



## 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	.....2
Quick start	.....2
Platform overview	.....2
Platform interfaces and charging	.....3-4-5
Computer and camera	.....6-7
System Architecture	.....8
Simple GUI	.....9-10
RTMIX multi-robot interface	.....11-14
Cameras	.....15-16
Connecting to the robot	.....17-18
Networking	.....19-20
Network configuration	.....21-24
Remote access	.....25-26
File transfer	.....27-28
Communication protocols	.....29-31
The CDROM	.....32
The standard CPU	.....33

## 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 RJ45 cable for the IP camera**

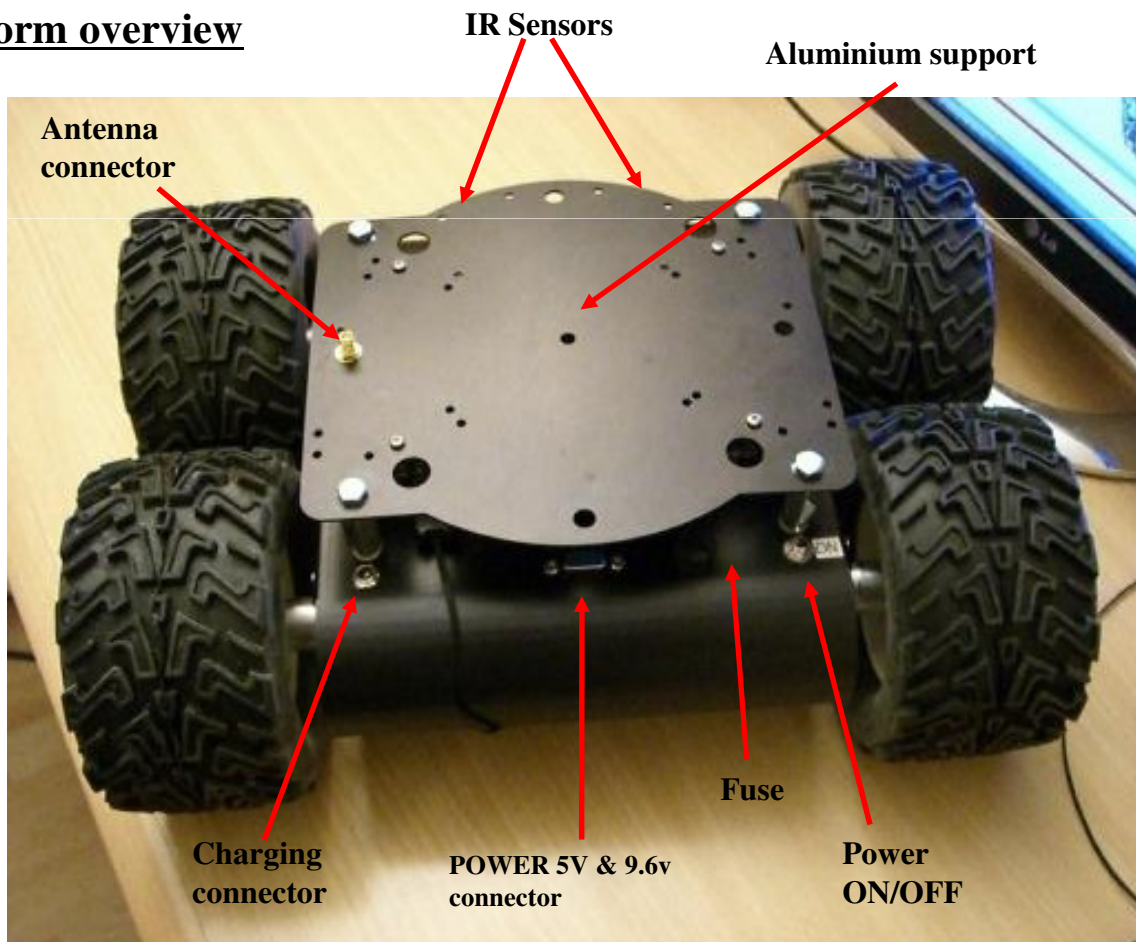
**4 wheels and a screw driver**

## Quick start

- 1- Install the RTMIX multi-robot interface or Simple GUI (see page 8).
- 2- Switch ON the robot or robots.
- 3- Set you IP settings (see page 12) for example:  
192.168.1.25 mask 255.255.255.0
- 4- Connect to the robots' ad-hoc network (see page 13)
- 5- Launch GUI for controlling the robot



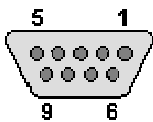
## Platform overview



## Platform interfaces

### **The 5V,GND and 9.6V power output:**

A 5V output, GND and a 9.6V filtered battery output are accessible at a **female** DSUB-9 connector located at the rear of the robot. Pin 1-2 are 9.6V, pin 4-5 are 5V and 6-7-8-9 are GND output respectively. The 5V output can't give more than 4A and a maximum of 6A is recommended too for the 9.6V output. 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.



### **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.



**The ON/OFF switch:**

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

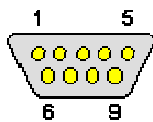
**The Fuse:**

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                  | 5- extra ADC |
| 2- ADC IR Sharp Left   | 6- SDA (I2C) |
| 3- ADC IR Sharp2 Right | 7-8- GND     |
| 4- extra ADC           | 9- SCL (I2C) |



**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 and finally switch the charger ON and set the charging current.



### **Caution:**

Charge the robot at 2.5A on regular use.

To charge the robot, switch off the power, plug the charger, push once the red button and when the red light is blinking fast, tune the potentiometer to the desired Amp, the charger will stop charging automatically.

The selector permit you to view battery voltage, charging Amp's, and mA/ hour injected in the batteries.

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.

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

Do not let the battery without charging for more then 4-6 months, if so they can become dangerous.

## Computer and camera installation

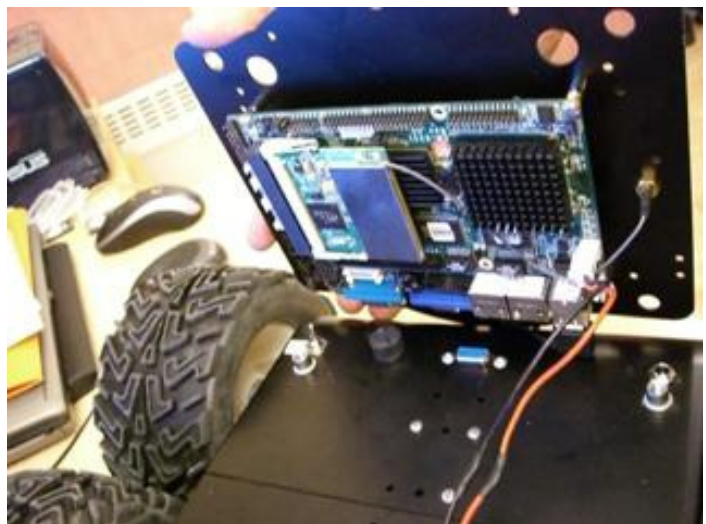
The platform is sold with an **Pan & Tilt IP camera or webcam and an embedded computer** which model can vary depending of the version. Those are independent elements from the platform which can be replaced by any other model. For more information about your particular camera and embedded computer please 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:

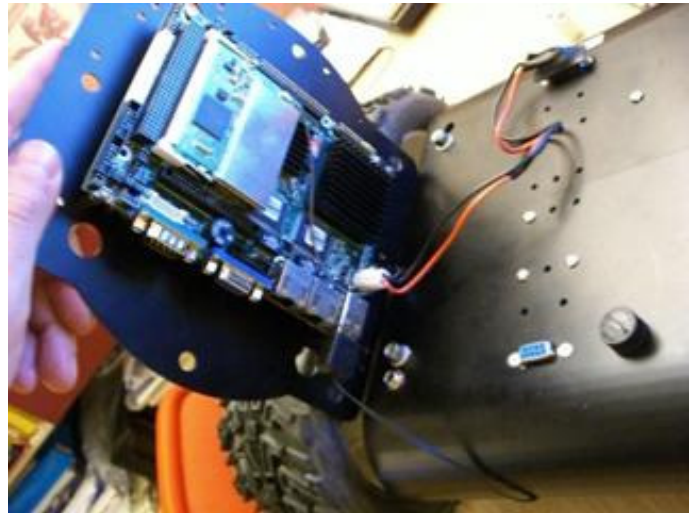
The upper aluminium support can be Unscrew :  
:



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



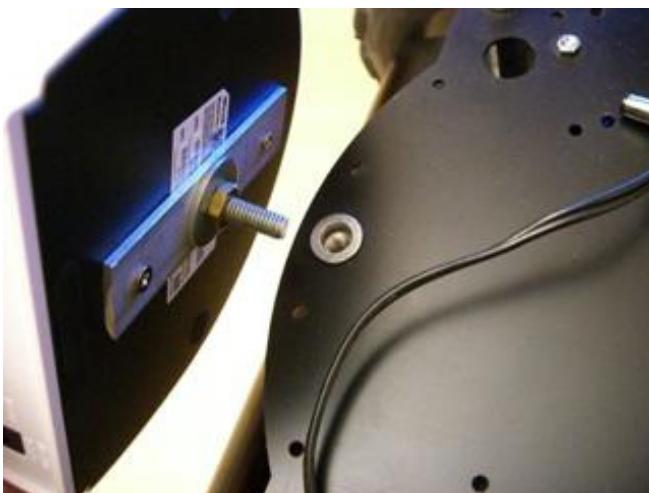
The power cable of the computer is connected to the appropriate connector:



The Control Command USB connector is connected to the computer.



