

Bluetooth-Based Display for GPS Data

Jay interfaced a Wintec Bluetooth module to an LCD and a PIC16F88, which connects to a miniature Bluetooth GPS receiver. The display shows the date, time, speed, direction, location, and number of the satellites used by the GPS receiver.

Cell phones are ubiquitous these days. Phones incorporating digital cameras are rapidly replacing basic models. The latest cell phone to be making headway, however, is the Bluetooth-enabled model that permits hands-free talking. It features a lightweight headset without any dangling and tangling wires.

Bluetooth is a low-power wireless networking technology that's currently finding its way into many products. This project demonstrates the ease with which Bluetooth may be implemented in PIC-based applications. I coupled a BASIC-programmed PIC with a Bluetooth module to wirelessly connect to a miniature Bluetooth GPS receiver (see Photo 1). The system decodes the GPS data and displays position, speed, and heading information on an LCD.

Although it doesn't provide a graphical moving map display or routing instructions, my system can easily replace a PDA or laptop to provide basic GPS information. It can also form the core of an automated vehicle tracking system.

WIRELESS OVERVIEW

GPS uses a network of 24 NAVSTAR satellites to determine a unit's position anywhere on Earth. A GPS receiver decodes simultaneous transmissions from multiple satellites and then reports position data to an accuracy of several meters. GPS technology is routinely used for military applications, aviation and surface navigation, and vehicle tracking. GPS receivers have decreased markedly in both cost and

physical size as their sensitivities and capabilities have increased.

Bluetooth is a short-range wireless connectivity technology. It's used for wireless headsets, keyboards, mice, short-range networking, digital camera interfacing, automotive diagnostics, and a host of other applications.

(Wires are absent from such applications, but functionality is improved.) The most intriguing device, however, is a Bluetooth GPS receiver that isn't much larger than a couple of 9-V batteries, including its antenna and power supply. It can wirelessly connect to a PDA, a laptop for mobile navigation systems, or to my system to display basic GPS information.

NEMESIS-BASED SYSTEM

I used a Kronos Robotics Nemesis microcontroller, which is a Microchip Technology PIC16F88, for this project. It is preconfigured to permit you to program it in Athena, a PIC version of the BASIC language. The language and



Photo 1—Take a look at the PIC16F88-based Bluetooth GPS platform. A Wintec WBTV42-D-SPP Bluetooth module sits to the right of the PIC16F88 Nemesis and above the RS-232 interface MAX232 chip. The project wirelessly connects to a GPS module and displays date and time, speed and heading, and latitude and longitude data.

development platform are available for free on the Kronos Robotics web site.

The processor incorporates a built-in buffered UART for serial communications and an LCD driver. It also includes a host of PIC standards such as timers, PWMs, and digital and analog I/O. I interfaced the Nemesis to a Wintec WBTV42-D-SPP Bluetooth module in order to wirelessly connect to a

Bluetooth GPS receiver, process the data packets, and display the data on an LCD.

In addition to providing a stand-alone Bluetooth GPS display, this project also provides a test platform for experimenting with either the Bluetooth module or the Nemesis microcontroller. Three modes of operation are available. They're selected by setting two three-pin headers on the circuit board. Figure 1 illustrates the principle components and the system's operating modes.

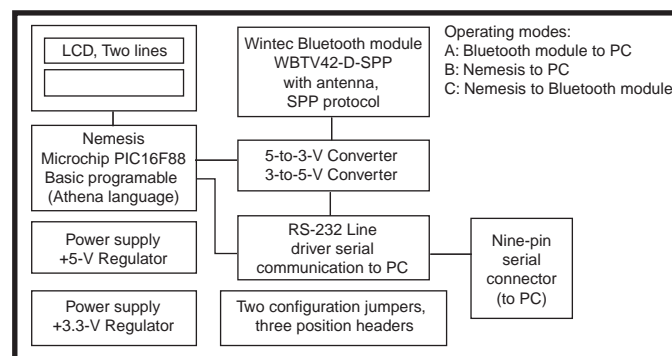


Figure 1—The project's three operating modes are determined by setting two three-pin headers on the circuit board. The Bluetooth module may be controlled via ASCII text commands from an RS-232-connected PC (mode A). In mode B, the Nemesis PIC may be programmed via a PC. In mode C, the PIC uses the Bluetooth module to connect to a Bluetooth GPS and display its data.

