# Quick Start Guide

Thank you for choosing the **Wifibot Lab** platform for your robotic application.

- Before using the platform, please read with care this manual
- Keep this manual in a safe place for any future reference
- For updated information about this product visit the official site of wifibot http://www.wifibot.com

## Index

## Package contents

Make sure to be in possession of all the articles mentioned below. If any of them should be missing, contact your reseller as soon as possible.

**Platform + SBC CPU**
**Pan & Tilt IP camera or Web Cam**
**Battery charger**
**Wifibot CDROM**
**Camera CD-ROM and documentation**
**1x charging cable**
**1x WIFI Access point**
**4 wheels and a screw driver**

## Quick start

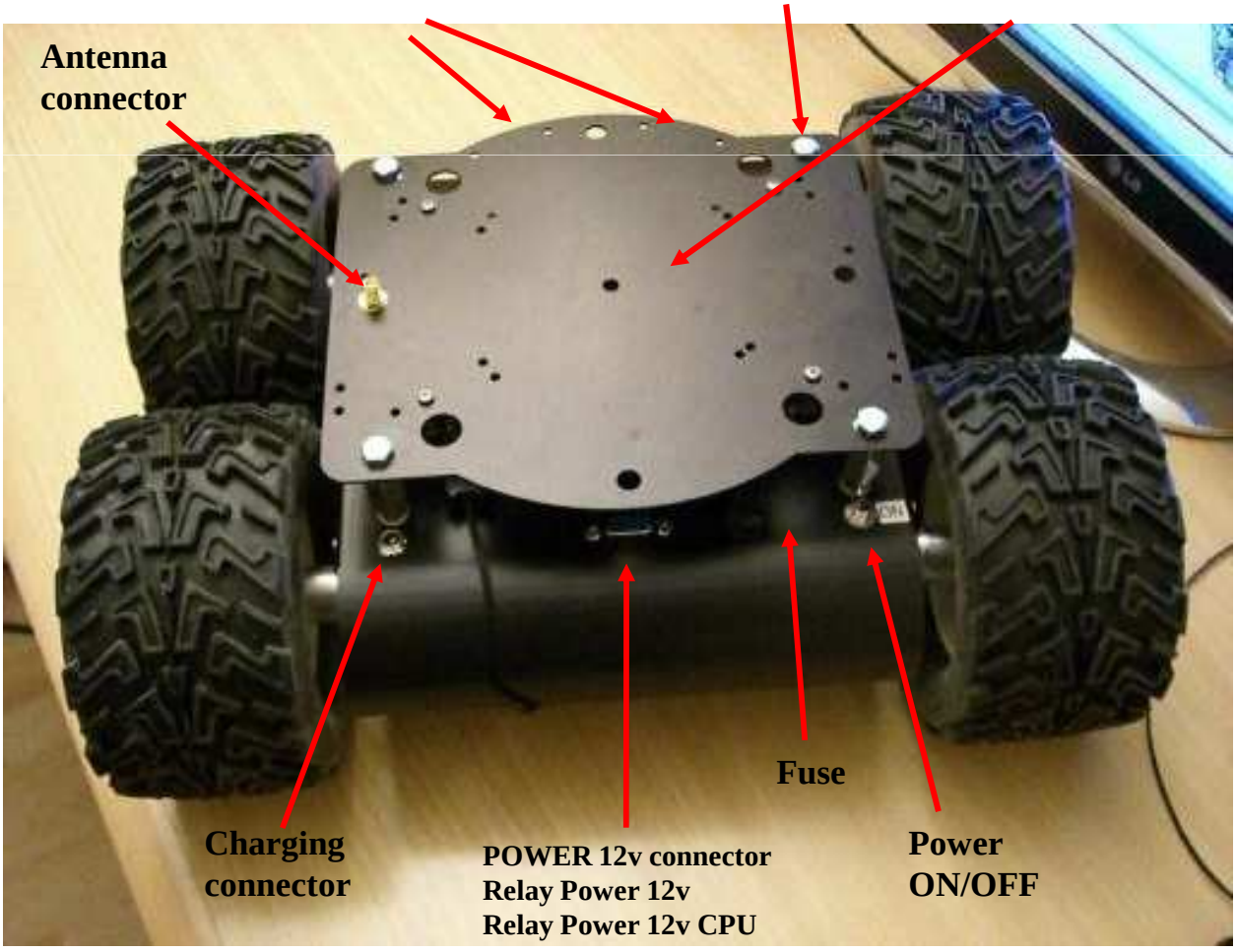
**1**- Install Simple GUI (see page 11).
**2**- Switch ON the robot or robots.
**3**- Switch On the Access point
**4**- Set you IP settings (see page 12) for example: 192.168.1.25 mask 255.255.255.0 or use DHCP
**5**- Connect to the robots' ad-hoc network (see page 13)
**6**- Launch GUI for controlling the robot

**Support + CPU Atom D510**

## Platform overview

**IR sensors**

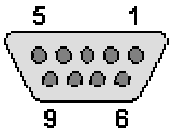
**15v DC Power connector**



# Platform interfaces

**The GND and 12V bat power output:**

A GND and a 12V non stabilised but filtered battery output are accessible at a **female** DSUB-9 connector located at the rear of the robot. Pin 1-2 are 12V, pin 4-5 are 12V for the CPU and 6-7-8-9 are GND output respectively. 12v an can give a maximum of 6A. An incorrect use of this connector beyond those values (short circuit or other) can provoke a malfunction of the platform or of the DC/DC converter and even damage it.
Pin 3 is a 12v controlled from the RS232 or PC.

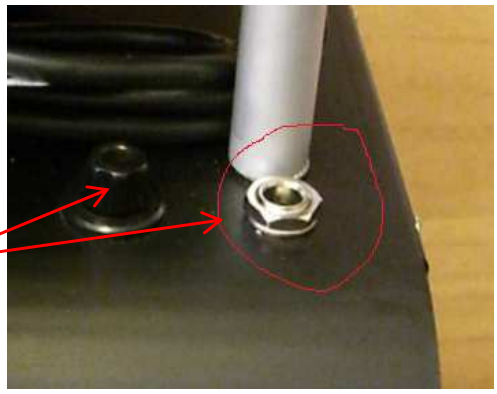




**CPU power connector :**

To power the CPU (8 to 24v)

**The charging connector:**

This connector located on the left at the rear of the robot, presents directly the + and – of the platform batteries. When charging the platform make sure the power switch is OFF as it drives a relay for connecting the connector to the batteries. If you want to power the robot externally with a protected power supply without his battery, just take off the fuse and use a 12v power supply or plug a DC 15v in the front connector.



**The DC 15v connector:**

You use this when you want to power up the robot from an external DC source (15v). An smooth commutation (even when the embedded pc is ON) occurs. After that the charging connector is free to be used for charging. Never power on the robot when charging and no dc 15v is present.

**The ON/OFF switch:**

The platform is switched ON and OFF by the interruptor located on the left at the back of the platform.

**The Fuse:**

The 10 Amp fuse is located on the right at the back of the platform.

## Interface connector:

This DSUB-9 **male** front connector presents a mix of input-output signals. The pinout is the following: (ADC are from 0 to 3.3v tolerant)

1- 5V

2- ADC IR Sharp Left

3- ADC IR Sharp2 Right

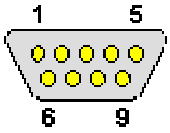
4- extra ADC

5- extra ADC

6- TX

7-8- GND

9- RX




**The antenna connector:**

This is the wi-fi antenna connector. Screw the antenna carefully on the connector till the end.

**Charging the robot:**

A battery charger is included with the platform. First make sure the platform and the charger are OFF, then connect the plugs of the charging cable (first on the charger and after on the robot) and finally switch the charger ON, check if your are in NIMH mode and press the green button for 5 second. The charger will stop


**Caution:**

Charge the robot at 3.8A on regular use.

Never discharge deeply the robot (around 0v).

When the robot is completely discharged, you will need some time to launch 2 times the charger, he will first charge the battery for 15 minutes and stop charging and you need then to relaunch the charger again, the charger will stop at 9400mAh automatically.

Charge the robot on a open area away from inflammable objects.

Do not let the battery without charging for more then 6 months, if so they can become dangerous.

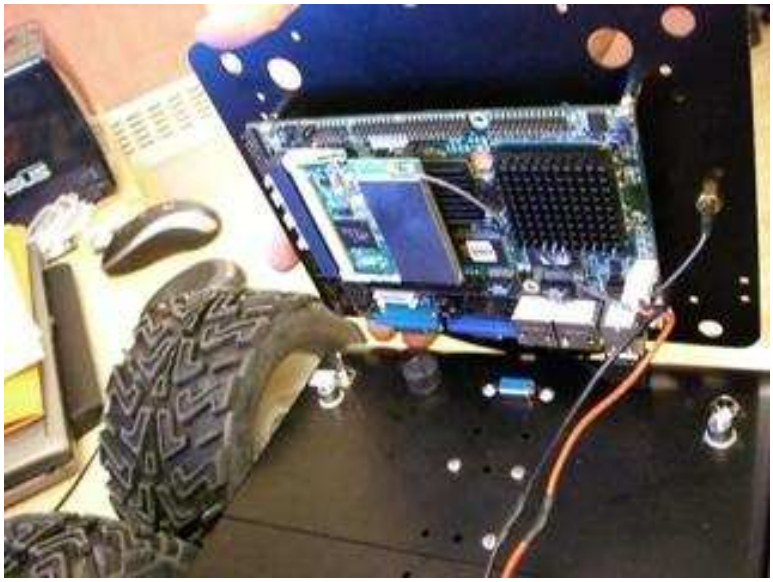
# Computer and camera installation

The platform is sold with an **Pan & Tilt IP camera or webcam and an embedded computer** which model can vary depending of the version. Those are independent elements from the platform which can be replaced by any other model. For more information about your particular camera and embedded computer please refer to their respective manuals included in the CD ROM of the robot. The top aluminium support witch is **already mounted** on the platform, has been thought for the fixation of those and other user components.