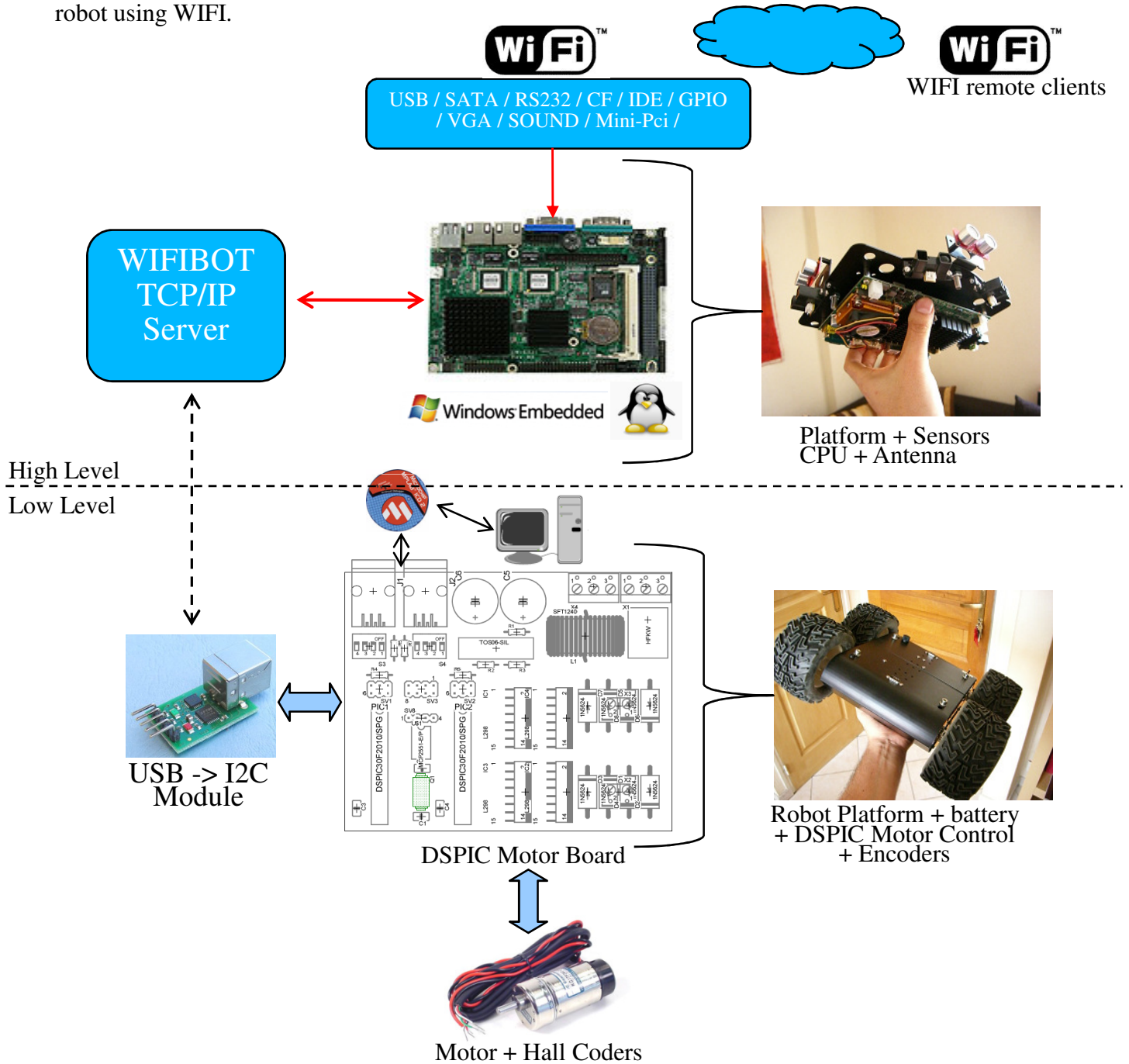
Screw back the aluminium support on top of the platform and screw the IP camera on top of the support.



## 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. An I2C to USB adapter is the link between the CPU and the low level.

Once plugged, Linux or windows see the I2C bus as a serial port. So to control the wheels or receive sensors data, you need to send or receive RS232 frame. **The protocol is explained at the end of this manual.** A simple TCP/IP gateway is provided with source code to see how it is simple to control the robot using WIFI.



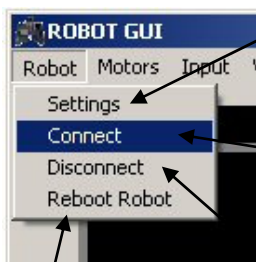
## The control software (TCP) Simple GUI:

### 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.

The menu options:

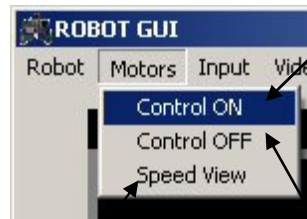


**Reboot:** Reboots the robot's CPU if available.

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

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

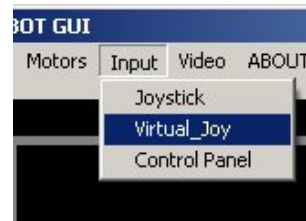
**Disconnect:** Stops the communication with the Control Server.



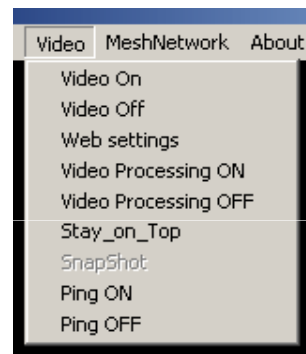
**Speed View:** Plots in real time the speed signal from the code wheels.

**Motor Control ON:** Activates the speed control, Input\_Left and Input\_Right set on the dialog will be applied.

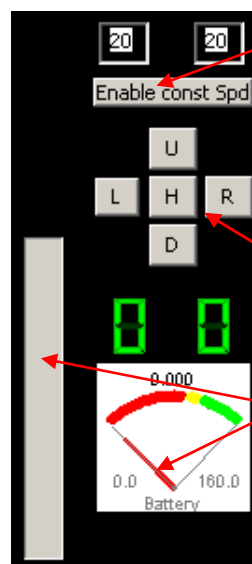
**Motor Control OFF:** Deactivates the speed control.



