HumaRobotics
making your personal robot smarter

Wifi Block

User documentation
Version 1.0

1 Introduction to the WifiBlock

The WifiBlock for Lego Mindstorms NXT is a standalone enclosure containing an electronic board with a Wifi device. Useful features are:

1. 2 leds (one green and one orange)
2. A NXT compatible female connector
3. An ON/OFF switch

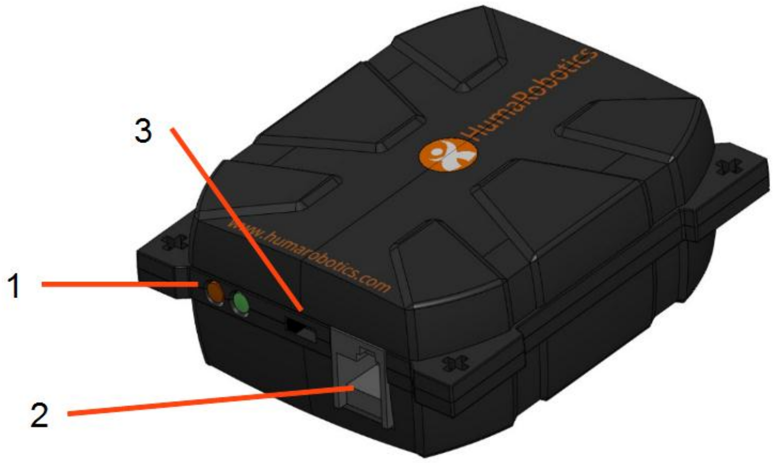


Figure 1: Overview of the WifiBlock

The WifiBlock is powered by 3 AA (LR6) batteries (not supplied) that must be inserted into the WifiBlock as shown on the battery hatch marking:

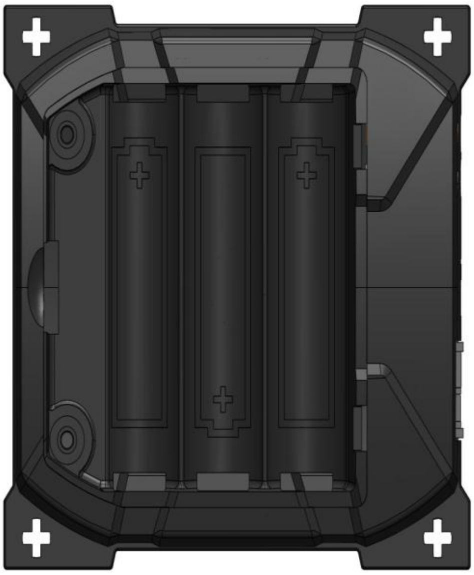


Figure 2: Battery hatch

The WifiBlock connects to your Lego Mindstorms NXT brick enabling it to have a complete TCP/IP connectivity with an access to the internet. With such equipment, your NXT robot will be able to send data to a computer over the internet or receive information from this computer.

Let’s introduce some use case for the WifiBlock.

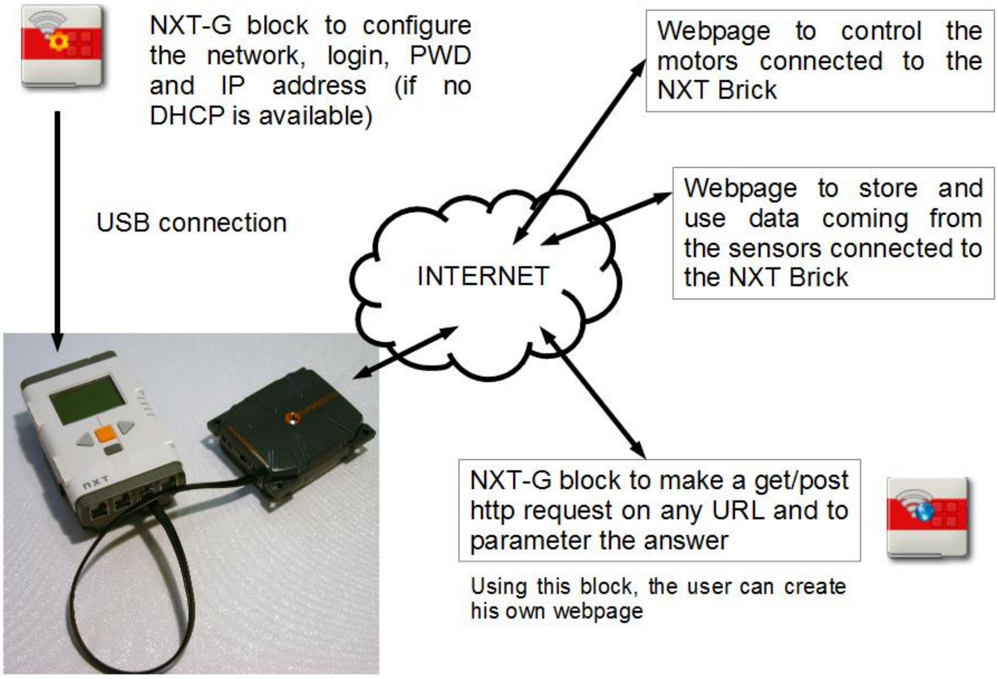

NXT-G block to configure the network, login, PWD and IP address (if no DHCP is available)
USB connection
INTERNET
Webpage to control the motors connected to the NXT Brick
Webpage to store and use data coming from the sensors connected to the NXT Brick
NXT-G block to make a get/post http request on any URL and to parameter the answer
Using this block, the user can create his own webpage

Figure 3: Use case for the WifiBlock

1.1 Data logging

The « data logging » is a function consisting in saving on a file (for example) a dataset of measures made by a system. Once saved, these measurements are available for further analysis or visualization. In the case of the Lego robot, it can be sensor measurements, for example ultrasounds telemeters data.

Data logging can of course be made directly on the Lego Brick writing a local file that will be stored in the memory. The main advantage brought by the WifiBlock is the ability to send data almost in real time to a computer that can use much more memory and computational power. A graphical visualization or more complex analysis could then be performed by the computer using the data received from the Lego robot, for example to build a representation of the robot environment.

1.2 Remote control

In this case, the computer sends information to the Lego Brick using the WifiBlock connection. It is possible to design a control interface on a computer and to send control commands to the robots over the wireless network.

1.3 A Robot-PC connection

As seen in the two previous examples, the WifiBlock creates a true bridge between the robot and connected computer, enabling unprecedented capabilities for the Lego Mindstorms NXT robot, like long distance remote control with sensor return.

2 Comparison between Wifi and other wireless communication technologies

2.1 Bluetooth

By default, the Lego Mindstorms NXT smart brick offers a wireless connection. It is the Bluetooth technology. The Wifi has several critical advantages compared to the Bluetooth:

- The bandwidth: Wifi allows to transmit more data, up to 11Mbps (theoretical max), to be compared with the 460 kbps maximum rate for NXT bluetooth (http://www.tau.ac.il/~stoledo/lego/btperformance.html)
- The range : The Bluetooth present on the NXT has a 10m range only. The Wifi range is much better (up to 50 or 100m according to the norms and the surrounding environment) and in addition, if we consider that the WifiBlock equipped robot can communicate over the internet, this range is no longer a limitation.

2.2 XBee

Xbee is a radio protocol with a bandwidth that is better than the Bluetooth but is still out of reach of the Wifi.

Even if the Xbee range is more important, it only remains a point to point connection with no TCP/IP connectivity and the lack of internet connection.

To learn more on the Xbee modules for Lego Mindstorms NXT, please refer to : http://www.generationrobots.com/wireless-communication-module-nxtbee-pro-dexter-industries-mindstorms-nxt,us,4,NXTBeePRO.cfm

3 Connecting and configuring the WifiBlock

3.1 First step: connect the WifiBlock

Make sure that the 3 batteries are correctly inserted in the WifiBlock.
Connect the WifiBlock using a standard Lego Mindstorms cable to one of the sensors ports of the Mindstorms. These ports are labeled from 1 to 4.

