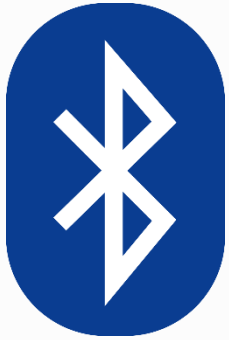
Bluetooth



Goal

- Wireless Communication
  - Designed to replace cables by enabling short-range wireless communication between devices
- Low Power Consumption
  - Created for energy-efficient connections, ideal for mobile devices
- Interoperability
  - Standardized technology to connect a wide range of devices, regardless of brand or platform
- Convenience
  - Simplifies data exchange and device pairing (e.g., headphones, keyboards, smartwatches)

Why Bluetooth?

- Comes from Harald "Bluetooth" Gormsson
  - A 10th-century Viking, king of Denmark
  - Known for uniting parts of Scandinavia
    - Just like Bluetooth technology unites communication between different devices
  - The name was suggested as a code name during development, but it stuck!
- The logo combines the runes for H (ᚼ) and B (ᛒ)

History

- 1994
  - Developed by Ericsson as a wireless alternative to RS-232 data cables
- 1998
  - Formation of the Bluetooth Special Interest Group (SIG) with companies like Intel, Nokia, IBM, and Toshiba
- 1999
  - First consumer Bluetooth device introduced – a hands-free mobile headset
- 2000s–2020s
  - Continuous evolution with new versions improving speed, range, security, and energy efficiency (e.g., Bluetooth Low Energy)
- Today
  - Widely used in smartphones, wearables, smart home devices, cars, and more

Versions' timeline: classic v1

- Bluetooth 1.0 & 1.1 (1999–2001)
  - Basic wireless connectivity
  - Max speed: 721 Kbps
  - Initial adoption, but had compatibility and security issues
- Bluetooth 1.2 (2003)
  - Faster connection setup
  - Improved resistance to interference
    - AFH – Adaptive Frequency Hopping

Versions' timeline: classic v2/3

- Bluetooth 2.0 + EDR (2004)
  - Enhanced Data Rate (EDR): up to 3 Mbps (2.1 Mbps useful)
  - Lower power consumption
  - More efficient data transmission
- Bluetooth 2.1 + EDR (2007)
  - Introduced Secure Simple Pairing (SSP) for
    - Easier and safer connections
  - Improved power efficiency
- Bluetooth 3.0 + HS (2009)
  - Added support for High Speed
    - Up to 24 Mbps using Wi-Fi for large data transfers
  - Better suited for transferring videos or large files

Versions' timeline: classic + BLE v4

- Bluetooth 4.0 (2010)
  - Introduced **Bluetooth Low Energy** (BLE)
  - Designed for smart devices and wearables
  - Maintained compatibility with classic Bluetooth
- Bluetooth 4.1 (2013)
  - Improved coexistence with LTE
  - Better device communication
    - Devices can act as both central and peripheral
- Bluetooth 4.2 (2014)
  - Enhanced privacy and security
  - Increased speed and packet capacity
  - Support for IPv6
    - Important for IoT

Versions' timeline: classic + BLE v5

- Bluetooth 5.0 (2016)
  - 2x speed (up to 2 Mbps with BLE), 4x range (up to ~240 m in ideal conditions)
  - 8x broadcast messaging capacity
  - Significant for smart homes and IoT devices
- Bluetooth 5.1 (2019)
  - Introduced direction finding (Angle of Arrival/Departure)
  - Improved device location accuracy (used in indoor navigation)
- Bluetooth 5.2 (2020)
  - Introduced LE Audio: better sound quality, lower power usage
  - Multi-stream audio support (for truly wireless earbuds)
  - Enhanced Isochronous Channels (important for synchronized streaming)
- Bluetooth 5.3 (2021)
  - Improved power control
  - Better connection management
  - Reduced interference

Versions: rate & range

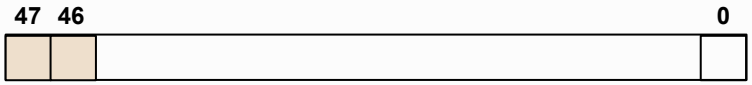
Version		Adoption year	Maximum rate (Mbit/s)		Max range (m)
Major	Minor		Classic	Low energy	
1	1.0	1999	0.7322	N/A	10
	1.1	2001	0.7322		
	1.2	2003	1		
2	2.0	2004	2.1		
	2.1	2007			
3	3.0	2009	2.4		
4	4.0	2009	3	1	60
	4.1	2013			
	4.2	2014			
5	5.0	2016	5	2	240
	5.1	2019			
	5.2	2020			
	5.3	2021			
	5.4	2023			

