Bluetooth

From ArchWiki

Bluetooth (http://www.bluetooth.org/) is a standard for the short-range wireless interconnection of cellular phones, computers, and other electronic devices. In Linux, the canonical implementation of the Bluetooth protocol stack is BlueZ (http://www.bluez.org/).

Related articles

- Bluez4
- Bluetooth mouse
- Bluetooth mouse configuration
- Bluetooth headset
- Blueman

Contents

Installation

Install the Bluetooth protocol stack `bluez (https://www.archlinux.org/packages/?name=bluez)` and the `bluez-utils (https://www.archlinux.org/packages/?name=bluez-utils)` package which provides the `bluetoothctl` utility from the official repositories. The `dbus` daemon, which is started automatically by *systemd*, is used to read settings and for PIN pairing, while the `bluetooth` daemon is required for the Bluetooth protocol.

Load the generic bluetooth driver, if not already loaded:

```
# modprobe btusb
```

To start the bluetooth systemd service use the command:

```
# systemctl start bluetooth
```

To enable the bluetooth service at boot time use the command:

```
# systemctl enable bluetooth
```

Note: By default the bluetooth daemon will only give out bnep0 devices to users that are a member of the lp group. Make sure to add your user to that group if you intend to connect to a bluetooth tether. You can change the group that is required in the file `/etc/dbus-1/system.d/bluetooth.conf`.

Note: Some Bluetooth adapters are bundled with a Wi-Fi card (e.g. Intel Centrino (http://www.intel.com/content/www/us/en/wireless-products/centrino-advanced-n-6235.html)). These require that the Wi-Fi card is first enabled (typically a keyboard shortcut on a laptop) in order to make the Bluetooth adapter visible to the kernel.

Note: Some Bluetooth cards (e.g. Broadcom) conflict with the network adapter. Thus, you need to make sure that your Bluetooth device get connected before the network service boot.

Configuration via the CLI

Bluetoothctl

Pairing a device from the shell is one of the most simplistic and reliable options. The exact procedure depends on the devices involved and their input functionality. What follows is a general outline of pairing a device using `/usr/bin/bluetoothctl`:

Start the `bluetoothctl` interactive command. There one can input `help` to get a list of available commands.

- Turn the power to the controller on by entering `power on`. It is off by default.
- Enter `devices` to get the MAC Address of the device with which to pair.
- Enter device discovery mode with `scan on` command if device is not yet on the list.
- Turn the agent on with `agent on`.
- Enter `pair` *`MAC Address`* to do the pairing (tab completion works).
- If using a device without a PIN, one may need to manually trust the device before it can reconnect successfully. Enter `trust` *`MAC Address`* to do so.
- Finally, use `connect` *`MAC_address`* to establish a connection.

An example session may look this way:

```
# bluetoothctl
[NEW] Controller 00:10:20:30:40:50 pi [default]
[bluetooth]# agent KeyboardOnly
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:10:20:30:40:50 Discovering: yes
[NEW] Device 00:12:34:56:78:90 myLino
[CHG] Device 00:12:34:56:78:90 LegacyPairing: yes
[bluetooth]# pair 00:12:34:56:78:90
Attempting to pair with 00:12:34:56:78:90
[CHG] Device 00:12:34:56:78:90 Connected: yes
[CHG] Device 00:12:34:56:78:90 Connected: no
[CHG] Device 00:12:34:56:78:90 Connected: yes
Request PIN code
[agent] Enter PIN code: 1234
[CHG] Device 00:12:34:56:78:90 Paired: yes
Pairing successful
[CHG] Device 00:12:34:56:78:90 Connected: no
```

In order to have the device active after a reboot, a udev rule is needed:

```
/etc/udev/rules.d/10-local.rules
```

```
# Set bluetooth power up
ACTION=="add", KERNEL=="hci0", RUN+="/usr/bin/hciconfig hci0 up"
```

After a suspend/resume-cycle, the device can be powered on automatically using something like this systemd service:

```
/etc/systemd/system/bluetooth-auto-power@.service
```

```
[Unit]
Description=Bluetooth auto power on
After=bluetooth.service sys-subsystem-bluetooth-devices-%i.device suspend.target

[Service]
Type=oneshot
#We could also do a 200 char long call to bluez via dbus. Except this does not work since bluez
#ExecStart=/usr/bin/dbus-send --system --type=method_call --dest=org.bluez /org/bluez/%I org.fre
ExecStart=/usr/bin/hciconfig %i up

[Install]
WantedBy=suspend.target
```

Configuration with a Graphical Front-end

The following packages allow for a graphical interface to customize Bluetooth.

GNOME Bluetooth

GNOME Bluetooth (https://wiki.gnome.org/Projects/GnomeBluetooth) is a fork of the old *bluez-gnome* and is focused on integration with the GNOME desktop environment. The `gnome-bluetooth` (`https://www.archlinux.org/packages/?name=gnome-bluetooth`) package provides the back-end, `gnome-shell` (`https://www.archlinux.org/packages/?name=gnome-shell`) provides the status monitor applet, and `gnome-control-center` (`https://www.archlinux.org/packages/?name=gnome-control-center`) provides the configuration front-end GUI that can be accessed by typing Bluetooth on the Activities overview, or with the `gnome-control-center bluetooth` command.