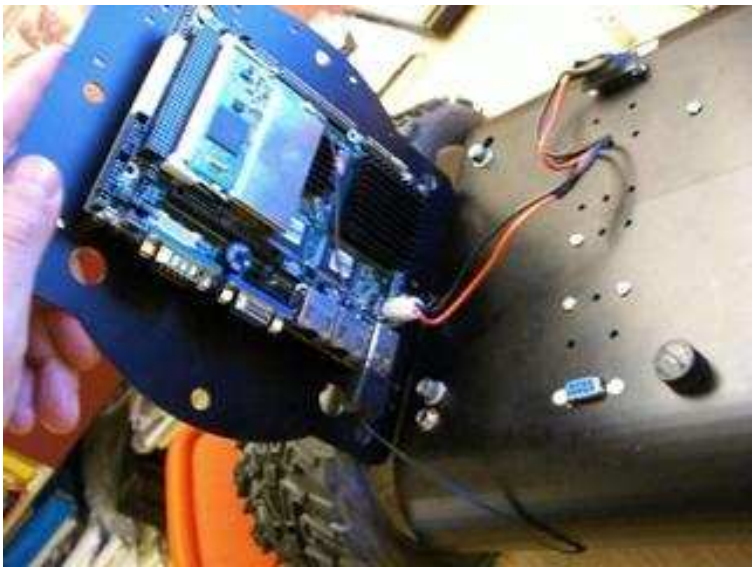
Their installation takes place as follows:

The upper aluminium support can be Unscrew :

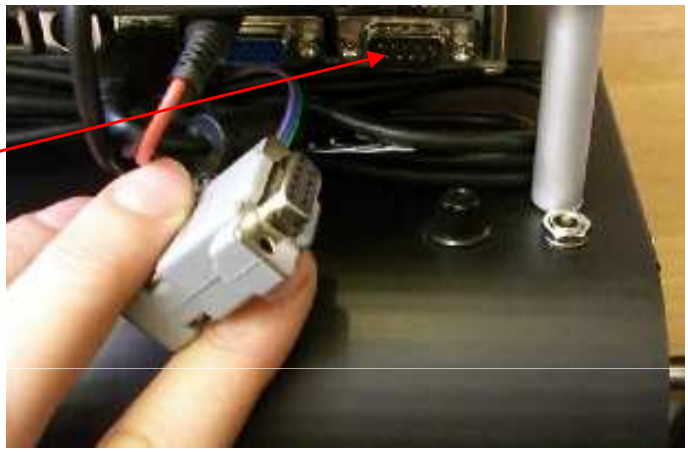
The embedded computer is fixed on the down part of the support :



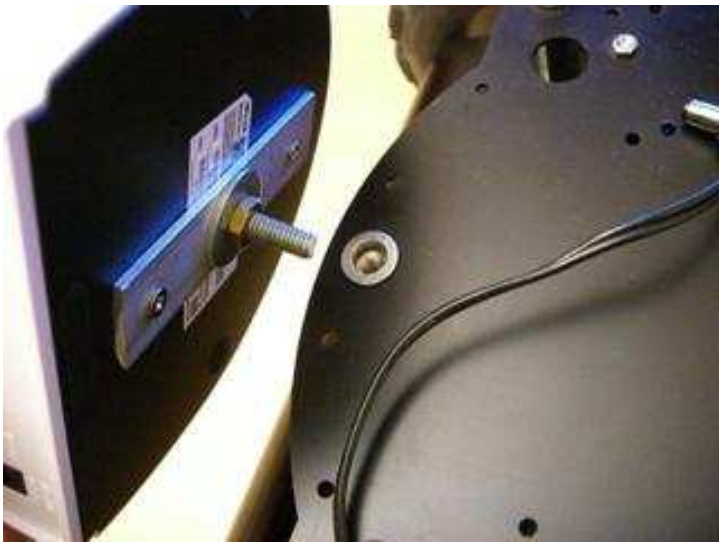
The power cable of the computer is connected to the appropriate connector (rear):



The Control Command RS232connector is connected to the computer.

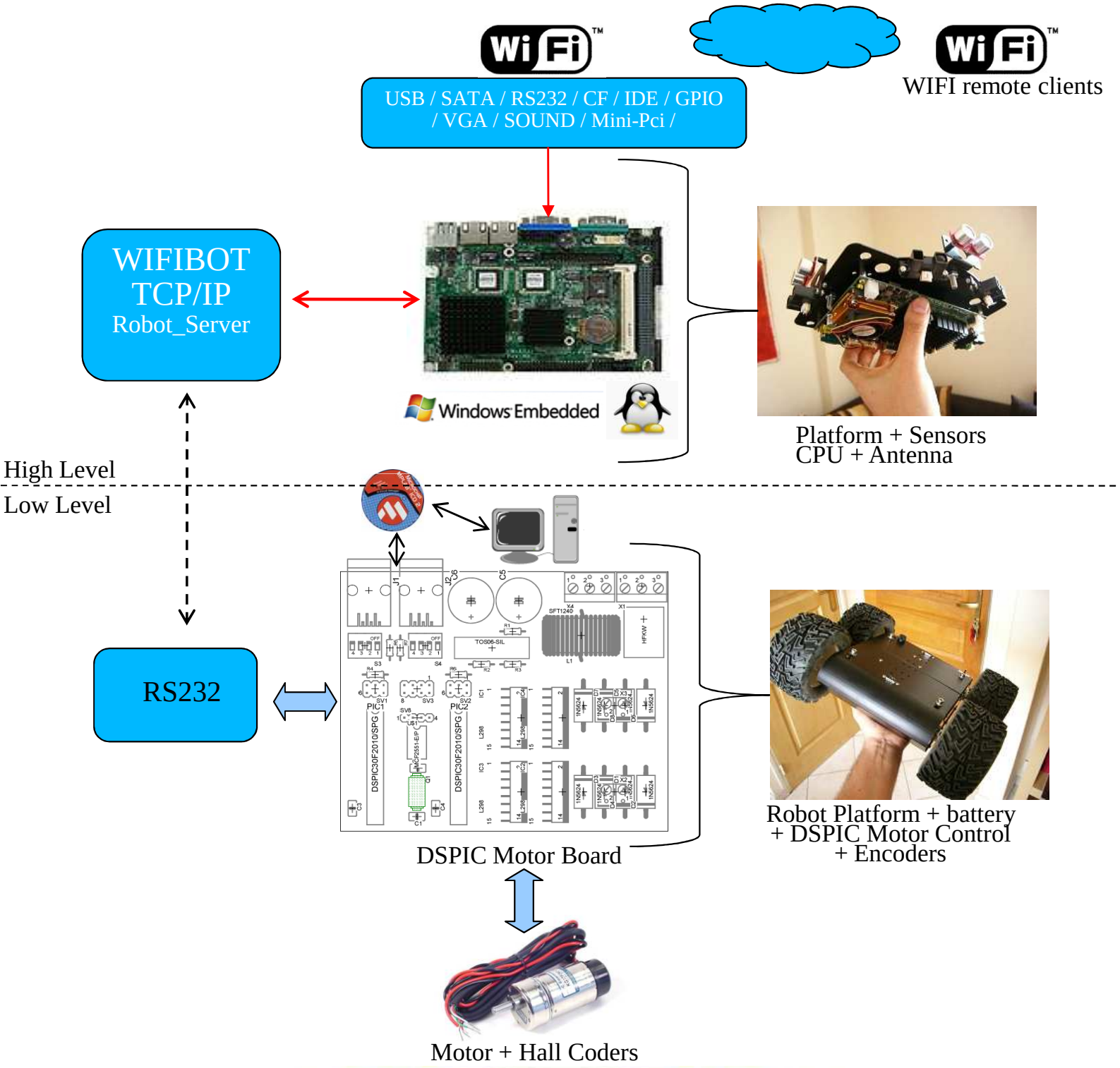


Screw back the aluminium support on top of the platform and screw the IP camera on top of the support.



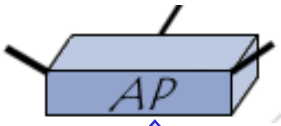
# System architecture:

This architecture is composed by 2 Parts : The **high level** composed by the sensors and the CPU (or other custom devices), and the **low level** composed by a ICD2 capable DSPIC motor board controller. A RS232 port is the link between the CPU and the low level.
Once plugged, Linux or windows can send and received data from serial port and control the wheels or receive sensors data. **The protocol another document.** A simple TCP/IP gateway is provided with source code to see how it is simple to control the robot using WIFI.