Bluetooth fundamental logical concepts

- Addresses and names
- Roles
- Network types
- Pairing and bonding
- Security modes and levels
- Bluetooth Low Energy (BLE) concepts
- Profiles
- Protocol stack

Bluetooth addresses

- MAC address
  - 48-bit value
- Public addresses
  - Fixed and unique

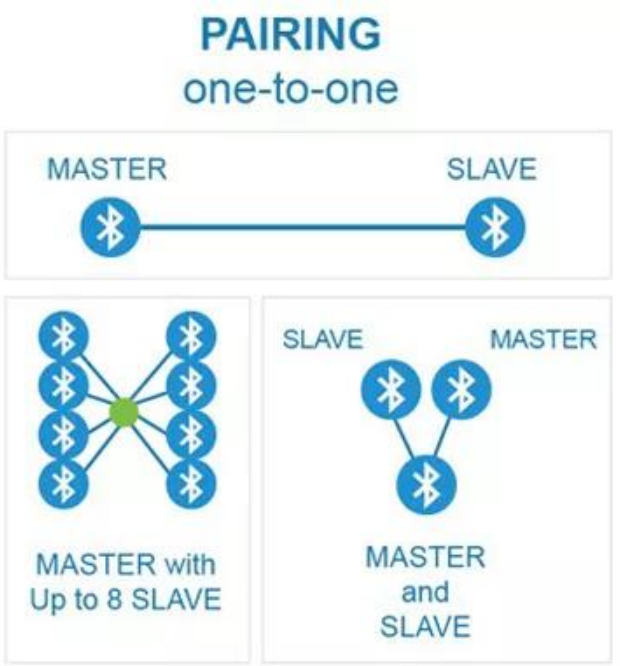


0	0	**NRPA**
0	1	**RPA**
1	0	**Static random**
1	1	**Public**

- Random addresses
  - Used in BLE for privacy
  - Static
    - Kept across power cycles
  - Dynamic
    - Frequently regenerated and used only for a short period
    - **Resolvable (RPA)**
      - Allow private recognition
    - **Non-resolvable (NRPA)**
      - Complete privacy

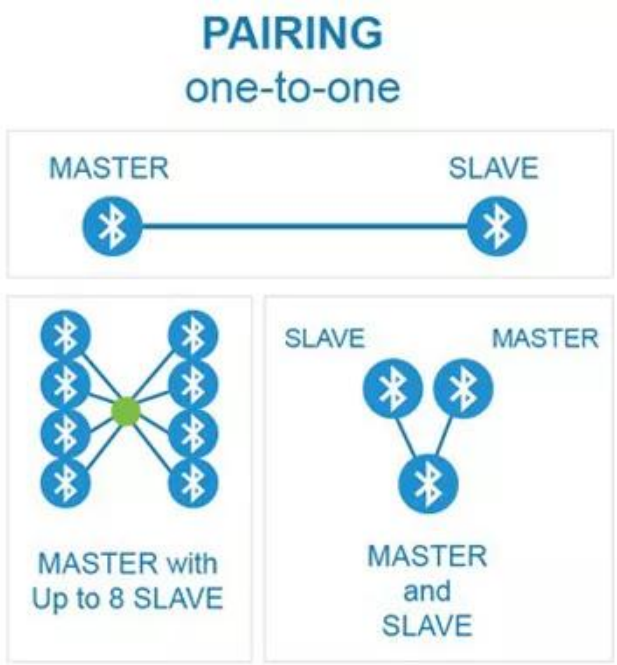
Bluetooth networks (classic)

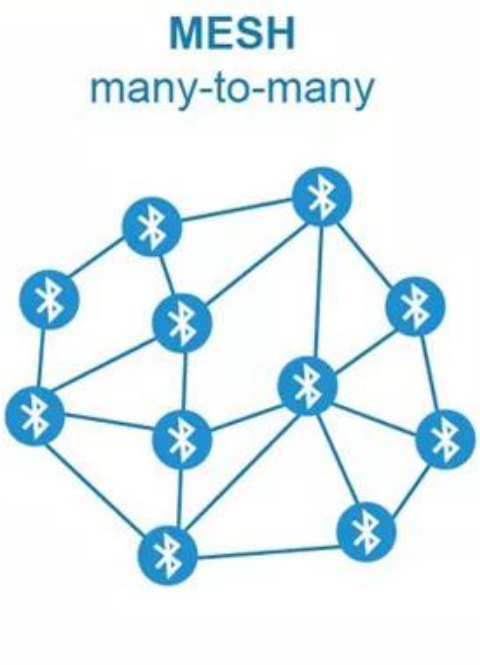
- Piconet
  - Small network with a master and up to 7 active slaves
- Scatternet
  - A network of multiple piconets
  - Shared devices acting as bridges
  - Mainly academic…



https://www.iot-rf.com/how-bluetooth-module-works-m-b14001-ble-module-introduction.html

