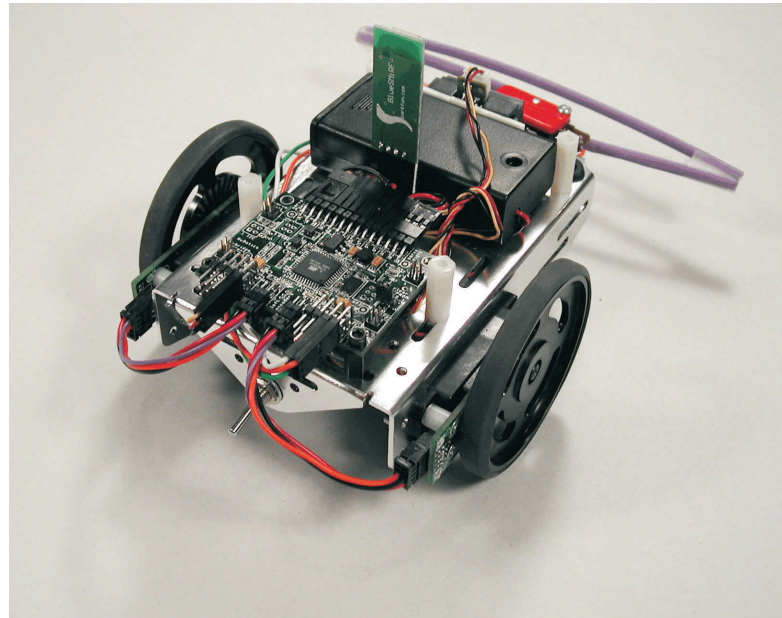


Application Note #2

10/3/2005

Wireless Communications with WheelCommander WC-132

The Nubotics WC-132 WheelCommander differential drive motion controller is ideal for use with wireless control of small robots from desktop computers. This application note explains how to control a WC-132 using a BlueRadios Bluetooth module packaged by Sparkfun. Control of the BlueRadios module is built in to the companion NdiCmd.DLL. This Dynamic Link Library is used by Windows programs such as the Wheel Commander's WC Wizard to provide easy access to the WheelCommander's functionality, whether through a conventional RS-232 serial port, a USB-based serial port, or wirelessly as explained below.



Bluetooth-enabled Boe-Bot with WheelCommander.

Bluetooth Serial Ports

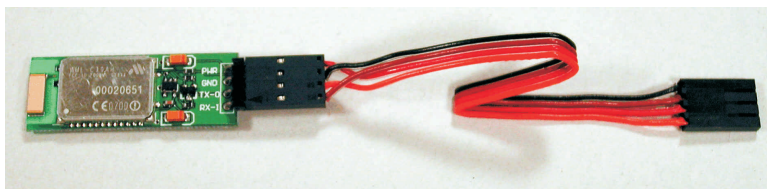
Bluetooth is the name of an international standard for 2.4GHz wireless communications, originally designed for wirelessly replacing cables between mobile computing devices and their peripherals. It has evolved to provide everything from wireless audio to spontaneously formed personal area networks (piconets). It does so using a layered protocol stack with separately defined profiles for each type of device, much of which is not useful for our application. However, by encapsulating the serial port profile, BlueRadios has made Bluetooth very useful for small robots.

BlueSMiRF Module

The BlueSMiRF module from Sparkfun uses a BR-C30 module from BlueRadios. It packages it with a voltage regulator and level shifting circuitry to enable 3-6v tolerant connections. The connector has only 4 pins, just like the communications connector on the WC-132, except for a slight difference in pinout. So, you will need to make a cable (female, 4 pin, .100" pitch connectors on each end) that connects the BlueSMiRF to the WheelCommander.

