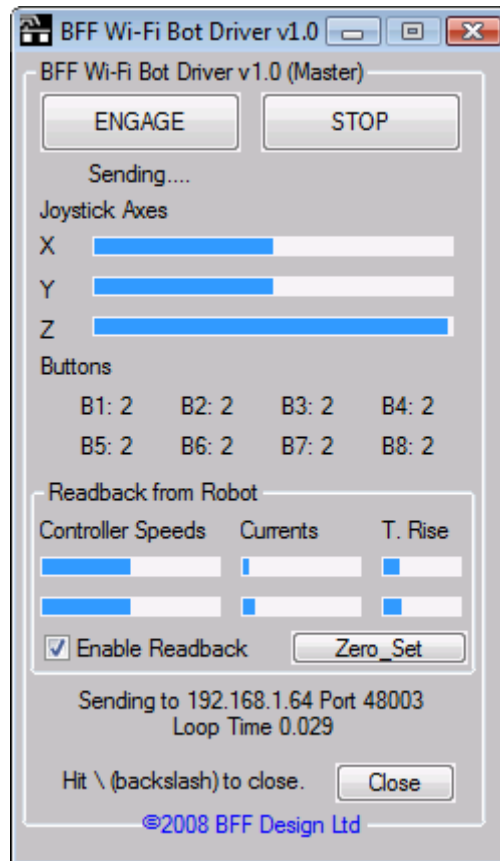


BFF Wi-Fi Robot Driver



User Manual (Draft)

(For Beta V1.0 Driver)

Ian Hopper (BFF Design Ltd)

© 2008, BFF Design Ltd, All rights reserved.

Table of Contents

BFF Wi-Fi Bot Driver – User's Guide.....	3
1. What is it & what does it do?.....	3
2. Installation and Configuration.....	4
3. Operation.....	6
4. Serial Communications Formats.....	8
5. LICENSE.....	9

BFF Wi-Fi Bot Driver – User's Guide

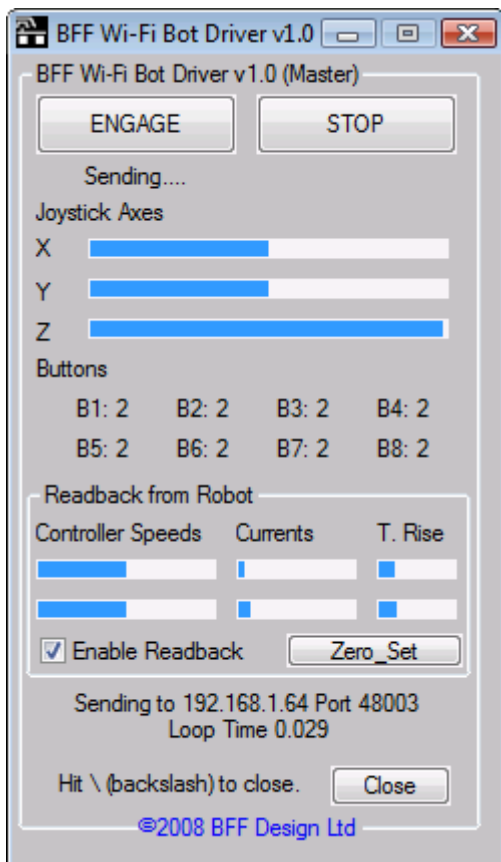
1. What is it & what does it do?

The BFF Wi-Fi Bot Driver software's was written to provide 2-way communications over a Wireless LAN between a fixed "master" PC and a laptop or Notebook PC carried on your mobile skid steered robot. It was written for use with the skid-steered BFF DIY Wi-Fi Bot however it could be used with any micro controller based control system which can read and write data in the serial format used by the software.

The software consists two separate elements; the BFF_Bot_Sender which is installed on the fixed master PC, and the BFF_Bot_Receiver which is installed on the mobile slave PC.

