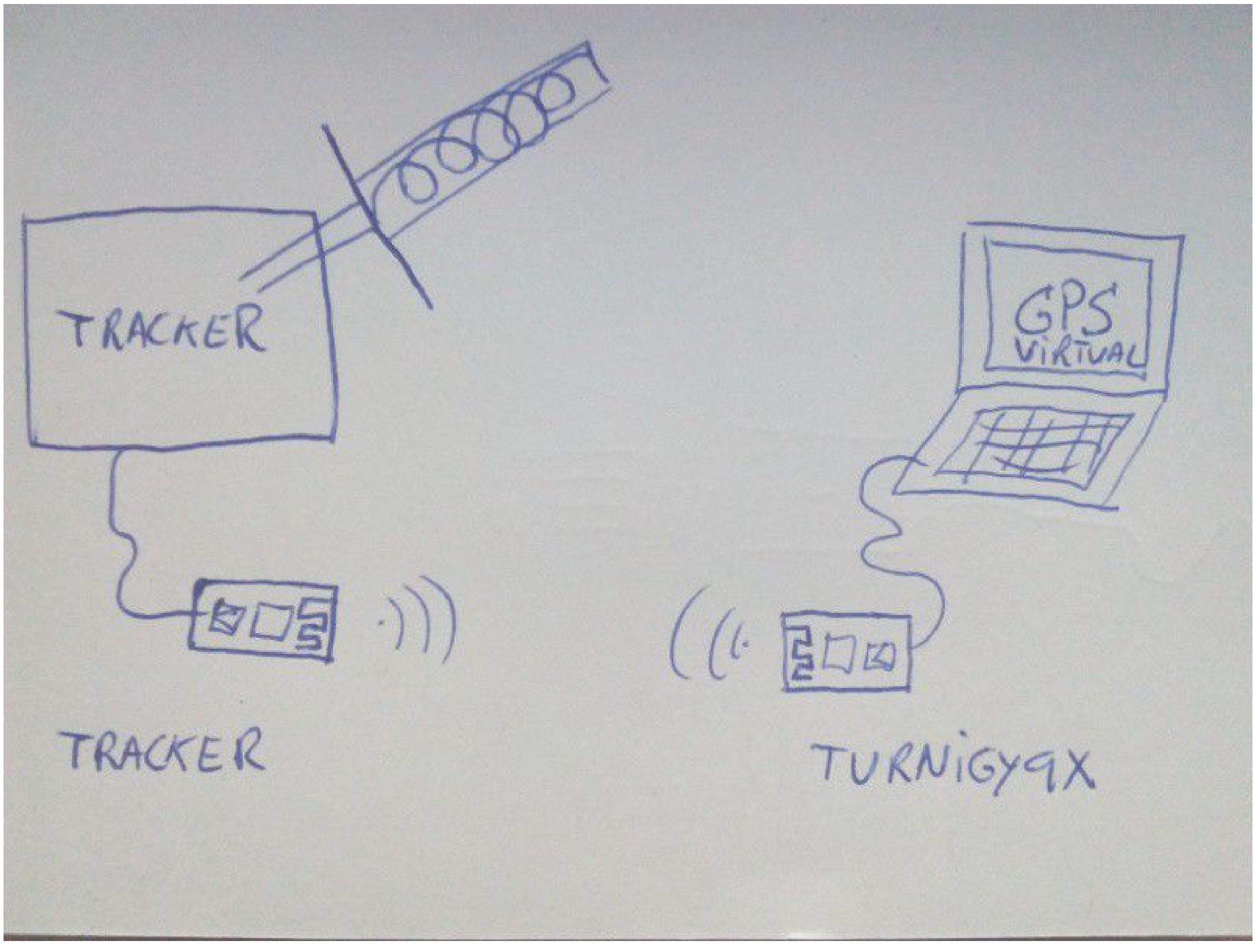
CONFIGURE BLUETOOTH MODULES FOR amv-open360tracker

This tutorial explains how to configure two Bluetooth modules, one master and another slave, to connect the amv-open360tracker to other devices to providing telemetry data necessary for AT to track.