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



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

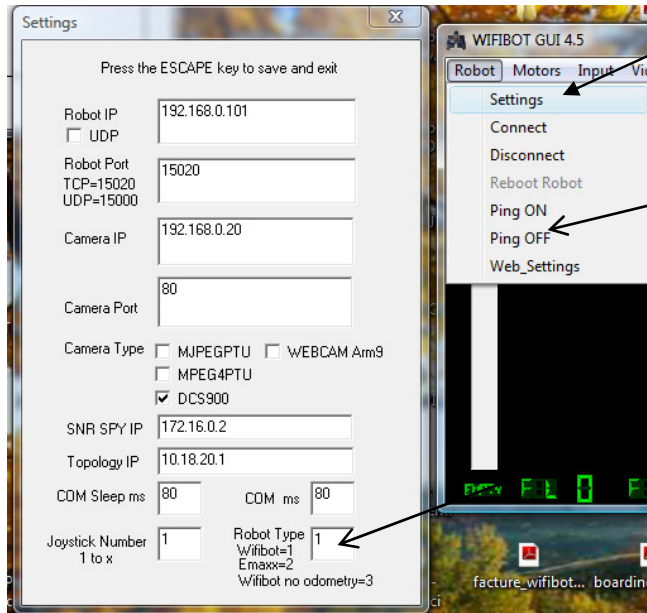


**Current input:** shows the current input or allows to set it manually with keyboard.

**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 and the speed of the robot in tics.

# The control software (TCP) Simple GUI:



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

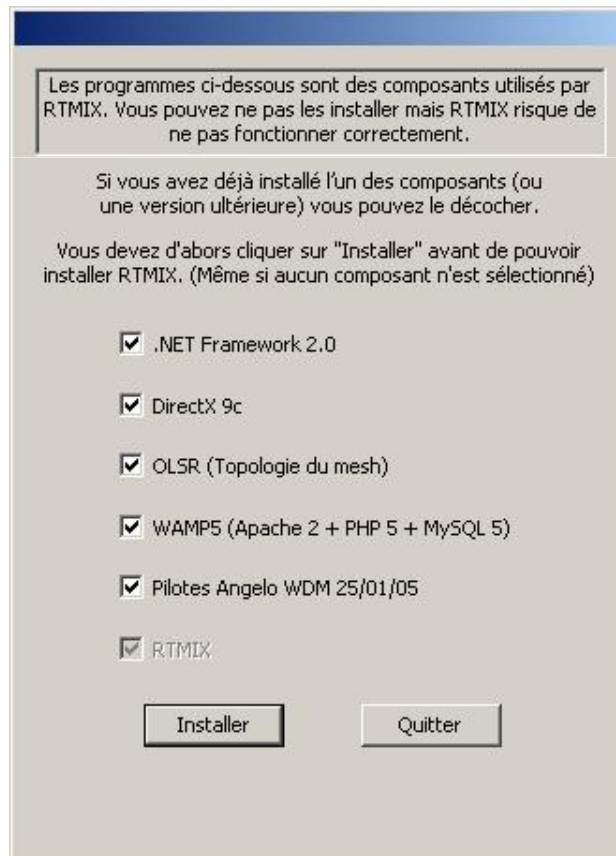
**Ping Robot**

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

## RTMIX multi-robot interface:

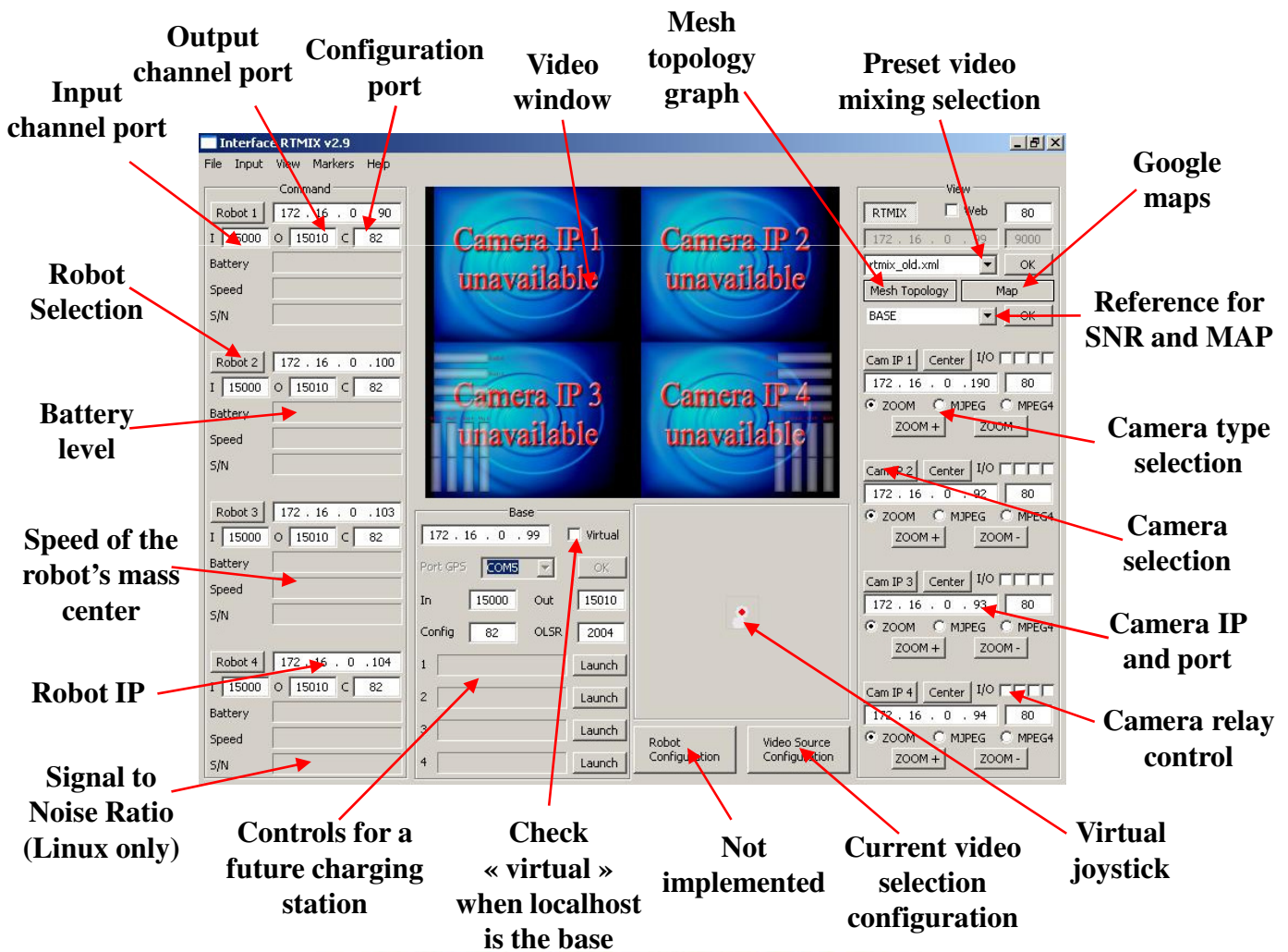
### Installation:

- 1- Unpack the zip file.
- 2- Double-click on Setup.
- 3- Unselect from the program list those you may already have installed on your computer.
- 4- Click on the « install » button.
- 5- When asked to reboot your computer always say « no ».
- 6- Once installation is done, create a folder « rtmix » in the web shared folder of the web server installed on your computer.  
If you have followed the standard installation this would be « C:/wamp/www/ »
- 10- Copy the content of «C:/Program Files/RTMIX/monsite/» into the «rtmix» folder you just created, ex:«C:/wamp/www/rtmix/»
- 11- Reboot your computer.

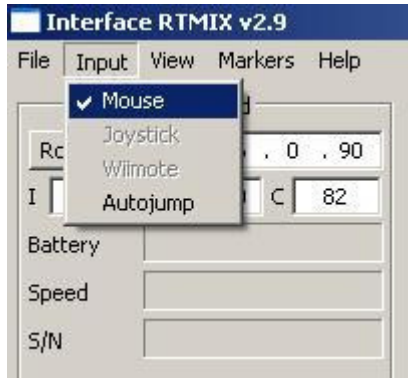


## Features:

- The RTMIX multi-robot interface allows the user to control a team of up to four robots.
- The interface is best viewed with a 1024x768 resolution.
- Platform related controls are located on the left while visual related ones are located on the right.
- Before operation the user has to make sure all IP addresses and ports are correctly set.
- Selected robots can be controlled using the virtual joystick, a joystick or a wiimote, selecting more than one robot at a time will have as a result having all of them receiving the same command.
- Cameras can be selected individually or blended in one image with RTMIX. Preset mixing layouts can be selected and new ones added thanks to the “video source configuration” webpage.
- Certain functionalities need a reference which can be selected with a menu located just under the Mesh and Map buttons.
- Certain types of supported cameras have incorporated relays which are set with the I/O checkbox.
- The “base” can be either an external gateway or the control computer itself (check “virtual” for this).
- The Map button needs to have an active internet connection and will show the robot’s and the base position on a Google Maps webpage provided those are equipped with a GPS.



## Program menu:

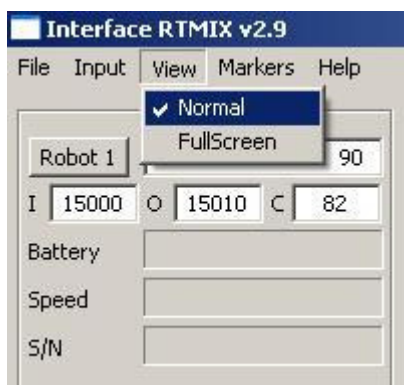
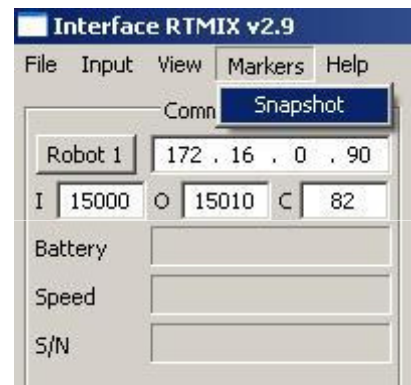


The **input** menu allows to choose between four modes of controlling the robots. Those are:

- mouse (virtual joystick)
- joystick
- wiimote
- autojump (virtual joystick)

This is a mode where the robot advances a certain distance with a set speed and radius.

The **markers** menu allows for now only to take snapshots of the current video window and to place a marker on the map at the location it was taken. The snapshot can then be seen by clicking on the marker.



The **view** menu allows to choose between two modes of presentation:

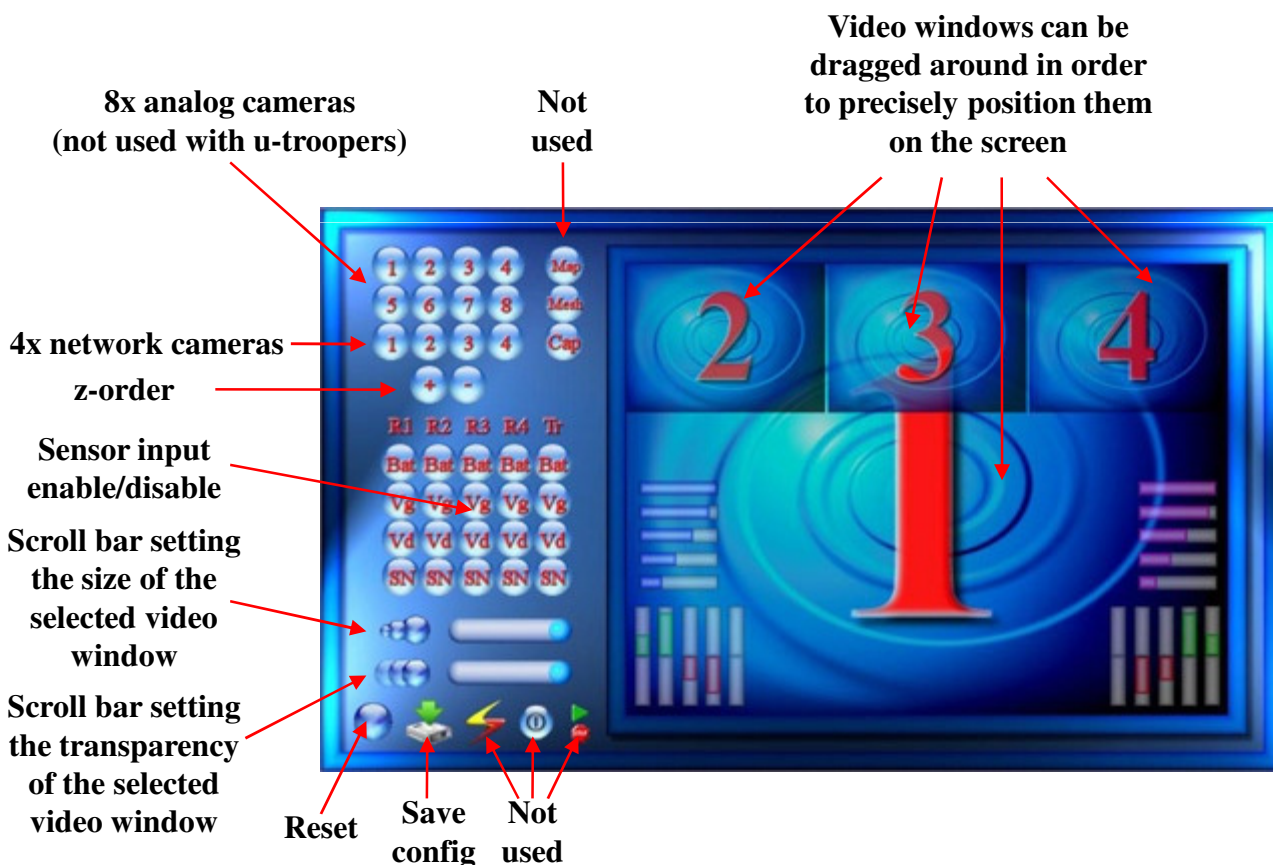
- Normal view (default)
- Fullscreen view

This is a mode where the video windows covers the whole screen. It is particularly useful when using configuration webpages or driving the robots.