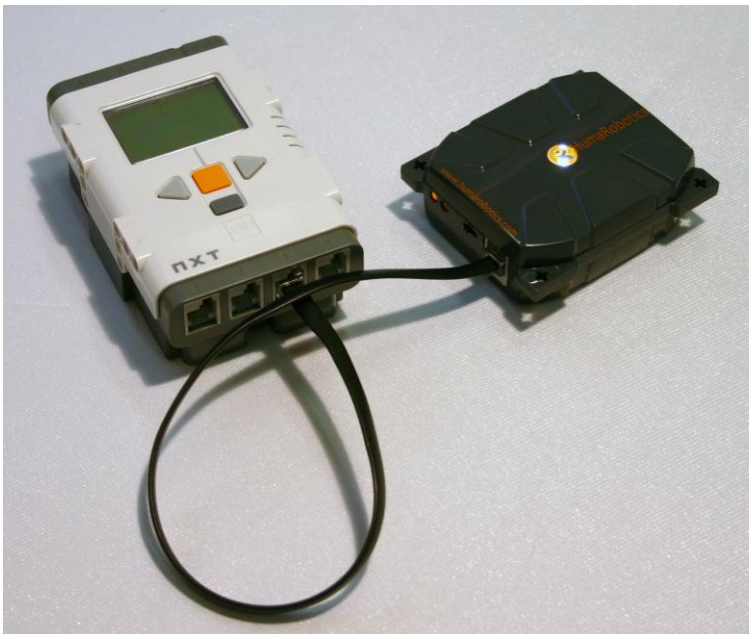


Figure 4: Connection between the Lego Mindstorm Brick and the WifiBlock

***Warning** : Never connect the WifiBlock to one of the motor ports (A, B and C ports). The voltage on these connectors is not suitable and could do irreversible damages your WifiBlock.*

3.2 Second step: Find out your network parameters

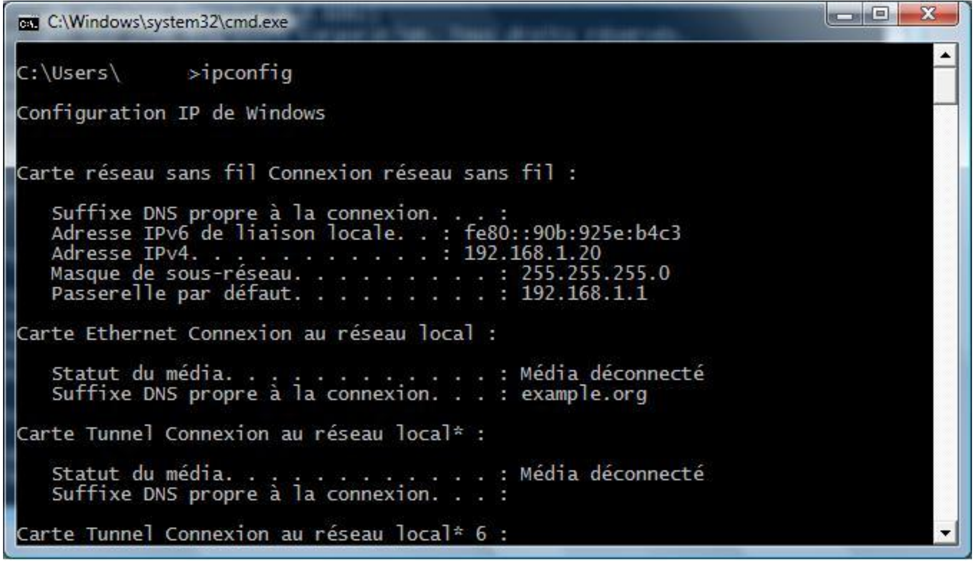
The WifiBlock doesn't offer a DHCP feature that would allow it to automatically obtain an IP address from your wireless router. Therefore, you will have to configure your WifiBlock to allow it to communicate with your Wifi network by providing the following information:

- An IP address for the WifiBlock
- A network mask
- A default gateway
- The login for your wifi
- The password for your wifi.

The following paragraphs will explain you step by step how to find these information.

3.2.1 Windows

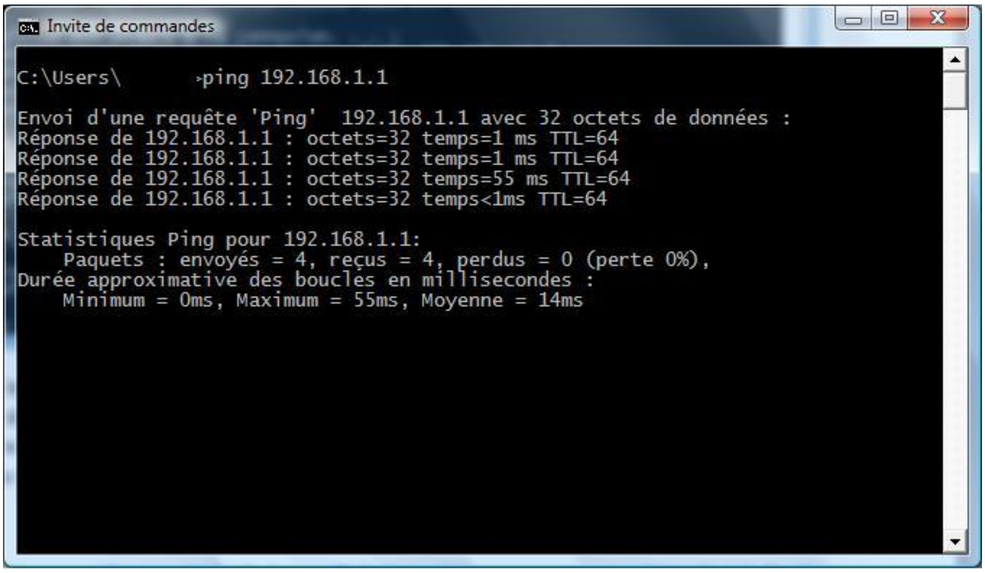
Launch a command prompt by entering « cmd » in the program panel of your computer (windows button + R to get the program panel).
Enter the “**ipconfig”** command. This command displays the network configuration of your computer and the parameters of your network as shown in the example below:



```
C:\Users\      >ipconfig

Configuration IP de Windows


Carte réseau sans fil Connexion réseau sans fil :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . : fe80::90b:925e:b4c3
   Adresse IPv4. . . . . . . . . . . : 192.168.1.20
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . : 192.168.1.1

Carte Ethernet Connexion au réseau local :

   Statut du média. . . . . . . . . . . . : Média déconnecté
   Suffixe DNS propre à la connexion. . . : example.org

Carte Tunnel Connexion au réseau local* :

   Statut du média. . . . . . . . . . . . : Média déconnecté
   Suffixe DNS propre à la connexion. . . :

Carte Tunnel Connexion au réseau local* 6 :
```

Figure 5: Ipconfig command result

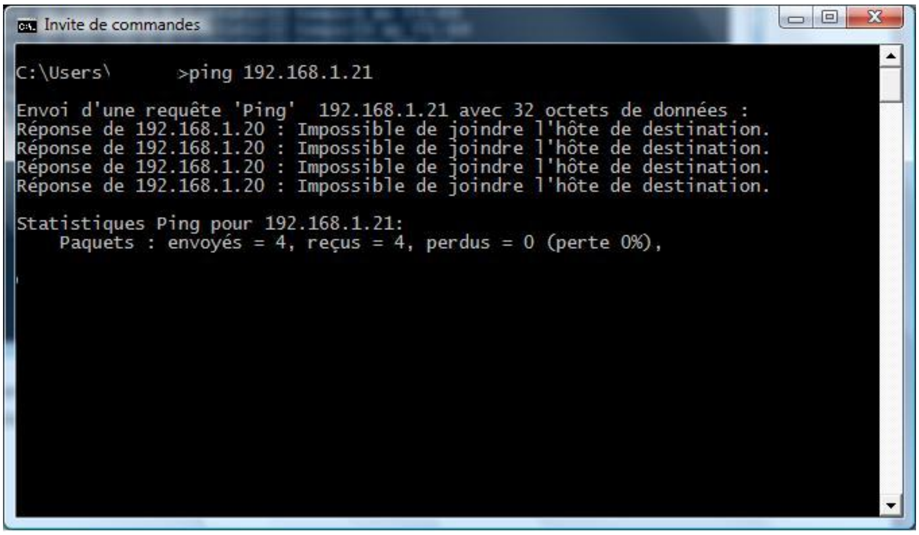
Find the wireless connection settings and write down your default **gateway** (in our example 192.168.1.1) as well as the value for your **network mask** (255.255.255.0 in the screenshot above). The **IP address** of the computer in our example is 192.168.1.20.
You now need to find a free IP address that will be attributed to the WifiBlock. Build a new IP address by taking your computer's IP and replacing the last number by another number between 1 and 255. Then type “**ping** ” and this IP in your command prompt. If the “**ping**” command gives an answer (Ctrl + C to stop the command), this IP address is already in use, so choose another one.



```
C:\Users\      >ping 192.168.1.1

Envoi d'une requête 'Ping'  192.168.1.1 avec 32 octets de données :
Réponse de 192.168.1.1 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps=55 ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.1.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 55ms, Moyenne = 14ms
```

Figure 6: The ping command indicate the presence of a device at the looked up address

If you get a “unable to reach the destination host” answer, the IP address is free, so write it down for your WifiBlock.



```
C:\Users\       >ping 192.168.1.21

Envoi d'une requête 'Ping'  192.168.1.21 avec 32 octets de données :
Réponse de 192.168.1.20 : Impossible de joindre l'hôte de destination.
Réponse de 192.168.1.20 : Impossible de joindre l'hôte de destination.
Réponse de 192.168.1.20 : Impossible de joindre l'hôte de destination.
Réponse de 192.168.1.20 : Impossible de joindre l'hôte de destination.

Statistiques Ping pour 192.168.1.21:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
```

Figure 7: the result of a « ping » command showing that no device replies on a given IP address.

