

Bluetooth-based firmware upgrade for the CoaX helicopter

Cédric Pradalier (cedric.pradalier@skybotix.com)

July 14, 2010

1 Purpose of this document

This document describes the procedure for upgrading the firmware of the CoaX helicopter using the bluetooth bootloader. It assumes a global knowledge of CoaX helicopter and its hardware, as described in http://www.skybotix.com/support/wiki/index.php/Main_Page.

The bootloader uses a bluetooth protocol emulating a wireless serial link (RFCOMM). Most of these functions have been heavily tested under Linux. Under windows, the support of bluetooth serial device is not extremely reliable, but the Python GUI has been seen working.

2 Using the Python GUI

Note that older CoaX boards (id \geq 30) have been delivered with a version of the bootloader firmware that is not compatible with the Python GUI. We strongly recommend upgrading the bootloader firmware in this case. This is explained in section 3.

2.1 Establishing the bluetooth serial connection under linux

The first step is to identify the address of the bluetooth device installed on the CoaX helicopter. The hcitool command can be used (on some systems you may need to prefix it by sudo, or even be root to use it):

```
#>hcitool scan
Scanning ...
10:00:E8:C0:AC:C8 CoaX_0041
```

The 6 numbers on the left are the address we are looking for. The rfcmm command can then be used to establish the connection (again, you may need to be root to run it):

```
#> rfcomm connect 0 10:00:E8:C0:AC:C8
Connected /dev/rfcomm0 to 10:00:E8:C0:AC:C8 on channel 1
Press CTRL-C for hangup
```

The argument “0” of rfcomm is the channel number. This channel will be linked to a device /dev/rfcomm0 (the zero is the same). The last argument is the device address.

During the first connection, the bluetooth system will require to enter a password. For each CoaX board, the password is the 4-digit number corresponding to the board id. For instance, the board CoaX_0041 has the string ”0041” for password. See figure 1 for an example.

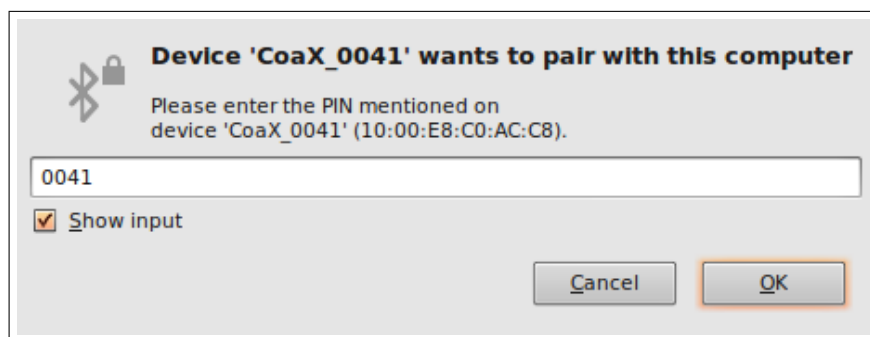


Figure 1: Authentication under linux (ubuntu 10.04).

When the connection is not needed anymore, the connection can be closed by hitting Ctrl-C.

2.2 Establishing the bluetooth serial connection under windows

Using the bluetooth panel should show the coax device. Double clicking on it will initialise the connection process and require the device password. For each CoaX board, the password is the 4-digit number corresponding to the board id. For instance, the board CoaX_0041 has the string ”0041” for password.

Once the connection is established, Windows has created a COM device. You need to open the device manager and search the serial port section for a new device. The name will be COM plus a number, usually greater than 4.

2.3 Uploading the firmware: principle

The next steps will assume that a bluetooth serial connection to the CoaX board has been established. It has been heavily tested under linux, and has been seen to work under windows XP.

The principle of the upload process is as follows:

- The bootloader preload a .hex file containing the firmware to upload. These files can be obtained from the Skybotix download page, or compiled by hand using gcc¹ or the Microchip development environment under windows.
- When the CoaX board boots, it sends a character indicating that it is ready to receive a firmware and waits a second for a proposal. If no proposal is received during this time, the boards goes on and runs the current firmware.
- If a firmware proposal is proposed, the board goes into the upload mode, and waits for the firmware packets. The PC application starts then to send the firmware packets, first erasing the existing memory, and then writing the new firmware.
- When the upload is completed, the board reboot itself, and starts the new firmware.

2.4 Uploading with the GUI

Under linux, the GUI can be started by going to the pybootloader directory (from the CD or a svn checkout), and running the BootloaderGUI.py program. Depending on your system, this may require to be prefixed by sudo. The gui is depicted in fig. 2.

The GUI is composed of several elements as detailed below:

- The file drop-box provides a way to select the firmware file. It must be a .hex file.
- The port entry should be modified to reflect the port used by the serial connection (usually /dev/rfcomm0 or COMxx).
- Flow Control must be checked when using a Xbee connection instead of the bluetooth. In this case, the Xbee module is generally visible on /dev/ttyUSB0. This has not been successfully tested under windows.
- Send Reset must not be checked on the CoaX board.
- The Reload button is used to reload a firmware file, for instance after recompiling it.
- The Check button will try to open the serial port and report success or failure.
- The Upload button starts the upload process once the file and the port have been configured. When the upload process starts, it first (re-)read the .hex file, then open the serial port, and wait for the ready token on the serial channel. This is triggered by hitting the reset button on the CoaX board.

The following text output shows some debug message output during the upload process. The sequence of “!_” indicates that the GUI is waiting for the board to send the “ready” token. Once this is started, the “X” symbols correspond to the packets requesting a memory erase on the dsPIC, whereas the ‘#’ symbols correspond to the packets containing the new firmware to write.

¹Version modified by Microchip for the dsPIC

3 Upgrading the bootloader firmware

The bootloader firmware can be found on the Skybotix download page. It must be programmed on the main dsPIC using the programming board and a Microchip programmer (ICD2 or ICD3). Once the bootloader is written to the dsPIC, then the procedure described in the previous section must be used to upload a firmware.

4 Using coaxuploadbt

This method only work for linux, for older version of the bootloader firmware (board id ; 30). The coaxuploadbt executable can be found and compiled in the btbootloader directory (from the CD or on the SVN repository). To compile it, just run make in the btbootloader directory. This program depends on the libbluetooth package (including the development files).

Running the uploader on the command line is done as follows:

```
#>./coaxuploadbt controller-coax002x.hex 41
coax bluetooth uploader (using libbluetooth) version 2
(c) 2006 - 2009 Olivier Michel, Stephane Magnenat, Cedric Pradalier
Program located at 000200
Upload size: 307 packets 19648 words (58944 bytes)
Added 39 erase packets
Scanning bluetooth:
10:00:E8:C0:AC:C8 CoaX_0041
Contacting Coax 41
Press enter and then reset on the robot
```

.....

Note that this method does not require to establish the serial connection beforehand. It will fail on CoaX boards equipped with the newer bootloader firmware due to a significant change in the bootloader protocol.