Bluetooth LE networks





https://www.iot-rf.com/how-bluetooth-module-works-m-b14001-ble-module-introduction.html

- Point-to-Point
  - Like classic Piconet
- Broadcast (one-to-many)
  - Mainly used for advertising
- Mesh (many-to-many)
  - Ad hoc connections

Bluetooth pairing and bonding

- The process by which two Bluetooth devices establish trust and a secure connection

- Pairing
  - The act of exchanging keys and setting up encryption

- Bonding
  - The storing of keys for future automatic reconnections

Bluetooth security modes and levels

- Bluetooth supports multiple security modes and levels to manage encryption & authentication

- Classic
  - Mode 1: No security
  - Modes 2/3/4: provide different levels of authentication, authorization, and encryption
- LE
  - Mode 1: Encrypted communication
  - Mode 2: Authentication, no encryption

Bluetooth LE concepts

- GATT (Generic Attribute Profile)
  - Defines how data is organized and transferred
- GAP (Generic Access Profile)
  - Defines roles and procedures for connecting
    - Advertising, scanning, etc.
- Characteristics & Services
  - BLE data is structured in a hierarchy
    - Devices have services
    - Services have characteristics (data points)

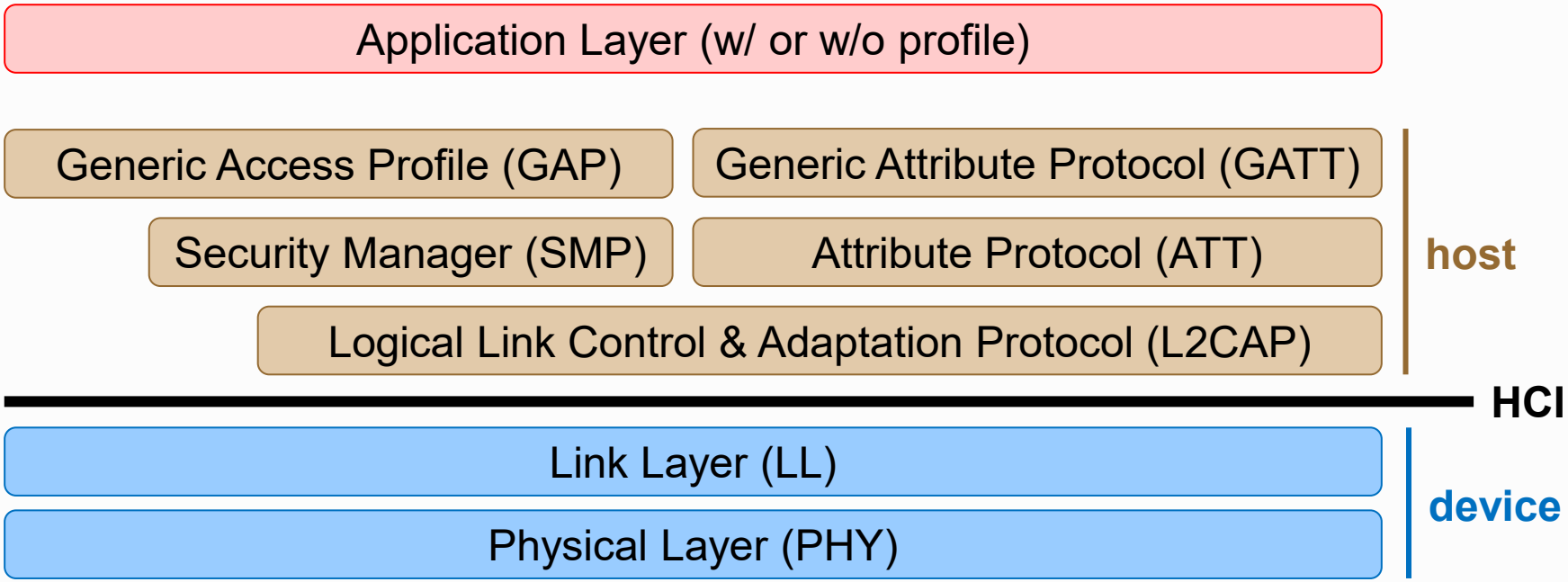
Bluetooth profiles

- A profile defines a specific use case or application for Bluetooth
  - e.g., audio streaming, file transfer
  - It outlines the protocols and procedures devices must use to ensure compatibility
- Service Discovery Protocol (SDP)
  - A protocol that allows a device to query another device for its services (profiles and capabilities)

