Quick Start Guide

Thank you for choosing **WiFiBoT SC** for your robotic application.

- Before using the robot, 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

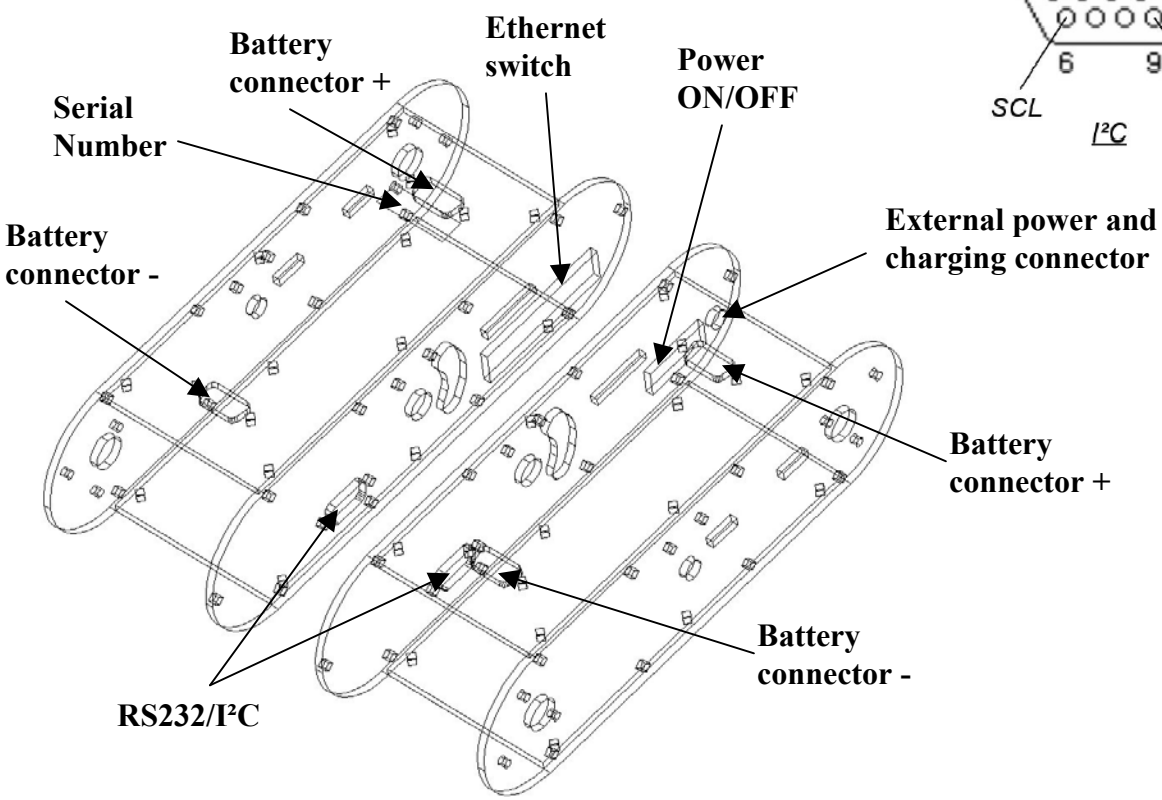
Make sure to be in possession of all the articles mentioned below. If any of them should be missing, contact your reseller as soon as possible.

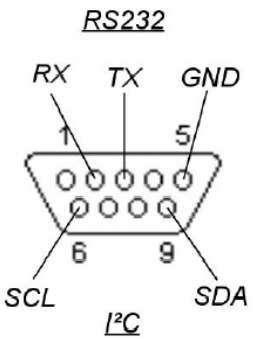
Robot
IP camera
Two battery packs
Battery charger
Wifibot CD-ROM
Camera CD-ROM and documentation
Embedded access point CD-ROM and documentation
4x charging cables
1x RJ45 cable for the IP camera

WARNING!
The DSUB-9 connector on the left of the robot presenting on its pins both the I²C and the RS232 interfaces, be careful when wiring a connection cable to prevent to connect those to unwanted tensions.

Connectors overview

The figure below shows the location of the different connectors of the robot:





Powering the robot

Powering the robot with batteries:

The robot gets its power normally from two battery packs with four Ni-MH cells each, with a capacity of 9500 mAH and a total nominal voltage of 9,6V. Located on the upper part of the platform, their location and clamping have been especially designed to facilitate their removal and replacement in an easy and simple way. It is enough to insert the packs in their connectors, their shape preventing any error, the only thing left is to close the clamps and the robot is ready to go. The robot comes with only two packs, additional packs are available separately.