The example that follows is a real case, in which we connect via Bluetooth the amvopen360tracker to the DJS FrSky module installed in a Turnigy 9x.

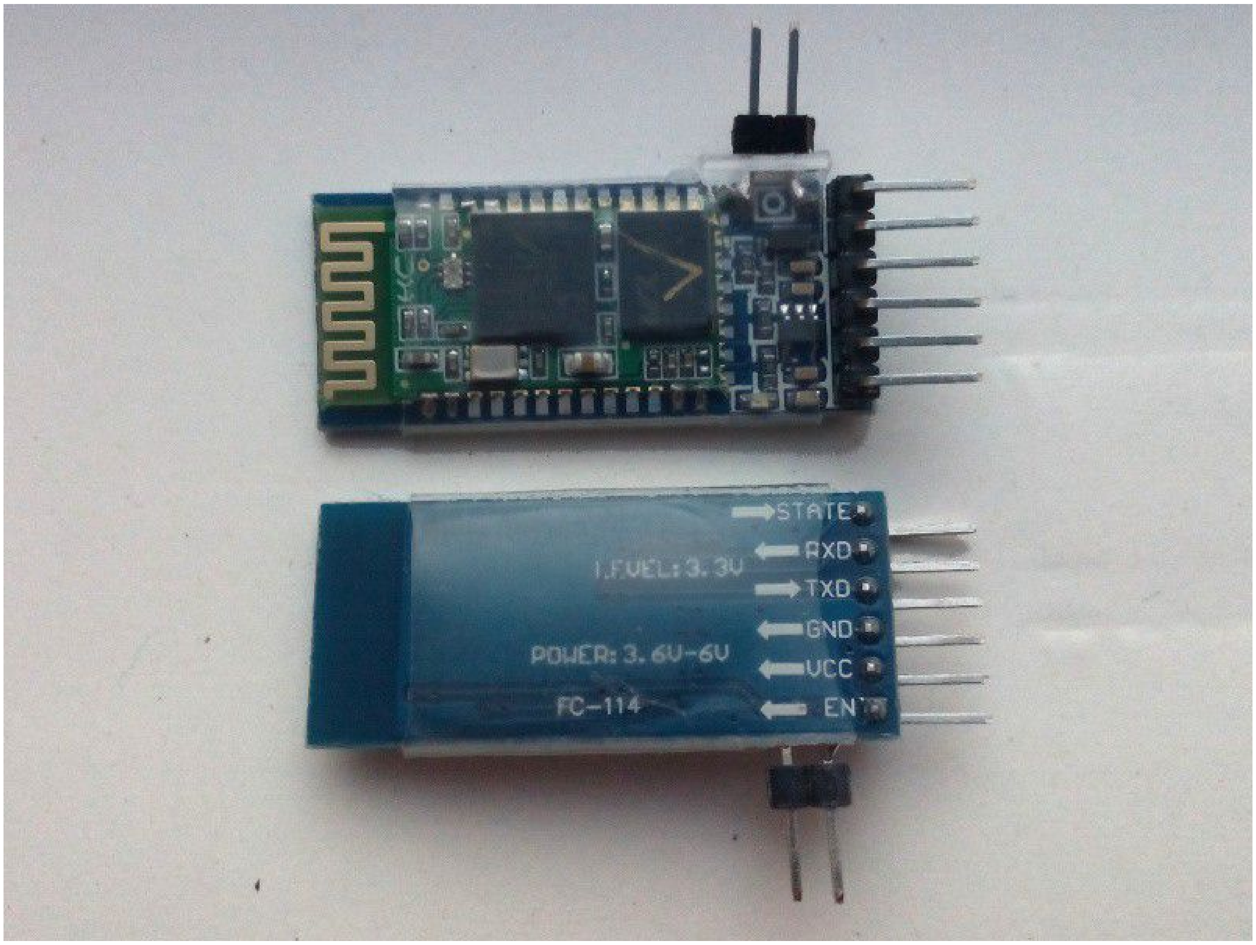


ACQUIRE MODULES HC-O5 / HC-06

There is a large variety and range of Bluetooth modules on the market. What to us we are interested in communicating our amv-open360tracker with the station, they are modules that meet these characteristics:

- That at least one of the two modules can be configured as Master (HC-05), being the other Slave (HC-06).
- That can be powered at 5 volts to connect to the CRIUS SE controller or some Arduino compatible board, such as Arduino UNO, Mega, Nano …

In my particular case, I have purchased two modules here that can be configured both as a Master and as a Slave through AT commands.



Note: I will use my Bluetooth modules to connect it to devices that supply 5v power supply, and works perfectly at the voltage levels of the TX / RX pins of the UART of Arduino and the CRIUS SE. One of them will be connected to the FrSKY DJT module that I have mounted on my Turnigy9X (after doing the corresponding mod). If you are going to connect modules Bluetooth with other devices, make sure that the TX / RX voltage and power levels are the right ones, otherwise you can burn something.

2. COMMAND AT MODE

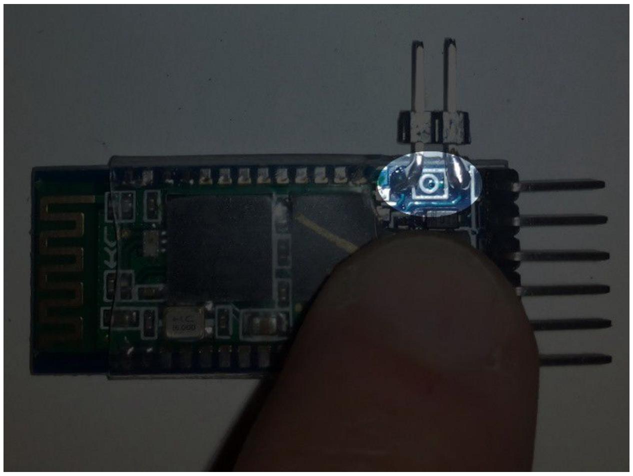
For our purpose, there are two types of Bluetooth modules to use: HC-05 and HC-06. The fundamental difference between both is the firmware they have installed, being able to configure the HC-05 in both master and slave mode, while the HC-06 only we could configure as a slave. All that in theory ...

In any case, the modules that we buy should be able to activate the mode of AT commands, and thus be able to configure the parameters involved in the communication, security, linked with other Bluetooth devices ...

Keep in mind that modules that can be configured as master support a greater number of AT commands than those that can be configured only as slaves, limited in the latter to very basic commands that only allow configuration the name of the device, the communication parameters and the key.