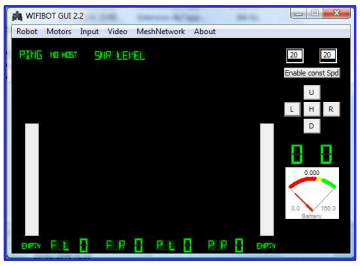
# *High level Architecture*

WIFI
a/b/g


Ethernet

Mini-PCI



Remote PC

Embedded CPU

Remote HMI

USB

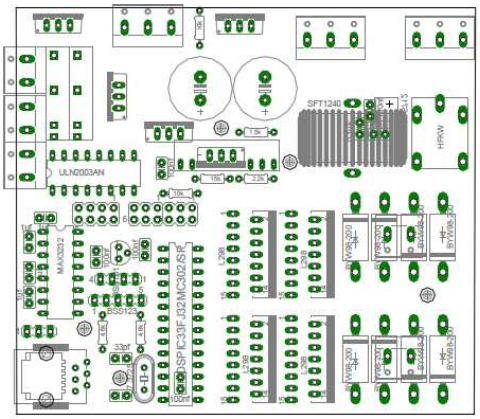
USB

RS232


ICD3 (option)
or bootloader

MJPEG Web Server

Relay



Camera

4m Lidar

Robot

DSPIC 33F Motor Board

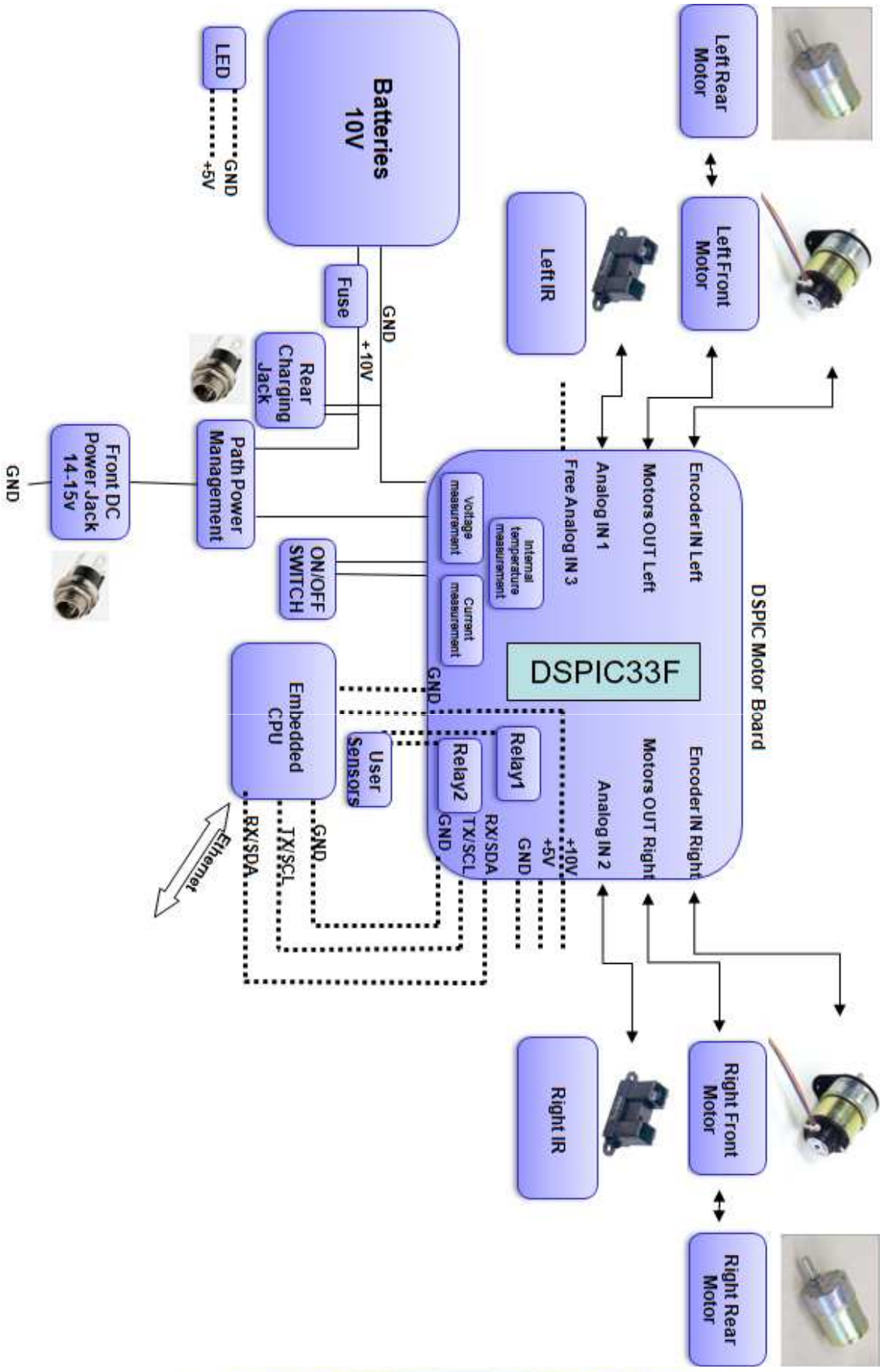
Analog

PWM

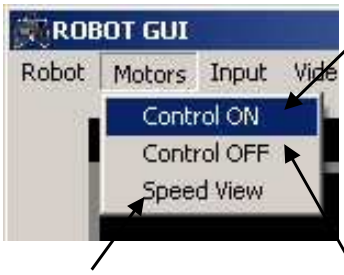
2 x IR Sensors

Motor + Hall Coder

# *Low Level Architecture*



# The control software (TCP/UDP) Simple GUI:




**Motor Control ON:** Activates the speed control, Input_Left and Input_Right set on the dialog will be applied.

**Speed View:** Plots in real time the speed signal from the code wheels.

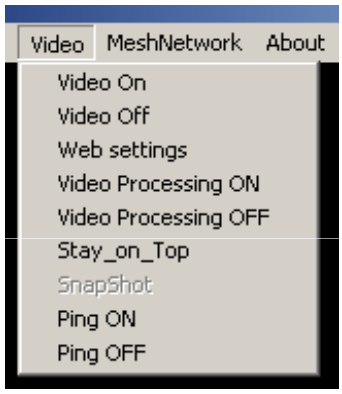
**Motor Control OFF:** Deactivates the speed control.

**The control software:**

The control software can be found in the CDROM in **..\Software\control software**

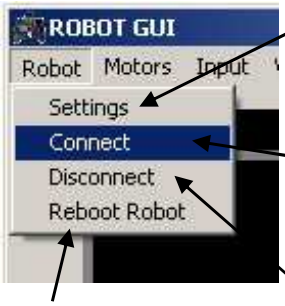
- Install if necessary the **Video Decoder** present in the same folder.
- Launch the **WifibotGUI** program.
- Click on **Robot** then **Settings**. The **Robot Settings** window appears.
- Set the **Control Server IP** and the **Control Server Port** which by default is **15020**.
- Set the **Camera IP** and the **Camera Port** which for the image is by default **80**.
- Select the proper **Camera Type**.
  If the camera type is not present use **Firefox or Internet explorer at port 8080 to view the image**.
- Click on **Video**, then select **VideoOn**. The image from the camera will appear.
- Click on **Robot** then **Connect.**
- Click on **Input** then select **Joystick** or **Virtual_joy**. The robot can now be operated.


**Input Selections** (control panel for calibrating the joystick)




**Video selections:** Allows to configure and control some options of the camera.

The menu options:

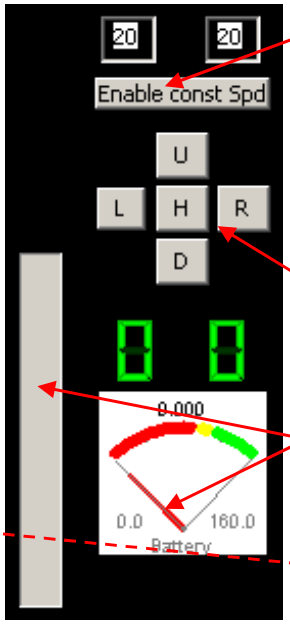



**Settings:** IP settings of the Control Server and the Camera.

**Connect:** Starts the communication with the Control Server.

**Disconnect:** Stops the communication with the Control Server.

**Reboot:** Reboots the robot's CPU if available.

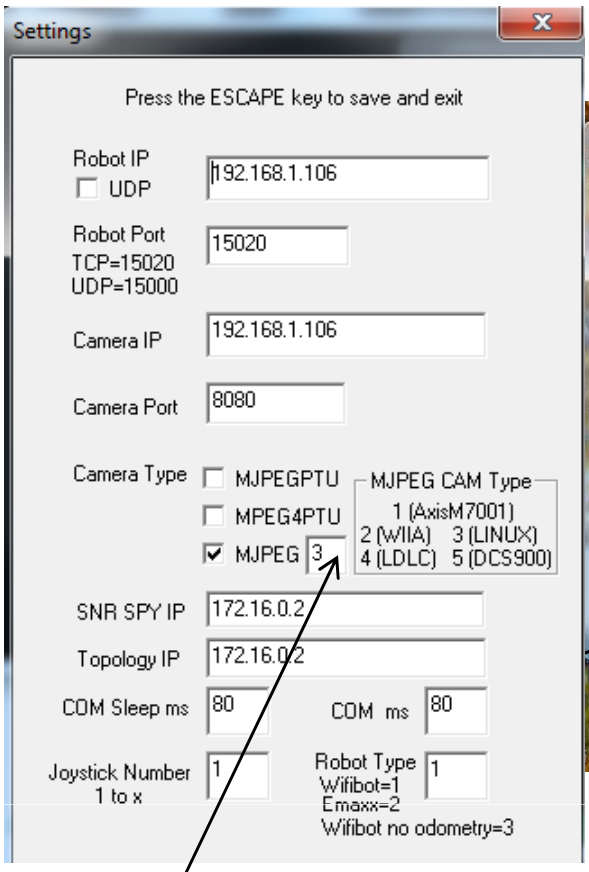


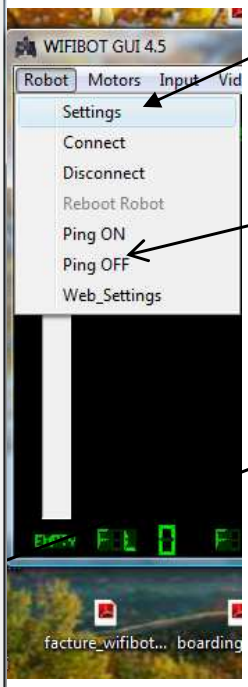

**Current input:** shows the current input or allows to set it manually with keyboard.

**Pan-Tilt camera control:** The red button takes the camera to the default position. You can click on the image too for moving the camera.

**Sensor feedback:** shows the data retrieved from the range sensors, the battery level, robot current and the speed of the robot in tics.