The Sender program reads joystick axis and button inputs from a joystick connected to the master PC on which it is installed and transmits them over the Wi-Fi network to the mobile PC where they are converted to a local serial output that is used to control the robot. It is the Receiver program installed on the mobile PC that does the data receiving and conversion. The Receiver program also accepts live read-back data from the robot and transmits this back to the master PC where it is displayed and/or recorded by the Sender program.

A standard joystick must be connected to the master PC and be operational. It must be properly calibrated using the normal XP / Vista game controller settings before the Driver software is used.



When used with the BFF DIY Wi-Fi Bot the joystick outputs are used to control speed and direction and to move an on-board camera. The read-back data carries motor controller speed, current and temperature data.

The transmitted data bytes could be used for other purposes depending on the micro-controller programming on the robot. Three joystick axis outputs and eight button outputs are transmitted in a UDP data packet to the robot, and 8 data bytes are transmitted back each data loop. The data refresh time is roughly 0.1 s giving about 10 loops per second but this may vary and can be faster.

The Sender program has a second mode of operation in which the joystick output is sent directly to a local serial (COM) port on the master PC. This can be used for wired debugging of the robot or for wireless serial transmission to a receiver on the robot.

2. Installation and Configuration.

IMPORTANT

You must calibrate the joystick using the normal calibration facilities in Vista or XP before using the Bot Driver software.

The software reads the joystick's X and Y axis positions on a scale of 0 to 100 with 50 as the mid or zero position. The fore/aft or Y axis is used to signal forward or reverse and speed, the left-right or X axis is used to signal left and right turns and speed of turn. The signals are mixed in the robot's SPU to generate speed and direction instructions for the motor speed controllers and an output of 50 for each axis corresponds to zero speed and zero turn.

If the joystick calibration is not centered properly then the startup speed and direction demand to the robot may not be zero and it may move unexpectedly.

INSTALLATION

Fairly simple..... copy the Bot_Sender directory and its contents to the master PC, and copy the Bot_Receiver directory and its contents to the mobile slave PC. The program .exes should be run from these directories – don't separate the .exes from the other files in the directories or they won't run.

Each program has its own .cfg configuration file. In these the LAN IP Address of the machines must be set – follow the format described in each .cfg file. The port numbers used by the software are also set here but unless you have to it's probably best not to change them.

The .cfg files are text files and can be opened with any text editor – eg notepad.

To find the IP Address of a PC open the "Command Prompt" from the program menu and type **ipconfig** , hit return and amongst the data returned there should be an IP number that looks something like 192.168.1.69. If there isn't the machine may not be present on the LAN. You will need to ensure Wi-Fi connectivity between your PC's before the driver software will work.

In the CNTRL-Slave.cfg file there are also settings for the COM port and baud rate to use for the serial read/write to the robot's micro-controller. The master config file must also have these set if the output "Mode" is set to 2.

The config file contents are typically -

For the Master:

```
-----  
48001  
48002  
192.168.1.69  
48003  
192.168.1.71  
COM5  
9600  
1  
0
```

The above items are -

- Write Port on this machine (Master PC)

- Read Port on this machine (Master PC)
- IP address of this machine
- Receiving Port on target PC (on robot)
- IP address of target PC (Slave) - all above are ignored if Write Mode = 2
- Local COM port to write data to if Mode = 2
- Baud rate to write at if Mode = 2
- Write Mode, 1 = UDP, 2 = Local Serial Port
- Data Trace Flag = 0 for no data output, = 1 for dump data to Bot-output.txt at program close

If the Data Trace Flag is set to 1 or 2 the read-back data from the robot is stored internally and dumped to a text file when the software is closed. Set to 1 to dump the processed current and temperature data returned by the BFF Wi-Fi Bot, set to 2 to dump the raw unprocessed byte data.

This should normally be set to 0, do not leave it as a non-zero value in normal use as it will clog up the program with internal variables and write very large files on close.

and for the **Slave**:

48003
 48004
 192.168.1.71
 48002
 192.168.1.69
 COM7
 9600

The above items are -

- Read Port on this machine (on robot)
- Write Port on this machine
- IP address of this machine
- Receiving Port on Master machine
- IP address of Master machine
- COM port to write data to
- Baud rate to write at

Don't alter the Baud rate settings unless you intend to change the programming of the microcontroller on the bot's SPU.