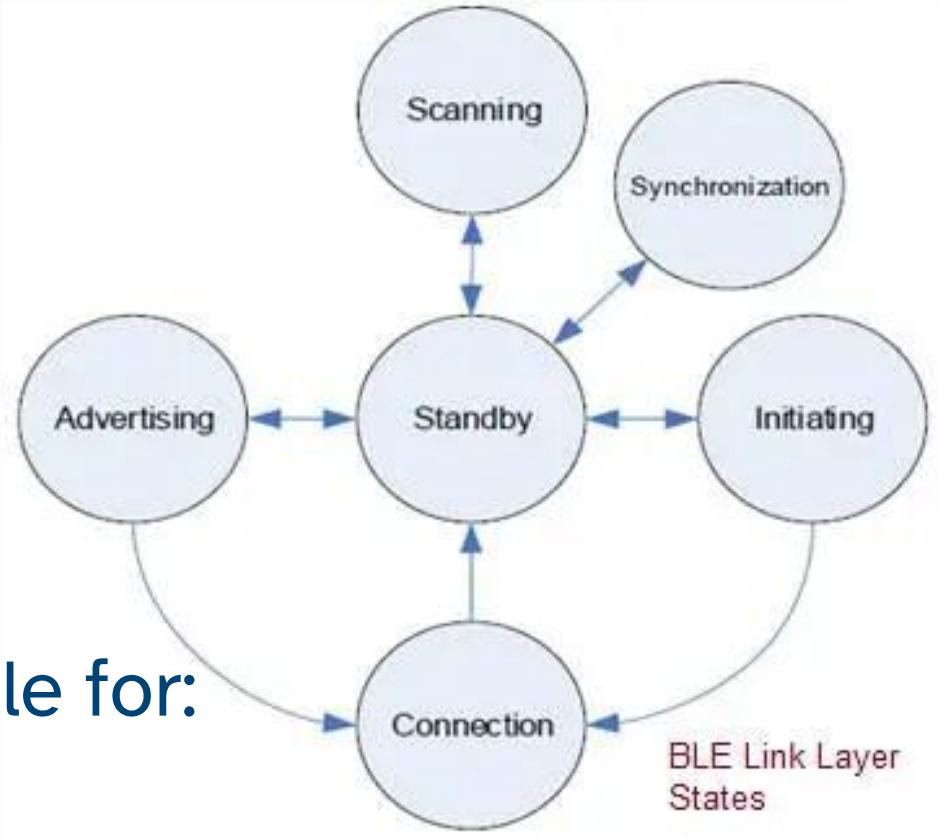
Bluetooth protocol stack

- Layered architecture that organizes how data is handled during communication
  - It consists of hardware and software layers from the physical radio to high-level profiles

Bluetooth LE protocol stack



BLE link layer: device states



- This layer is responsible for:
  - Advertising
  - Scanning
  - Creating and maintaining a connection

https://www.rfwireless-world.com/terminology/ble-protocol-stack-system-architecture

BLE host layers

- L2CAP (Logical Link Control & Adaptation Protocol)
  - Offers data encapsulation services to upper layers
  - Allows logical end-to-end data communication
- ATT (Attribute Protocol)
  - Allows a device to expose specific attributes
- GATT (Generic Attribute Protocol)
  - Service framework that specifies sub-procedures to use ATT
  - Data communications between two devices are handled through these sub-procedures
  - Applications and/or profiles will use GATT directly

BLE host layers (2/2)

- GAP (Generic Access Profile)
  - Directly interfaces with the App layer and/or profiles on it
  - Handles device discovery and connection-related services
  - Manages the initiation of security features

- SMP (Security Manager Protocol)
  - Provides methods for device pairing and key distributions
  - It offers services to other protocol stack layers to securely connect and exchange

BLE application layer

- The BLE protocol stack layers interact with applications and profiles as desired