# The control software (TCP)
# Simple GUI:



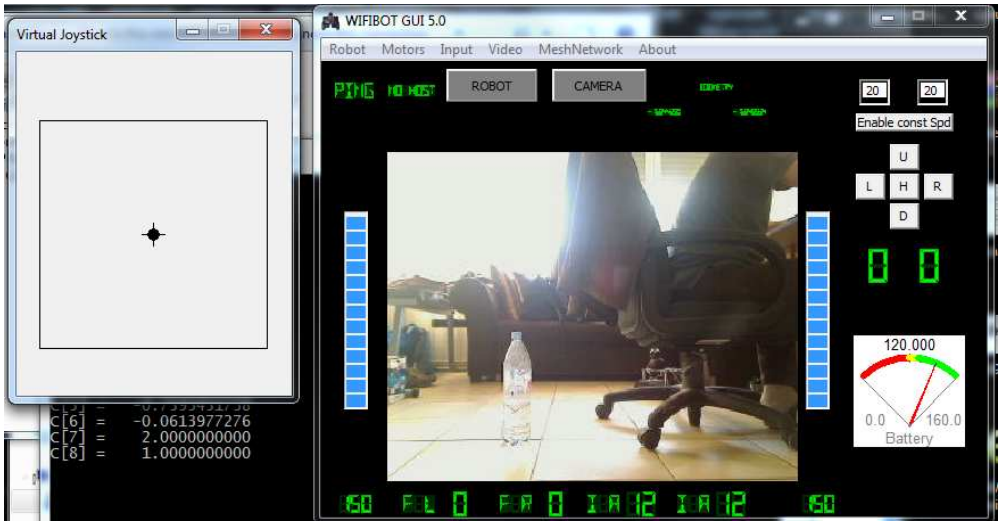

**Settings menu**
Set IP & port of robot
IP & port camera

**Ping Robot**

**Robot Type**
Set robot type
1 WIFIBOT Lab
2 Emaxx 4wd
3 WIFIBOT SC/4G

**Camera Type**

1 Axis MJPEG Server
2 MJPEG WIA Server
3 MJPEG Streamer
4 LDLC Camera
5 DCS900



## Camera Type and software clients compatibility:

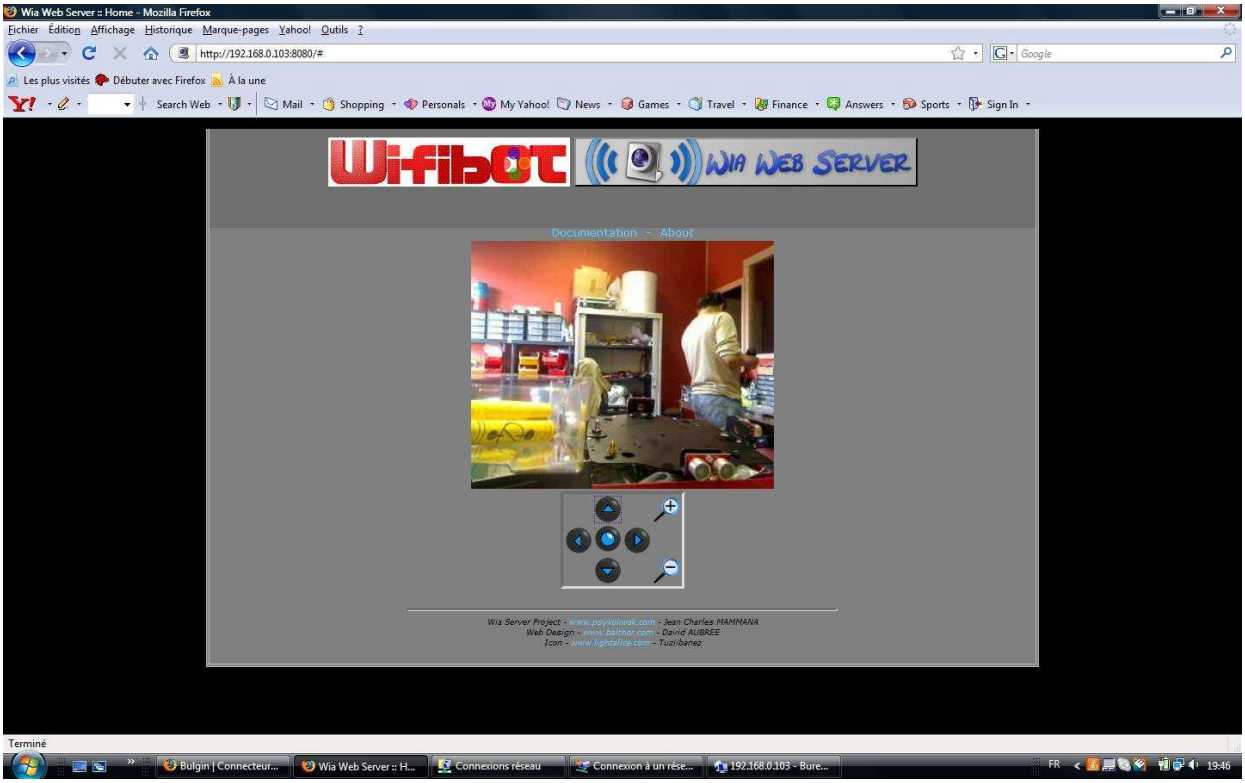
AXIS COMMUNICATIONS


**Simple GUI**

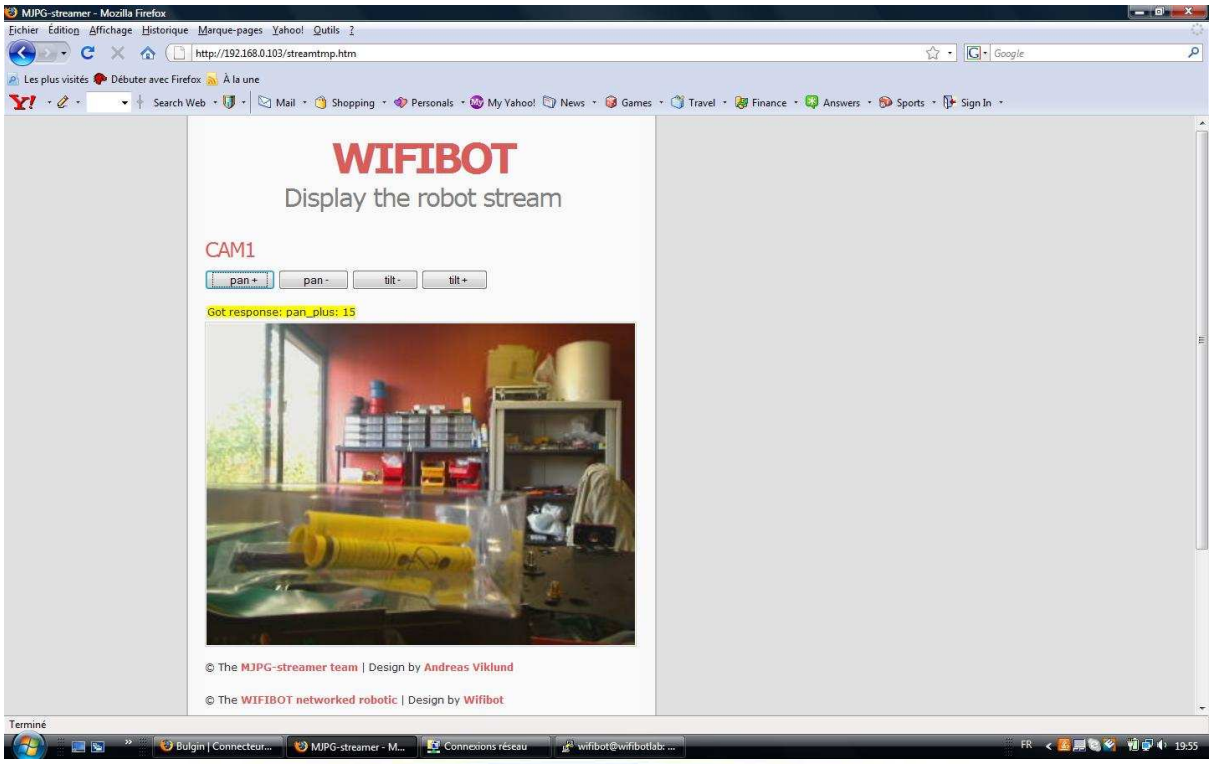

AXIS COMMUNICATIONS

# Web Cam Server web client:

## Windows WIA SERVER: http://192.168.1.XXX:8080




## Linux MJPEG-STREAMER: http://192.168.1.XXX:8080



# Connecting to the robot using wireless network:

By default, the robot has been pre-configured with a certain IP addresses and it connect to the provided access point (essid “wifibotlab”).

You need just to connect your control PC using DHCP to the access point. And then you can obtain a valid IP address to get into the robot network.

You can also adjust the IP settings of the network adapter of your computer manually. Make sure all the devices in a same network having to communicate with the robot have the same class of address.


SSID WIFIBOTLABAP
192.168.1.1
DHCP Server

192.168.1.XXX

If you are connecting to a robot **under Linux or Windows** with a **cable** directly to his Ethernet port, then enter 192.168.0.x on your PC (x can be any number between 1 and 254 except 250 and those used by the CPU and the camera of the robot).

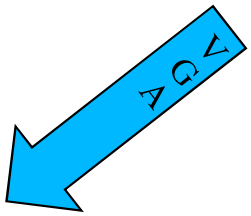
For example, a Wifibot Serial Number: **LABYYYXXX** will have as IP for **the CPU 192.168.0.XXX and 192.168.0.XXX:8080** for the camera if webcam, **192.168.0.20:80** if IP camera. Set the **Subnet Mask** to 255.255.255.0 and leave **Default gateway** and **DNS** empty.

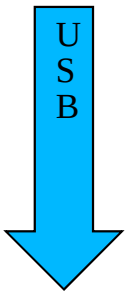
If you are connecting **wirelessly** to a robot: **Under windows** because we bridge the wired and the wireless interface, the IP are the same as previous wired mode.
**Under Linux we are on 192.168.1.XXX. We NAT the eth0 to the ath0.**

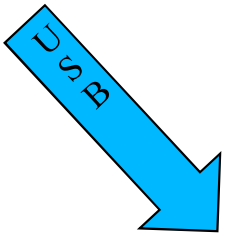
For example, a Wifibot Serial Number: LABYYYXXX will have as IP 192.168.1.XXX, and 192.168.1.XXX:8080 for the camera IP or webcam.
Set the **Subnet Mask** to 255.255.255.0

## Connecting to the robot using a vga screen and a keyboard:

The robot can act as
A regular PC. You can
Plug a screen and a
Keyboard.




# Networking

**Network architecture:**

In the Wifibot Lab the embedded CPU works as a gateway between the internal wired LAN and the external wifi WLAN. The CPU has at least one ethernet card and one wireless card that form two separate networks (LAN/WLAN).The LAN and the WLAN should have in general a different address class and therefore data needs to be routed between them. Depending if you have chosen a robot under Windows or Linux the problem of connecting the two networks has been solved differently. Under Windows this has been done by configuring a bridge between the network interfaces, by doing so the robot's CPU appears to have a unique network interface and uses one single IP address. Under Linux, the interconnection is done through Dynamic NAT (Network Address Translation) and the CPU uses two different IP adresses, one for the internal LAN and one for the WLAN. In both cases, all local components of the robots such the IP camera will have their own IP address within the LAN, but when it comes to accessing them from the WLAN the method will differ. Under Windows as there is in practice no distinction between the WLAN and the LAN, every internal component will be reached using its own IP address (see **Fig1**). Under Linux, only the robot's CPU WLAN IP address can be seen and any internal network element will have to be reached using this single IP. In order to be able to access the separate devices using a single IP, we will need to assign to each of them a separate port (see **Fig2**). This will require to configure the CPU with the proper routing table.

