

# Linux driver for HC-06 Bluetooth module

## 1.1 Overview

This project implements a Linux device driver for HC-06 bluetooth module, which is usually used to give different microcontrollers a bluetooth capability (see reference manual for HC-06 bluetooth device: [HC-06 module reference manual](#)).

The whole *gist* of this project will be based around using this hardware module directly with Linux machine, i.e. PC, rather than a microcontroller with UART interface. Thus, for this purpose, we will write a Linux module for this bluetooth device in order to be able to load it, whenever we need it, and test it on some bluetooth compatible devices.

The board that we used in this project, where the actual HC-06 module is placed on is an Arduino compatible board with description of it being provided here: [HC-06 board description](#). From now on, in the context of this project, this board itself will be called as HC-06, for the convenience of it.

In order to communicate with the board, we have to be able to convert USB communication to UART, thus FTDI module is used for it: [FTDI board description](#).

## 1.2 Pinout and connection

## 1.3 Interfacing with HC-06 module via command line interface

We can interface with HC-06 module via passing it commands (including the fact that there is a FTDI module between our machine and HC-06 to be able to convert USB communication to UART), as it is mentioned here: [HC-06 board description](#) with actual commands being described here: [HC-06 commands](#).

## 1.4 Linux device driver

As it is mentioned in [FTDI board description](#), Linux already has a Virtual COM Port (VCP) driver by default loaded for FTDI, which are [ftdi\\_sio](#) and [usbserial](#) (although, in my Linux installation, which is Arch 6.12.4-arch1-1, [usbserial](#) module wasn't loaded and only [ftdi\\_sio](#) was loaded). Thus, if we want to write our own driver for FTDI, we would have to unload those [ftdi\\_sio](#) and [usbserial](#) modules (only [ftdi\\_sio](#) in my case, as [usbserial](#) is not even loaded on my machine).

Once we connect our FTDI module to our machine a device file can be found in `/dev/` directory, in my case, it is `/dev/ttyUSB0` . This way, we can communicate with our FTDI module via writing to and reading from `/dev/ttyUSB0` file.

## 1.5 Communication via terminal

As it is explained in the HC-06 datasheet, it expects commands, when it is in AT mode, in an interval of 1 second with no newlines and carriage returns required, thus we should send the commands as a string, rather than immediately sending each character, like `minicom` and `screen` do by default.

That is why, an easy setup is setting our devices permissions via `chmod o+rw /dev/ttyUSB1` and opening 2 terminals:

1. One with `cat /dev/ttyUSB1` , where we will see all the messages transmitted from the bluetooth module.
2. One where we can send our commands via `echo` , e.g.  
`echo 'AT+NAMEABCDE'>/dev/ttyUSB1` .

Another option is to use PuTTY with option **Local line editing** set to **Force on**, as it is explained here: Setting PuTTY to send data serially only on *Enter* key press.

## 1.6 Issue with AT command not working

Command `AT` , which should make bluetooth module send `OK` , doesn't work on my device, although, setting name with `AT+NAMEMY-HC-06` worked and I got `OKsetname` back from the bluetooth module. The similar issue is described in this thread: [issue with HC-06 AT command](#).

## 1.7 USB bandwidth

A good article that describes microframes, packet sizes, and intervals: [USB bandwidth](#).

The used USB device in our FTDI module is USB Mini-B, which is considered a high-speed USB device, just like USB 2.0, as it is illustrated on the chart here: [USB types and their speeds](#).