

## Common Raspberry set up

- 1) Plug in USB bluetooth dongle
- 2) Connect to RPI
- 3) `sudo apt-get install bluetooth bluez-utils python-bluez -y`
- 4) `lsusb` should show you your BT module connected. Like this:  
*Bus 001 Device 004: ID 0a12:0001 Cambridge Silicon Radio, Ltd. Bluetooth Dongle (HCI mode)*  
Pay attention to HCI mode. We only support dongles with HCI.
- 5) Set up second BT device you want to communicate with (enable BT on cellphone or power up UART module)
- 6) `hcitool scan` should show you Bt device you've just enabled  
Write down it's MAC address shown by scan.
- 7) `sudo nano /etc/bluetooth/rfcomm.conf`  
Edit file. It should look like this:

```
#
# RFCOMM configuration file.
#
rfcomm0 {
# Automatically bind the device at startup
    bind yes;
# Bluetooth address of the device
    device <writed down MAC goes here>;
# RFCOMM channel for the connection
    channel 1;
# Description of the connection
    comment "LS-ONE";
}
```

- 8) Create or edit file `/var/lib/bluetooth/xx:xx:xx:xx:xx:xx/pincodes`

Just press TAB right after bluetooth.

This file defines pincodes for paired devices.

Add your second device and assign pincode to it.

Ex: `sudo echo "98:D3:31:B0:80:6C 1234" >> /var/lib/bluetooth/xx:xx:xx:xx:xx:xx/pincodes`

- 9) Edit `/etc/bluetooth/main.conf`

Disable pnat on one of the first lines:

*DisablePlugins = pnat*

- 10) Restart bluetooth `sudo /etc/init.d/bluetooth restart`

## Setting up BT connection btw UART dongle + arduino and raspberry + USB dongle

- 1) Execute common Raspberry setup
- 2) Develop simple proxy btw SerialMonitor and raspberry. This can easily be done using SoftwareSerial (use it for connecting with BT module on arduino).
- 3) Open SerialMonitor on arduino
- 4) Execute on raspberry: `echo "TEST" > /dev/rfcomm0`
- 5) You should see TEST appeared in serial monitor.
- 6) If you want to send something from arduino to RPI then `cat /dev/rfcomm0` Write something to SerialMonitor on arduino.
- 7) Sending messages from python
  - a) Go <http://people.csail.mit.edu/albert/bluez-intro/x232.html>
  - b) Take `rfcomm-client.py` and move it on raspberry. Write MAC to `bd_addr`.

6c) `python rfcomm-client.py`

6d) You should see “hello!!!” in the terminal.

### Setting up BT connection btw Cellphone and raspberry + USB dongle

1) Make USB dongle visible for other devices (move to SLAVE mode in other words)

`sudo hciconfig hci0 piscan`

2) Go <http://people.csail.mit.edu/albert/bluez-intro/x232.html>

3) Take `rfcomm-server.py` example, edit it to send data if you want.

4) `python rfcomm-server.py`

5) Take your cellphone, pair with raspberry.

6) Launch app that can monitor Bluetooth traffic (BlueTerm in android)

7) Send or receive something

### Troubleshooting

1) Devices do not pair. Try to add your device to trusted.

`bluez-test-device trusted XX:XX:XX:XX:XX:XX yes`

2) You need your MAC address but you do not know how to get it.

`hcitool dev`