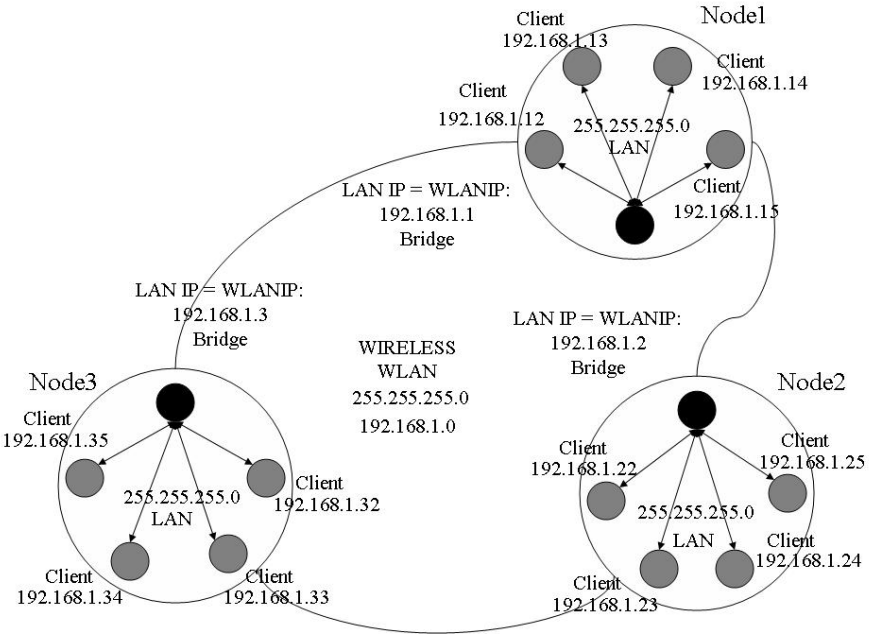



**Fig 1**

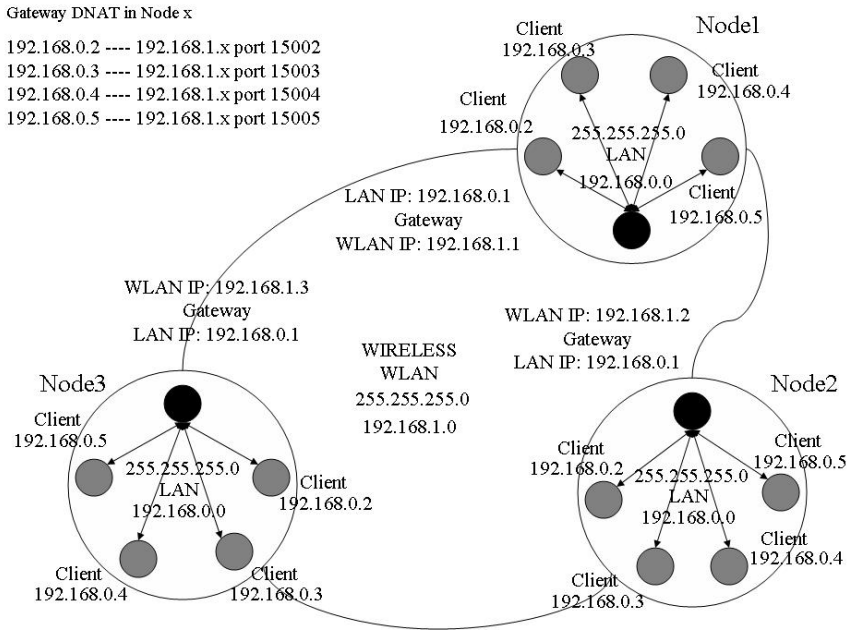
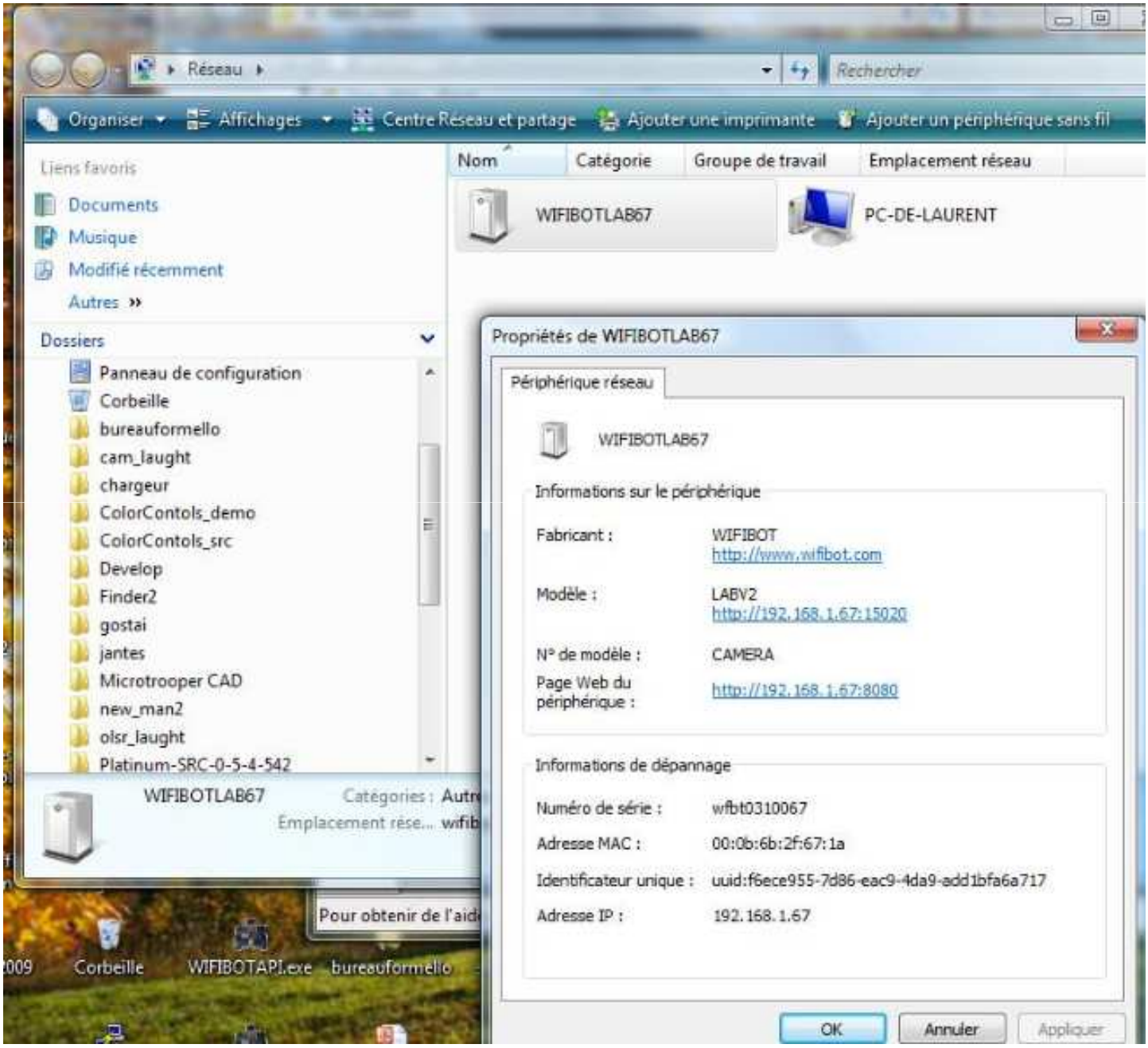



**Fig 2**

**UPNP:**

**For a Linux or Windows robot**
**An UPNP server expose the**
**robot data:**



**WLAN modes:**

Let's have here a quick overview of the different modes Wi-Fi adapters can be configured :

-Master (Access Point) not used
-Infrastructure Managed (default mode)
-Ad-hoc without routing algorithm
-Ad-hoc with the OLSR routing algorithm (Mesh Networking)

In infrastructure mode we have a master/slave structure where all the data is centralized in one device called access point (server/master) to which different adapters (clients/slaves/managed) connect. A client cannot talk directly to another but has to pass by the access point which will forward the data to the destination. Several access points can be connected together with cables extending in this way the zone covered by the wireless network. This is the most common setup for a Wi-Fi network (see **Fig1**).

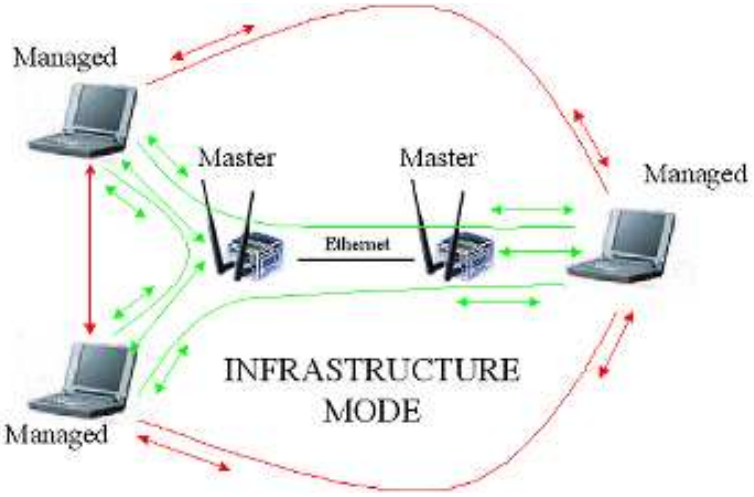



**Fig 1**

In ad-hoc mode we do not have any central management, each client can talk directly to the other. This mode works fine for networks with few elements. Without any routing algorithm, each element needs to have a direct radio link with the others in order to communicate, no data will be forwarded (see **Fig2**). If a routing algorithm such as OLSR or BATMAN is added, you obtain a self-organizing mesh network in which message forwarding is possible wirelessly between different nodes, connecting in this way devices which are not within direct radio range (see **Fig3**). This allows to extend the zone covered without the need of any cable.The network is completely dynamic, routing tables are rewritten automatically and dynamically as the network changes. If a new OLSR or BATMAN enabled device appears, it will be automatically detected and merged to the routing tables of each node. This is especially useful for mobile networks that can change over time like for example in a multi-robot application.

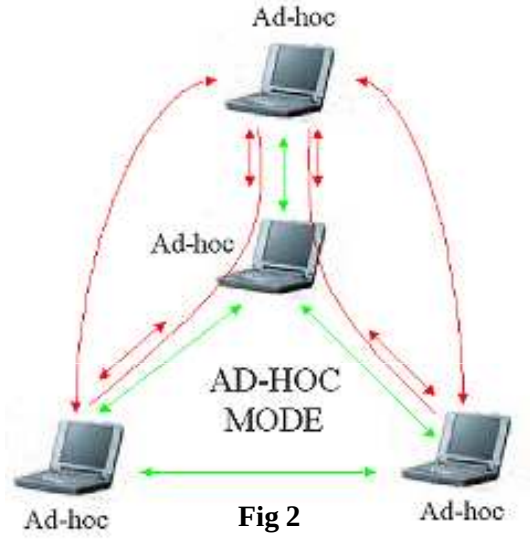



**Fig 2**

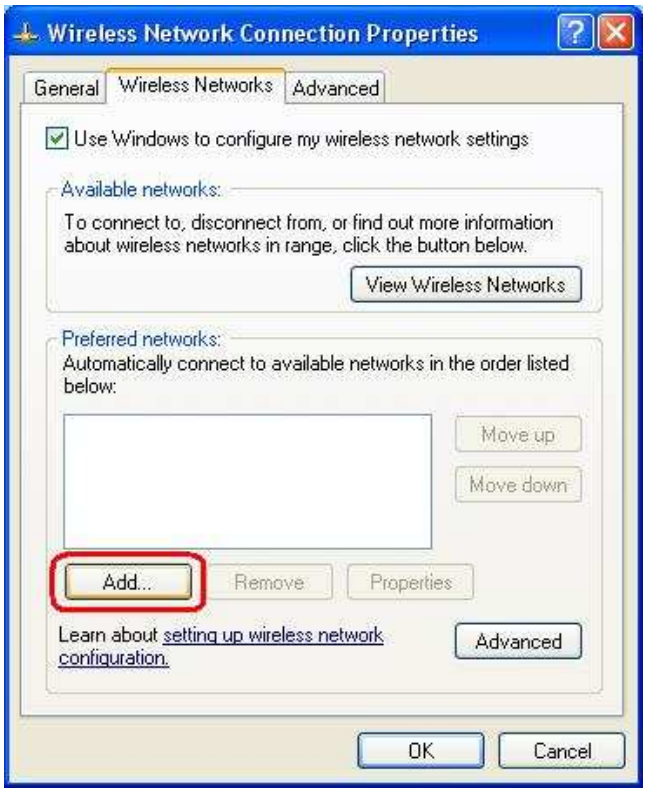

**Fig 3**

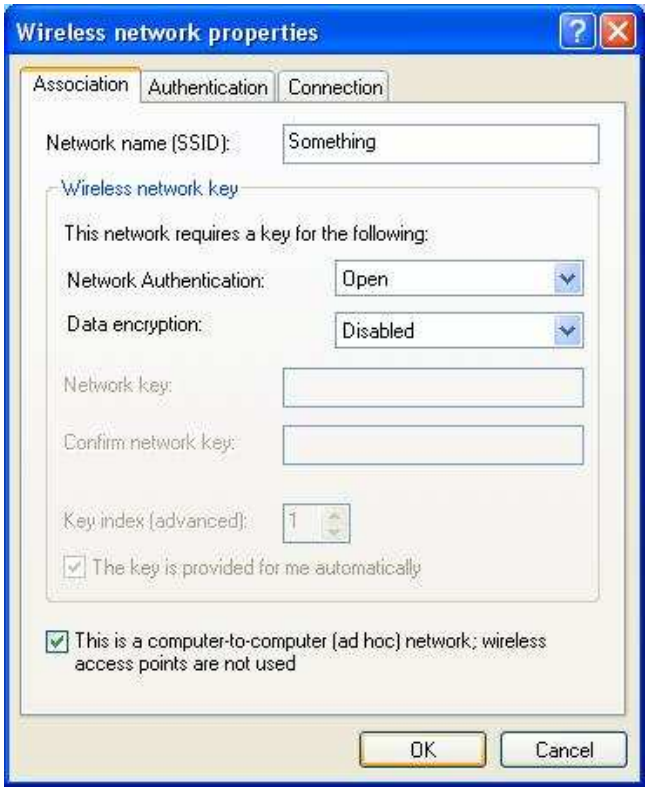
# Network configuration

**Robot under Windows XP**

By default all robots come already configured and ready to work with the provided access point.