- Any profiles/applications run on top of the GAP/GATT layers of the BLE protocol stack
  - It handles device discovery and connection-related services for the BLE device

Pairing methods (classic & LE)

- Just Works
  - No authentication or MitM protection
  - Used in low-UI devices (e.g., earbuds)
- Passkey Entry
  - One device displays a 6-digit code; the other enters it
  - MitM protection
- Numeric Comparison
  - Both devices show a number; user confirms if they match
  - MitM protection
- Out-of-Band (OOB)
  - Pairing info exchanged through another medium (e.g., NFC)
  - Most secure

LE Security mode 1: levels

1. No security
   - No pairing
2. Unauthenticated pairing w/ encryption
   - Just works pairing, vulnerable to MitM
3. Authenticated pairing w/ encryption
   - Passkey entry / numeric comparison / OOB pairing
4. Authenticated LE secure connections w/ encryption (v4.2+)
   - Same as 3
   - Uses ECDH for key exchange
5. Same as 4 with keypress tracking (v5.2+)

LE Security mode 2: levels

1. Unauthenticated pairing w/ data signing
   - Just works pairing, vulnerable to MitM
2. Authenticated pairing w/ data signing
   - Passkey entry / numeric comparison / OOB pairing

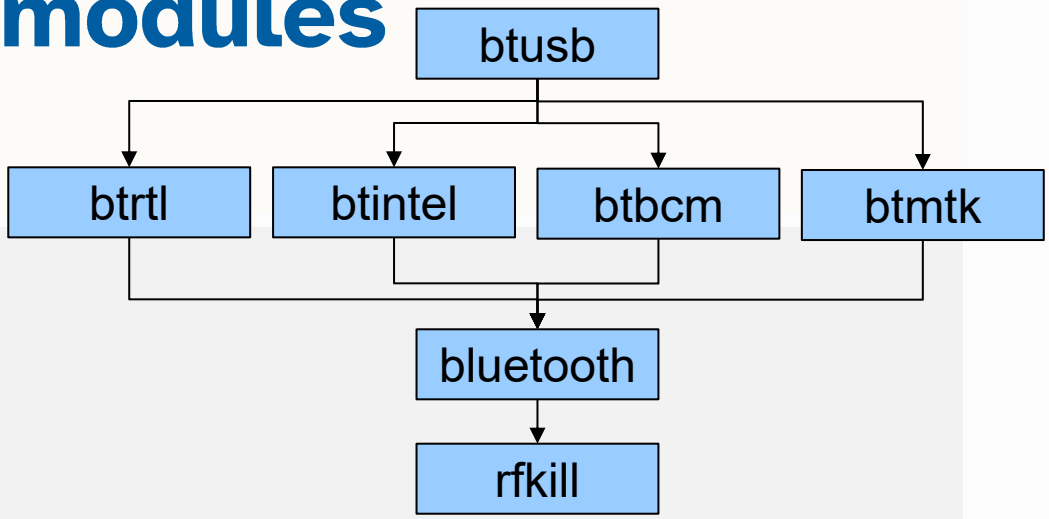
Bluetooth stacks

- There are alternative stack implementations
  - That follow the standard specifications
- Linux
  - BlueZ is the official stack in Linux distributions