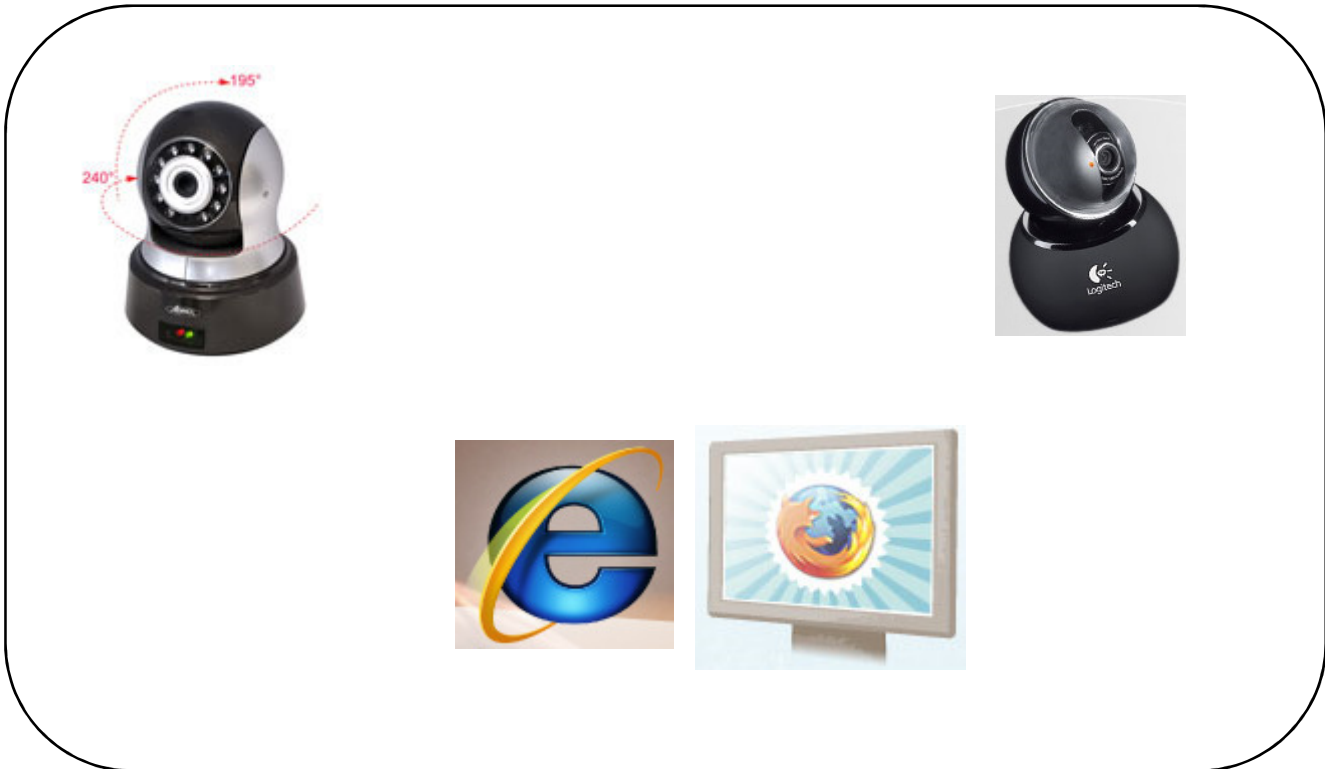
## Configuration:

The configuration webpage is accessible clicking on the “video source configuration” button when RTMIX is selected. The video mixing capabilities were originally designed to be embedded in larger robots and controlled remotely. They were only later adapted to the robolab system, this is the reason why some options such as the analog cameras are not applicable. To create a custom layout to be added to the interface menu do the following:

- Select the cameras and sensors to be displayed.
- Drag the cameras and the sensors at the desired location.
- Set the size, opacity and z-order of the different cameras.
- Press the save button and introduce a name.

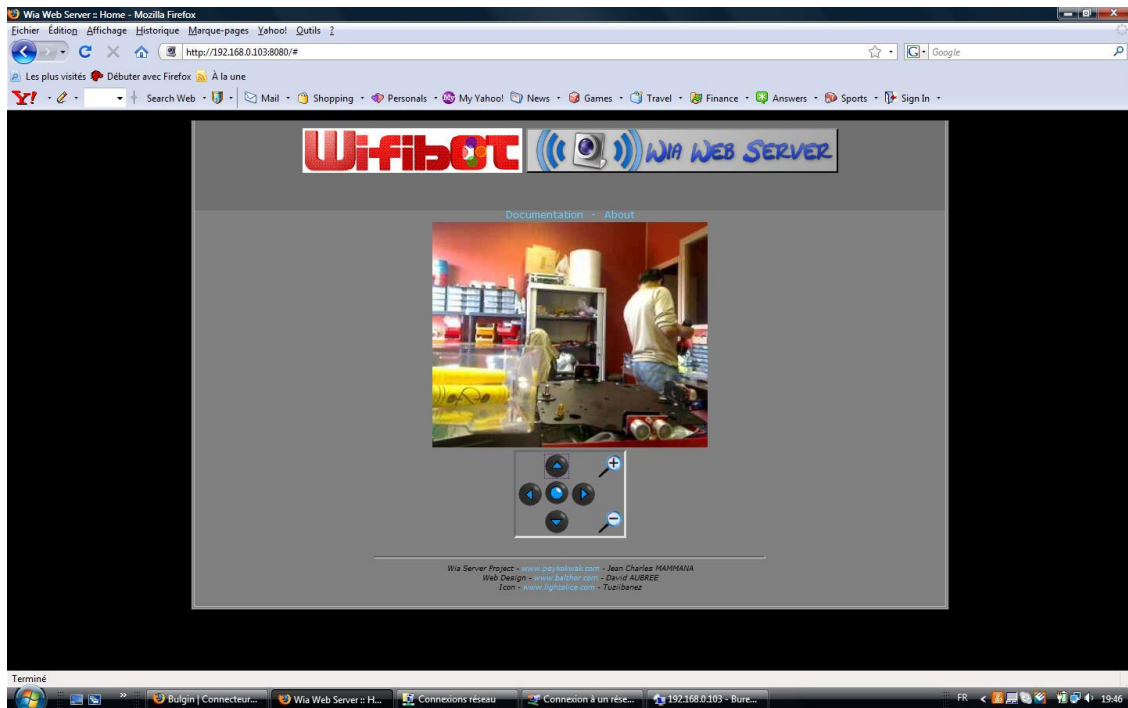


# Camera Type and software clients compatibility:

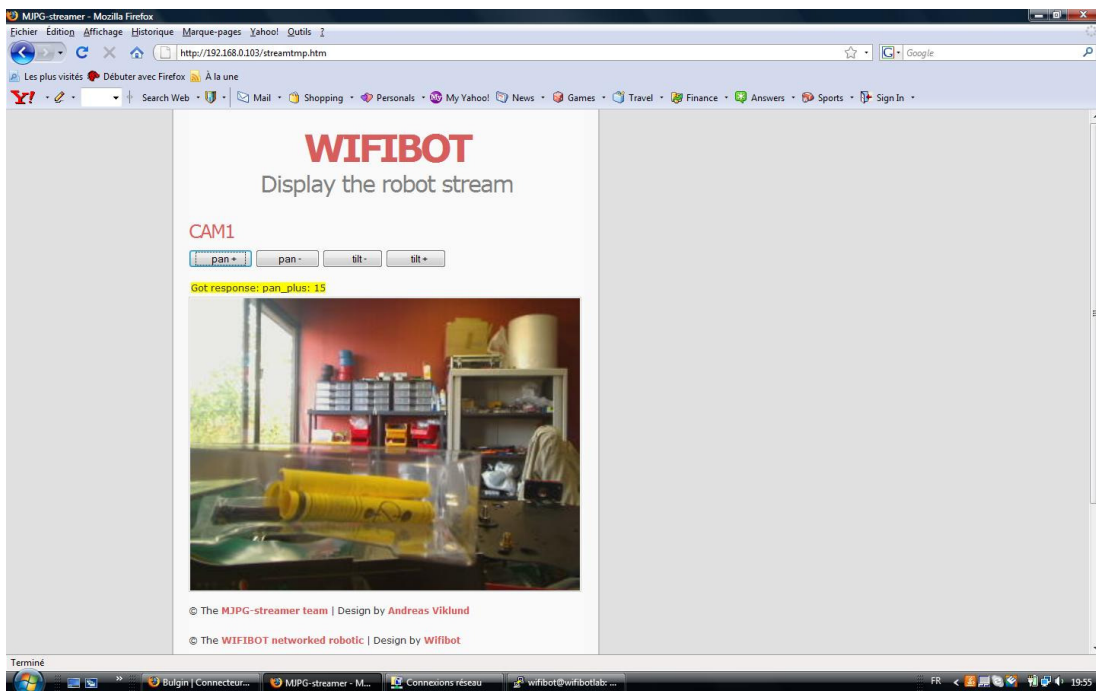


## Web Cam Server web client:

Windows **WIA SERVER**: <http://192.168.0.XXX:8080>



Linux **MJPEG-STREAMER**: <http://192.168.1.XXX:8080>



## Connecting to the robot:

### Configuring your ethernet/wireless adapter:

By default, the robot has been pre-configured with certain IP addresses. Before connecting to the robot you may need to adjust the IP settings of the network adapter of your computer. Make sure all the devices in a same network having to communicate with the robot have the same class of address.

To adjust the TCP/IP settings of the network adapter:

1 - Right-click on **My Network Places** in the **Start** menu, then select **Properties** from the pop-up menu. **The Network and Dial-up Connections** window appears. (Fig1)

2 - Create a network bridge between your interfaces or else disable all the network adapters except the one you want to use for connecting to the robot (Ethernet or Wi-Fi). Right-click the network adapter, then select **Properties** from the pop-up menu. (in Fig1)

3 - Double-click the **Internet Protocol (TCP/IP)** item to display the **Internet Protocol (TCP/IP) Properties** window. (in Fig2)

4 - Check the **Use the following IP address** option, then enter the IP address for the network adapter. **Set IP address** depending on the case : (in Fig3)

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**.