Powering the robot from an external source:

When developing custom applications it is often more practical to use an external power source rather than to have to constantly charge the batteries. Take out the batteries, then plug the included cable into the connector located next to the power switch of the robot and connect the robot to a lab DC power source at a voltage between 9 and 12 V. Make sure the power source can deliver several amps, especially if you plan to test the motors.

Power connectors

The 5V/9.6V power output:

The robot is equipped with two 5V 2A DC/DC converters. One of the converters is reserved exclusively to power the different internal components of the robot. The second converter is available to the user through the general power connector dedicated to external modules such an IP camera. A direct connection to the batteries is made available on this connector as well. The 5V output can't give more than 2A and a maximum of 10A is recommended for the 9.6V output. An incorrect use of this connector beyond those values (short circuit or other) can provoke a malfunction of the robot or of the DC/DC converter and even damage those.

Note: The WiFiBoT company will not in any case be considered responsible for any damage provoked by any incorrect use of this connector. Any reparation necessary for any damage caused by the incorrect use of this connector will not be covered by the warranty.

The external power and charging connector:

This connector presents directly the + and – of the robot and has a double use. On one side it allows to directly power the robot with an external source without having to use the batteries. The second use is to charge the batteries on the robot itself when no additional packs are available.

Note : When charging the robot make sure the power switch is OFF so the charger does not find the robot in march.

Battery installation

Insertion:

Locate the connectors and their direction on both the robot and the two battery packs. Insert the batteries till the end and secure them by closing the lateral clamps.

Extraction:

Open the clamps and pull the battery packs up.

Charge:

A battery charger is included with the robot and can be used for charging the batteries in two different ways:

Externally: This charging mode allows a continuous use of the robot by doing a rotation of several battery packs. Insert first each plug in the corresponding color on the charger side and then connect the three cables included for this purpose to the batteries as shown on the photo. (red color with the battery +, see page 3)

On the robot: This mode is recommended when the user has only one set of battery packs. First make sure the robot is OFF, then connect the plugs of the charging cable on the side of the charger and then on the charging connector located next to the power switch of the robot.

Camera installation

The robot is sold with an IP camera which model can vary depending of the robot version, it is a complement and is not part of the robot itself. It is an independent peripheral which can be replaced by any other camera model or network peripheral. For more information about your particular camera please refer to its manual included in the CD-ROM. Nevertheless its installation is similar in all models.

Place the camera on the central support:

Screw the camera on its support and if it applies, adjust manually the desired position angle.

Connect the Ethernet cable to the camera :

Connect the included Ethernet cable to the RJ45 port located at the back of the camera and to one of the ports of the embedded switch.

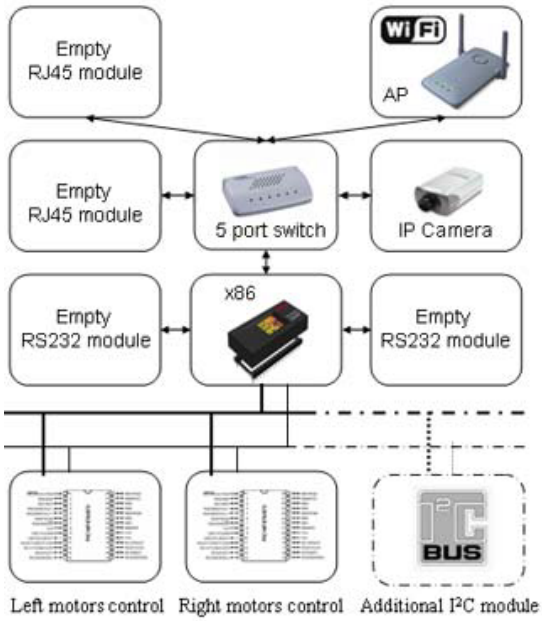
Connect the power cable of the camera:

Connect the proper power output (5V/9.6V) of the robot to the camera power input located at the back of the camera.

General structure of the robot

System architecture:

The system architecture is very simple, it is build around a double bus Ethernet-I²C and a central processor that acts as a bridge between the two. The Ethernet bus is then accessible from the outside through a Wi-Fi access point. In general the embedded LAN is used for peripherals of a certain importance such the IP camera while the I²C bus is useful for connecting more simple modules based on micro-controllers. To finish, the robot features two RS232 ports which can be bridged to upper levels as well. This makes possible to add to the robot commercial modules as well as "home made" ones based on simple micro-controllers.



The heart of the system:

The heart of the robot is the embedded controller SC12 from BECK, some of its interesting features are:

-186-20MHZ 16 bit CPU
-512 Kbytes RAM, 512 Kbytes Flash
-RTOS with Flash file system
-Ethernet,RS232 and I²C interfaces
-File transfer through serial and Ethernet links
-TCP/IP, PPP, HTTP, FTP, Telnet, POP3, SMTP et DHCP

The controller is totally programmable and all the documentation is available on the companion CD and on the website of BECK www.beck-ipc.com for updated material. Some programming examples are included in the CD as well.