3.2.2 Linux

Launch a command prompt and run the “**ifconfig”** command. This command displays the network configuration of your computer and the parameters of your network. Find the wireless connection settings and write down your **default gateway**, the value for your **network mask** and the **IP address** of the computer.

You now need to find a free IP address that will be attributed to the WifiBlock. Build a new IP address by taking your computer's IP and replacing the last number by another number between 1 and 255. Then type “**ping** ” and this IP in your command prompt. If the “**ping**” command gives an answer (Ctrl + C to stop the command), this IP address is already in use, so choose another one.
If you get a “unable to reach the destination host” answer, the IP address is free, so write it down for your WifiBlock.

3.2.3 Mac

Launch a terminal, located in /Applications/Utilities/, then type the “**ifconfig**” command. IP addresses are on lines that start by 'inet' (to get only these lines, you can type “**ifconfig | grep inet**”). Find the line that looks like:

```
inet 192.168.1.20 netmask 0xffffff00 broadcast 192.168.1.1
```

The IP address will be between “inet” and “netmask”.
Warning: there will always be a line with the 127.0.0.1 IP, which is your machine loopback address, that you should not take into account for wifi connection.

Write down your computer's **IP address** (in our example 192.168.1.20), your default **gateway** (in our example 192.168.1.1) and the value for your **network mask** (0xffffff00, equivalent to

255.255.255.0 if you divide the hexadecimal number by groups of two digits and convert them to decimal numbers).

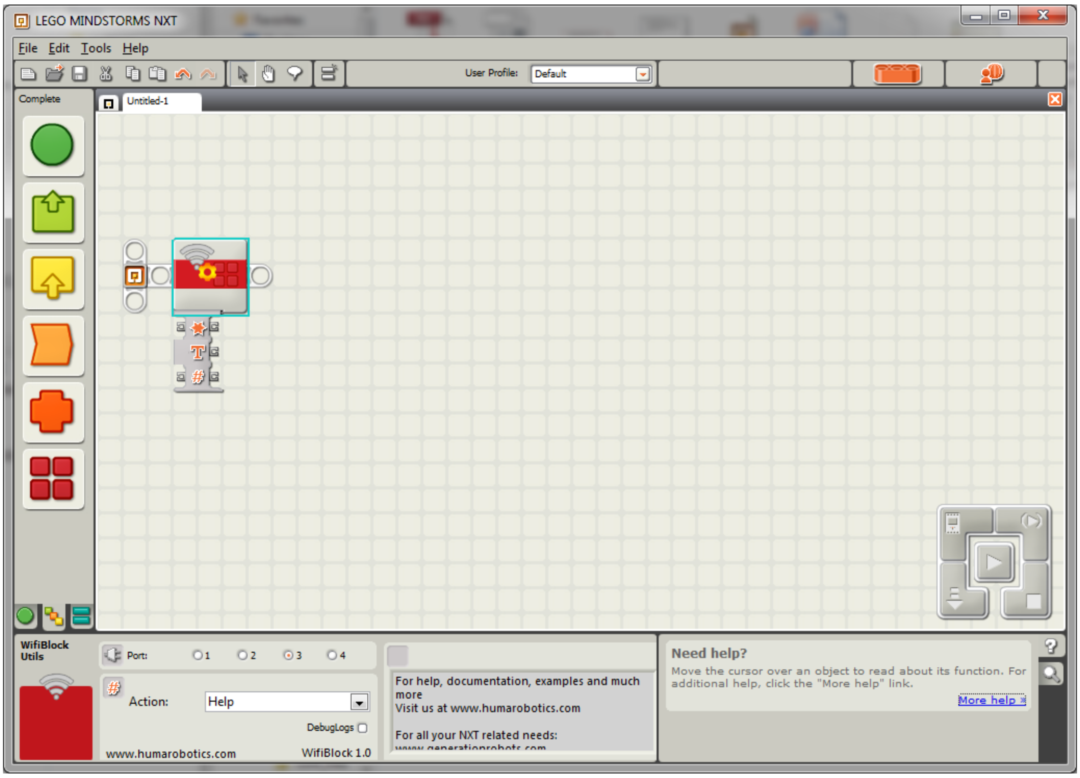
You now need to find a free IP address that will be attributed to the WifiBlock. Build a new IP address by taking your computer's IP and replacing the last number by another number between 1 and 255. Then type "**ping** " and this IP in your command prompt. If the "**ping**" command gives an answer, this IP address is already in use, so choose another one.
If you get a "unable to reach the destination host" answer, the IP address is free, so write it down for your WifiBlock.

3.3 Third step : Configuring the WifiBlock

Now that you have determined your network parameters, you have to provide to the WifiBlock the information in order to allow it to connect to your network.

Launch NXT-G, the programming environment for Lego Mindstorms. To configure the WifiBlock and allow it to connect to the web, we are going to use the block called **WifiBlockUtils**.

You can find this block, as well as the "WifiBlock Communication" and the "WifiBlock Query" ones that we will use later on the Humarobotics website (here). You will need to download them and to import them into NXT-G.
Please refer to the appropriate appendix of this document or to the NXT-G user manual in order to know how to add third-party blocks to NXT-G.



Port: ○1 ○2 ●3 ○4

Action: Help

DebugLogs □

www.humarobotics.com WifiBlock 1.0

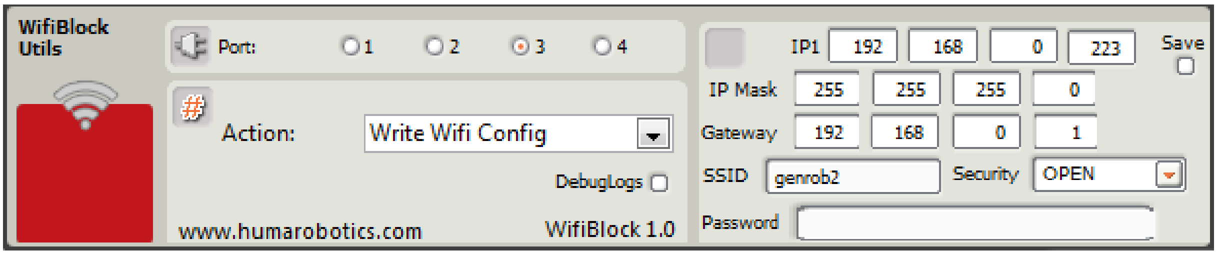
For help, documentation, examples and much more
Visit us at www.humarobotics.com

For all your NXT related needs:
www.generationrobots.com

Need help?
Move the cursor over an object to read about its function. For additional help, click the "More help" link.
More help »

Figure 8: The WifiBlockUtils block in NXT-G

This block allows writing the IP configuration of the WifiBlock, used to make a connection to your network. To enter the parameters choose “Write Wifi Config” from the “Action” dropdown menu. You’ll then see the panel displaying various parameters that have to be fill out.