Fig 1

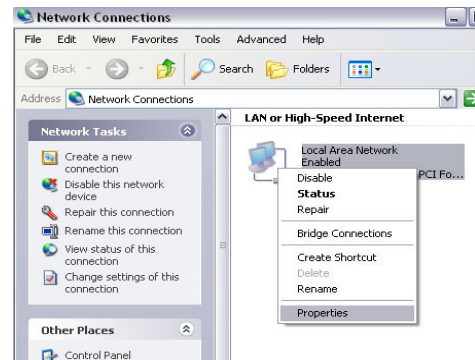


Fig 2

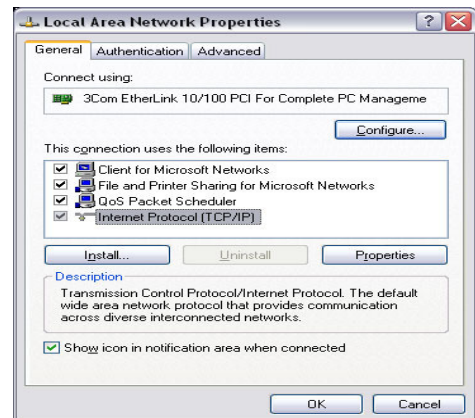
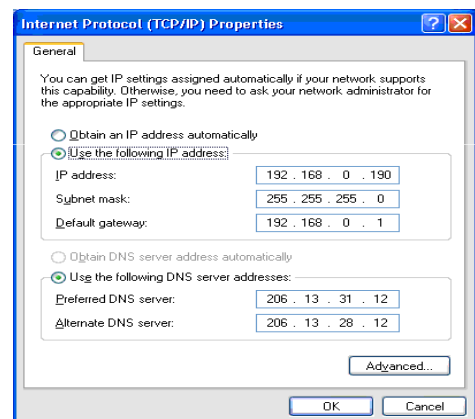


Fig 3



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  
5 - Click **OK** when finished.

## Connecting your wireless adapter to the robot:

Once you have adjusted the TCP/IP settings, if you are using a cable and providing the robot is switched **ON** then you are already connected but if you are connecting wirelessly you have to make sure your wireless adapter is connected to the robot and not to something else. This can be done through windows or using the software provided with your adapter.

To connect your wireless adapter to the robot using windows, follow these steps:

- Switch **ON** the robot and wait a few seconds.
- In the **Network and Dial-up Connections** window, right-click on the wireless network adapter, then select **Properties** in the pop-up menu. The **Wireless Network Connection Properties** window appears. (Fig1)
- Click the **Wireless Networks** tab. A list of wireless access points appears in the **Available networks** box. (in Fig1)
- If the **wifibot** network is not listed in the **Available networks**, then click **Refresh** till it is. (in Fig1)
- Check the **Use Windows to configure my wireless network settings** option. (in Fig1)
- Click **OK** when finished.
- Right-click on the wireless network adapter again, then select **View Available Wireless Networks** from the pop-up menu (in Fig2).
- The **Connect to Wireless Network** dialog box opens with a list of available networks in the **Available networks** box. (in Fig3)
- Select the **wifibot** network from the list, then check **Allow me to connect to the selected wireless network, even though it is not secure** option. (in Fig3)
- Click **Connect** (in Fig3), a pop-up window at the bottom of the screen should appear indicating you are connected. (in Fig4)

Fig 1

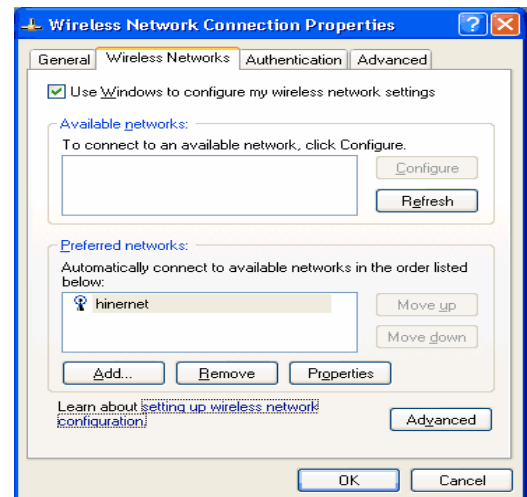


Fig 2

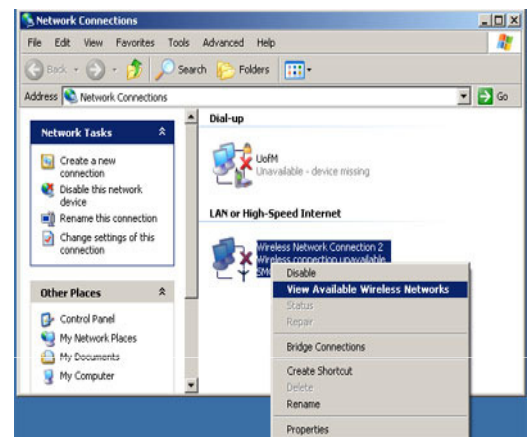


Fig 3

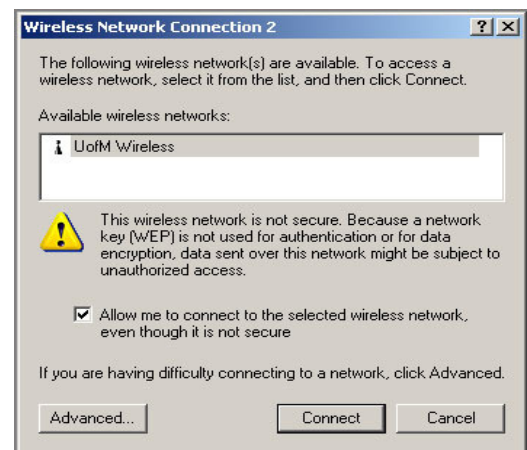
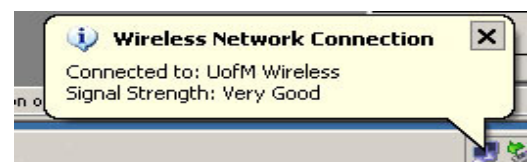


Fig 4



# 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 addresses, 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.

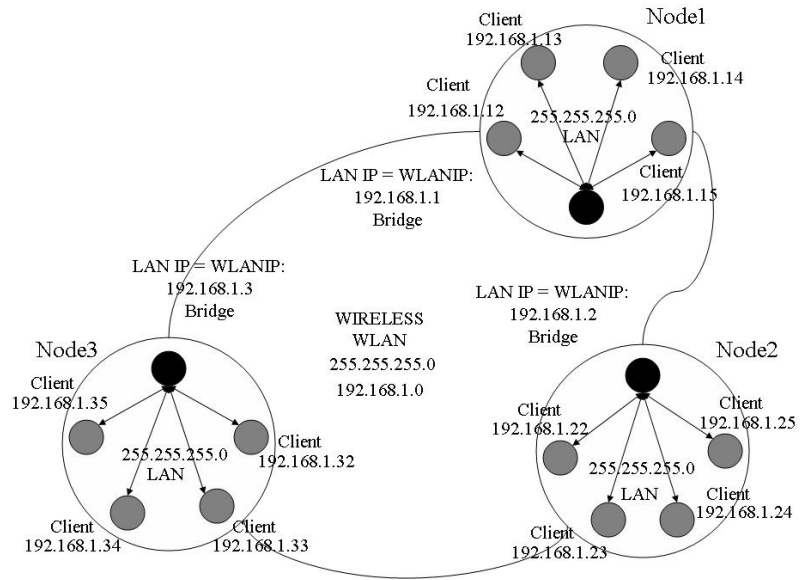


Fig 1

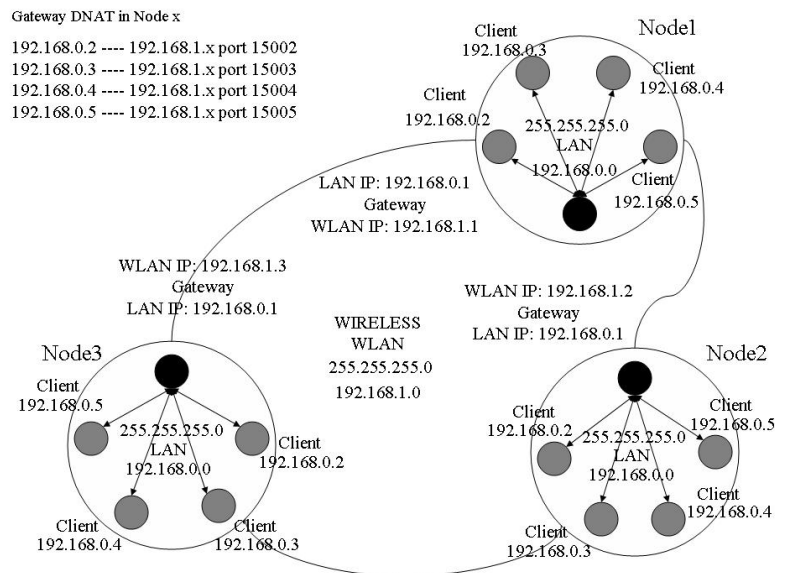


Fig 2

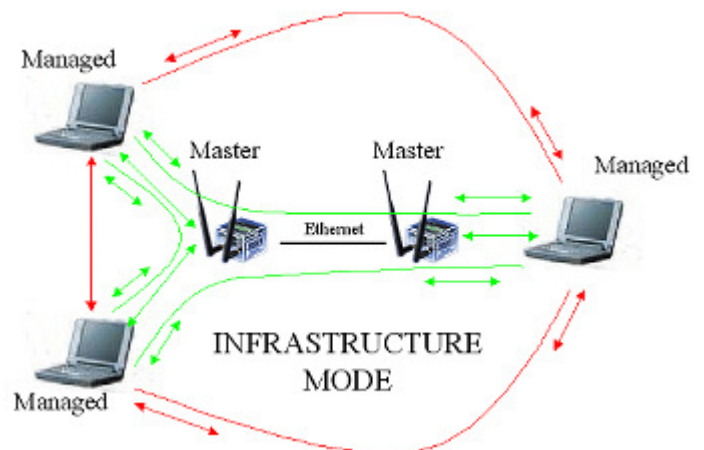
## WLAN modes:

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

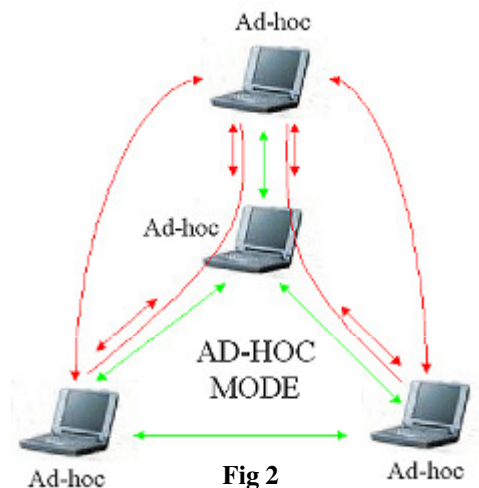
- Infrastructure Master (Access Point)
- Infrastructure Managed (Adapter/Bridge)
- Ad-hoc without routing algorithm
- Ad-hoc with the OLSR routing algorithm (Mesh Networking)

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**).