Communication interfaces

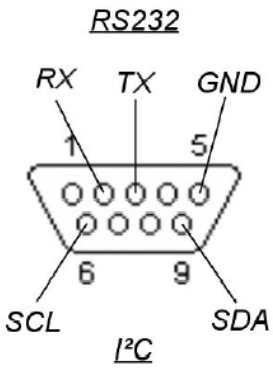
The Ethernet switch:

A 10/100 Ethernet 5 port switch interconnects the different high level peripherals of the robot forming an embedded LAN. Two ports are necessarily taken by the access point and the SC12 controller. One port is generally used by the IP camera but can be made available depending of the application. Finally, the two last ports are free to add other peripherals if needed, those working in a completely transparent way and independently from the robot.

The I²C bus:

The I²C bus interconnects the micro-controllers in charge of the low level modules to the central processor that acts as a bridge with the upper levels of the robot architecture. The SC12 controller works always as the master and all communications necessarily pass by it. Up to 127 devices can be connected and the maximum clock frequency of the bus is of 30kHz. The I²C is accessible on pin 5 (GND), 6(SCL) and 9(SDA) of the DSUB-9 connector located on the LEFT of the robot. The address of the chip in charge of the left wheels is 0xA2, the one of the right wheels is 0xA4.

WARNING!
The DSUB-9 connector on the left of the robot presenting on its pins both the I²C and the RS232 interfaces, be careful when wiring a connection cable to prevent to connect those to unwanted tensions.



The RS232 ports:

Two RS232 ports (RX,TX and GND) are available. Port COM1 on the DSUB-9 connector of the left, port COM2 on the DSUB-9 connector of the right. Both ports present a standard PC signal, for peripherals using TTL an interface such as the MAX232 is needed.

Embedded sensors

Speed control:

The four motors can be controlled in open or close loop depending of the needs of the user. Every wheel has an external laser cut code wheel of 300 sectors which signal is recuperated by an optical sensor. The signal is then filtered and sent to four independent PID speed controllers. When remotely controlling the robot, the close loop is done by the operator and therefore is not very useful at the wheel level but it is generally needed in autonomous applications. In both open and close loop control, the speed of each wheel can be retrieved as the number of sector per 1/25 of second but we note here that the "tank like" design of the robot does not allow consistent odometry calculations from the wheel speed.

IR ranging sensors:

The robot is equipped with two IR sensors with a maximal range around 1,30m. They work by triangulation and give an analog output inversely proportional to the distance to the obstacle. The output voltage is then digitalized over 8bits by a micro-controller and sent to the central processor. Those sensors can have different uses like for example to trigger emergency stops or during docking procedures. The output of the sensors is not linear and follows a certain curb. For more information refer to the sensor datasheet.

Battery level:

The robot autonomy is around 1h30 but can vary depending of how it is used. For this and in order to monitor the battery level, an 8 bit A/D converter has been directly connected to the I/O ports of the SC12 providing us this information.

Networking

Network addressing:

In the WiFiBoT SC, the embedded access point (AP) acts as a bridge between the robot's embedded network components and external clients. All the connected elements from different robots and other computers merge in one single network. This means several robots can be seen in fact as a single distributed entity where robots can access transparently the systems of others like if it was their own.

The default network address is 192.168.0.0 that is, make sure all the clients of the same network having to communicate with the robot have an IP address of the type 192.168.0.x. The default addresses of the different elements in the robot are (where xx are the last two figures of the serial number):

CPU: 192.168.0.1xx port 15000
IP Camera: 192.168.0.xx port 80
Access Point: 192.168.0.2xx

The addresses and other settings of those elements can be configured respectively through an http interface for the camera and the access point and through the INI file of the embedded calculator. For more information please refer to their respective manuals.

The wireless modes:

The robot is equipped with a Wi-Fi access point configurable in 3 different working modes.

-Infrastructure Master (Access Point)
-Infrastructure Managed (Adapter/Bridge)
-Ad-hoc

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 (see **Fig1**). In standard Wi-Fi, several access points connected together with cables can extend the zone covered by the wireless network by letting clients « roam» between them. Because of the limitations of the embedded access point in managed mode, the WiFiBoT SC does **NOT** have this ability to « roam ». Further firmware upgrades from the manufacturer (ASUS) may one day add this functionality.