The BT module may be connected to a PC and operated via commands entered using HyperTerminal. The PIC and LCD are not used in this mode. Commands to the Bluetooth module are sent as text strings (e.g., AT+ZV Discovery). The responses also appear in an easy-to-read text format. In this mode, the circuit board provides the power supply and RS-232 serial communications interface to a PC. Listing 1 shows instructions for connecting the Bluetooth module to the GPS receiver and the resultant datastream.

You may also connect the Nemesis microcontroller to a PC via the mode configuration headers. Doing so enables you to program the PIC or use the board for further experimentation or program development.

The third mode (the operational mode) connects the Bluetooth module directly to the Nemesis. The Nemesis

uses the Bluetooth module to detect BT devices in its vicinity, bond to the Bluetooth GPS unit, and initiate a serial port protocol (SPP) connection with the device. The SPP connection enables the GPS module to wirelessly send data packets to the Nemesis, which then decodes and displays the information.

The circuit board includes the Nemesis microcontroller, the WBTV42-D-SPP Bluetooth module, a RS-232 driver chip, 3.3- and 5-V power supplies, configuration headers, and an LCD connector. Power, LCD backlight, and two push button switches are also included, forming a self-contained test bed. One push button switch cycles through the various display modes and is used to set the local time zone. The other push button switch is used to reset the Bluetooth module.

You may power the board with a 9- to 12-V AC/DC power supply, a 9-V nonrechargeable battery, or a 12-V vehicle supply. As a test bed, the LCD is mounted above the circuit board, thereby enabling you to read the display while having access to the circuitry, switches, and LEDs. However, you may also mount it on the backside of the circuit board (overlying it) to facilitate the mounting of the board within a smaller case.

NEMESIS SOFTWARE

There is a multitude of different predefined GPS NMEA data packets. My GPS receiver supports five packet types, each identified by its unique header. The \$GPRMC packet provides the time, date, latitude, longitude, speed, and heading information. This information is sent as ASCII text strings once per second. The Nemesis parses the data packet, captures the data, and displays it on the LCD. The speed is reported in knots, which is then converted to miles per hour. The direction, reported in degrees, is converted to an alpha character (e.g., N/NE/E). The direction of travel, up to 359 degrees, is scaled by 2 in order to permit single-byte integer math (0 to 255) to be used when calculating the alpha character direction value.

The GPS receiver reports the time in Coordinated Universal Time (UTC) format. To display the local time, you must add an offset to the hours to match your local time zone. Eastern Standard Time, for example, is UTC less 5 h, and it changes twice a year to accommodate daylight savings time. Rather than reprogramming the PIC for every time zone or time change, you may enter Set Up mode by pressing the Display mode push button switch while powering up the unit. You may then cycle through the 0- to ±12-h offset, which is displayed on the LCD, to set the correct offset for your local time zone. You exit Set Up mode by turning off the device. The local time is displayed after you reboot the device. The offset is stored in the PIC's EEPROM.

A two-line LCD driven by the PIC is used to present the GPS data. A push button switch cycles through three