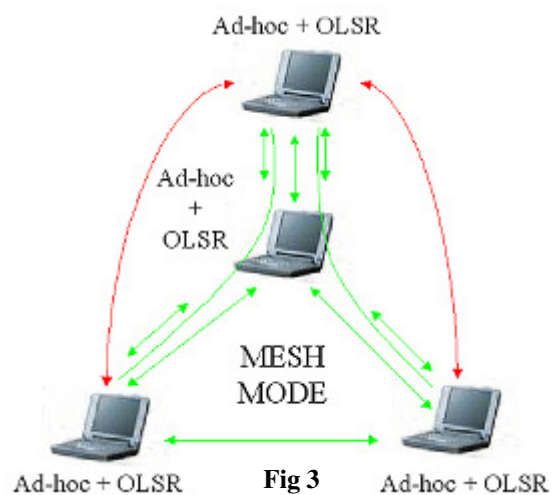
In ad-hoc mode we do not have any central management, each client can talk directly to the other. This mode works fine for networks with few elements. Without any routing algorithm, each element needs to have a direct radio link with the others in order to communicate, no data will be forwarded (see **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 1**



**Fig 2**



**Fig 3**

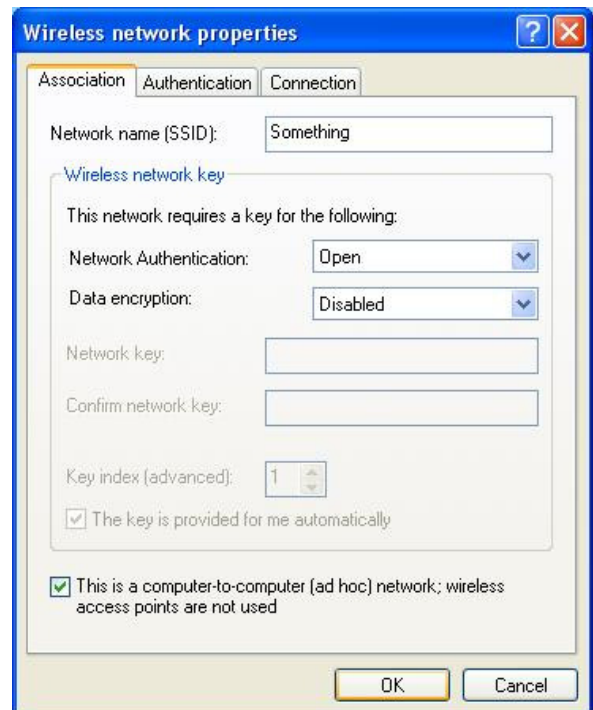
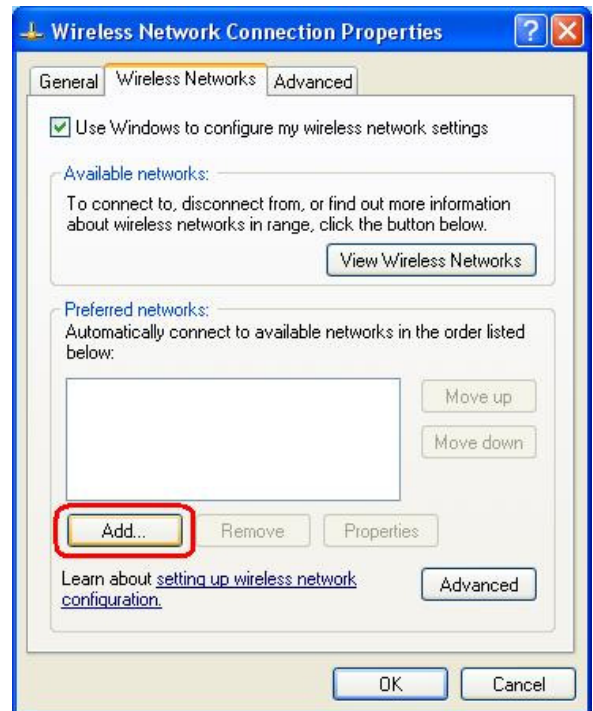
## Network configuration

### Robot under Windows XP

By default all robots come already configured and ready to work. 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

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

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

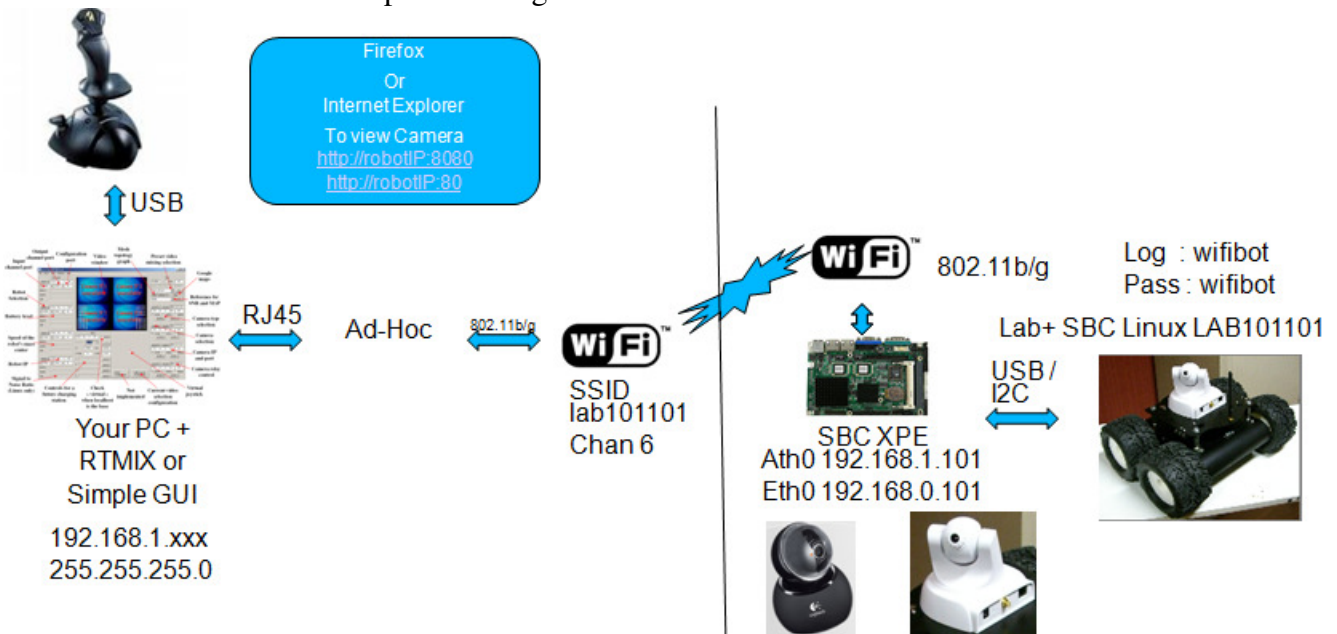
**/usr/sbin/wifibot-init**  
**/usr/sbin/robot\_server**

The I2C modules for the robot or the ultrasonic sensors are in /dev/ttyUSB0 and /dev/ttyUSB1

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 ath00 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:

```
interfacesClient - WinVi
Fichier Edition Recherche Options Fenêtres Aide
auto lo
iface lo inet loopback

auto ath0
iface ath0 inet static
# ensure ath0 is down (never fails because of "true")
pre-up wlanconfig ath0 destroy || true
# set up the ath0 device in AP mode before bringing up the interface (unless you're using AutoCreate)
pre-up wlanconfig ath0 create wlandev wifi0 wlanmode sta
address 192.168.1.106
netmask 255.255.255.0
gateway 192.168.1.2
wireless-essid linksys
pre-up iwconfig ath0 essid linksys
# IF you use WEP, put the key here:
#wireless-key 1234-1234-1234-1234

auto eth0
iface eth0 inet static
address 192.168.0.106
netmask 255.255.255.0

#auto eth1
#iface eth1 inet static
#address 192.168.2.105
#netmask 255.255.255.0
~
<C:\Users\laurent\Desktop\x86_inst904\x86_inst\network\interfacesClient> [Lf] 26 lignes, 67
```

- 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:

```
interfacesAdHoc - WinVi
Fichier Edition Recherche Options Fenêtres Aide
auto lo
iface lo inet loopback

auto ath0
iface ath0 inet static
# ensure ath0 is down (never fails because of "true")
pre-up wlanconfig ath0 destroy || true
# set up the ath0 device in AP mode before bringing up the interface (unless you're using AutoCreate)
pre-up wlanconfig ath0 create wlandev wifi0 wlanmode ad-hoc
address 192.168.1.106
netmask 255.255.255.0
gateway 192.168.1.2
wireless-channel 10

wireless-essid lab100106
pre-up iwconfig ath0 essid lab100106
# IF you use WEP, put the key here:
#wireless-key 1234-1234-1234-1234

auto eth0
iface eth0 inet static
address 192.168.0.106
netmask 255.255.255.0

#auto eth1
#iface eth1 inet static
#address 192.168.2.105
#netmask 255.255.255.0
~
<C:\Users\laurent\Desktop\x86_inst904\x86_inst\network\interfacesAdHoc> [Lf] 28 lignes, 67
```

- 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 “S31wifibot” 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 setting 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)

```
wifibot-init* - WinVi
Fichier Edition Recherche Options Fenêtres Aide
# ! /bin/bash
sysctl -w net.ipv4.ip_forward=1
source /etc/wifibot.ini
sudo iptables -t nat -A PREROUTING -i ath0 -p tcp -d $robotwlanip --dport $camport1 -j DNAT --to $camip:$camport1
sudo iptables -t nat -A PREROUTING -i ath0 -p tcp -d $robotwlanip --dport $camport2 -j DNAT --to $camip:$camport2
sudo iptables -t nat -A PREROUTING -i ath0 -p udp -d $robotwlanip --dport $camport3 -j DNAT --to $camip:$camport3
sudo iptables -t nat -A PREROUTING -i ath0 -p udp -d $robotwlanip --dport $camport4 -j DNAT --to $camip:$camport4
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
~
~
~
<C:\Users\laurent\Desktop\86_inst\904\86_inst\bin\wifibot-init* [Lf] 16 lignes, 914 caractères
```

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:

- 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**).