BlueSMiRF to WheelCommander Cable Pinout

Signal	BlueSMiRF	WheelCommander
Power	pin 1 (Pwr)	pin 2 (+5v)
Ground	pin 2 (Gnd)	pin 4 (Gnd)
TX- BlueSMiRF to WC	pin 3 (TX-O) output	pin 1 (RX) input
RX- WC to BlueSMiRF	pin 4 (RX-I) input	pin 3 (TX) output



Recommended Parts List

Quantity	Digikey Part#	Description
8	WM2557-ND	Molex/Waldom C-Grid Box Crimp Terminals
2	WM2802-ND	Molex/Waldom C-Grid 4 position 1 row crimp connector housings
2"x4 conductor	24-30 AWG wire	26 AWG rainbow ribbon cable (.050" pitch) works well, and helps you identify each signal

BlueRadios Configuration Requirements

The BlueRadios module can be configured from either the wired connection or over the wireless link, using modem-like “AT” commands (see the [BlueRadios AT Commands User Guide](#)). Rather than build knowledge of this Bluetooth module into the WheelCommander firmware, we chose to take advantage of the wireless configuration ability for configuration. The NdiCmd.DLL exposes a function, EnableBlueRadio(), which allows you to turn on the DLL’s ability to modify the BlueRadio modules behavior to suit the WheelCommander. This is exposed on the WC Wizard’s Comm Settings tab in the BlueRadios section.

Regardless of BlueRadios mode, NdiCmd.DLL performs a number of steps to establish a connection with the WheelCommander.

1. Enumerate all serial ports that can be opened
2. Ignore those that the user has disabled (e.g., via the checkmarks in the Comm Settings dialog next to each port number)
3. Iterate through all enabled ports:
 - a. Open the port
 - b. Set the baud rate and other parameters
 - c. Attempt a WheelCommander Sync command
 - d. If successful, attempt a WheelCommander Echo command
 - e. If successful, attempt a WheelCommander Name command
 - f. Declare the device found.

When NdiCmd.DLL's BlueRadio mode is enabled, NdiCmd.DLL does some additional tasks between steps 3b and 3c:

1. Request the BlueRadio module to enter command mode (+++)
2. Set the module to short response form (ATSW24,1,0,0,0)
3. Optionally, set the module baudrate to match the PC (ATSW20,157,0,0,0 sets it to 38400 baud, for example)
4. Optionally, write the baudrate to non-volatile memory
5. Put the module into normal or fast data mode

Step 2 must be done to prevent communications conflicts between the BlueRadio module's connection announcement messages and the WheelCommander's command processor. Step 3 is needed whenever changing the baud rate of the PC, but should not be done before changing the baud rate in the WheelCommander to the new baud rate first – otherwise, you will not be able to communicate with it (see Baud Rate Changes below). Fast data mode enables quicker response to WheelCommander communications, but locks out further configuration of the BlueRadio module until power is cycled.

See the [WheelCommander API Guide](#) for details of this and other API functions. Once properly configured, you can now control your robot from your PC up to 100 meters away!

Baud Rate Changes

There is a certain sequence that must be followed when changing the baud rate. It is important first to realize that all parties must be set to the same baud rate, and, they must be set in the correct order, so that communication can occur.

First, a connection must be made at a preestablished baud rate. The WheelCommander is factory-configured for 38400 baud. Using WC Wizard, set the PC serial port's baud rate to 38400, check on Blue Radio Link, and check on Change BlueRadio Baud, and optionally check BlueRadio Fast Data Mode and BlueRadio Write to EEPROM. This should enable the Search procedure to work, and you should get a connection. Then select a new RS232 Baud Rate, and hit the Change button; the steps below are done for you. It should now be connected at the new rate.

If you are writing your own wireless host software, do the following. The WheelCommander must first have its baud rate changed via the command string "F030x" – set constant at address 3 (baud rate) to the desired baud rate (05=57600, for example); this change is immediate. Next, change the BlueRadios baud rate (+++; ATSW20,236,0,0,0; ATMD sets it to 57600) to match. Finally, change the host serial port's baud rate. Of course, you should then retest the link by performing Sync and Echo commands to make sure everything is ok.

If you make a mistake and cannot establish a connection, you can do one of two things. One, set JP1 and power cycle the WheelCommander to clear the EEPROM and select 38400 baud. Or two, use a normal wired connection to configure the WheelCommander directly to the proper baud using the 'F030x' command, then try again. You can also use a terminal emulation program such as Hyperterminal to manually modify the BlueRadios baud rate as shown in step 3 above.