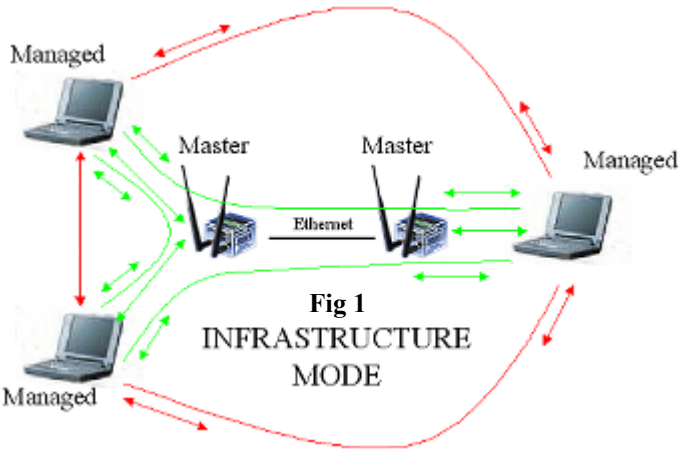


Fig 1

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. Each element needs to have a direct radio link with the others in order to communicate, no data will be forwarded (see **Fig2**). We recommend this mode if you want to connect to the robot directly from a computer.

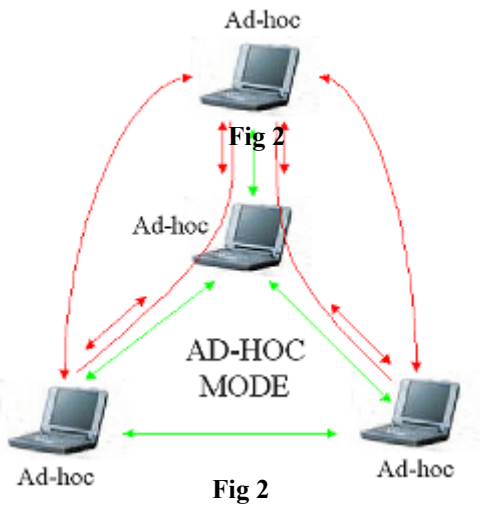


Fig 2

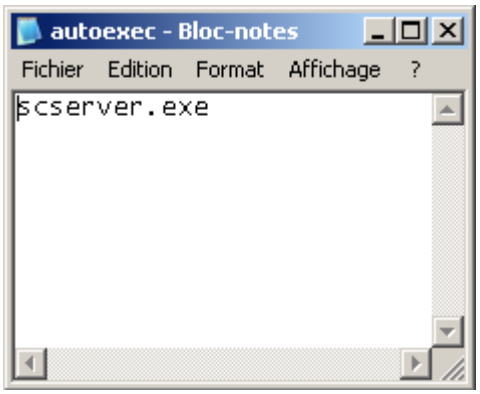
For more information about setting the wireless modes, please read the included access point documentation.

Configuring the robot:

Now that we have seen the different networking issues we will see how to configure the different parameters involved on the robot. For more information please read the documentation included in the CDROM **..\sc12documentation**.

There are 2 important configuration files we need to manage in the robot:

..\autoexec.bat
..\chip.ini

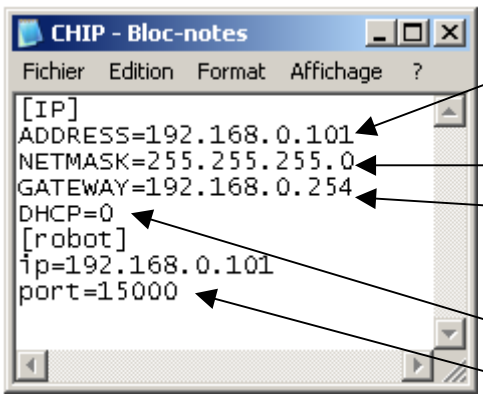


The "autoexec.bat" configuration file:

This file is very simple, all it does is to indicate what programs should launch upon start-up.By default it contains the name of the control server. **Note**: **it is important not to include your own programs here until you are sure they work. If it happens that one of them makes the SC12 crash, the only way to recover the chip would be to flash it again.**

The "chip.ini"configuration file:

In this file we should specify the IP settings of the robot and its control server.



- Address: this is the IP of the SC12, remember it has to be coherent with the network's class address.
- Netmask
- The gateway if you are plannig to make the robot communicate with elements located outside the network.
- DHCP:set to 0 if we assign the IP manually.
- Data to be recovered by the control server for its configuration. Here we will receive and send data using port 15000.

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 - 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**)

Whether you are connecting to the robot with a cable directly on its embedded switch or wirelessly through Wi-Fi, enter 192.168.0.x (x can be any number between 1 and 254 except those used by the processor, the access point and the camera of the robot). For example, a Wifibot Serial Number: S/N Y-AXX will have as IP for the CPU 192.168.0.1XX, 192.168.0.XX for the camera and 192.168.0.2XX for the access point.