Information is given here for those users willing to make changes in the network configuration. When working under Windows the robolab can only be configured in the managed and ad-hoc modes. For configuring the IP settings in managed mode or connecting the robolab to an AP or an already created ad-hoc network please follow the steps detailed in the "connecting to the robot" section. In addition to those steps, it is recommended to create from the robot itself the ad-hoc network to be used:




1. Open **Network Connections**, Select your Wireless card right click on it and select **Properties.**
2. Click the **Wireless Networks** tab.
3. Enable **Use Microsoft Windows to configure my wireless network settings**
4. Click **Add…**
5. For **Network name (SSID)** type: **wifibot**
6. For **Data encryption** select **Disabled**
7. Enable **This a computer-to-computer (ad hoc) network**
8. Click **Ok** to close the '**Wireless network properties**' window
9. Click **Ok** to close the '**Wireless Network Connection Properties**' window
10. Using your test computer wireless adapter, view the available wireless networks, check the list and validate that you can see your newly configured **wifibot** network. If it is configured, try to connect to it. If you cannot find your new network verify the settings are correct.

**Robot under Linux**

Under Linux all modes are possible but master mode seems buggy since Xubuntu 9.04. We will see here the different parameters involved in the configuration of the robot. There are a few important configuration files we need to manage in the robot:

**/etc/network/interfaces**

| | | |
|---|---|---|
| **/etc/init.d/wifibot-init** | launch **/usr/sbin/wifibot-init** | the script witch set the NAT |
| **/etc/init.d/wifibot-server** | launch /usr/sbin/robot_server | the robot server for control |
| **/etc/init.d/wifibot-mjpeg** | launch **mjpeg-streamer** | the webcam server |

**/etc/wifibot.ini** Some variables for /usr/sbin/wifibot-ini

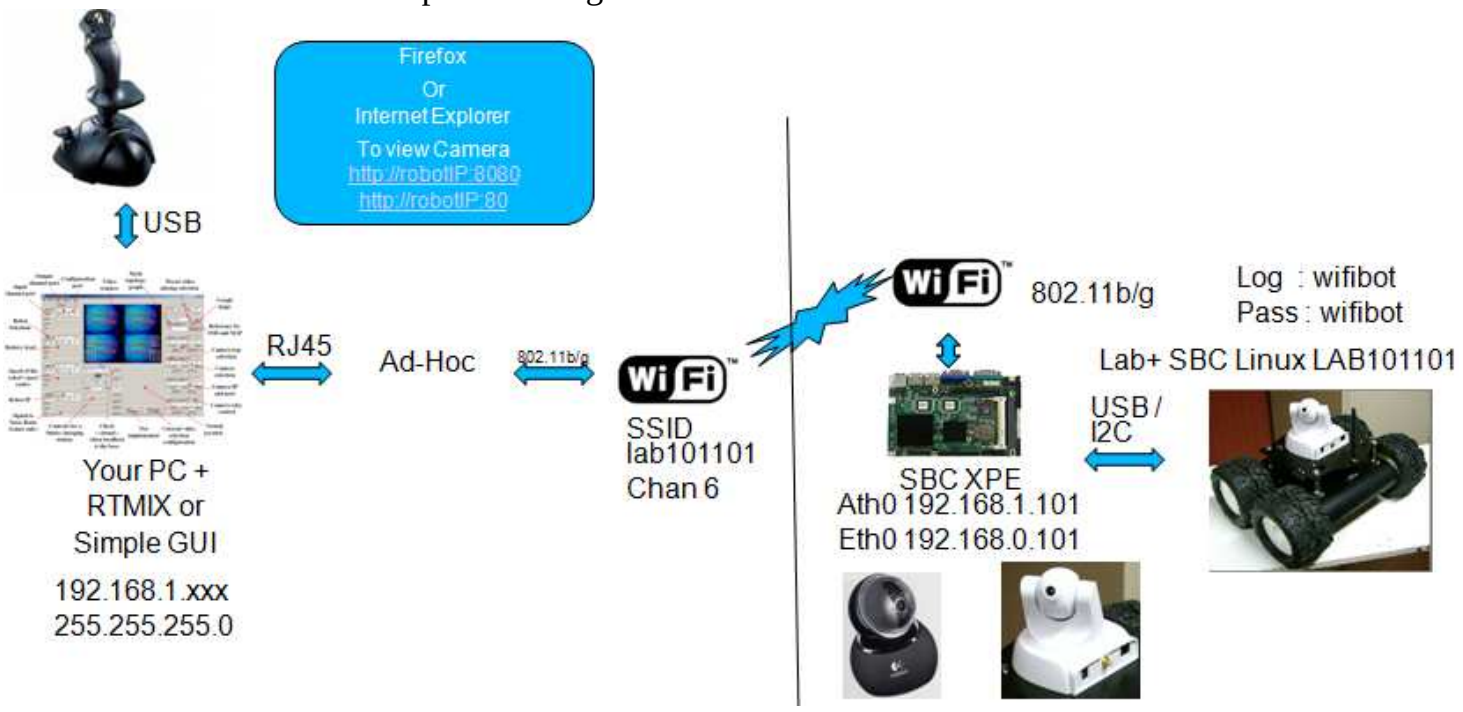
**/usr/sbin/wifibot-init**
**/usr/sbin/robot_server**

The I2C modules for the robot or the ultrasonic sensors are in /dev/ttyUSB0 and /dev/ttyUSB1
**/dev/ttyS0 is the RS232 COM1**

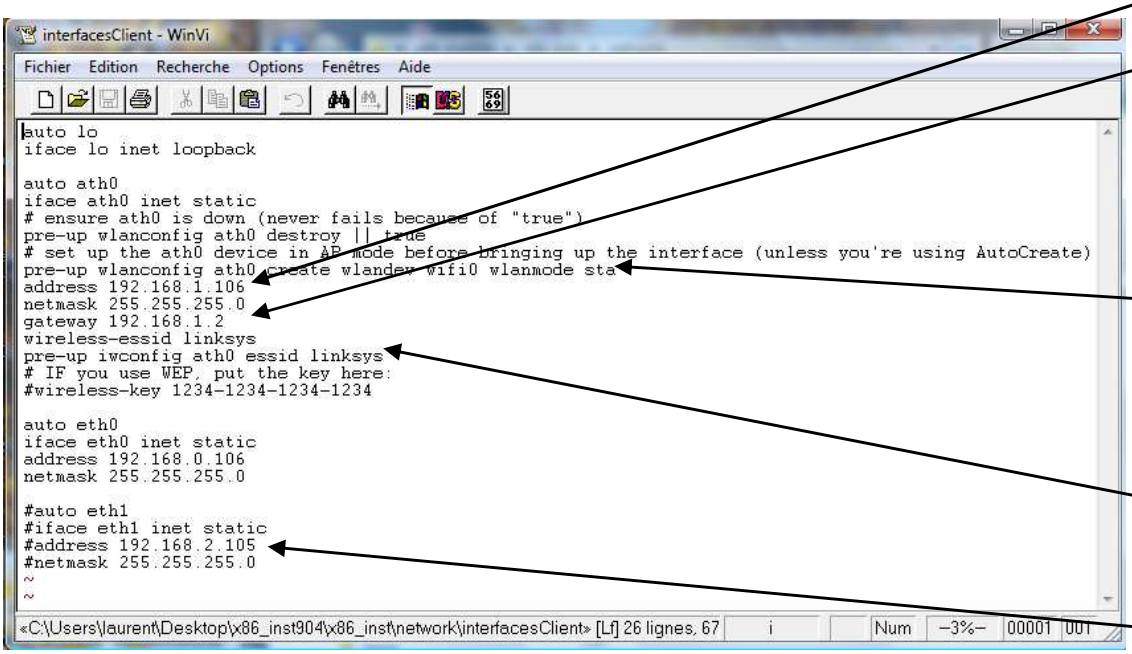
These files can either be edited outside the robot and then transferred or directly edited on the robot.

**The "interfaces" configuration file:**

This file allows to specify the IP settings of the different network interfaces present on the robot and the wireless settings when it applies. All the Wifibot Lab have the eth0 interface for the LAN and the ath0 interface for the WLAN connection. As we have seen there are 2 possible wireless modes, here we will show an example of configuration of this file for each one of those modes.



Infrastructure Managed:




•WLAN IP settings

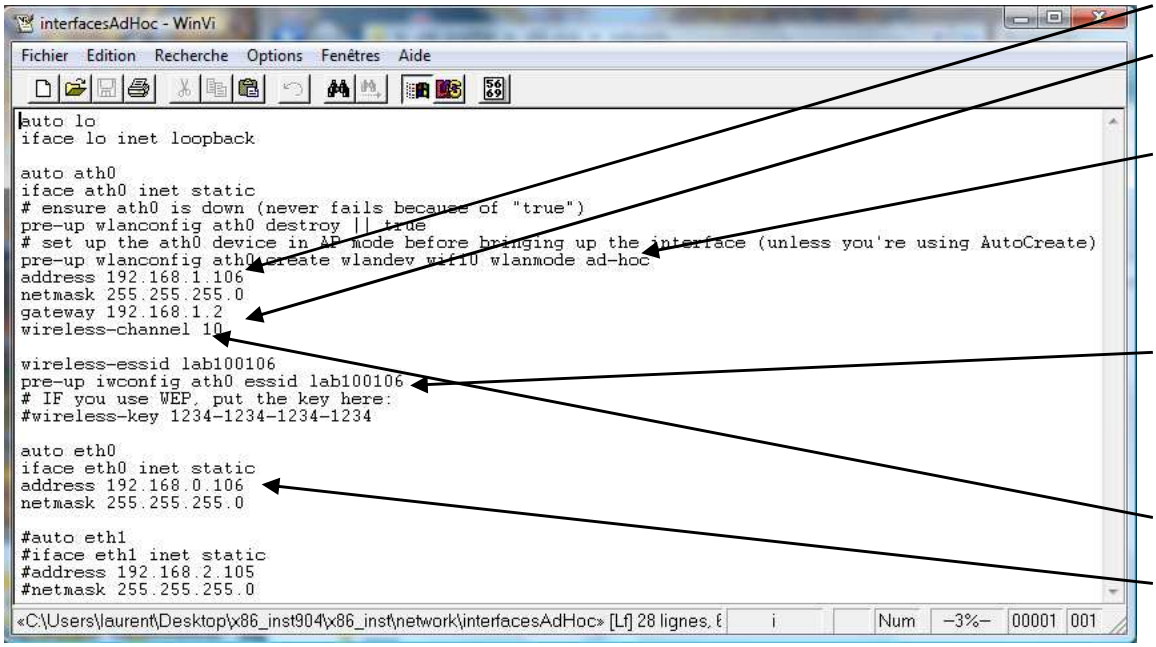
• If the wireless network is further to be connected to another network (i.e Internet) we need to specify the gateway.

• wireless mode specified to **managed,** that is, it will connect to an existing wireless network.

• SSID, that is the name of the wireless network the robot will connect.

•LAN IP settings

Ad-hoc:




•WLAN IP settings

• The gateway IP if there is one.

• wireless mode specified to **ad-hoc,** that is,all devices will talk directly to each other.

• SSID, that is the name of the wireless network the robot will connect.

• channel to be used,

• LAN IP settings

**The "/usr/sbin/wifibot-ini" configuration file:**

This is the last step of configuration, we will first specify the routing tables of the NAT address translation, this is needed to make the embedded devices visible from outside the robot. Then we set the commands to launch the control server and some computer dependent drivers at boot time.