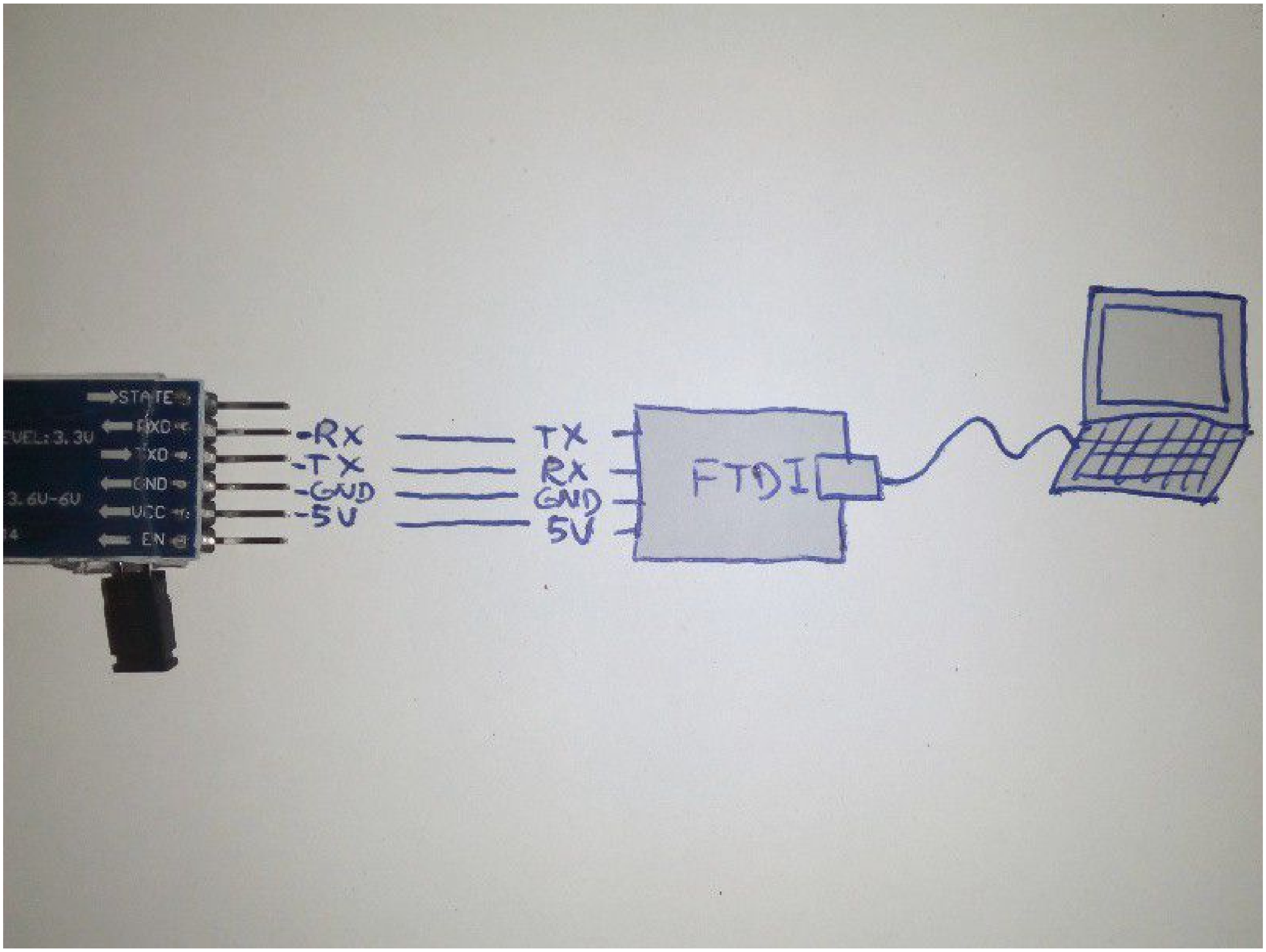
To be able to enter AT commands mode, we need our modules to have a button that allows us to enter this configuration mode. But in the market there is everything, those who have it, those who do not have it but in return incorporate pins, those who have it screen-printed but it seems that someone has forgotten to put the button ... even there are no They can not be configured with AT commands even if you try to solder directly on the board of the processor. Having the module's datasheet it could be elucidated which pads to weld, but still thus, if the firm is stuck, trying to put them in AT mode can be an odyssey. So the best is to buy them making sure they bring the little button already incorporated.
The variability is such that at first glance you do not know if what you have in front is an HC-05 or a HC-06, so I strongly recommend reading the specifications of the modules before buying it. It must clearly say if it can be configured using AT commands and if it is configurable as a teacher. And of course, let's be clear that we can work with him with 5 volt power supply and signal levels of the Arduino pins. If you have doubts always consult the seller information.
If we look at the next photo, we will see that the button does not come, but in the description of the product of the online store it was clearly indicated that master slave were configurable. Before I put my hand I consulted the seller and he solved the doubt. Yes they are configurable, but you have to solder the button.