WifiBlock Utils

Port: ○ 1 ○ 2 ● 3 ○ 4

Action: Write Wifi Config

DebugLogs □

www.humarobotics.com WifiBlock 1.0

IP1 192 168 0 223

Save □

IP Mask 255 255 255 0

Gateway 192 168 0 1

SSID genrob2 Security OPEN

Password

Figure 9: Configuration panel for the WifiBlock

The data to be filled out are described in the following table:

Data	Description
Port	Number of the Port on the NXT where the WifiBlock is plugged. Make sure that you have set up this to the correct value as it is a very common mistake.
Action	Drop down menu allowing to choose the action that will be performed by the WifiBlock. In our case, we have selected « Write Wifi Config » as we want to provide connection information to the WifiBlock.
IP	This is the IP address that we want to assign to the WifiBlock. This address has been determined in the previous step using the ping command.
IP Mask	This is the addressing mask for your IP address. This is the one that we discovered using the ipconfig or ifconfig command. The value of this mask depends on your network configuration.
Gateway	This is the IP address of your network router (in the case of a personal network, it is often the IP address of your modem or Wifi router). This address has also been discovered using the ipconfig or ifconfig command.
SSID	This is the name of the wifi network you are going to connect to. This is the name that you see when hovering the Wifi Icon on your computer (if of course it is connected to a Wifi network).
Security	This is the encryption and authentication protocol for your Wifi network. **OPEN** : No Security and no encryption. Your network is entirely open and anyone or any device can connect to it. **WEP** : fixed key encryption. It is a basic setting for many networks. **WPA** : More secure than the WEP security setting. **WPA 2** : 2nd version of the WPA setting, offering advanced security. Please make sure to select the right security mode to connect the WifiBlock. Any error at this stage will result in a connection error as the various encryption modes are not compatible.
Password	This is the password (or passphrase) of your Wifi network. Please refer to the documentation provided with your Wifi router or modem. Here again, make sure that you have entered the right password.

Table 1: IP configuration data fields

Enter the parameters you determined during step two.

If you check the “Save” box, your Wifi configuration will be saved and will remain until you choose to modify it. Therefore, if you shut your WifiBlock down and then up, it will try to connect directly to the Wifi using the previous configuration.

Once all the parameters are entered, make sure that the NXT brick is switched on and connected to your computer using the USB cable and that the WifiBlock is also switched on and plugged to the correct sensor port. Then click on the “play” button of NXT-G in order to launch the compilation of the program and the transmission to your NXT brick.

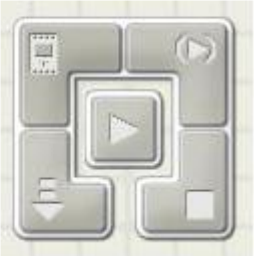


Figure 10: "Download and run" button at the center of the command panel of NXT-G

In order to check with your configuration has been sent to the WifiBlock and if it has successfully connected to your network, you have to look at the LEDs placed on the side of the block.

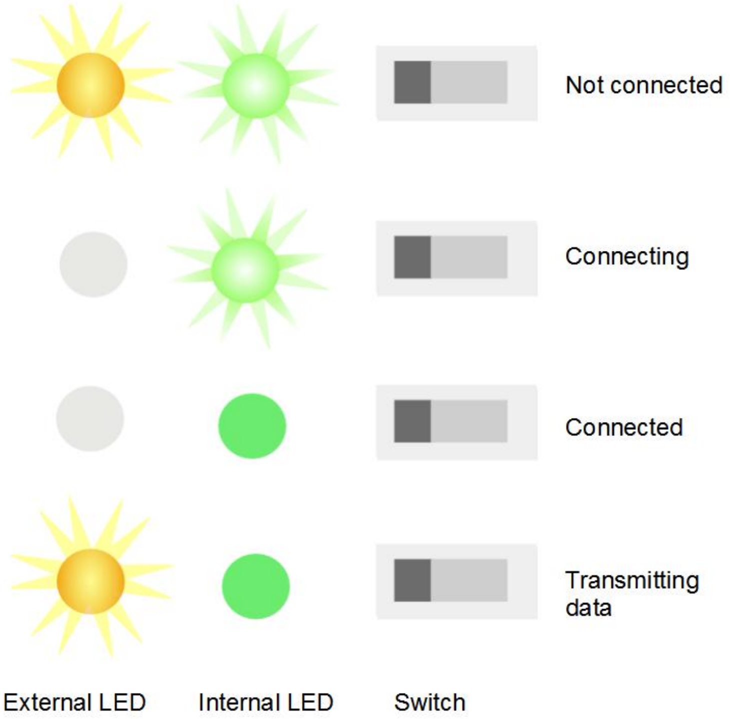


Figure 11: Visualization of the connection state with LEDs

When you switch your WifiBlock on, the two LEDs blink alternatively, showing that the device is not connected. Once you inject the IP configuration, the orange LED shuts down while the green one still blinks, showing a connection attempt.

Once connected, the green LED stays on stable. The orange LED will blink if the WifiBlock receives or emits information. To see this, try to ping your WikiBlock using its IP address in you computer's command prompt and you will see the orange LED blink.

Connection time depends on your Wifi encryption. When there is no security (OPEN network), connection is quick. On the contrary, if you have a strongly secured network (like WPA2), connection time can increase up to 30 seconds because the calculation of the encrypted key takes more resources and time.

3.4 Advanced functionality of the WifiBlockUtils block

The “WifiBlockUtils” block offers other possibilities than writing the Wifi configuration on the WifiBlock.

The 'DebugLogs' box allows you to ask the NXT Brick to store logs in a file, that you can retrieve for debug purpose (see Troubleshooting section).

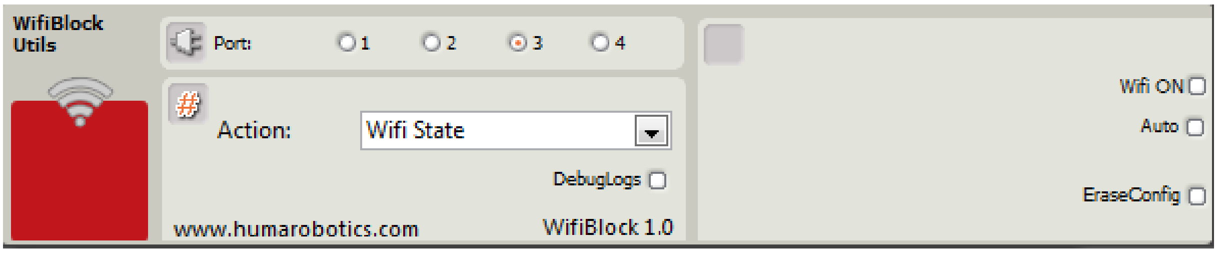
3.4.1 *The “Wifi State” action*

This action allows you to do some actions on the WifiBlock once it is configured.

Actions	Description
Wifi ON	Enable Wifi (checked: wifi enabled)
Auto	Auto reconnection if connection lost (checked: auto reconnection activated)
EraseConfig	Erase the saved Wifi configuration (checked: config will be erased)

Table 2: "Wifi state" actions

Remark: You can do these actions when writing the Wifi configuration.



WifiBlock Utils

Port: ○ 1 ○ 2 ● 3 ○ 4

Action: Wifi State

DebugLogs □

www.humarobotics.com

WifiBlock 1.0

Wifi ON □

Auto □

EraseConfig □

Figure 12: The “Wifi State” configuration panel

3.4.2 *“Write Wifi Config” with dynamic parameters*

In the previous example, we hard coded the Wifi configuration. But if you want some setting to depend from what happened previously in your program, you can dynamically change the parameters using the block's data hub.

To open the data hub of a block, click on its lower edge. You get access to the following settings:

Data	Data type	Description
Action	*Number*	1: help 2: Write Wifi Config 3: Wifi State 4: Read WifiBlock
Value to Read	*Number*	Value to be read in the WifiBlock registers, see next paragraph
Security	*Number*	0: Open 1: WEP 2: WPA 3: WPA2
SSID	*Text*	The Wifi network name
Password	*Text*	The Wifi network password
IP1, IP2, IP3, IP4	*Numbers*	IP address, divided into 4 numbers
IP mask, MSK2, MSK3, MSK4	*Numbers*	IP mask, divided into 4 numbers
Gateway, GW2, GW3, GW4	*Numbers*	Gateway, divided into 4 numbers
Wifi On	*Boolean*	Wifi enabled
Auto Reconnect	*Boolean*	Auto reconnection enabled
Save	*Boolean*	The Wifi configuration will be saved
EraseConfig	*Boolean*	Wifi configuration will be erased
DebugLogs	*Boolean*	Debug logs enabled

Table 3: Data accessible by the "WifiBlockUtils" data hub

3.4.3 *The “Read WifiBlock” action*

When you choose the “ReadWifiBlock” action, you have read access to all values saved in the WifiBlock. Select one using the “Value to read” dropdown menu. The data will be accessible in the “Buffer” output of the data hub, with a *Text* data type.