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**).

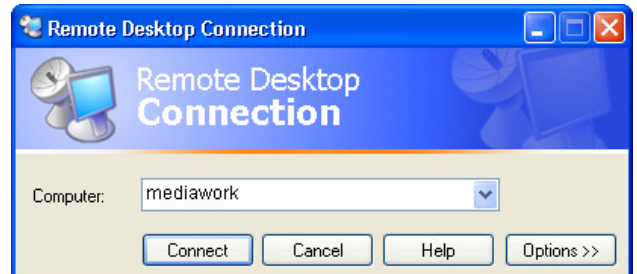


Fig 1



Fig 2

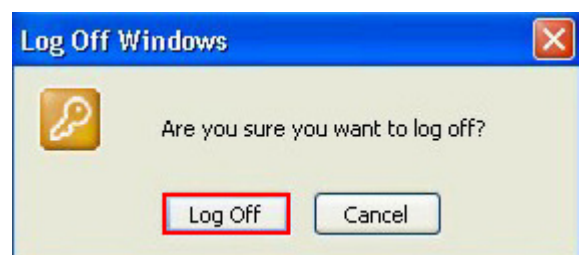


Fig 3

## 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:

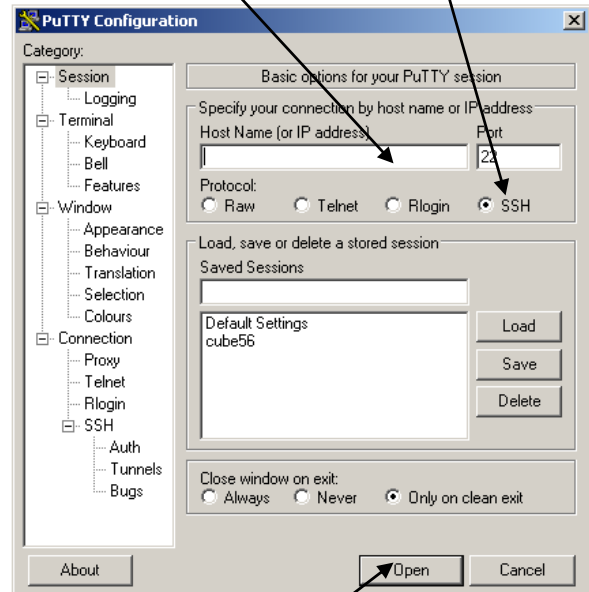
- Open the SSH client.
- Enter the CPU IP address (the default port is 22) and then start the connection.
- The first time a connection is established, the program will ask for confirmation.
- Enter login: **wifibot**.
- Enter password: **wifibot**

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

Connect to the robot in the following steps:

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

2 - Check the **SSH** option.

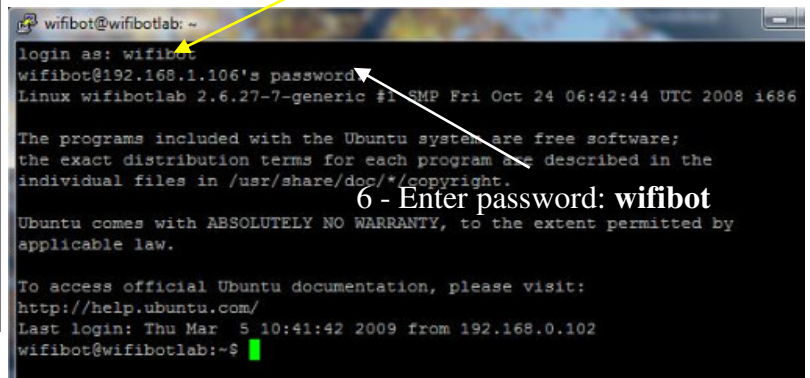


3 - Click **Open** to start the connection.



4 - Confirm the connection.

5 - Enter login: **wifibot**



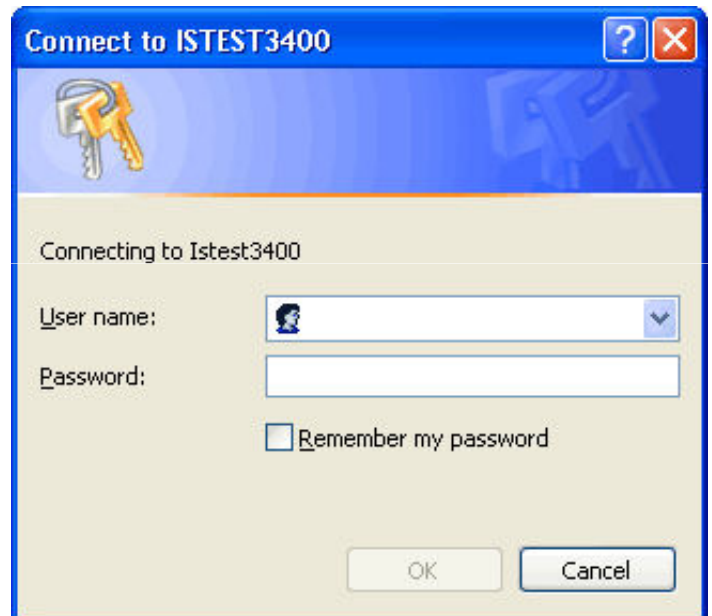
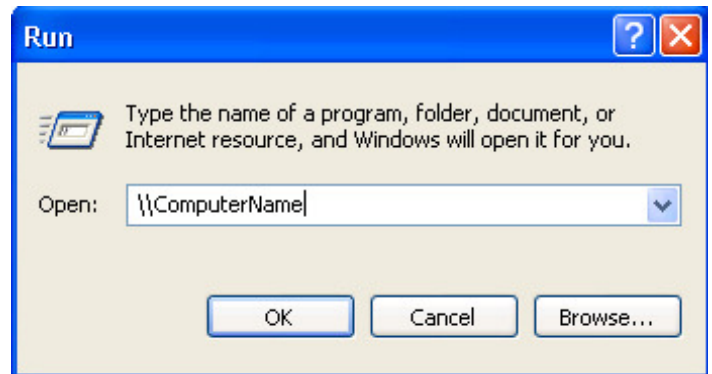
6 - Enter password: **wifibot**

## 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 add software or 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.

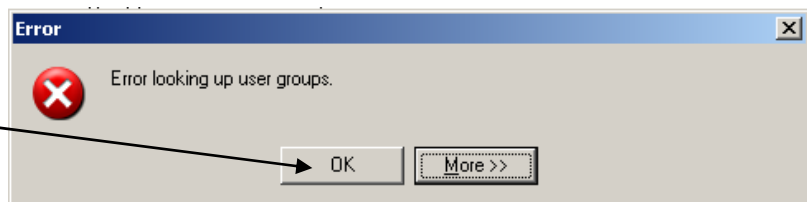
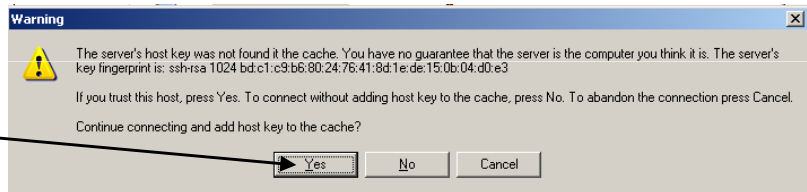
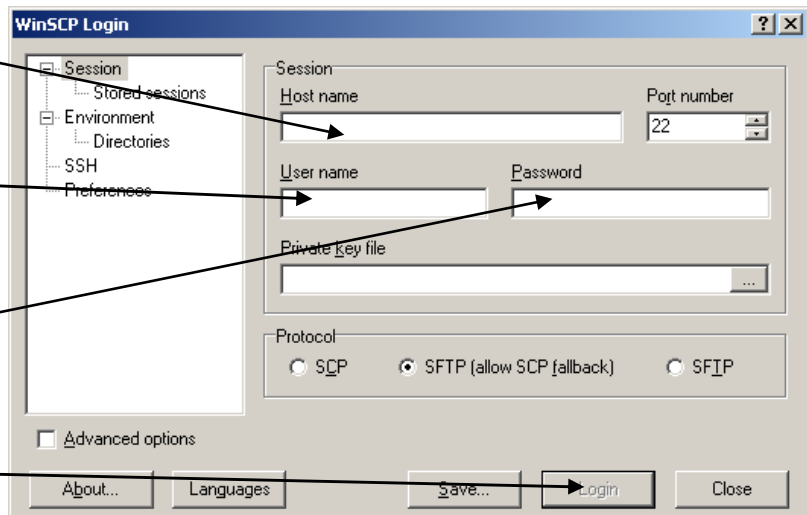
2 - Enter User name :  
**wifibot**

3 - Enter password: **wifibot**

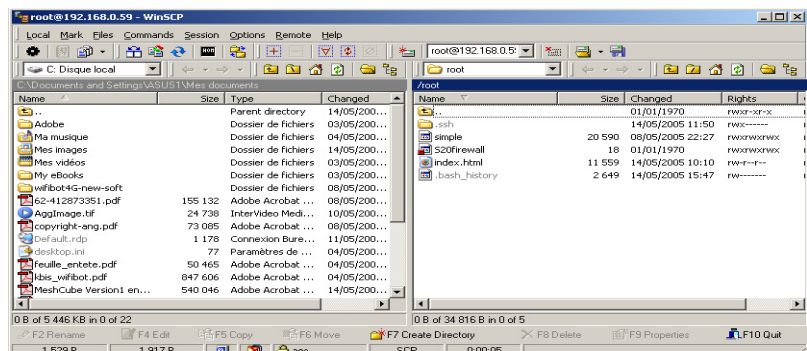
4 - Click **Login** to  
initiate the connection.

5 - Confirm the  
connection.

6 - Confirm again.



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



## Communication protocols

The robolab has been thought as an open robotic tool and as such it allows the user to have an absolute control over the higher layers. The user can work at two different levels with a robolab, on one side it is possible to program on the embedded computer your own autonomous behaviours and on the other you can consider the robot as a network peripheral and program all processes on a remote computer to finally only send the final commands to the robot or robots to perform.