Linux bluetooth experience:

```
[102320.145195] usb 1-2.1: new full-speed USB device number 4 using uhci_hcd
[102320.426990] usb 1-2.1: New USB device found, idVendor=8087, idProduct=0033, bcdDevice= 0.00
[102320.427012] usb 1-2.1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[102320.682460] Bluetooth: Core ver 2.22
[102320.682677] NET: Registered PF_BLUETOOTH protocol family
[102320.682701] Bluetooth: HCI device and connection manager initialized
[102320.682755] Bluetooth: HCI socket layer initialized
[102320.682789] Bluetooth: L2CAP socket layer initialized
[102320.682834] Bluetooth: SCO socket layer initialized
[102320.723779] usbcore: registered new interface driver btusb
[102320.730384] Bluetooth: hci0: Device revision is 0
[102320.730438] Bluetooth: hci0: Secure boot is enabled
[102320.730454] Bluetooth: hci0: OTP lock is enabled
[102320.730468] Bluetooth: hci0: API lock is enabled
[102320.730482] Bluetooth: hci0: Debug lock is disabled
[102320.730496] Bluetooth: hci0: Minimum firmware build 1 week 10 2014
[102320.730511] Bluetooth: hci0: Bootloader timestamp 2022.18 buildtype 1 build 49266
[102320.822557] Bluetooth: hci0: Found device firmware: intel/ibt-0180-0041.sfi
[102320.822623] Bluetooth: hci0: Boot Address: 0x100800
[102320.822638] Bluetooth: hci0: Firmware Version: 20-49.24
[102321.106938] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[102321.106980] Bluetooth: BNEP filters: protocol multicast
[102321.107045] Bluetooth: BNEP socket layer initialized
[102326.455339] Bluetooth: hci0: Waiting for firmware download to complete
[102326.455562] Bluetooth: hci0: Firmware loaded in 5959324 usecs
[102326.455739] Bluetooth: hci0: Waiting for device to boot
[102326.498708] Bluetooth: hci0: Malformed MSFT vendor event: 0x02
[102326.498746] Bluetooth: hci0: Device booted in 45630 usecs
[102326.521010] Bluetooth: hci0: Found Intel DDC parameters: intel/ibt-0180-0041.ddc
[102326.524618] Bluetooth: hci0: Applying Intel DDC parameters completed
[102326.526676] Bluetooth: hci0: No support for BT device in ACPI firmware
[102326.529610] Bluetooth: hci0: Firmware timestamp 2024.48 buildtype 1 build 3604
[102326.529617] Bluetooth: hci0: Firmware SHA1: 0xc115e35a
[102326.535589] Bluetooth: hci0: Fseq status: Success (0x00)
[102326.535598] Bluetooth: hci0: Fseq executed: 00.00.03.94
[102326.535603] Bluetooth: hci0: Fseq BT Top: 00.00.03.94
[102326.696980] Bluetooth: MGMT ver 1.23
[102326.746928] Bluetooth: RFCOMM TTY layer initialized
[102326.746973] Bluetooth: RFCOMM socket layer initialized
[102326.746999] Bluetooth: RFCOMM ver 1.11
```

Linux Bluetooth modules




```
$ lsmod | egrep "Used|blue|bt"
Module                  Size  Used by
btusb                  77824  0
btrtl                  36864  1 btusb
btintel                69632  1 btusb
btbcm                  24576  1 btusb
btmtk                  32768  1 btusb
bluetooth            1044480  44 btrtl,btmtk,btintel,btbcm,bnep,btusb,rfcomm
rfkill                 40960  5 bluetooth
$ for m in `lsmod | egrep "blue|bt" | awk '{print $1}'`;
> do echo -n $m ""
> modinfo $m | grep description
> done
btusb description:    Generic Bluetooth USB driver ver 0.8
btrtl description:    Bluetooth support for Realtek devices ver 0.1
btintel description:    Bluetooth support for Intel devices ver 0.1
btbcm description:    Bluetooth support for Broadcom devices ver 0.1
btmtk description:    Bluetooth support for MediaTek devices ver 0.1
bluetooth description:    Bluetooth Core ver 2.22
rfkill description:    RF switch support
```

Linux Bluetooth daemon bluetoothd

- This daemon manages all Bluetooth devices
  - Application use libraries to interact with devices
  - The libraries use the daemon to interact with the devices

```
$ ps -axww | egrep "PID|bluetoothd"
    PID TTY      STAT   TIME COMMAND
  63729 ?        Ss     0:00 /usr/libexec/bluetooth/bluetoothd
```

High-level Bluetooth management

- Bluetooth devices can be managed through bluetoothctl

```
$ bluetoothctl list
Controller E0:8F:4C:F6:7C:77 budgie [default]
```

- Details about known devices are recorded in /var/lib/bluetooth

```
$ ls -l /var/lib/bluetooth/
total 4
drwx------. 1 root root 36 Sep  9 15:31 E0:8F:4C:F6:7C:77
drwxr-xr-x. 1 root root  0 Apr  2 01:00 mesh
```

High-level Bluetooth management: Using bluetoothctl

```
[root@budgie E0:8F:4C:F6:7C:77]# bluetoothctl
[NEW] Media /org/bluez/hci0
          SupportedUUIDs: 0000110a-0000-1000-8000-00805f9b34fb
          SupportedUUIDs: 0000110b-0000-1000-8000-00805f9b34fb
Agent registered
[bluetoothctl]>
```

- These are the 128-bit UUIDs of entities supported by this Linux host
  - Services
    - A2DP (Advanced Audio Distribution Profile) source
    - A2DP (Advanced Audio Distribution Profile) sink
  - Characteristics
  - Descriptors

