



applicomIO® Linux solution 3.0.0



-- Documentation --

SUMMARY

1. Before to start	3
2. Features & Architecture	3
2.1. Features	3
2.2. Architecture	3
3. Build application driver for your kernel.....	4
4. Board configuration & initialisation	4
4.1. Initialisation through serial connection (except PCIIBSIO.....	5
4.2. Initialisation by the network using Remote TCP/IP.....	5
4.3. Initialisation with the recorder/player feature	5
4.4. Initialisation with dual boot system	6
4.5. Initialisation by downloading in flash under Windows NT	6
5. Remote TCP/IP mail server.....	6
5.1. Configuration of the client side	6
5.2. Configuration of the server side	7
6. Stop Remote TCP/IP mail server and deamons	7
7. Starting the oem solution.....	8
8. Build your application for applicomIO®	8

1. Before to start

For the all manipulation, it's necessary to use a konsole in **super user** mode.

2. Features & Architecture

2.1. Features

The *Linux solution* allows you to develop and to run your own applications using the applicomIO® API. The *Linux solution* contains the different layers (kernel drivers, API and utilities software). All the supplied layers are already built (on Linux kernel 2.6 with glibc 2.3.3) and can be used without recompilation except the driver layer.

However, the kernel driver source-codes are supplied to be able to recompile them for you own Linux system. The driver directory contains all the source files and the makefile needed to recompile this driver. This documentation doesn't include the description of the applicomIO APIs. For this, refer to the product documentation under Windows (Reference Manual).

2.2. Architecture

Two different kinds of mechanism are used to communicate with the applicomIO boards.

- The first mechanism is named "**mail access**". It allows to send some messages to the board and to wait for answers. This mechanism is used by the following features:
 - The applicom® API (applicomio.a)
 - The initialisation process (*with the player tool*)
 - Some diagnostic tools (*with pcmon, etc*)
 - Few functions of the applicomIO API (free messaging mode functions only)
 - Remote TCP/IP mail server

The "mail access" mechanism is implemented inside the "**User library**" named **applicomio** (applicomIO product).

- The second is named "**I/O access**". It allows getting directly the I/O data in the dual port memory of the board. This mechanism is used by the following features:
 - The applicomIO® API

The "**I/O access**" mechanism is implemented inside the "**I/O library**" named **appio**.

For applicomIO product, according to your needs, two "**mail access libraries**" are available on this CD-ROM. So, you can use:

- A mono-process solution (in the sub-directory **monoprocess**)
 - This solution allows one and only one application to use the "**mail access**". For this solution, you don't need to start and use the two serialization daemons.
- A multi-process solution (in the sub-directory **multiprocess**)
 - This solution allows several applications to use the "**mail access**" simultaneously. For this, you need to start and use the two serialization daemons.

Notes: - This previous information is about the "mail access libraries only". The "I/O access" library (appio) is usable by **only one soft logic application**.

- All the tools and libraries are supplied compiled

3. Build application driver for your kernel

Before recompile the driver "applicomio.ko", you have to change the version of the kernel in the file Makefile.

```
*****
obj-m += ../src/applicomio.ko
EXTRA_CFLAGS += -I$(PWD)/../../apincl -Ilib/modules/2.6.4-52-default/build/include -O
EXTRA_CFLAGS += -DMODULE -DU_DIRECTORY_SLASH -DDEBUG -D__KERNEL__ -
DOS_UNIX -DOS_LINUX -DPRODUCT_IO -DTYPE_APPLICATION -DTYPE_DRIVER -
DADK_KERNEL -Wall -Wstrict-prototypes -fomit-frame-pointer -O2 -fno-strength-reduce -pipe -g -
falign-loops=2 -falign-jumps=2 -falign-functions=2
default:
$(shell rm *.ko)
$(MAKE) -C/lib/modules/2.6.4-52-default/build/ SUBDIRS=$(PWD) modules
$(shell cp ../src/applicomio.ko applicomio.ko)
```

For know the version of the kernel , write the command "uname -r" in a Konsole and replace in the file Makefile your own version of the kernel .

And call the command "make" in a Konsole to build the driver "applicomio.ko".

4. Board configuration & initialisation

No tool is available today for the board configuration under Linux. So, the configuration part is done under windows, from a computer containing a version:

- **applicomIO 3.0** or upper for the applicomIO product.

Several ways to configure the boards are possible regarding on your needs and on your material.

To initialise your boards, use one of this methods:

4.1. Initialisation through serial connection (except PCIIBSIO and PCI-DPIO boards) :

- Step 1 (on Linux):
 - o Insert the board on your Linux machine.
 - o Connect the serial channel with the COM port of your NT computer.
- Step 2 (on WinNT):
 - o Start the applicomio configuration **console**.
 - o Choose the remote (serial link) mode in the configuration manager.
 - o Configure the board and the network.
 - o Download the configuration in the flash.

The Initialization will be automatically executed from the flash each time the power is switched on.

Note : With this method, you can also use all the configuration and diagnostic tools remotely (network configuration detection, network diagnostic...) exactly as if you are under Windows.