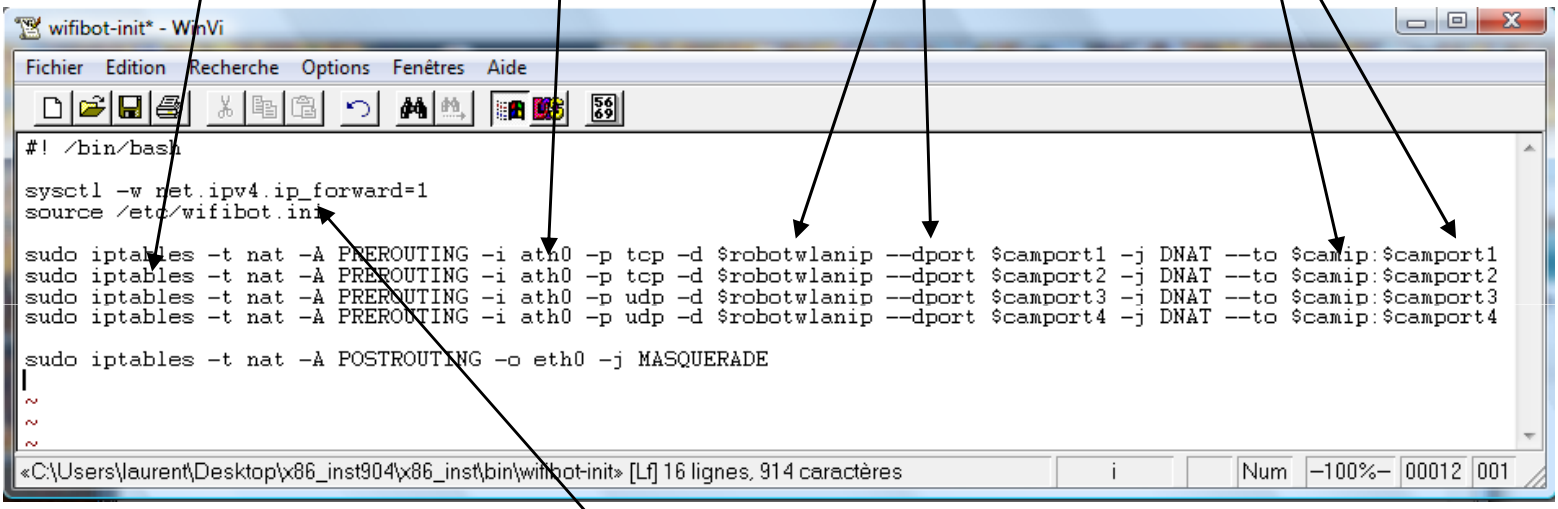
Information about IPTABLES/NAT can be found at http://www.netfilter.org/

We use « iptables » for seeting the NAT

The interface we are « NATting »

The WLAN interface **IP** and **ports** we will connect to from outside.

The LAN device **IP** and **ports** we want to reach (i.e the camera)




Read the /etc/wifibot.ini file to get NAT info

**IMPORTANT NOTE!!!!:** When editing the configuration files under windows, use the "WinVi32.exe" text editor ONLY, it is included in the CD ROM at **\software\WinVi32**
It is important to use it to respect the Linux format, specially when editing configuration files. Another option is to edit the files directly on the CPU under Linux with the installed "vi" editor, check http://www.linuxfibel.de/vi.htm for more information.

# Remote access

**Remote access to the desktop of a robot working under Windows XP:**

When working with the robolab, it is always possible to attach a screen, a mouse and a keyboard directly into the embedded computer but it is often more convenient to have access to the robot remotely over the network. If the robot works under Windows follow these steps:


**Fig 1**

1- Click Start, point to All Programs, and then point to Accessories.
2- In the Accessories menu,point to communications and then click Remote Desktop Connection.
3- In the Computer box, type the IP address or the name of the robot you want to connect to (**Fig 1**).
4- Click Connect.
5- When the Log On to Windows dialog box appears type **root** as the user name and **wifibot** as the password, and then click OK (**Fig 2**).


**Fig 2**

The Remote Desktop window opens, and you see the desktop settings, files, and programs that are on the robot. Your robot remains locked, and nobody can access it without a password. In addition, no one will be able to see the work you are doing remotely.

To end your Remote Desktop session:

1. Click Start, and then click Log Off at the bottom of the Start menu.
2. When prompted, click Log Off (**Fig 3**).


**Fig 3**

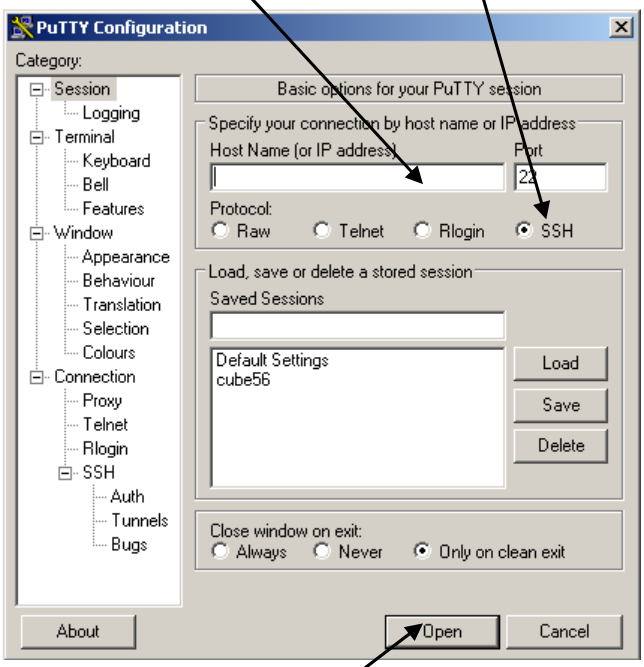
**Remote access to the command line of a robot working under Linux :**

To remotely log into the robot's Linux operating system we will make use of a protocol called SSH (Secure Shell) which facilitates encrypted communication across networks. This requires a SSH client program. Whichever the SSH client you use, the procedure is similar:

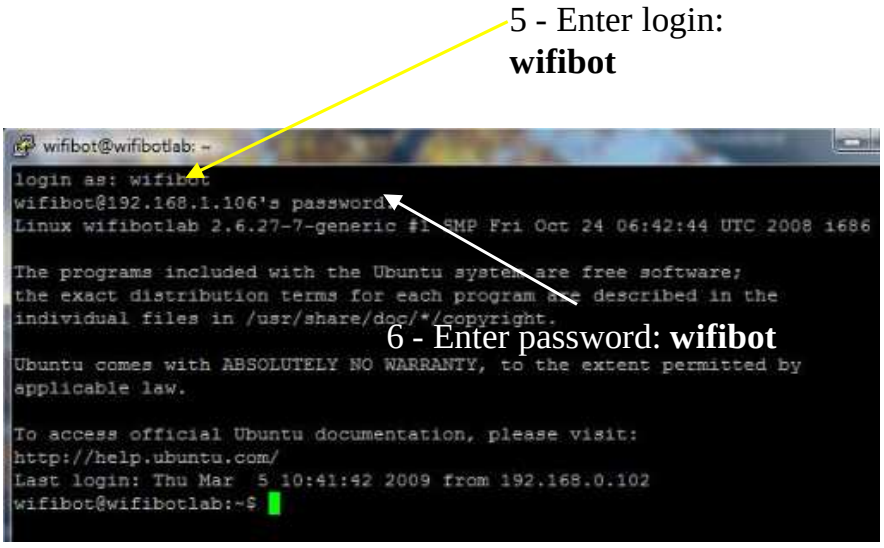
- Open the SSH client.
- Enter the CPU IP address (the default port is 22) and then start the connection.
- The first time a connection is established, the program will ask for confirmation.
- Enter login: **wifibot**.
- Enter password: **wifibot**

For your convenience the CDROM includes a free SSH client you can find in **\software\putty**

Connect to the robot in the following steps:





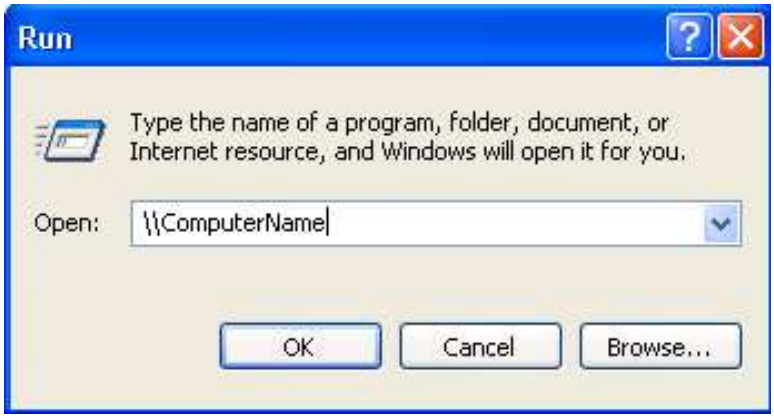


# File transfer

**Transferring files to a robot working under Windows XP:**

For transferring files we will make use of the file sharing capabilities of Windows XP.
From your computer in order to connect to the robot and transfer files do the following:

1. Click **Start > Run.**
2. In the **Open** field type **\\Robotname** or **\\IP address**
3. In the window that appears, type in the username **root** and password **wifibot**
4. Click **OK**
5. Only "\data" folder is shared and not protected with fbwfmgr





**Important Notice for Xpe:**

**To change the configuration to the Compact Flash except the /data folder witch is writable**

**In the command terminal:**

**fbwfmgr /disable and reboot to change CF**
**fbwfmgr /enable and reboot to reprotect CF**

## Transferring files to a robot working under Linux:

For transferring files we will use the SFTP protocol,this requires an SCP client program. For your convenience you will find in the CD ROM a free SCP client in **\software\WinSCP**

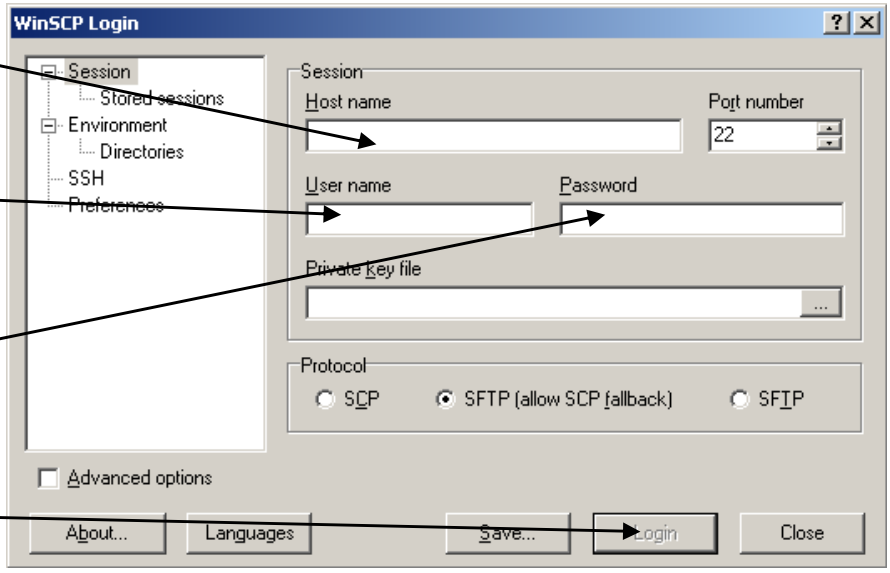
Connect to the robot in the following steps:

1 - Enter the **IP address** here.