High-level Bluetooth management: Discovery of nearby LE devices

```
[bluetoothctl]> can.transport le
[bluetoothctl]> scan on
SetDiscoveryFilter success
hci0 type 7 discovering on
Discovery started
[CHG] Controller E0:8F:4C:F6:7C:77 Discovering: yes
[NEW] Device F4:7B:5E:80:EC:26 DTVBluetooth
[NEW] Device 60:07:C4:12:1D:04 OPPO Watch Free 1D04
[NEW] Device 70:1F:3C:B0:5A:6E André's Tab S6 Lite
[bluetoothctl]>
```

- Devices are discovered with a scan
  - It is a statistical process, because several radio frequencies are used
  - Scanning can use BR/EDR, LE or both
  - In the example above we only used BLE scanning

High-level Bluetooth management: Cache of discovered devices

- Devices discovered with a scan are registered in a directory
  - /var/lib/bluetooth/[local MAC addr]/cache
- Each file in this directory as a MAC address as name

```
$ ls -l /var/lib/bluetooth/E0:8F:4C:F6:7C:77/cache
total 12
-rw-------. 1 root root 36 Sep 10 11:36 60:07:C4:12:1D:04
-rw-------. 1 root root 36 Sep 10 11:36 70:1F:3C:B0:5A:6E
-rw-------. 1 root root 28 Sep 10 11:36 F4:7B:5E:80:EC:26
```

- Each file contains the name indicated by the device

```
$ cat /var/lib/bluetooth/E0:8F:4C:F6:7C:77/cache/70:1F:3C:B0:5A:6E
[General]
Name=André's Tab S6 Lite
```

High-level Bluetooth management: Device pairing

```
[bluetoothctl]> connect 70:1F:3C:B0:5A:6E
Attempting to connect to 70:1F:3C:B0:5A:6E
hci0 device_flags_changed: 70:1F:3C:B0:5A:6E (BR/EDR)
     supp: 0x00000001  curr: 0x00000000
hci0 70:1F:3C:B0:5A:6E type BR/EDR connected eir_len 5
[CHG] Device 70:1F:3C:B0:5A:6E Connected: yes
[CHG] Device 70:1F:3C:B0:5A:6E Modalias: bluetooth:v0075p1200d1436
00001105-0000-1000-8000-00805f9b34fb
0000110a-0000-1000-8000-00805f9b34fb
0000110c-0000-1000-8000-00805f9b34fb
0000110e-0000-1000-8000-00805f9b34fb
00001112-0000-1000-8000-00805f9b34fb
00001115-0000-1000-8000-00805f9b34fb
00001116-0000-1000-8000-00805f9b34fb
0000111f-0000-1000-8000-00805f9b34fb
00001200-0000-1000-8000-00805f9b34fb
00001800-0000-1000-8000-00805f9b34fb
00001801-0000-1000-8000-00805f9b34fb
a82efa21-ae5c-3dde-9bbc-f16da7b16c5a
[CHG] Device 70:1F:3C:B0:5A:6E ServicesResolved: yes
Request confirmation
[agent] Confirm passkey 205319 (yes/no): yes
```

High-level Bluetooth management: Device bonding upon pairing

```
hci0 new_link_key 70:1F:3C:B0:5A:6E type 0x08 pin_len 0 store_hint 1
[CHG] Device 70:1F:3C:B0:5A:6E Bonded: yes
[CHG] Device 70:1F:3C:B0:5A:6E Paired: yes
[CHG] Device 70:1F:3C:B0:5A:6E AddressType: public
Failed to connect: org.bluez.Error.Failed br-connection-unknown
hci0 70:1F:3C:B0:5A:6E type BR/EDR disconnected with reason 3
[CHG] Device 70:1F:3C:B0:5A:6E ServicesResolved: no
[SIGNAL] org.bluez.Device1.Disconnected org.bluez.Reason.Remote Connection terminated by remote
user
[CHG] Device 70:1F:3C:B0:5A:6E Connected: no
```

- Devices got paired and bonded
  - The connection is then broken
  - And reinitiated with bonded keys
    - To authenticate the devices with each other

High-level Bluetooth management: Device bonding data

- Bonding data is stored in a file
  - /var/lib/bluetooth/[local MAC]/[remote MAC]/info

```
$ cat /var/lib/bluetooth/E0:8F:4C:F6:7C:77/70:1F:3C:B0:5A:6E/info
[General]
Name=André's Tab S6 Lite
Class=0x5a020c
SupportedTechnologies=BR/EDR;LE;
Trusted=false
Blocked=false
CablePairing=false
…
[LinkKey]
Key=4D27A7F48F052830CE82D98177F5712E
Type=8
PINLength=0
```

