



White Paper

Using the iCON711 on Linux

About this document

Confidentiality

All data and information contained or disclosed by this document is confidential and proprietary of Option NV, and all rights therein are expressly reserved. By accepting this document, the recipient agrees that this information is held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without prior and written permission of Option NV.

Version History

Date	Version	Author	Revision	Approval	Remarks
Oct 28, 2011	v001draft	F. Arboleda	<i>Not reviewed</i>	<i>Not approved</i>	Initial version
Oct 28, 2011	v002ext	F. Arboleda	P. Vandeneede		Review

Table of contents

1	INTRODUCTION	3
2	MANUAL PROCEDURE	4
2.1	Switching from CD mode to MODEM mode	4
2.2	Link the usbserial driver	4
3	AUTOMATIC PROCEDURE	7
3.1	Only using a udev rule	7
3.2	Loading the driver at startup and using a udev rule	8

1 INTRODUCTION

The purpose of this document is to explain how to configure the Linux system to work together with the iCON711.

The explanation on this document assumes that the user has some experience using Linux shell commands and that all the necessary commands are available for the user. An active internet connection might be necessary.

The following explanation has been written using Ubuntu 11.04 (Kernel 2.6.38-11-generic).

2 MANUAL PROCEDURE

2.1 Switching from CD mode to MODEM mode

When connecting the iCON711 into the Linux host system, the device will appear in the system with product ID 4007. This product ID refers to the iCON711 on CD configuration, but since the CD does not contain an installer or drivers for Linux the user will not be able to use the device as a modem.

The user can check this product ID by executing the following command on a Linux terminal:

lsusb|grep Option

The command will respond with a line like the following:

Bus 001 Device 006: ID 0af0:4007 Option

This line mentions the vendor ID (0af0) and the product ID (4007) for the iCON711.

The user needs to switch the device from CD mode (4007) to MODEM mode (4005). This can be done by executing the following command:

sudo usb_modeswitch -v 0x0af0 -p 0x4007 -S -R -W

Note that this command will give an error at the very end of the execution mentioning that the control message sent failed. This error is normal and will not affect the actual use of the device in the system.

If at this moment the user check again for the device ID's in the system using the ***lsusb|grep Option***, we will find that the device has switched from product ID 4007 to product ID 4005:

Bus 001 Device 007: ID 0af0:4005 Option

The product is now in MODEM mode and is ready.

2.2 Link the usbserial driver

The last step to get the modem to work in Linux is to link the ***usbserial*** driver to this MODEM device. This is needed because the ***usbserial*** driver does not know that it can handle this specific device.

Note that in some system configurations the **usbserial** driver is already in use (e.g. if another serial device is already attach to the system). In this case, we need to first remove the current module and then reload it again with our parameters.

To verify whether the **usbserial** driver is already loaded in the system, the user can execute the following command:

lsmod|grep usbserial

If there is any answer to that command, it means that the module is already loaded. The answer of an already loaded module looks like this:

usbserial 37116 0

To remove the module the user needs to execute this command:

sudo rmmod usbserial

The user can now restart the **usbserial** driver using the following command and parameters:

sudo modprobe usbserial vendor=0x0af0 product=0x4005

The device is now installed on the host system and the user should be able to see the ttyUSB interfaces on the host PC:

ls -l /dev/ttyUSB*

```
crw-rw---- 1 root dialout 188, 0 2011-10-28 12:00 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 2011-10-28 12:00 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 2 2011-10-28 12:00 /dev/ttyUSB2
crw-rw---- 1 root dialout 188, 3 2011-10-28 12:00 /dev/ttyUSB3
```

The answer to the same command before linking the **usbserial** driver to our device would have been:

ls: cannot access /dev/ttyUSB*: No such file or directory

It is important to mention that the device will always enumerate in the same order. In other words, the user can always be sure that every interface of the device is always mapped to the same **/dev/ttyUSB** interface. The actual correspondence is as follows:

```
/dev/ttyUSB0 -> Diagnostic port
/dev/ttyUSB1 -> Modem port
/dev/ttyUSB2 -> Application port
/dev/ttyUSB3 -> NDIS port
```

Out of these four ports, there are only two that are accessible to the user. The other two are not supported by the **usbserial** driver. The two working ports are:

/dev/ttyUSB1 -> Modem port
/dev/ttyUSB2 -> Application port

The application port can be use as an AT command port.

The user has to make sure that no other application is using these ports to actually be able to use them. Examples of applications that may be using these ports are the **network-manager** and the **modem-manager**.

Note that the numbers of the **/dev/ttyUSB** interfaces may change depending in the configuration of your host pc. For example if another **usbserial** device is attached to the system, it will also create a **/dev/ttyUSB** interface that is not related to our device.

3 AUTOMATIC PROCEDURE

In the previous section, a manual procedure had been explained. This manual procedure is OK when the user is trying the solution for the first time, but it can become very tedious if the user has to perform all the steps every single time he want to make a test. There is a way to add all these rules into the system so they are performed automatically.

There are two ways of doing this. These two methods are explained on sections 3.1 and 3.2.

The main difference between these two methods is:

With the First method (section 3.1): the user needs to ensure that the **usbserial** driver is not loaded before the device is inserted, but the user will have access to the uSD card. Four interfaces will be enumerated as explained on the previous section.

With the Second method (section 3.2): the system will load the **usbserial** drivers correctly at startup so the user can keep all the serial interfaces connected on the host system, but the user will not be able to access the uSD on the iCON711. The device will enumerate two more interfaces (6 instead of 4) that will not be accessible by the user. The order of the first four interfaces remains as explained on the previous section.

3.1 Only using a udev rule

Step 1: Make sure the **usbserial** driver is not loaded on the system prior to the connection of the iCON711 (Do not connect the device yet). Don't connect any other device in the system that may use the **usbserial** driver until the modem has been installed.

Step 2: Create a new rule file under **/etc/udev/rules.d/** containing the following two lines (See Ref 1):

```
ACTION=="add", SUBSYSTEMS=="usb", ATTRS{idVendor}=="0af0",  
ATTRS{idProduct}=="4007", RUN+="/usr/sbin/usb_modeswitch -v  
0x%x{idVendor} -p 0x%x{idProduct} -S -R"
```

```
ACTION=="add", SUBSYSTEMS=="usb", ATTRS{idVendor}=="0af0",  
ATTRS{idProduct}=="4005", RUN+="/sbin/modprobe usbserial  
vendor=0x%x{idVendor} product=0x%x{idProduct}"
```

Step 3: Restart the udev rules by executing the following command:

sudo udevadm control --reload-rules

Step 4: Connect the device into the Linux host system. After this step, the user should be able to see the device interfaces in the system.

3.2 Loading the driver at startup and using a udev rule

Step 1: Adding a line to **/etc/modules** to load the **usbserial** driver at startup. The user needs to append the following line into the **modules** file (See Ref 2 for an example of the file) :

usbserial vendor=0x0af0 product=0x4005

Step 2: Create a new rule file under **/etc/udev/rules.d/** containing the following line (See Ref 3):

```
ACTION=="add", SUBSYSTEMS=="usb", ATTRS{idVendor}=="0af0",  
ATTRS{idProduct}=="4007", RUN+="/usr/sbin/usb_modeswitch -v  
0x%s{idVendor} -p 0x%s{idProduct} -S -R"
```

Step 3: Restart the system.

Step 4: After the restart, connect the device into the Linux host system.

References

Ref	Document	
1	98-icon711_sol1.rules	Option
2	modules	Option
3	98-icon711_sol2.rules	Option