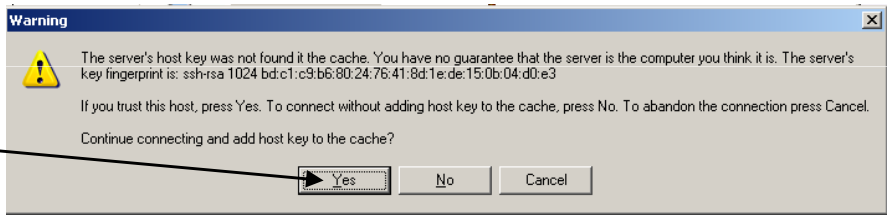
2 - Enter User name : **wifibot**

3 - Enter password: **wifibot**

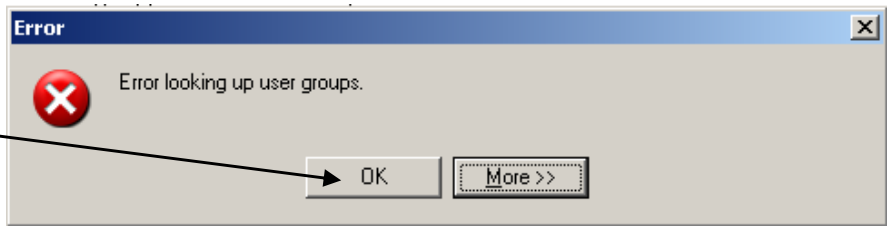
4 - Click **Login** to initiate the connection.



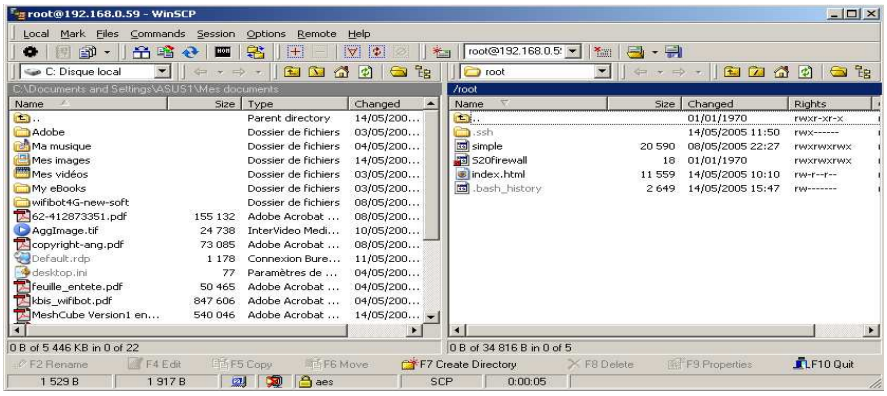

5 - Confirm the connection.




6 - Confirm again.

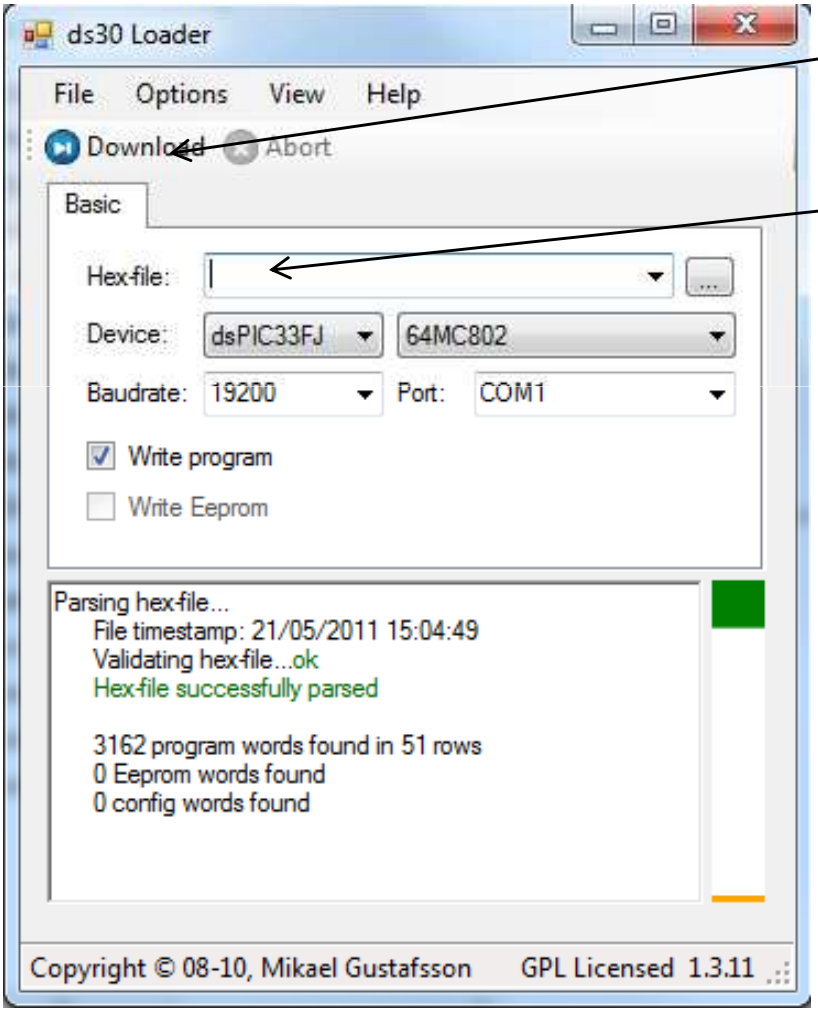



You are now connected and you can start transferring the files.



## Chassis update:

We can upgrade the chassis using Microchip ICD2 ICD3 programmer or using the embedded bootloader DS30 using ds30 Loader GUI.exe software.




Plug your pc on the robot serial port, push the Download button and quickly switch ON the robot. The upgrade will start automatically.

.hex File for update

# The standard CPU

## LE-376

***3.5" embedded board with Intel® Atom™ dual-core Solution***

| Specification | |
|---|---|
| **Form Factor** | 3.5" embedded board |
| **CPU** | Intel® Atom™ D510, 1.66GHz, 1MB cache (LE-376A)<br>Package type : Micro-FCBGA (FCBGA559) |
| **Memory** | 1 x 200-pin DDR2-667 SO-DIMM up to 4GB (LE-376H/A)<br>Support Non-ECC, unbuffered memory only |
| **Chipset** | Intel® ICH8M |
| **Real Time Clock** | Chipset integrated RTC with onboard lithium battery |
| **Watchdog Timer** | Generates a system reset internal timer for 1min/sec ~ 255min/sec |
| **Power Management** | ACPI 2.0 compliant, supports power saving mode |
| **Integrated Graphics** | Intel® integrated extreme GMA 3150 Technology |
| **Video Memory** | Up to 384MB shared with system memory |
| **LVDS Interface** | Chipset Integrated 18-bit single channel LVDS |
| **Serial ATA Interface** | 2 x SATAII interface with 300MB/s transfer rate |
| **Solid State Disk** | 1 x Compact Flash Type-II |
| **Audio Interface** | Intel® ICH8M integrated with Realtek ALC888 HD Codec |
| **LAN Interface** | 3 x Intel® 82583V Gigabit Ethernet controller |
| **Expansion Interface** | 1 x PCIe mini card & 1 x Mini-PCI socket |
| **Internal I/O Port** | 1 x Audio, 4 x USB2.0, 1 x LVDS, 1 x LCD Inverter, 1 x LPT<br>1 x RS232/4224/5, 4 x RS232, 1 x SMBUS, 1 x IrDA |
| **External I/O Port** | 1 x USB, 3 x RJ45 LAN, 1 x DB15 VGA, 1 x RS232 |
| **Power Requirement** | DC 9V ~ 24V input |

## The optional CPU (core I5 520M or core I7 620M)

# Industrial Single Board Computer

## 3.5" Miniboard

### LS-377

**Support Intel® Core™ i7, Core™ i5 and Core™ i3 CPU with DDRIII SO-DIMM, CRT, LVDS, DVI, Gigabit LAN, Mini PCI, PCI Express mini card, Serial ATAII, 7.1Channel HD Audio**

| | |
|---|---|
| **Form Factor** | 3.5" Miniboard |
| **CPU** | Intel® Core™ i7, Core™ i5, Core™ i3, Celeron®, and Pentium® Mobile Processor<br>Package type: rPGA988A |
| **Memory** | 1 x DDRIII SO-DIMM 800/1066 MHz up to 4GB |
| **Chipset** | Intel QM57 |
| **Real Time Clock** | Chipset integrated RTC with onboard lithium battery |
| **Watchdog Timer** | Generates a system reset with internal timer for 1min/s ~ 255min/s |
| **Power Management** | Supports ACPI 2.0 compliant. |
| **Serial ATA Interface** | 2 x serial ATAII interface with 300MB/s transfer rate |
| **VGA Interface** | Onboard VGA (depend on CPU) |
| **LVDS Interface** | Onboard 24-bit dual channel LVDS connector with +3.3V/+5V/+12V supply |
| **DVI Interface** | DVI interface |
| **Audio Interface** | Realtek ALC888 HD Audio |
| **LAN Interface** | 1 x Intel 82574L Gigabit LAN |
| **GPIO Interface** | Onboard programmable 8-bit Digital I/O interface |
| **Extended Interface** | 1 x Mini PCIE socket, 1 x Mini PCI socket to support Mini PCI Type IIIA |
| **Internal I/O Port** | 1 x RS232/422/485, 1 x SMBUS, 1 x GPIO, 4 x USB ports, 1 x IrDA, 1 x LVDS, 1 x DVI, 1 x LCD, 2 x Serial ATA, 1 x LCD Inverter, 1 x HD Audio, 1 x DIO, 1 x DCOUT and 1 x CDIN |
| **External I/O Port** | 1 x PS/2, 1 x LAN ports, 1 x VGA port, 2 x USB2.0 ports, 1 x RS232 port |
| **Power Requirement** | 9~24V full range DC Input |
| **Dimension** | 146mm x 101mm |
| **Temperature** | Operating within 0~60 centigrade<br>Storage within -20~85 centigrade |

# The CDROM

The CDROM included with the robot contains the documentation and sample programs for the robot. Its contains three folders:

**Documentation:**

In this folder you will find the original documentation of the platform (this file), the embedded computer and the camera.

**Software:**

This folder contains the robot control software installation program, a copy of the robot's embedded server, as well as several tools necessary to manage the robot. The location in the robot for the embedded server (robot_server.exe) is the "data" folder of the Compact Flash, the user is free to replace this file with his or her own program for giving the robot autonomous behaviours etc.
Under linux you will find the server here /usr/sbin/robot_server

**Code Samples:**

Here you can find the source code of the robot's server for Windows and Linux as well as samples for the programming of remote control applications on computers under both Windows and Linux.