Don't alter the Port numbers unless you need to.

Setting the master and mobile PC IP addresses to the same will allow both programs to run on the same machine. This may be of use when setting up the software and robot.

3. Operation

Read the notes provided on the suggested construction and setup sequence for the robot and ensure that your system is properly debugged before the robot drive is fully enabled electrically and mechanically.

Operation of the software is fairly simple.

Start the Sender program on the fixed PC and the Receiver program on the mobile PC. Both programs will start with the data flow on STOP. On the Sender program this disables its UDP output to the network (or output to local COM port), and for the Receiver program this disables its serial output to the microcontroller on the bot.

Communications will not be fully enabled until both programs are switched to ENGAGE. This is a safety feature and allows the flow of data to be interrupted at either end of the data chain.

Both programs are also provided with a "Close" button, however the \ (backslash) key is also programmed as a hot key to close the driver with a single keystroke if required.

The Receiver program will also halt its output to the bot microcontroller if the Sender program is on STOP and no UDP data is being received; this allows you to control the bot fully from the fixed PC – click STOP on the sender program and the bot should stop. Problems with the Wi-Fi network causing a block on the UDP data flow will also cause the Receiver program to stop the robot.

The Sender (Master) program window -

Data export ENGAGE and STOP

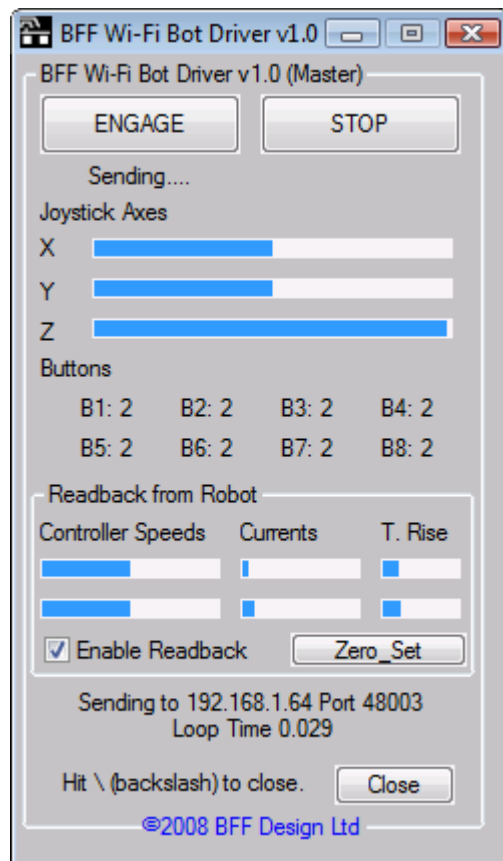
Current Joystick Axis Inputs

Current Joystick Button Inputs

Readback data area – active if enabled
Motor controller speeds, currents and temp rises.
Current scale is 0-20 Amps, Temperature rise is 0-50 C
Readback Enable check box and Zero Set for Current
and Temperature readback