What I've done is solder a couple of pins to use a jumper, with which I bridle when I want to configure it, and I remove it once configured.



Enter AT command mode

To do this, we place the jumper as shown in the photo, and connect our bluetooth module to the computer through a USB FTDI module. It is very important to place the jumper before providing power to the FTDI.



Once we connect it to the computer, the led of the bluetooth module will blink slower, in my case once every 2 seconds. This tells us that you have entered the mode of AT commands, and is ready to receive configuration commands.

But before starting to send orders, in Arduino:

1. We select the COM port corresponding to our USB FTDI module.
2. We launched the Series Monitor.
3. Select "Both NL & CR" in the drop-down box on the left.
4. We select 38400 baud (the baud rate will depend on each manufacturer of the bluetooth device when it enters AT command mode).

3. TRACKER BLUETOOTH MODULE: SLAVE

We will start configuring our module for the tracker, and we will put it in slave mode.
First we write the command:
AT
and we send it as is, without any parameter. The answer should be Ok if everything is correctly connected and the bauds are correct. A command that can be useful:
AT + VERSION
which in this case returns us
+ VERSION: 2.0-20100601
With what we know the version of the firm that has been loaded, we can use it to search internet more information on the network about the module that we have and the commands for Configure.

Name of the module
In order to distinguish one module from the other, it is convenient to give each one a name. In this case let's put the name AMV-OPEN360TRACKER to the bluetooth module for the tracker. The command is:

```
AT + NAME = AMV-OPEN360TRACKER
```

To verify that the name has been changed correctly:

```
AT + NAME
```

Communications parameters
We are going to configure the basic communication parameters of the serial port below, so that the CRIUS SE is understood with the bluetooth module, that is, we are going to configure the UART of the bluetooth module:

```
AT + UART = 9600,0,0
```

The two zeros separated by comma indicate, the first we use 1 stop bit, and the second that we do not use parity. To verify that the configuration has been changed correctly:

```
AT + UART
```

Password
The Bluetooth modules usually come configured with a password for defects, which It is usually 0000 or 1234 usually. To prevent our tracker from being linked by "intruders", or simply avoid the possibility of linking with other Bluetooth modules, it is convenient to change the password. The command is:

```
AT + PSWD = 6677
```

This same password will be configured in the other Bluetooth module.
Slave Mode For the module of the tracker to work as a slave and can be linked from the other module. In case you have an HC-06 module that can not be configured as we will use the command:

```
AT + ROLE = 0
```

To check it, we type:

```
AT + ROLE
```

With all this, we would have the module for the configured tracker. Let's review now the list of all the commands that we have used:

```
AT + NAME = AMV-OPEN360TRACKER
AT + UART = 9600,0,0
AT + PSWD = 6677
AT + ROLE = 0
```

4. MODULE BLUETOOTH OF THE ISSUER: MASTER

Although in principle we are going to use the module to connect it to the computer and send the frames from the NMEA simulator, let's configure it as if we were to connect it to the module FrSky DJT of the station. We will configure this module in Master mode, and the configuration procedure is similar to the one described for the tracker module, with some difference in the values, and using these lines:

```
AT + NAME = TURNIGY9X
AT + UART = 9600,0,0
AT + PSWD = 6677
AT + ROLE = 1
AT + CMODE = 1
```

The command AT + ROLE = 1 is the one that configures it as a master. And the command AT + CMODE = 1 makes this module can be automatically connected to the bluetooth module which receives a better signal, as long as their passwords are identical, of course (it is for that said to put passwords different from the ones it brings by default).

With all this, we would have our modules ready to link automatically and establish a serial communication channel at 9600 baud. To use it remove the jumpers and connect each one to the corresponding device. In the case that you have purchased other modules, it is very likely that these commands serve. If not, you would have to locate the manufacturer's information to find the equivalent AT commands. There are other ways to configure the modules, this is just one of them.