Pair Bluetooth Controllers - manual setup

First ensure that Bluetooth is enabled on your device. The line `controllers.bluetooth.enabled=1` should be present in your `batocera.conf` and not commented out. If it is missing, add it in. Note that doing this won't suddenly make a non-functional Bluetooth dongle start working, only the ones that are already compatible with the Linux kernel and the drivers included in Batocera.

Establish an SSH connection (or open up Xterm) to start the manual connection process. Get started by typing `bluetoothctl` and hitting [Enter].

Now you are in the Bluetooth control unit. Type following commands and confirm every single one with [Enter]

```
agent on
default-agent
power on
scan on
```

Now set the controller in pairing mode! This strongly depends from your controller type.

An example of the output you would see from running all the commands successfully:

```
[root@BATOCERA /userdata/system]# bluetoothctl
Agent registered
[bluetooth]# agent on
Agent is already registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# scan on
Discovery started
[CHG] Controller 43:45:C0:00:1F:AC Discovering: yes
[NEW] Device E4:17:D8:C2:0B:0E 8BitDo M30 Modkit
```

> 💡 If you're having trouble detecting devices, check out the Bluetooth section on the troubleshooting page.

Now we want to connect to our controller. So we type following commands to connect to discovered controller with MAC address E4:17:D8:C2:0B:0E properly. Wait a few seconds after entering the commands in order to ensure they are processed successfully.

```
pair E4:17:D8:C2:0B:0E
connect E4:17:D8:C2:0B:0E
trust E4:17:D8:C2:0B:0E
```

This would be the output from our example:

```
[bluetooth]# pair E4:17:D8:C2:0B:0E
Attempting to pair with E4:17:D8:C2:0B:0E
[CHG] Device E4:17:D8:C2:0B:0E Connected: yes
[CHG] Device E4:17:D8:C2:0B:0E Modalias: usb:v2DC8p5101d0100
[CHG] Device E4:17:D8:C2:0B:0E UUIDs: 00001124-0000-1000-8000-00805f9b34fb
[CHG] Device E4:17:D8:C2:0B:0E UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device E4:17:D8:C2:0B:0E ServicesResolved: yes
[CHG] Device E4:17:D8:C2:0B:0E Paired: yes
Pairing successful

[bluetooth]# connect E4:17:D8:C2:0B:0E
Attempting to connect to E4:17:D8:C2:0B:0E
Failed to connect: org.bluez.Error.Failed
[CHG] Device E4:17:D8:C2:0B:0E Connected: yes
Authorize service (yes/no): yes

Authorize service 00001124-0000-1000-8000-00805f9b34fb
[8BitDo M30 Modkit]# trust E4:17:D8:C2:0B:0E
[CHG] Device E4:17:D8:C2:0B:0E Trusted: yes
Changing E4:17:D8:C2:0B:0E trust succeeded
```

Type `devices` to see a list of devices configured in your BT setup.
Type `remove MAC-ADRESS` to remove single devices
Type `yes` if you are asked for authorization progress or enter PIN

If you have multiple Bluetooth adapters on your machine and you want to tell which one is used, you can use the command `hcitool dev` to list them all. On my setup, `hci0` is the internal BT on my Pi4, and `hci1` is an additional USB BT module:

```
# hcitool dev
Devices:
    hci1    90:80:62:00:88:00
    hci0    DC:A6:32:04:10:7F
# bluetoothctl
Agent registered
[bluetooth]# scan on
Discovery started
[CHG] Controller 90:80:62:00:88:00 Discovering: yes
```

Batocera is automatically pairing a device I don't want it to!

Some devices out there break Bluetooth protocol and are always in "pairing" mode, common with modern "smart" TVs. Since Batocera will automatically connect to any device in pair mode, this could cause unintended connections. This can be avoided by utilizing the `live_devices` function.

Demonstration needed!

Run:

```
batocera-bluetooth live_devices
```

Note down which device you want to connect to. Then:

```
batocera-bluetooth trust [MAC address]
```

Blacklisting a device in Batocera v33

> ⚠️ The functionality of this has been supplanted by `live_devices` described above. These instructions are retained here specifically for Batocera **v33** and no other version.

Demonstration needed!

There might be some device that is perpetually in pairing mode that you don't want your Batocera machine to connect to (such as your neighbor's "smart" TV). You can block such devices by utilizing the blacklist.

In order to do this via SSH, run the following command:

```
batocera-bluetooth blacklist <device address> <device name>
```

For example:

🔧 Fix Me!

To show a list of all the currently blacklisted devices:

```
batocera-bluetooth list_blacklistable
```

To remove a device from the blacklist:

```
batocera-bluetooth unblacklist <device address>
```

Using batocera-bluetooth

> Still needs further investigation/testing.

```
batocera-bluetooth list                    Print the list of currently
trusted and saved Bluetooth devices.
batocera-bluetooth list_blacklistable      Print the list of currently
blacklisted Bluetooth devices.
batocera-bluetooth trust                   Trust a new, randomly(?)
selected device. Good luck!
batocera-bluetooth remove <device address>      Removes (untrusts)
specified Bluetooth device from the list and removes its connection from
memory.
batocera-bluetooth save                    Saves the newly trusted
device to memory.
batocera-bluetooth restore                 Undoes any trusted but not
currently saved device and restores from memory.
batocera-bluetooth live_devices            Lists all current nearby
devices in pairing mode.
```

v33 specific commands

```
batocera-bluetooth blacklist <device address> <device name>  Adds a
device to the blacklist, preventing it from being paired to.
batocera-bluetooth unblacklist <device address>  Removes specified
Bluetooth device from the blacklist.
```