Mon 2016-02-29 04:56:27 UTC; 1s ago
     Docs: man:bluetoothd(8)
 Main PID: 715 (bluetoothd)
   Status: "Running"
   CGroup: /system.slice/bluetooth.service
           └─715 /usr/local/libexec/bluetooth/bluetoothd
```

You can stop the service by running:

```
sudo systemctl stop bluetooth
```

And again the status command should now show the bluez service as inactive.

If you just want to intermittently use bluez tools & programs you could manually start and stop the service as shown above. Make sure to start the bluez service before running programs that depend on it.

However you probably want the bluez service to run automatically when the Raspberry Pi boots. To enable the bluez service to run at boot run the following command:

```
sudo systemctl enable bluetooth
```

Now reboot the Pi and run the status command again to check the bluez bluetooth service is running.

If you ever want to disable the bluez service from automatically starting, run the following command:

```
sudo systemctl disable bluetooth
```

Enable Bluetooth Low Energy Features

One final configuration change you can make is to enable the bluetooth low energy features in bluez. These are special APIs that allow bluez to interact with bluetooth low energy devices, however they're still in development and put behind an experimental flag that must be enabled first.

To enable bluez's experimental features like BLE you can modify the bluez service configuration. Edit this configuration by running:

```
sudo nano /lib/systemd/system/bluetooth.service
```

You should see a configuration file similar to the following:

```
[Unit]
Description=Bluetooth service
Documentation=man:bluetoothd(8)
ConditionPathIsDirectory=/sys/class/bluetooth

[Service]
Type=dbus
BusName=org.bluez
ExecStart=/usr/local/libexec/bluetooth/bluetoothd
NotifyAccess=main
#WatchdogSec=10
#Restart=on-failure
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
LimitNPROC=1

[Install]
WantedBy=bluetooth.target
Alias=dbus-org.bluez.service
```

Enable the experimental features by adding --experimental to the ExecStart line, for example the configuration should look like:

```
[Unit]
Description=Bluetooth service
Documentation=man:bluetoothd(8)
ConditionPathIsDirectory=/sys/class/bluetooth

[Service]
Type=dbus
BusName=org.bluez
ExecStart=/usr/local/libexec/bluetooth/bluetoothd --experimental
NotifyAccess=main
#WatchdogSec=10
#Restart=on-failure
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
LimitNPROC=1

[Install]
WantedBy=bluetooth.target
Alias=dbus-org.bluez.service
```

Save the file and exit the editor by pressing Ctrl-o, enter, then Ctrl-x.

Now tell systemd to reload its configuration files by running:

```
sudo systemctl daemon-reload
```

If the bluez service was previously running you'll want to restart it by running:

```
sudo systemctl restart bluetooth
```

Or if the service wasn't running you can start it with the start command previously shown.

Once the bluez service is running you can check the experimental features are enabled by running the status command again. You should see output similar to the following (notice the --experimental on the last line):

```
● bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; disabled)
   Active: active (running) since Mon 2016-02-29 05:15:55 UTC; 4s ago
     Docs: man:bluetoothd(8)
 Main PID: 1022 (bluetoothd)
   Status: "Running"
   CGroup: /system.slice/bluetooth.service
           └─1022 /usr/local/libexec/bluetooth/bluetoothd --experimental
```

That's all there is to configuring and running bluez on the Raspberry Pi! At this point you're ready to start using bluez tools like bluetoothctl, hcitool, etc.