Fig 1

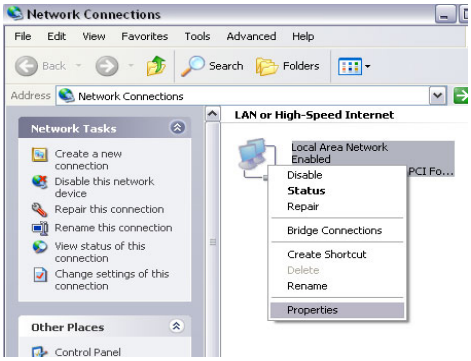


Fig 2

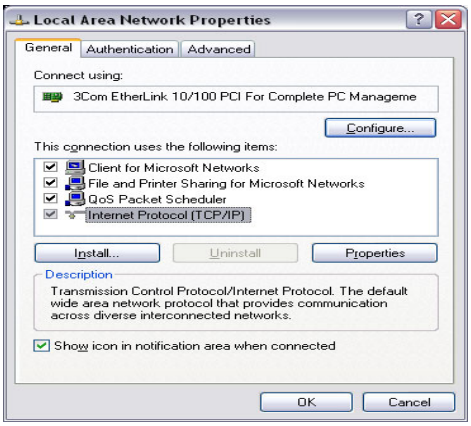
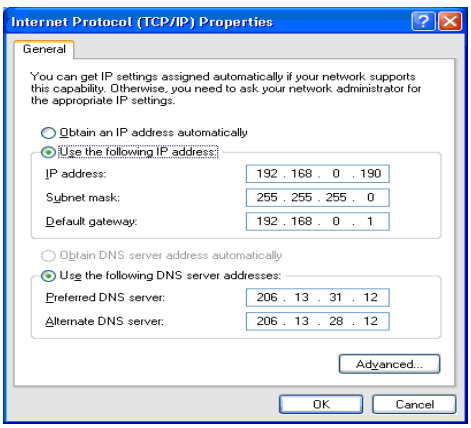


Fig 3



Those number are therefore not available. Set the **Subnet Mask** to 255.255.255.0 and leave **Default gateway** and **DNS** empty.

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:

1. Switch **ON** the robot and wait a few seconds.
2. 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**)
3. Click the **Wireless Networks** tab. A list of wireless access points appears in the **Available networks** box. (in **Fig1**)
4. If the wifibot network is not listed in the **Available networks**, then click **Refresh** till it is. (in **Fig1**)
5. Check the **Use Windows to configure my wireless network settings** option.(in **Fig1**)
6. Click **OK** when finished.
7. 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**)
8. 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**)
9. Click **Connect** (in **Fig3**), a pop-up window at the bottom of the screen should appear indicating you are connected. (in **Fig4**)

Fig 1

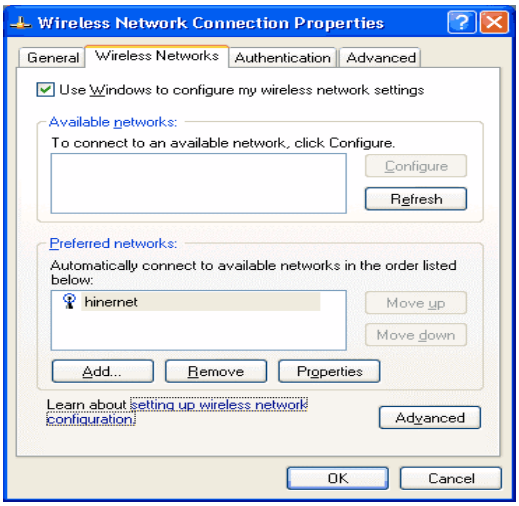


Fig 2

Fig 3

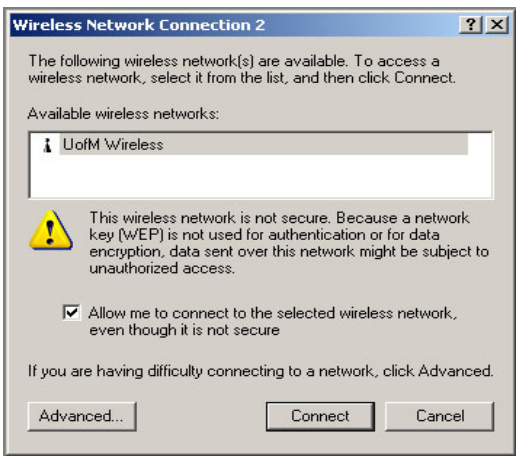
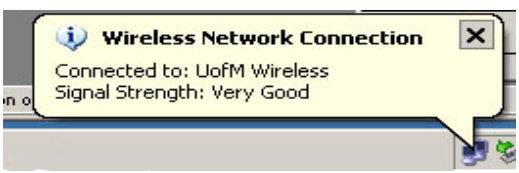