The following table describes the accessible data. The numbers correspond to the numbers you have to enter in the “Value to read” field of the data hub (see previous paragraph):

Data		Description
Read Version	1	Returns the firmware version
Read Product ID	2	Returns the product ID, should be HRWB
Read Sensor Type	3	Should return “WIFI”. Generic NXT-G function.
Read Voltage	4	Returns **10 times** the battery voltage. Don't forget to divide this value by 10 to get the real voltage.
Read Mac	5	Returns the MAC address of the wifi device (works only if the WifiBlock is connected to a network)
Read IP	6	Returns the configured IP address
Read IP Mask	7	Returns the IP mask of the Wifi network
Read Gateway	8	Returns the gateway of the Wifi network
Read SSID	9	Returns the SSID of the Wifi network
Read Security	10	Returns the Wifi safety type (OPEN, WEP, WPA, WPA2)
Read Passphrase	11	Returns the password of the Wifi configuration
Read Get Result	12	Returns the result of the latest GET request
Read Date	13	Gives the date
Read Get Status	14	Returns the GET status Bit[0] : Port ready (0: KO, 1: OK) Bit[1]: Reserved Bit[2]: Send request (1: send) Bit[3]: Request done (1: done) Bit[6..4]: Reserved Bit[7]: Request error (1: error)
Read Wifi Block Status	15	Returns the Wifi status Bit[0] : Enable Wifi (0: dis, 1: en) Bit[1]: Wifi save config (1: save) Bit[2]: Start scan (1: start) Bit[3]: Scan done (0: busy, 1: done) Bit[4]: Wifi status (1: connected) Bit[5]: Disable Rs232 debug (1: no Rs232) Bit[6]: Re-connecton (0: dis, 1: en) Bit[7]: Wifi erase config (1: erase)
Read HTTP response	16	Returns the Http last response code (3 bytes)
Read WifiRetryNb	17	Returns the number of Wifi connection retry
Read Get Length	18	Return the length of the last data received from a GET request
Read Get IP	19	Returns the IP address of the last GET request
Read Get Port	20	Returns the port of the last GET request

Data		Description
Read Get URL	21	Returns the URL of the last GET request (the part after IP address, that you enter in the “data” field, see the GET request paragraph)
Read Get Data	22	Returns the last data received from a GET request
Read Post Status	23	Returns the POST status Bit[0] : Port ready (0: KO, 1: OK) Bit[1]: Reserved Bit[2]: Send request (1: send) Bit[3]: Request done (1: done) Bit[4]: Enable Dataflash Storage Bit[5]: Enable Dataflash Retrans Bit[6]: Reserved Bit[7]: Request error (1: error)
Read Post Length	24	Return the length of the last data transmitted by a POST request
Read Post IP	25	Returns the IP address of the last POST request
Read Post Port	26	Returns the port of the last POST request
Read Post URL	27	Returns the URL of the last POST request (the part after IP address, that you enter in the “data” field, see the GET request paragraph)
Read Post Data	28	Returns the last data transmitted by a POST request
Read UDP Status	29	Returns the UDP status Bit[0] : Port enable (0: dis, 1: en) Bit[1]: Send Tx data (1: send) Bit[2]: Send Tx done (1: done) Bit[3]: Rx data ready Bit[4]: Rx data read Bit[6..5]: Reserved Bit[7]: Send error (1: error)
Read UDP dest IP	30	Returns IP address for UDP
Read UDP dest port	31	Returns destination port for UDP
Read UDP src port	32	Returns source port for UDP
Read UDP Tx Data Length	33	Returns the length of the Tx data
Read UDP Tx Data	34	Returns the Tx data
Read UDP Rx Data Length	35	Returns the length of the Rx data
Read UDP Rx Data	36	Returns the Rx data
Read TCP status	37	Returns the TCP status Bit[0] : Port ready (0: KO, 1: OK) Bit[1]: Send request (1: send) Bit[2]: Request done (1: done) Bit[6..3]: Reserved Bit[7]: Send error (1: error)
Read TCP IP	38	Returns IP address for TCP
Read TCP port	39	Returns port for TCP
Read TCP Tx Data Length	40	Returns the length of the Tx data

Data		Description
Read TCP Tx Data	41	Returns the Tx data
Read TCP Rx Data Length	42	Returns the length of the Rx data
Read TCP Rx Data	43	Returns the Rx data
Read Dataflash Status	44	Returns the Dataflash status Bit[0]: Write data (1: write) Bit[1]: Write done (1: done) Bit[2]: Read data (1:read) Bit[3]: Read done (1:done) Bit[6..4]: Reserved Bit[7]: Dataflash error
Read Dataflash Addr	45	Return the Dataflash address

Table 4: Data accessible by the "Read Wifi Block" action

4 Sending requests using the Wifi Block

In this part, we describe how the WifiBlock asks and gets a piece of information on the web.

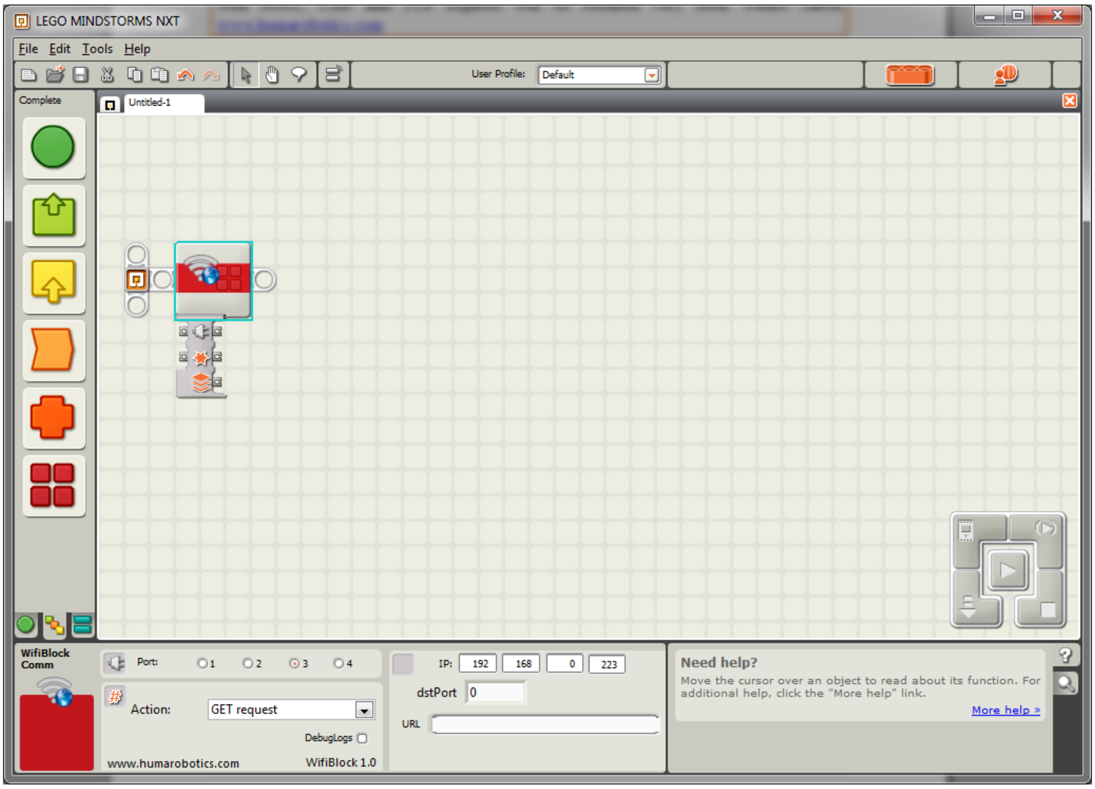
Remark: At the moment, only GET requests are available with NXT-G blocks, a new block with POST, UDP and TCP requests will be released very soon. Please check www.humarobotics.com

4.1 Sending a GET request using the Wifi Block

A GET request asks data from an internet resource using its URL.

4.1.1 Using the "WifiBlockCommunication" block to send a GET request

If you haven't done it already, import the "WifiBlockComm" NXT-G block and drag and drop it on a worksheet, like in the screenshot below.



Port: ○1 ○2 ●3 ○4

Action: GET request

DebugLogs □

www.humarobotics.com WifiBlock 1.0

IP: 192 168 0 223

dstPort 0

URL

Need help?

Move the cursor over an object to read about its function. For additional help, click the "More help" link.