Activity commentary text

Driver Close button



The Receiver (Slave) program window -

Data export ENGAGE and STOP

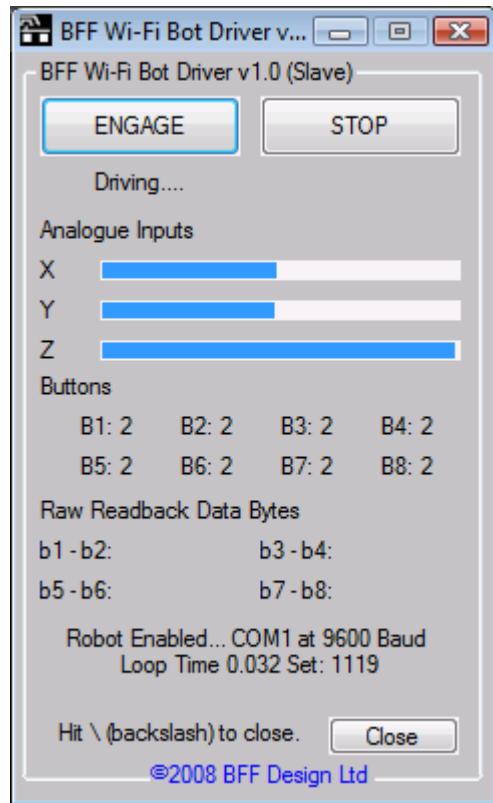
Current Joystick Axis Inputs

Current Joystick Button Inputs, 1=depressed

The data readback from the bot microcontroller – the raw 8 bytes

Activity commentary

Driver Close button



If the Sender program is active and UDP data is being received over the network the Joystick axis and button inputs shown in the Receiver program window should be active. You can use the display to confirm that joystick movements are being received by the mobile PC and therefore are available to drive the robot.

Readback data will only be available if the Readback is enabled on the Sender program and the microcontroller based SPU on the bot is connected powered-up and active. If the SPU is not active you will notice much larger Loop Times as the attempt to read serial data from the SPU times-out.

4. Serial Communications Formats

The serial format is 8 bit, 1 stop bit, no parity to the COM port stated at the baud rate stated in the configuration file.

The serial output from the Sender program to the bot is 14 bytes:

“AB” axis1 axis2 axis3 but1 but2 but3 but4 but5 but6 but7 but8 0x0D

AB is the start of data identifier, axis1-3 are the joystick axis positions on a 1-100 scale, but1-8 are the positions of buttons 1 to 8 on the joystick where =1 for button depressed and =2 for not depressed. 0X0D is the CR character.

The BFF Wi-Fi Bot SPU uses the axis 1&2 data for fwd/reverse and turn data, and buttons 2-5 for camera pan and tilt information. The programming of the PICAXE microcontroller dictates how the data is used. If you wish to use different joystick buttons to activate camera pan and tilt then the symbol declaration section at the top of the PICAXE .bas program can be altered to suit.

When the data Readback check box is enabled in the Sender program the Receiver program will expect a return serial output from the microcontroller each data loop in the following format:

“P” byte1 byte2 byte3 byte4 byte5 byte6 byte7 byte8

No processing is done on this data in the Receiver program – it is sent directly back to the Sender program for display or recording. For the BFF Wi-Fi Bot SPU it represents motor controller speed, current and temperature data – see the chip .bas programming for details.

5. LICENSE

This SOFTWARE PRODUCT is licensed as separate components.

Its component parts may be separated. The script components (BFF_Bot_Sender and BFF_Bot_Receiver) are licensed under the terms of the EULA below. The script's interpreter component is licensed under the terms of the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>).

Credit given to: AutoHotKey ©2003-2007 Chris Mallett, portions ©Autolt Team (<http://www.autohotkey.com/> and <http://www.autoitscript.com/>).

EULA

BFF_Bot_Sender.exe and BFF_Bot_Receiver.exe have been provided WITHOUT ANY WARRANTY without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. You assume all responsibility for the consequences of your use of BFF_Bot_Sender.exe or BFF_Bot_Receiver.exe.

BFF_Bot_Sender.exe and BFF_Bot_Receiver.exe are licensed for your personal NON-COMMERCIAL use only. I am happy to negotiate a reasonable charge with you if you wish to use BFF_Bot_Sender.exe or BFF_Bot_Receiver.exe for the purposes of commercial development including but not limited to research & development, design feasibility study, prototype development, production and/or inclusion within or packaging with domestic or commercial products.

You may not further distribute BFF_Bot_Sender.exe or BFF_Bot_Receiver.exe. You may not reverse engineer or modify BFF_Bot_Sender.exe or BFF_Bot_Receiver.exe.

Removal or modification of the License files will disable the software.

BFF_Bot_Sender is copyright ©, 2008 BFF Design Ltd.

BFF_Bot_Receiver is copyright ©, 2008 BFF Design Ltd.