High-level Bluetooth management: Device connection after bonding

```
hci0 70:1F:3C:B0:5A:6E type BR/EDR connected eir_len 5
[CHG] Device 70:1F:3C:B0:5A:6E Connected: yes
00001105-0000-1000-8000-00805f9b34fb
0000110a-0000-1000-8000-00805f9b34fb
0000110c-0000-1000-8000-00805f9b34fb
0000110e-0000-1000-8000-00805f9b34fb
00001112-0000-1000-8000-00805f9b34fb
00001115-0000-1000-8000-00805f9b34fb
00001116-0000-1000-8000-00805f9b34fb
0000111f-0000-1000-8000-00805f9b34fb
00001200-0000-1000-8000-00805f9b34fb
00001800-0000-1000-8000-00805f9b34fb
00001801-0000-1000-8000-00805f9b34fb
a23d00bc-217c-123b-9c00-fc44577136ee
a82efa21-ae5c-3dde-9bbc-f16da7b16c5a
[CHG] Device 70:1F:3C:B0:5A:6E ServicesResolved: yes
```

- Exported UUIDs are discovered again
  - And services are resolved
  - This means that endpoints are locally created to access those services

High-level Bluetooth management: Remote UUID interpretation

UUID	Entity
00001105-0000-1000-8000-00805f9b34fb	OBEX Object Push
0000110a-0000-1000-8000-00805f9b34fb	Audio Source
0000110c-0000-1000-8000-00805f9b34fb	A/V Remote Control Target
0000110e-0000-1000-8000-00805f9b34fb	A/V Remote Control
00001112-0000-1000-8000-00805f9b34fb	Headset AG
00001115-0000-1000-8000-00805f9b34fb	Personal Area Network User (PANU)
00001116-0000-1000-8000-00805f9b34fb	Network Access Point (NAP)
0000111f-0000-1000-8000-00805f9b34fb	Handsfree Audio Gateway
00001200-0000-1000-8000-00805f9b34fb	PnP Information
00001800-0000-1000-8000-00805f9b34fb	Generic Access Profile
00001801-0000-1000-8000-00805f9b34fb	Generic Attribute Profile
a23d00bc-217c-123b-9c00-fc44577136ee	Vendor Specific
a82efa21-ae5c-3dde-9bbc-f16da7b16c5a	Vendor Specific

High-level Bluetooth management: D-Bus setup for Lib/App interaction

```
[NEW] Endpoint /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep1
[NEW] Endpoint /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep2
[NEW] Endpoint /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep3
[NEW] Endpoint /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep4
[NEW] Endpoint /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep5
[NEW] Transport /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/fd4
[NEW] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 [default]
```

- These are D-Bus entities
  - They are a sort of D-Bus objects, with methods and properties
- The Linux host recognizes a remote AVRCP (Audio/Video Remote Control Profile) player
  - And lists all its characteristics
- More details on this endpoints can be observed with d-feet or d-spy
  - You can use this tool to execute command (e.g., connect or disconnect)

High-level Bluetooth management: D-Bus player properties

```
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Playlist is nil
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Repeat: off
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Shuffle: off
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Type: Audio
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Subtype: Audio Book
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Status: playing
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Name: Media Player
[CHG] Transport /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep2/fd4 Volume: 0x004d (77)
[CHG] Transport /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep2/fd4 State: pending
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.Title: Another Man's Woman - Live
At Hammersmith Odeon / 1975
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.TrackNumber: 0x00000001 (1)
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.NumberOfTracks: 0x00000001 (1)
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.Duration: 0x00070cb0 (462000)
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.Album: Crime Of The Century
(Deluxe)
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.Artist: Supertramp
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Track.Genre:
[CHG] Transport /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/sep2/fd4 State: active
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Position: 0x0003a37f (238463)
[CHG] Player /org/bluez/hci0/dev_70_1F_3C_B0_5A_6E/avrcp/player0 Position: 0x0003a3df (238559)
```

- You can see that I was listening a (quite old!) Supertramp album ☺

High-level Bluetooth management: Using a remote device

- The remote device asks to use Linux as an A2DP sink
  - The answer was yes
  - And the Samsung tablet sound output started to the Linux host

```
Authorize service
[agent] Authorize service 0000111e-0000-1000-8000-00805f9b34fb (yes/no): yes
[André's Tab S6 Lite]>
```