Fig 4



Whether you are connected to the robot directly though a cable or wirelessly you can now decide to:

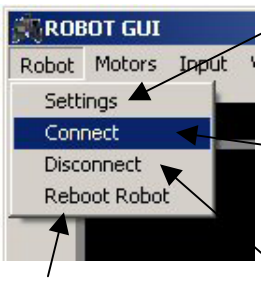
- Control the robot through our control software.
- Access to the robot's operating system to configure the robot.
- Transfer files between your computer and the robot's flash memory.

The control software:

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

1. Install the **Video Decoder** present in the same folder.
2. Launch the **WifibotGUI** program.
3. Click on **Robot** then **Settings**. The **Robot Settings** window appears.
4. Set the **Control Server IP** and the **Control Server Port** which by default is **15000**.
5. Set the **Camera IP** and the **Camera Port** which for the image is by default **80**.
6. Click **OK** when finished.
7. Click on **Video**, then select the proper camera **ON**. The image from the camera will appear.
8. Click on **Robot** then **Connect.**
9. Click on **Input** then select **Joystick** or **Virtual_joy**. The robot can now be operated.

The menu options:

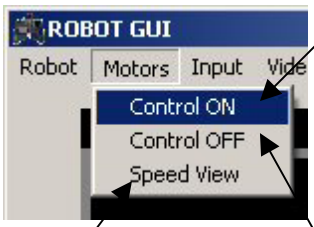


Settings: IP settings of the Control Server and the Camera.

Connect: Starts the communication with the Control Server.

Disconnect: Stops the communication with the Control Server.

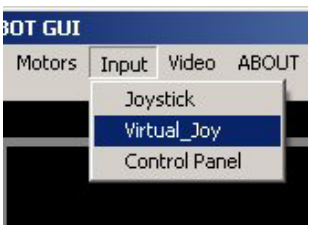
Reboot: Reboots the robot's CPU.



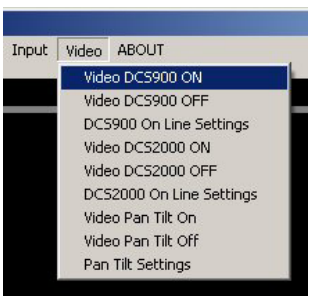
Motor Control ON: Activates the speed control, Input_Left and Input_Right set on the dialog will be applied.

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

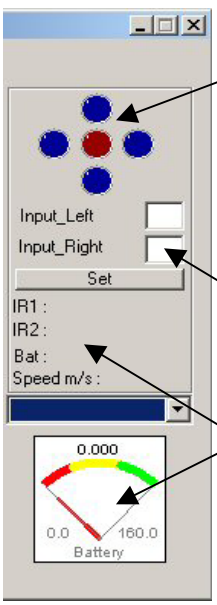
Motor Control OFF: Deactivates the speed control.



Input Selections (control panel for calibrating the joystick)



Video Source selections: Choose your camera from the selection.