Users who are not using GNOME Shell can install `gnome-bluetooth-applet-git` (`https://aur.archlinux.org/packages/gnome-bluetooth-applet-git/`) from AUR, which provides the old status monitor applet, and allows setup devices and transfer of files by clicking the Bluetooth icon. Just make sure that `bluetooth-applet` is autostarted with your session.

You can also launch the following commands directly:

- `bluetooth-sendto` : send files to a remote device
- `bluetooth-wizard` : for new devices to be paired

To add a Bluetooth entry to the *SendTo* menu in Thunar's file properties menu, see instructions here (http://docs.xfce.org/xfce/thunar/send-to). (The command that needs to be configured is `bluetooth-sendto %F`)

BlueDevil

The Bluetooth tool for KDE is BlueDevil (https://projects.kde.org/projects/extragear/base/bluedevil). It can be installed with the package `bluedevil (https://www.archlinux.org/packages/?name=bluedevil)`, available in the official repositories.

Make sure `bluetooth` daemon is running, as described above. A Bluetooth icon should be visible in both Dolphin and in the system tray, from which users may configure BlueDevil and detect Bluetooth devices by clicking the icon. An interface is also available from the KDE System Settings.

Blueman

See Blueman.

Using Obex for sending and receiving files

ObexFS

Another option, rather than using KDE or Gnome Bluetooth packages, is ObexFS which allows for the mounting of phones which are treated like any other filesystem.

> **Note:** To use ObexFS, one needs a device that provides an ObexFTP service.

Install `obexfs (https://www.archlinux.org/packages/?name=obexfs)` and mount supported phones by running:

```
$ obexfs -b devices_MAC_address /mountpoint
```

Once you have finished, to unmount the device use the command:

```
$ fusermount -u /mountpoint
```

For more mounting options see http://dev.zuckschwerdt.org/openobex/wiki/ObexFs

> **Note:** Ensure that the bluetooth device you are mounting is **not** set to mount *read-only*. You should be able to do this from the device's settings. If the device is mounted *read-only* you may encounter a permissions error when trying to transfer files to the device.

ObexFTP Transfers

If your device supports the Obex FTP service but you do not wish to mount the device you can transfer files to and from the device using the obexftp command.

> **Note:** If you installed `obexfs (https://www.archlinux.org/packages/?name=obexfs)` earlier then `obexftp (https://www.archlinux.org/packages/?name=obexftp)` should have also been installed as a dependency.

To send a file to a device run the command:

```
$ obexftp -b devices_MAC_address -p /path/to/file
```

To retrieve a file from a device run the command:

```
$ obexftp -b devices_MAC_address -g filename
```

Note: Ensure that the file you are retrieving is in the device's *exchange folder*. If the file is in a subfolder of the exchange folder then provide the correct path in the command.

Obex Object Push

For devices that do not support Obex FTP service, check if Obex Object Push is supported.

```
# sdptool browse XX:XX:XX:XX:XX:XX
```

Read the output, look for Obex Object Push, remember the channel for this service. If supported, one can use `ussp-push (https://www.archlinux.org/packages/?name=ussp-push)` to send files to this device:

```
# ussp-push XX:XX:XX:XX:XX:XX@CHANNEL file wanted_file_name_on_phone
```

Examples

All examples have been moved to the bluez4 article. They need to be checked and fixed for use with bluez5.

Troubleshooting

gnome-bluetooth

If you see this when trying to enable receiving files in bluetooth-properties:

```
Bluetooth OBEX start failed: Invalid path
Bluetooth FTP start failed: Invalid path
```

Then install `xdg-user-dirs (https://www.archlinux.org/packages/?name=xdg-user-dirs)` and issue:

```
$ xdg-user-dirs-update
```

You can edit the paths using:

```
$ vi ~/.config/user-dirs.dirs
```

Bluetooth USB Dongle

If you are using a USB dongle, you should check that your Bluetooth dongle is recognized. You can do that by running `journalctl -f` when plugging in the USB dongle (or inspecting `/var/log/messages.log`). It should look something like the following (look out for hci):

```
Feb 20 15:00:24 hostname kernel: [ 2661.349823] usb 4-1: new full-speed USB device number 3 usin
Feb 20 15:00:24 hostname bluetoothd[4568]: HCI dev 0 registered
Feb 20 15:00:24 hostname bluetoothd[4568]: Listening for HCI events on hci0
Feb 20 15:00:25 hostname bluetoothd[4568]: HCI dev 0 up
Feb 20 15:00:25 hostname bluetoothd[4568]: Adapter /org/bluez/4568/hci0 has been enabled
```