The only level that is not accessible to the user is the microcontroller that takes in charge the motor control unless the user plug a ICD2/ICD3 debugger. The programmer will have therefore to stick with the established communication protocol between the embedded computer and the microcontroller. For a description of this protocol please see the dedicated annex and the API. Users are free though to make use of the two existing protocols between the embedded computer and the provided control interfaces for their own projects or to create a new one altogether that would suit better their application needs. This low level communication is only a mater of sending and receiving RS232 frame over the USB/I2C module.

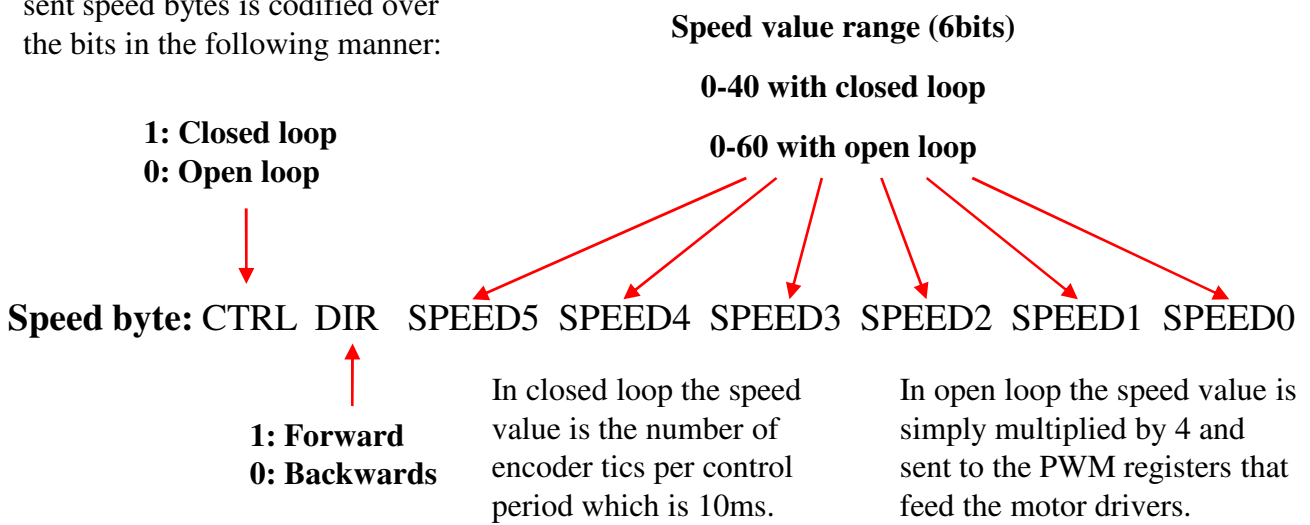
**The computer-WIFIBOT control interface protocol:**

Here the communication takes place through one TCP socket. The robot is the server and the interface the client. The communication channel is through robot's **port 15020** and unlike the other protocol it is used for both sending and receiving data.

Messages to port **15020**:

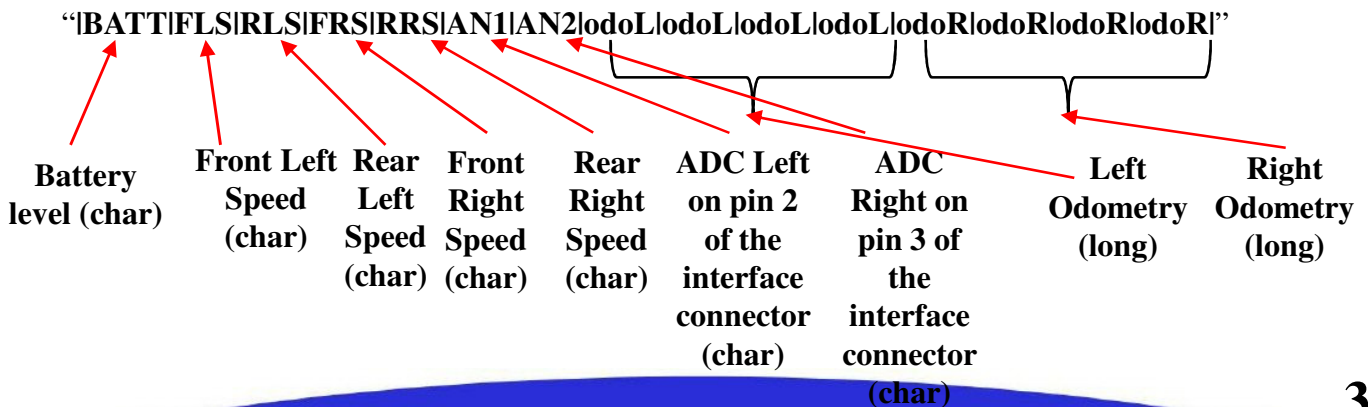
"**comg|comd**" this is simply the two speed command for the motor where "**comg**" is the speed for the wheels on the left (char) and "**comd**" is the speed for the wheels on the right (char).

The information contained in the sent speed bytes is codified over the bits in the following manner:



Messages from port **15020**:

The robot sends us a buffer of 14 char with sensor values. Note: if the robot is the two encoder version, then the rear speeds values are a copy of the front encoders values.



```

buffso_send[0]=(char)(bat+20);
buffso_send[1]=(char)dataL.SpeedFront;
buffso_send[2]=(char)dataL.SpeedFront;
buffso_send[3]=(char)dataR.SpeedFront;
buffso_send[4]=(char)dataR.SpeedFront;
buffso_send[5]=(char)dataL.IR;
buffso_send[6]=(char)dataR.IR;
buffso_send[7]=(char)dataL.odometry;
buffso_send[8]=(char)(dataL.odometry >> 8);
buffso_send[9]=(char)(dataL.odometry >> 16);
buffso_send[10]=(char)(dataL.odometry >> 24);
buffso_send[11]=(char)dataR.odometry;
buffso_send[12]=(char)(dataR.odometry >> 8);
buffso_send[13]=(char)(dataR.odometry >> 16);
buffso_send[14]=(char)(dataR.odometry >> 24);

```

### The computer- RTMIX control interface protocol:

Here the communication takes place through two channels of **UDP sockets**. The first channel (robot's **port 15000**) is for sending commands to the robot while second (robot's **port 15010**) is to request data from it. Communication is done by the use of different and specific messages that the robot separately processes in order to take the proper action. We will present here only the basic set of messages necessary for your program to deal with the control interface, please refer to the source code for more detailed information.

Messages to port **15000**:

“**!l!n!l!t!**” this has to be answered with “**!o!k!**” to let the control interface know we are really there

“**!s!p!e!l!d! !R!S! !L!S!**” this needs no answer. RS stands for right speed and LS for left speed.

Messages to port **15010**:

“**!l!n!l!t!**” this has to be answered with “**!o!k!**” to let the control interface know we are really there

“**!d!a!l!t!a! !x!x!x!.!x!x!x!.!x!x!x!.!x!x!x!**” this is a request for sensor data, the message includes the IP address of the reference for calculating the SNR. This needs to be answered with:

“**!B!A!T!T!R!S!I!S!N!R! !x!x!x!.!x!x!x!.!x!x!x!.!x!x!x! !L!A!T!...byte 39!L!N!G!...!**”

Battery level (char)	Speed feedback (char)	SNR (char)	Robot's IP address (char)	Latitude (float)	Longitude (float)
-------------------------	--------------------------	---------------	------------------------------	---------------------	----------------------

## **The CDROM**

The CDROM included with the robot contains the documentation and sample programs for the robot. It 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 RTMIX remote multi-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 behaviors etc.

### **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.

## The standard CPU

# Industrial Single Board Computer

## 3.5 inches Miniboard

### LE-374

Intel Atom processor  
Onboard VGA, LVDS , DVI , Giga LAN,  
Mini PCI, USB2.0, SATA  
AC97 Audio and Compact Flash socket



<b>Form Factor</b>	3.5 inches Miniboard
<b>CPU</b>	Intel Atom N270 processor 1.6GHz Package type: FCBGA8 , Front side bus: 533MHz
<b>Memory</b>	One 200-pin DDR2 SO-DIMM SDRAM up to 2GB Non-ECC, unbuffered memory supported only
<b>Chipset</b>	Intel 945GSE & ICH7M
<b>Real Time Clock</b>	Chipset integrated RTC with onboard lithium battery
<b>Watchdog Timer</b>	Generates a system reset with internal timer for 1min(sec) ~ 255min(sec)
<b>Power Management</b>	ACPI 1.0 compliant, supports power saving mode
<b>VGA Interface</b>	Intel integrated extreme GMA 950(Graphic Media Accelerator) Technology
<b>Video Memory</b>	Up to 224MB shared with system memory
<b>LVDS interface</b>	Onboard 18-bit dual channel LVDS connector
<b>Serial ATA interface</b>	2 x serial ATA interface with 150MB/s transfer rate
<b>Solid State Disk</b>	IDE supports 44-Pin Disk On Module with +5V power supply One Compact Flash Type II socket
<b>DVI Interface</b>	Onboard Chrontel CH7307C DVI Transmitter for DVI interface
<b>Audio Interface</b>	Intel ICH7-M integrated with Realtek ALC655 5.1CH AC97 Codec
<b>LAN Interface</b>	1 x Intel 82574L Gigabit Ethernet controllers
<b>GPIO interface</b>	Onboard programmable 8-bit Digital I/O interface
<b>Extended Interface</b>	1 x Mini-PCI socket
<b>Internal I/O Port</b>	1 x IrDA, 1 x GPIO, 1 x AUDIO, 1 x CDIN, 1 x LVDS, 2 x USB2.0 1 x RS232/RS422/485, 1 x DVI, 1 x LCD Inverter, 2 x SATA , 1 x HDTV
<b>External I/O Port</b>	1 x COM, 1 x VGA, 1 x RJ45, 2 x USB2.0, 1 x PS2
<b>Power Requirement</b>	9~24V full range DC Input
<b>Dimension</b>	146mm x 101mm
<b>Temperature</b>	Operating within 0~60 centigrade Storage within -20~85 centigrade