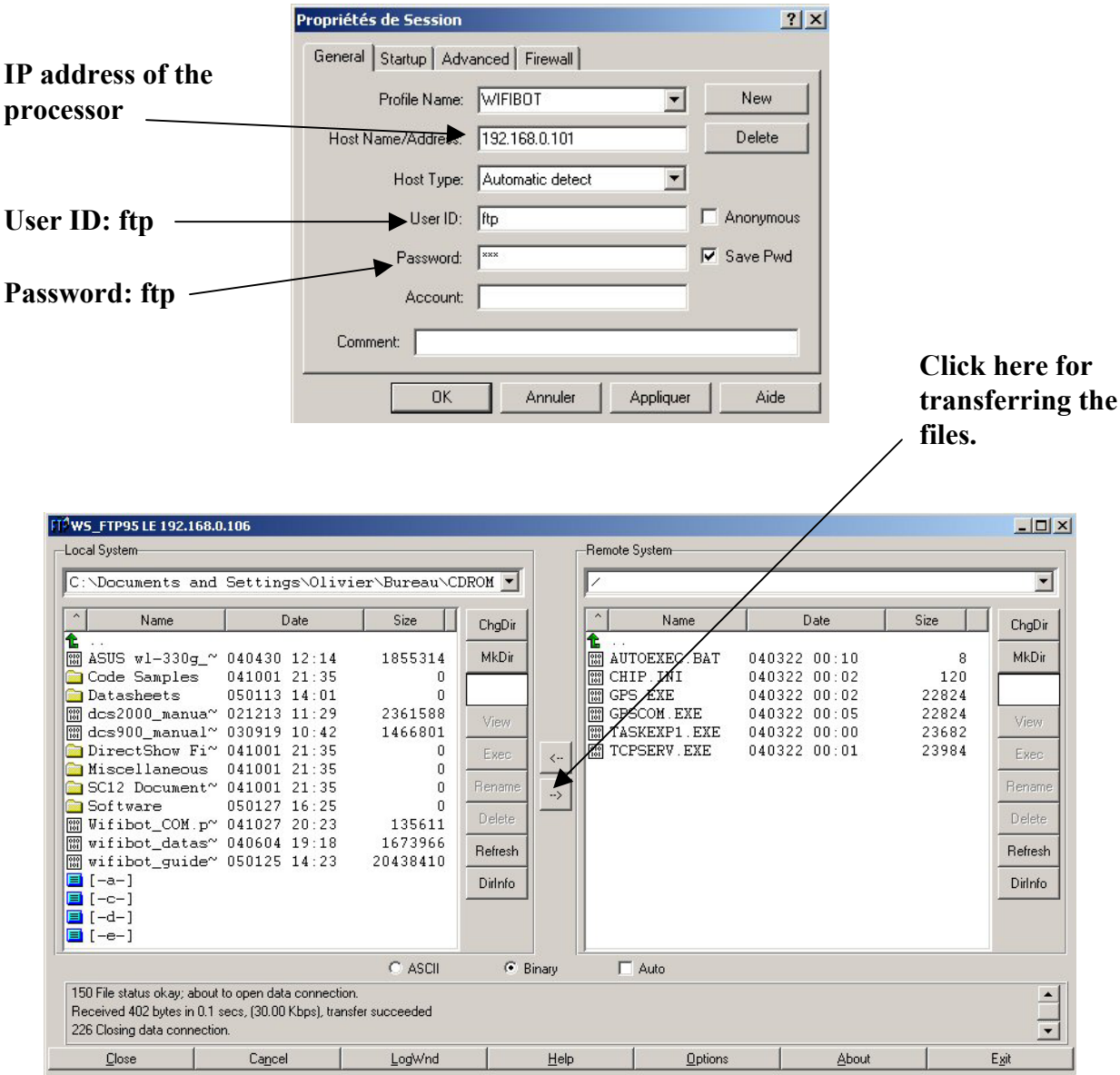
Pan-Tilt camera control: The red button takes the camera to the default position. You can click on the image too for moving the camera.

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

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

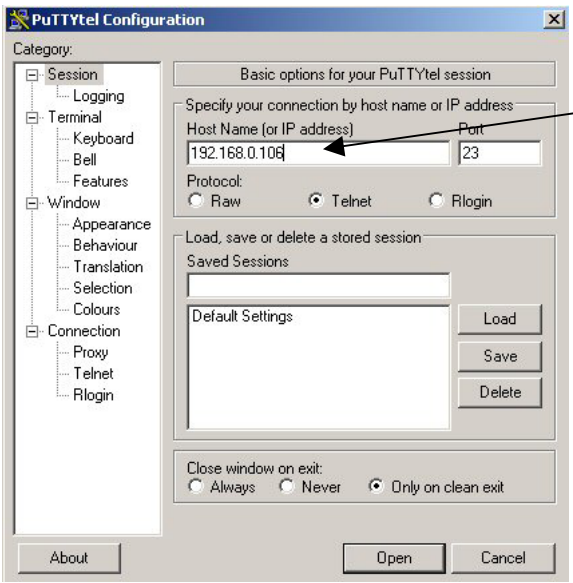
File transfer by FTP:

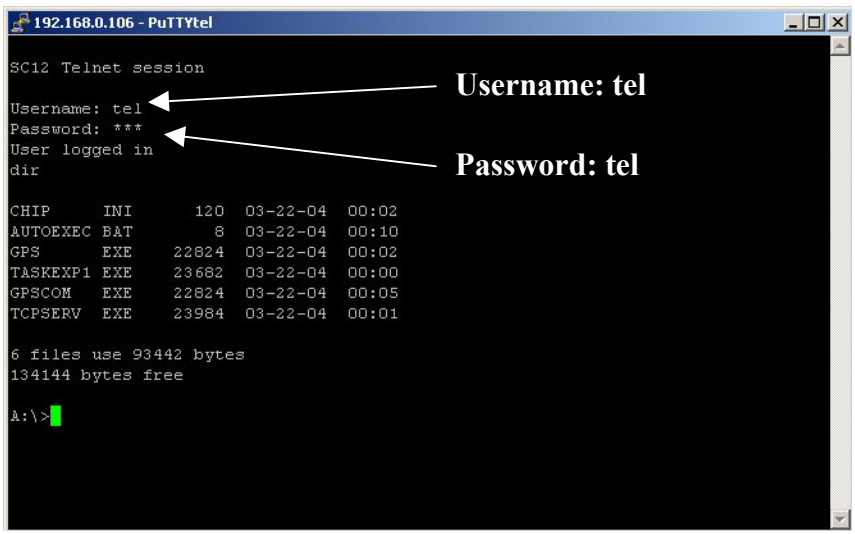
To transfer your files and programs to be embedded in the robot the simplest way is to use its FTP server. You can find a free FTP client in the companion CD **..\software\ftp**. The only parameters you will need to enter for connecting are the controller's IP address: **192.168.0.1xx**, the user ID: **ftp** and the password: **ftp**.