4.2. Initialisation by the network using Remote TCP/IP mail server

See section 5

4.3. Initialisation with the recorder/player feature

- Step 1 (on WinNT):
 - o Start the applicomio configuration console.
 - o Choose the simulation mode in the configuration manager and select the initialization recording feature
 - o Configure manually the board and the network
 - o Start the Download in flash or **PCINITIO** command. A file named **applicomioflash.ply (applicomio.ply)** will be generated
- Step 2 (on Linux):
 - o Transfer the generated player file (TCP/IP, disk ...).
 - o Play this file in the applicomIO board with the **playerio** utility software.

Notes: - With this method, you must use the configuration tool in simulation mode.

So, you cannot use the network detection feature during the configuration step.

- If **applicomioflash.ply** is used the Initialization will be automatically executed from the flash each time the power is switched on.

4.4. Initialisation with dual boot system

- Step 1 (PC started with WinNT):
 - Install the board on the computer.
 - Configure the board and the network.
 - Initialize the applicomIO board with **PCINITIO**.
- Step 2 (PC started with Linux):
 - Reset the computer without cutting the power.

Note: The configuration must be played in the board each time the power is switched on.

4.5. Initialisation by downloading in flash under Windows NT

- Step 1 (on WinNT):
 - Install the board.
 - Configure the board and the network.
 - Download the configuration in the flash.
- Step 2 (on Linux):
 - Insert and use the flashed board.

Note: The Initialization will be automatically executed from the flash each time the power is switched on.

5. Remote TCP/IP mail server

This component allows you to do the following operations through your Lan Network:

- configuration with the applicomio console,
- board(s) initialisation with pcinitio tool,
- diagnostic with diagnostic tool,

from a Windows system to your applicomio solution running on a Linux system.

The communication takes place using a standard network adapter and over the TCP/IP protocol and to enable this feature, the user has to configure both the client side (Windows console) and the server side (Linux system).

5.1. Configuration of the client side

Start the applicomio configuration console and choose "Configuration Manager" from the "File" menu. Create a new configuration pressing the "new" button, select the "On Remote machine (LAN TCP/IP)" mode of the configuration tool and specify the TCP/IP connection parameters:

1. PC name or IP address,
2. TCP port (should be the same as server side) by default set as 5002,
3. Select the "Automatic Transferring of the Configuration to the Remote Machine" check box, if you want to save a copy of configuration files in the Linux PC,
4. Select (by default) the "Record Initialization on Disk" to generate the ".ply" file(recorder / player feature). This file will be transferred on Linux PC if you have check "Automatic Transferring of the Configuration to the Remote Machine",
5. Restart the applicomio configuration console.

Notes: - If you check "Automatic Transferring of the Configuration to the Remote Machine", your configuration will be saved on the Linux PC during the board(s) initialisation.

- For retrieving the configuration from the Linux PC, select "Imported configuration" from the "File" menu.

5.2. Configuration of the server side

When you launch the server, you have to precise :

- the path where the configuration files will be saved,
- TCP port (should be the same as client) by default set as 5002
- the IP address of the PC client (*).

Example: `./servertcpipio ./config 5002 192.168.7.125`

(*) The IP address is an protection option. If you set this parameter, the Remote TCP/IP mail server will accept connection only from this IP.

6. Stop Remote TCP/IP mail server and deamons

- For stopping the server, read the server ID with the command "ps -A" and after run the command : `kill -3 "server ID"`

- For stopping the deamons, you must run the clearaccessio module with the option "deamons".

Example: `./clearaccessio deamons`

7. Starting the oem solution

Before running your application, you have to start applicomIO® product. To do this, follow these steps:

For *applicomIO Linux solution*:

- Start applicomIO® driver: in the driver directory, use: **insmod applicomio.ko** (*)(**)
- In the case of a multi-process solution, start the **apwriteio** daemon: in **Deemons** sub-directory, use: **apwriteio &**
- If not already made (flashed configuration in boards), initialize your boards (Tools: playerio)
- Now, you can run your application.

(*) For this, you have to compile first the corresponding driver provided on the applicom Linux CD and create the devices access to the applicom board with the following instruction (Note : do not use the driver provided with the sources of your linux kernel) :

```
.\> mknod /dev/ac c 10 157
```

(**) For ISA or PC104 boards, when you are loading the applicomIO® driver you have to add the irq number of the boards. All applicomIO® installed on a same computer are sharing the same interrupt number. To define this interrupt number:

```
insmod applicomstd.ko irq=5
```

8. Build your application for applicomIO® solution

In first step, you have to know what kind of global solution you need and what is the rule of the application you are compiling.

If your application uses:

- **IO data access** (see windows product documentation), you have to:
 - Include the header file [appio.h] in your source code to call the IO data access API
 - Link with the library [appio.a] in your makefile.
- **Messaging mode** (see windows product documentation), you have to:
 - Include the header file [applicomIO.h] in your source code to call the Messaging mode API.
 - Link with the library [applicomio.a] in your makefile.

If **only one** application uses the messaging mode (or any other feature based on "**mail access**") simultaneously:

- You can use the files contained in the directory [\monoprocess\aplib\io]

If **several** applications can use the messaging mode (or any other feature based on "**mail access**") simultaneously:

- You must use the files contained in the directory [\multiprocess\aplib\io].

• In this case, before to start your application, you have to start the daemon [apwriteio] in the directory [\multiprocess\daemons]. The command id [\apwriteio &]. This daemon will start automatically another daemon named [apreadio]. These two daemons are needed to serialize the board access (for mail access mode)