More help »

Figure 12: The "WifiBlockCommunication" NXT-G block

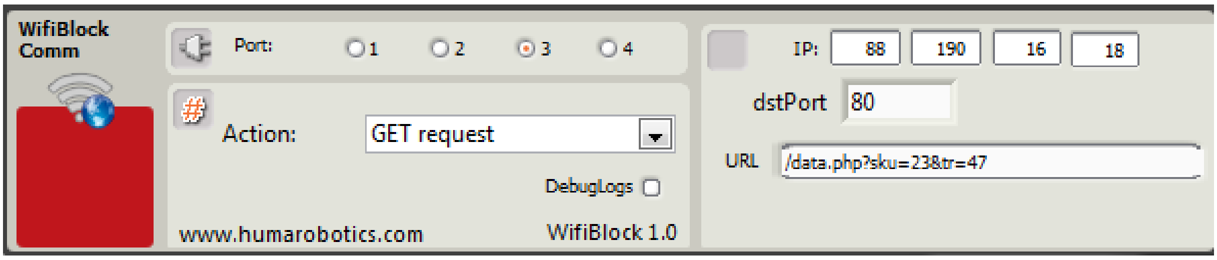
This block works in the same way as the "WifiBlockUtils" used before. The configuration panel allows you to choose the action to be done. Select "GET request" to make a GET http request on the internet.

Like for the previous block, it is important to configure correctly the connection port. The right part of the panel contains the following fields:

- IP: The IP web server on which you want to make the request
- dstPort: The destination port used for connection. The default port for internet communication is 80.
- URL: character chain that will be placed after IP address, containing html page (or any kind of extension) you want to call, including parameters.

You can check the 'DebugLogs' box so your WifiBlock will store logs in a file in the NXT Brick, for debug purpose.

For example, if you wish to call http://www.humarobotics.com/data.php?sku=23&tr=47 webpage, you divide the URL into the IP part, 88.190.16.18 for “www.humarobotics.com” and the additional data and parameters, here “/data.php?sku=23&tr=47”. Your configuration panel will look like below:



Port: ○ 1 ○ 2 ● 3 ○ 4

Action: GET request

DebugLogs □

IP: 88 190 16 18

dstPort 80

URL /data.php?sku=23&tr=47

Figure 13: Configuration panel of the “WifiBlockCommunication” to do a Get request on “http://www.humarobotics.com/data.php?sku=23&tr=47”

To know the IP address of a website, you can ping in from your command prompt (“ping www.humarobotics.com”).

You have to be sure that the server answers if called by its IP address, particularly in the case of virtual hosts. To test this, you can call the IP address in your web browser and see if you get an answer (for example “http://88.190.16.18” won't give you an error)

4.1.2 Displaying the result

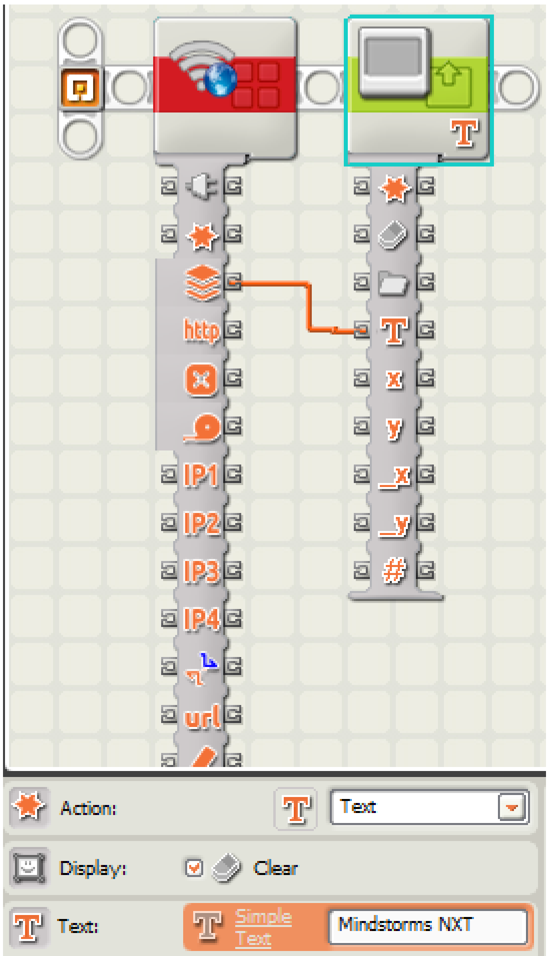
To visualize the return data from the GET request, you can use a ”Display” block, so the return data will be displayed on the NXT Brick's screen.

Drag and drop a “Display” block after the “WifiBlockCommunication” one. Then connect the “*Buffer*” output from the “WifiBlockCommunication” block to the “*Text*” input of the “Display” one, like in the figure 14.

You can also retrieve the http response code, using the HttpCode output (“*Text*” output). It should be 200 if the request was successful. Check this page to interpret other codes: http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

The two other outputs are “Error” (“*Boolean*” type), telling you if any error occurred and “DataLength” (“*Number*” type), containing the length in byte of the request's result.

In the configuration panel of the “Display” block, choose “Text” from the “Action” dropdown menu.



Action: Text

Display: ☑ Clear

Text: Simple Text Mindstorms NXT

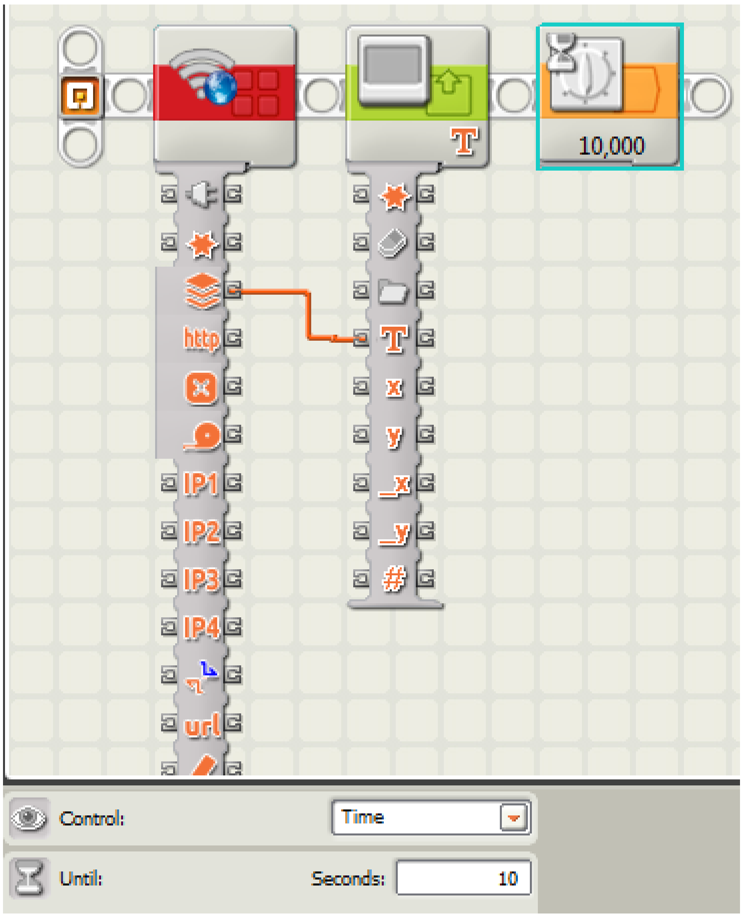
Figure 14: Data transmission between the “WifiBlockCommunication” block and the “Display” one

The problem with this is, you won't have time to see what is displayed on the brick screen because the program will stop right after displaying the result. This is a classic problem with NXT-G.

Therefore, you will have to add a “Wait” block after the “Display” one to have some time to visualize your GET request's result.

In the configuration panel of the “Wait” block, select “time” in the “Control” dropdown menu and set the number of seconds you want, for example 10, in the “Until” field.

Finally, you get this program:



Control: Time

Until: Seconds: 10

Figure 15: NXT-G code displaying on the brick screen the result of a http GET request

Download and run your program from the command panel.

> *Remark*: This block in interesting because it allows you to send data to a web server and get data in answer with only one request.