If you only get the first two lines, you may see that it found the device but you need to bring it up. Example:

```
hciconfig -a hci0
```

```
hci0:   Type: USB
        BD Address: 00:00:00:00:00:00 ACL MTU: 0:0 SCO MTU: 0:0
        DOWN
        RX bytes:0 acl:0 sco:0 events:0 errors:0
        TX bytes:0 acl:0 sco:0 commands:0 errors:
```

```
# hciconfig hci0 up
```

```
hciconfig -a hci0
```

```
hci0:   Type: USB
        BD Address: 00:02:72:C4:7C:06 ACL MTU: 377:10 SCO MTU: 64:8
        UP RUNNING
        RX bytes:348 acl:0 sco:0 events:11 errors:0
        TX bytes:38 acl:0 sco:0 commands:11 errors:0
```

If this fails with an error like:

```
Operation not possible due to RF-kill
```

it could be due either to the `rfkill` utility, in which case it should be resolved with

```
# rfkill unblock all
```

or, it could simply be the hardware switch of the computer. The hardware bluetooth switch (at least sometimes) controls access to USB bluetooth dongles also. Flip/press this switch and try bringing the device up again.

To verify that the device was detected you can use `hcitool` which is part of the `bluez-utils` . You can get a list of available devices and their identifiers and their MAC address by issuing:

```
$ hcitool dev
```

```
Devices:
        hci0    00:1B:DC:0F:DB:40
```

More detailed information about the device can be retrieved by using `hciconfig` .

```
$ hciconfig -a hci0
```

```
hci0:   Type: USB
        BD Address: 00:1B:DC:0F:DB:40 ACL MTU: 310:10 SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:1226 acl:0 sco:0 events:27 errors:0
        TX bytes:351 acl:0 sco:0 commands:26 errors:0
        Features: 0xff 0xff 0x8f 0xfe 0x9b 0xf9 0x00 0x80
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: 'BlueZ (0)'
        Class: 0x000100
        Service Classes: Unspecified
        Device Class: Computer, Uncategorized
        HCI Ver: 2.0 (0x3) HCI Rev: 0xc5c LMP Ver: 2.0 (0x3) LMP Subver: 0xc5c
```

```
        Manufacturer: Cambridge Silicon Radio (10)
```

Logitech Bluetooth USB Dongle

There are Logitech dongles (ex. Logitech MX5000) that can work in two modes Embedded and HCI. In embedded mode dongle emulates a USB device so it seems to your PC that your using a normal USB mouse/keyoard.

If you hold the little red Button on the USB BT mini-receiver it will enable the other mode. Hold the red button on the BT dongle and plug it into the computer, and after 3-5 seconds of holding the button, the Bluetooth icon will appear in the system tray. Discussion (http://ubuntuforums.org/showthread.php?t=1332197)

hcitool scan: Device not found

- On some Dell laptops (e.g. Studio 15) you have to switch the Bluetooth mode from HID to HCI. Install the `bluez-hid2hci (https://www.archlinux.org/packages/?name=bluez-hid2hci)` package, then udev should do this automatically. Alternatively, you can run this command to switch to HCI manually:

```
# /usr/lib/udev/hid2hci
```

- If the device will not show up and you have a Windows operating system on your machine, try booting it and enable the bluetooth adapter from windows.
- Sometimes also this simple command helps:

```
# hciconfig hci0 up
```

rfkill unblock: Do not unblock

If your device still soft blocked and you run connman, try this:

```
$ connmanctl enable bluetooth
```

My computer is not visible

Cannot discover computer from your phone? Enable PSCAN and ISCAN:

```
# enable PSCAN and ISCAN
$ hciconfig hci0 piscan
# check it worked
```

```
$ hciconfig
```

```
hci0:   Type: USB
        BD Address: 00:12:34:56:78:9A ACL MTU: 192:8 SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:20425 acl:115 sco:0 events:526 errors:0
        TX bytes:5543 acl:84 sco:0 commands:340 errors:0
```

Note: Check DiscoverableTimeout and PairableTimeout in `/etc/bluetooth/main.conf`

Try changing device class in `/etc/bluetooth/main.conf` as following:

```
# Default device class. Only the major and minor device class bits are
# considered.
#Class = 0x000100 (from default config)
Class = 0x100100
```

This was the only solution to make my computer visible for my phone.

Logitech keyboard does not pair

If you do not get the passkey when you try to pair your Logitech keyboard, type the following command:

```
# hciconfig hci0 sspmode 0
```

If after pairing, the keyboard still does not connect, check the output of `hcidump -at`. If the latter indicates repeatedly connections-disconnections like the following message:

```
  status 0x00 handle 11 reason 0x13
  Reason: Remote User Terminated Connection
```

then, the only solution for now is to install the old Bluetooth stack.

HSP/HFP profiles

bluez5 has no support for the HSP/HFP profiles (telephony headset for TeamSpeak, Skype, etc.), see bluez4 discussion for details.

Until support is implemented, it is necessary to downgrade to bluez4 and install `pulseaudio-bluez4 (https://aur.archlinux.org/packages/pulseaudio-bluez4/)` from the AUR.

Retrieved from "https://wiki.archlinux.org/index.php?title=Bluetooth&oldid=334920"
Category: Bluetooth