Remote access and Telnet:

Once your files have been transferred, if you want to start a program on the SC12 from a remote computer you will have to make use of its embedded Telnet server. For your convenience a Telnet client has been included in the companion CD at **..\software\telnet** . The only parameters you will need to enter for connecting are the controller's IP address: **192.168.0.1xx**, the user ID: **tel** and the password: **tel**. You can now launch your embedded software application.





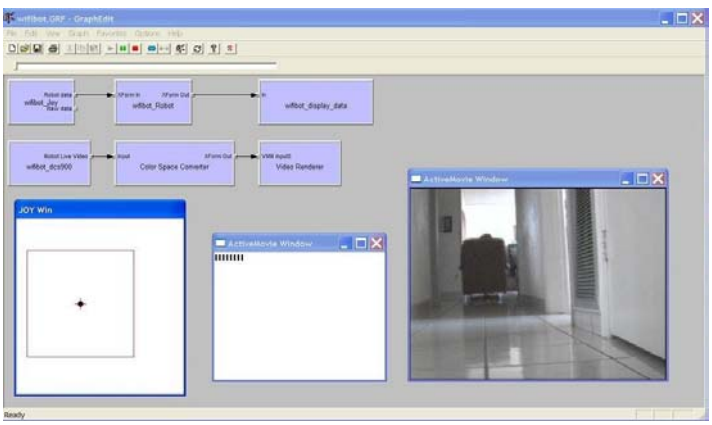
Robot programming

WiFiBoT SC has been designed in order to make it an open robotic tool that could be used for experimentation in the computer science and electronic fields. There are three main possible ways to work with a WiFiBoT SC:

Users interested in programming embedded applications can program the SC12 with the Borland C++ 5.0 16 bit compiler and give the robot autonomous behaviors. Interface functions in C are given for setting the speed and for retrieving the signal from the code wheels, the battery level and the range value from the sensors (**wifibot SC low level control.pdf** and **..\code samples\sc12 wifibot server\wifibot.c**). The CD-ROM includes programming examples of the different communication interfaces of the SC12 controller (bus I²C, RS232, TCP/IP) as well as the original documentation and examples of the manufacturer.

Another way to see the robot is as just a network peripheral. A detailed description of the communication protocol with the embedded server of the robot is included in the CD-ROM (**wifibot-pc protocol.pdf**) so the user can communicate with the robot via TCP/IP without having to even enter into the programming of the embedded calculator. Commands can be sent and sensor data retrieved from one or more robots to a remote computer and in this way realize complicated processes and calculations impossible to do on the limited SC12.

To finish, we have developed an interface based on DirectShow filters that has the advantage to be modular and used in the form of graphs inside a graphical interface (graphedit) or integrated within the user software as COM objects. The user can further program his own filters for controlling the robot or for processing the video feedback etc. using the free SDK from Microsoft and then interconnect those with the ones of the robot.



The CD-ROM

The companion CD includes documentation, programs and programming examples. The root folder contains the most important manuals and documentation for configuring the robot. For the rest the CD is structured in six folders:

SC12 documentation:

This contains the documentation and the programming examples that can be found on the website of BECK www.beck-ipc.com

Software:

This folder contains the robot control software, graphedit from DirectShow, the embedded server as well as an FTP and a Telnet client.

Code Samples:

Here you can find code samples for both the programming of embedded applications and external ones on a remote computer. The projects for the SC12 were realized with a Borland C++ 5.0 compiler while for the examples under windows Visual Studio 6.0 was used.

DirectShow filters:

This contains the different filters that form the robot interface under DirectShow. The installation program installs and declares the filters automatically. The filters can be used in graphedit or in your own software application.

Datasheets:

Here you will find the datasheets of some of the components of the robot.

Miscellaneous:

Different documentation that can be useful when making mechatronic projects.