4.2 Sending a POST request using the Wifi Block

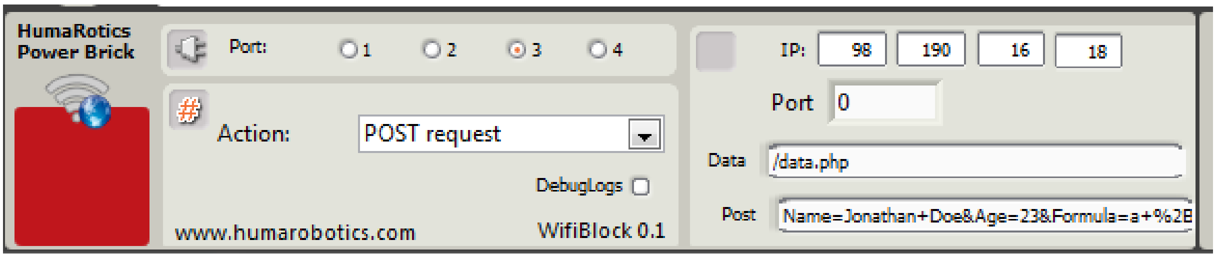
> *Remark*: At the moment, only GET requests are available with NXT-G blocks, a new block with POST, UDP and TCP requests will be released very soon. Please check www.humarobotics.com

A POST request allows you to send data to a server. You can do your request the same way as the GET request, but you have an additional field at your disposal, the 'Post' field.

In this field, you can enter all the data you want to transmit to the server, encoded in application/x-www-form-urlencoded type. You have several key-value pairs with possibly duplicate keys, the key being separated from the value by '='. The character '&' is used to separate key-values pairs and '+' for spaces, other non-alphanumeric characters being encoded with URL encoding.

See http://en.wikipedia.org/wiki/POST_(HTTP) for an example

The data in the 'Post' field will be included in the message body, where a GET request only sends a message header and a URL.



HumaRotics Power Brick

Port: ○ 1 ○ 2 ● 3 ○ 4

Action: POST request

DebugLogs □

www.humarobotics.com

WifiBlock 0.1

IP: 98 190 16 18

Port 0

Data /data.php

Post Name=Jonathan+Doe&Age=23&Formula=a+%2B

Figure 16: The “WifiBlockCommunication” block configured for a POST request

4.3 UDP and TCP data transmission and reception using the Wifi Block

4.3.1 Usages of UDP and TCP

The **User Data Protocol** (UDP) and the **Transmission Data Protocol** (TCP) are both core protocols of the Internet Protocol Suite, the set of network protocols used for the internet.

UDP is simpler and does not require *handshaking dialog* (packet exchange to initiate the transmission). Therefore it is quick but unreliable: data can arrive out of order, duplicated or be missing without notice.

TCP is a bit slower as it needs three packet exchange to before starting data transmission. But this protocol ensures a reliable, ordered delivery of the data. It is the most used protocol for internet applications.

4.3.2 UDP transmission and reception

Remark: At the moment, only GET requests are available with NXT-G blocks, a new block with POST, UDP and TCP requests will be released very soon. Please check www.humarobotics.com

4.3.3 TCP transmission and reception

Remark: At the moment, only GET requests are available with NXT-G blocks, a new block with POST, UDP and TCP requests will be released very soon. Please check www.humarobotics.com

4.4 Do parametrized requests

Like for the Wifi configuration, you can dynamically set the parameters for your requests by using the data hub. The available parameters are:

Data	Data type	Description
Port	*Number*	The NXT port where the WifiBlock is connected
Action	*Number*	1: GET request *(coming soon* *2: POST request* *3: UDP Tx* *4: UDP Rx* *5: TCP Tx* *6: TCP Rx)*
IP1	*Number*	First part of the IP address
IP2	*Number*	Second part of the IP address
IP3	*Number*	Third part of the IP address
IP4	*Number*	Fourth part of the IP address
dstPort	*Number*	The port used for connection
URL	*Text*	The additional data (html page with parameters...) that will be added after IP address
PostData	*Text*	The data to transmit when you use the POST request
DebugLogs	*Boolean*	The logs will be stored in a file on the brick, for debug purpose

Table 5: Description of the "WifiBlockCommunication" data hub fields

Like for any data hub in NXT-G, you can retrieve this data unchanged as output of the hub. There is only one additional output, the *Buffer* one, containing the result of the request.

5 Troobleshooting

In this part, we will show you some possibilities to debug your programs, understand what went wrong during communication and hopefully what to do to repair it.

First check that the port on witch you plugged the WifiBlock on the NXT Brick and the port you parametrized in your blocks are the same, because it is a very common mistake.

5.1 The WifiBlockQuery block

This block can be downloaded on the Humarobotics website and should be imported into your NXT-G software following the procedure described in appendix.

Using this block, you can ask a set of questions to your WifiBlock and receive Boolean (yes/no) responses. This allows you to check if your WifiBlock sends and receive information correctly and to precise where a problem can be.

The DebugLogs box allows you to ask the brick to store logs in a file for debug purpose.

The list of available questions is the following:

TO CHECK AND COMPLETE

Question		Description
Error on GET port ?	1	
Error on POST port ?	2	
Error on UDP port ?	3	
Error on TCP port ?	4	
Error on Dataflash port ?	5	
Wifi enabled ?	6	If 0, use a WifiBlockUtils block with “Wifi State” action to activate Wifi
Scan in progress ?	7	If 1, scanning for Wifi, please wait. Should no last more than ??
Wifi connected ?	8	If 0, check your Wifi config and your Wifi On parameter.
Re-connection enabled ?	9	
GET port ready ?	10	If 0 and no GET request error, transmission in progress
GET request done ?	11	
GET request error ?	12	If 1, could not find destination
POST port ready ?	13	If 0 and no POST request error, transmission in progress
POST request done ?	14	
POST request error ?	15	
Dataflash storage enabled ?	16	
Dataflash retransmit enabled ?	17	
UDP port enabled ?	18	

Question		Description
UDP Tx done ?	19	
UDP data ready ?	20	
UDP send error ?	21	
TCP port ready ?	22	
TCP request done ?	23	
TCP send error ?	24	
Dataflash write done ?	25	
Dataflash read done ?	26	
Dataflash error ?	27	

Table 6: Data accessible by the "Read Wifi Block" action

5.2 The debug logs

In all the blocks we presented here, there was a possibility to active the 'DebugLogs'. You can retrieve this logs by uploading the 'WifiBlockLogs.txt' file of your brick. You can find it in the “Other” section. (follow the procedure in the appendix).

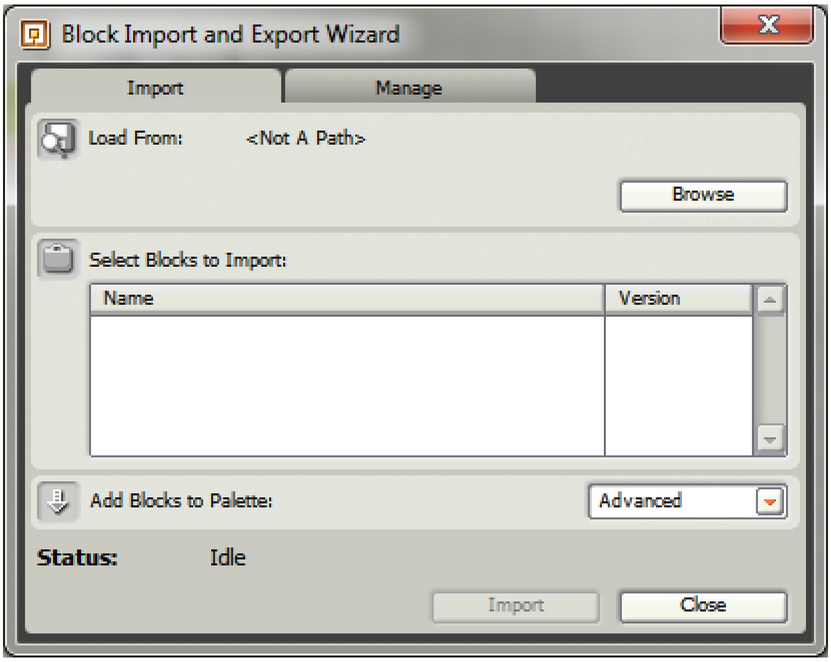
The information you can get by looking at the logs are TO COMPLETE

6 Appendix: import NXT-G blocks

To import new blocks to your block palette, follow these steps:

- In NXT-G, the graphical programming environment for Lego Mindstorms, open *“Tools/Block Import and Export Wizard”*

The following window appears:



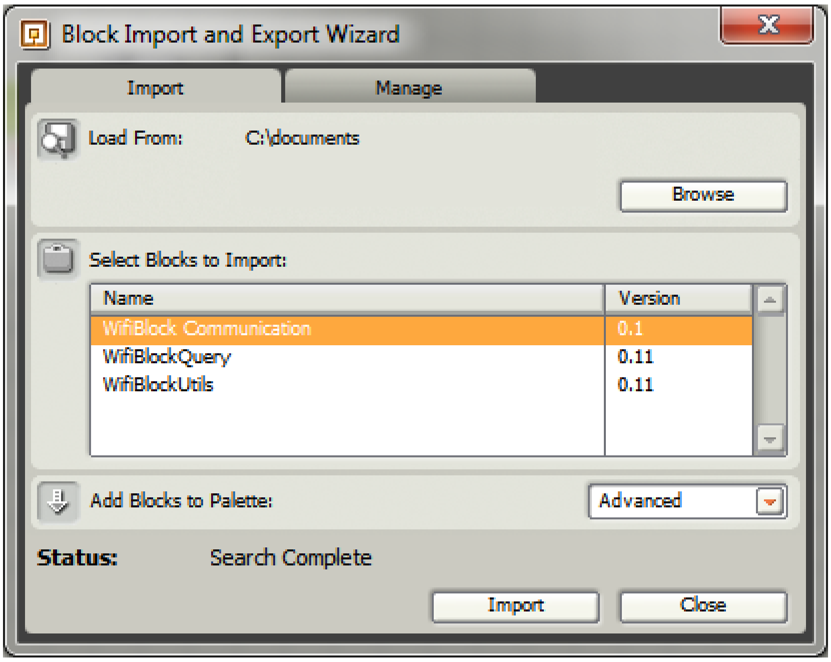
Load From: <Not A Path>

Add Blocks to Palette: Advanced

Status: Idle

Figure 17: Block import and export wizard

- With the 'Browse' button, select the folder where the source code of your block lies, i.e. where you unzipped the folders you downloaded from the Humarobotics website, containing the blocks presented in this documentation.
- The name of the found block is displayed in the list of blocks to import.
- Select in the “Add Blocks to Palette” dropdown menu in which palette you want to import your blocks.
- The 'Import' button become clickable only when you select the block to import from the blocks list.



Load From: C:\documents

Select Blocks to Import:

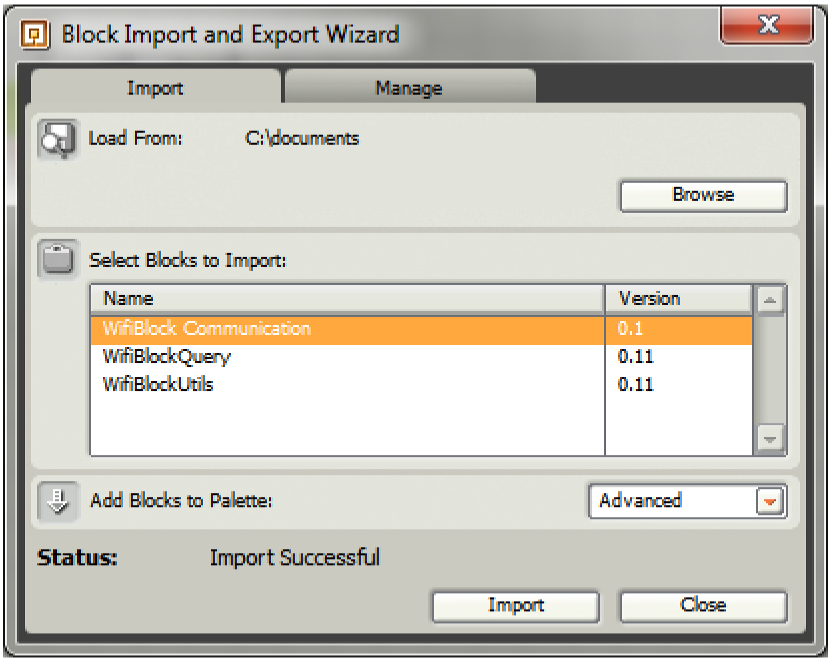
Name	Version
WifiBlock Communication	0.1
WifiBlockQuery	0.11
WifiBlockUtils	0.11

Add Blocks to Palette: Advanced

Status: Search Complete

Figure 18: Import wizard before importation

- Click on the “Import” button.
- The “Status” text is updated to “Import successful”.



Load From: C:\documents

Select Blocks to Import:

Name	Version
WifiBlock Communication	0.1
WifiBlockQuery	0.11
WifiBlockUtils	0.11

Add Blocks to Palette: Advanced

Status: Import Successful

Figure 19: Import successful

Do this for each block you want to import or update.

7 Appendix: upload a file from the NXT Brick

With your NXT Brick connected to your computer and switched on, click on the “NXT Window” button.

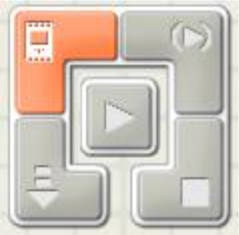
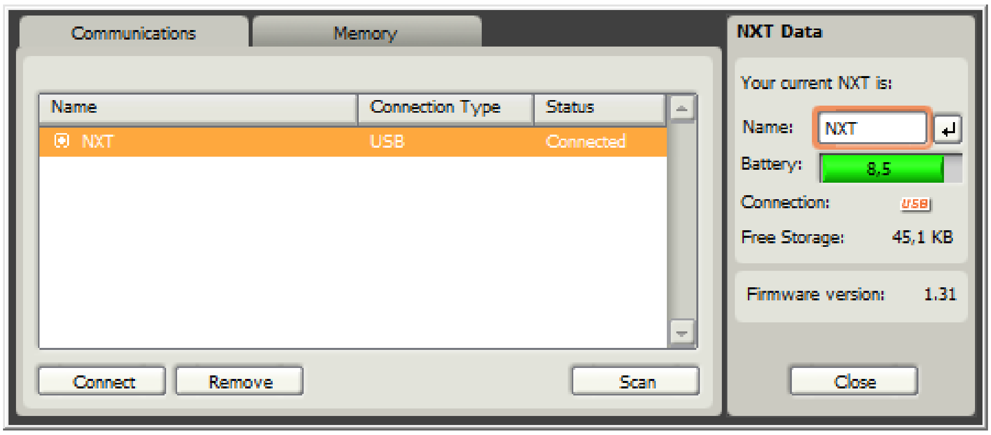


Figure 20: The “NXT Window” button.

A new window will open, showing information on connected devices. Find your Brick and you can see its name, battery level and firmware version.



Name	Connection Type	Status
NXT	USB	Connected

Your current NXT is:

Name: NXT

Battery: 8,5

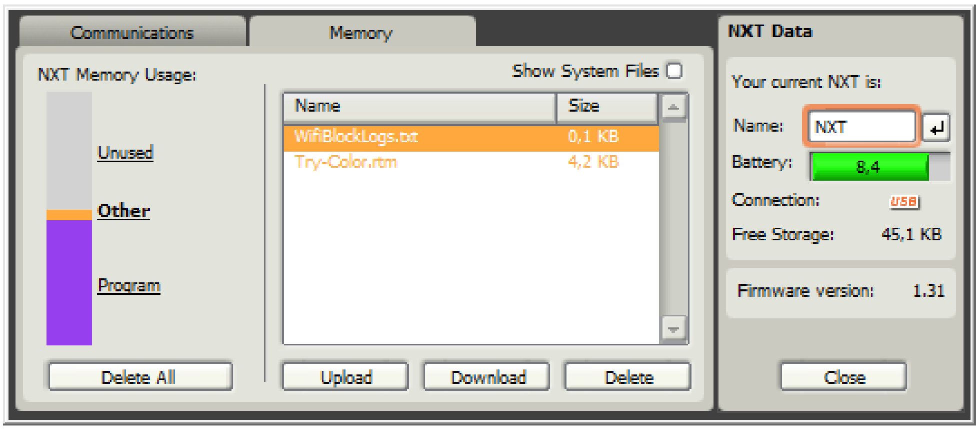
Connection: USB

Free Storage: 45,1 KB

Firmware version: 1.31

Figure 21: NXT window

Go to the “Memory” tab to access to the documents inside your Brick. Click on the names of the different sections (here “Unused”, “Other” and “Program”) to see the contents.



Show System Files □

Name	Size
WifiBlockLogs.txt	0,1 KB
Try-Color.rtm	4,2 KB

Your current NXT is:

Name: NXT

Battery: 8,4

Connection: USB

Free Storage: 45,1 KB

Firmware version: 1.31

Figure 22: NXT window with Memory tab

Select the file you want to retrieve and click on the “Upload” button. A window will ask you where to save in on your computer.
From this tab, you can also delete or download files (from your computer to the NXT Brick).