Listing 1—The WBTV42-D-SPP module may be directly controlled via user-typed commands on a PC. All you need are the Reset, Discovery, Bond, and SPPConnect commands to initiate the GPS datastream. Alternatively, the PIC can be used to automate this process and format the data for display on an LCD.

```

AT-ZV -CommandMode- 'Power Up message
AT-ZV BDAddress 000f70102065 'Wintec BT Module's BT Address

AT+ZV Reset 'Issue Software Reset command
AT-ZV ResetPending
AT-ZV -CommandMode-
AT-ZV BDAddress 000f70102065

AT+ZV Discovery 'Issue Discovery Command
AT-ZV InqPending
AT-ZV DiscoveryPending 5 '5 BT devices responded
AT-ZV BondPending 000f70102085
AT-ZV Device 000f70102085 "WINTEC Serial Port" NoSvcs 'Another BT module
AT-ZV BondPending 00092d2d6000
AT-ZV Device 00092d2d6000 "JCPDA" NoSvcs 'PDA's BT
AT-ZV BondPending 00081b0ca81b
AT-ZV Device 00081b0ca81b "BT GPS" NoSvcs 'BT GPS Receiver
AT-ZV BondPending 00038947b9a7
AT-ZV Device 00038947b9a7 "M3000 by Plantronics" NoSvcs 'BT Headset
AT-ZV BondPending 000c783017be
AT-ZV Device 000c783017be "JCDe11XPS" NoSvcs 'PC's BT Dongle

AT+ZV Bond 00081b0ca81b 0000 'Issue Bond command to BT GPS
AT-ZV BondPending
AT-ZV BondOk 00081b0ca81b

AT+ZV SPPConnect 00081b0ca81b 'Issue SPP Connect command
AT-ZV ConnectionUp
AT-ZV -BypassMode- 'Auto-switch to Data Mode

$GPRMC,231338.168,V,4113.5389,N,08128.8709,W,0.00,0.00,080905,,A*6D
$GPVTG,0.00,T,,0.00,N,0.00,K,A*70
$GPGGA,231339.168,4113.5389,N,08128.8709,W,0,00,00.0,321.6,M,-35.6,M,,*57
$GPGSA,A,1,,,,,,,,,0.0,0.0,0.0*30
$GPGSV,3,1,09,03,,,,07,,,,08,,35,11,,,37*7C
$GPGSV,3,2,09,19,,,,26,,,,27,,27,28,,31*75
$GPGSV,3,3,09,29,,,,,,,,,,,,,*7B
$GPRMC,231339.168,V,4113.5389,N,08128.8709,W,0.00,0.00,080905,,A*6C
'Data stream continues from the BT GPS receiver.

```

display formats: time and date, speed and direction, and latitude and longitude. Photo 2 shows the project's real-time GPS coordinates.

The number of satellites being received and used for the position calculations varies from moment to moment. Between four and 12 are typically in sight above the horizon, providing an adequate signal for inclusion in the calculations. The more satellites used, the more accurately a position may be determined.

A minimum of four satellites is required for a three-dimensional position report because time is a fourth variable used in the GPS calculations. The position is then reported in a latitude, longitude, and altitude coordinate system. Fortunately, the GPS receiver does all of these calculations, and it provides the data to the connected device. The number of active satellites is included in both the date and time and the speed and direction display modes. The individual satellite ID codes and signal strengths are available, but they aren't being used at this time.

To fit the Bluetooth handshaking software, the GPS data parsing and calculations, and the time zone setup all in the PIC's memory, limited feedback is provided on the LCD during the connection phase. At power-up, the Nemesis instructs the WBT42-D-SPP Bluetooth module to connect to the GPS unit. Within approximately 30 s, the display comes alive with the current date and time. The BASIC program (NemBTGPSa.txt) is posted on the *Circuit Cellar* FTP site.

You may download a separate program (NemBTConnect.txt) to the Nemesis that provides more detailed feedback during the Bluetooth connection phase of operation. After the connection, it streams unformatted GPS data to the LCD. This program is useful for learning the actual Bluetooth device connection process. The included serial interface makes it easy to modify and download various programs to the Nemesis.



Photo 2—After first establishing a Bluetooth connection to a GPS receiver, the current latitude and longitude are available for display. The 16-channel Bluetooth GPS receiver provides the data.

The Bluetooth Special Interest Group was founded in 1998. The long 1,200-page Bluetooth V. 1.2 specification is available at www.bluetooth.org. The good news is that you need only a handful of commands to connect to a Bluetooth device and start its data flowing. The Discovery, Bond, and SPPConnect commands are all that is required. The interface's simplicity truly isolates you from the complexity buried within.

HARDWARE SPECIFICS

The project's hardware is illustrated in Figure 2. The Nemesis firmware includes an LCD driver. A low-cost, parallel interfaced LCD may be directly connected to the PIC and written to with simple BASIC instructions. This eliminates the need for using an LCD driver chip, writing your own driver code, and relying on a more expensive serially interfaced LCD. The circuit board incorporates an LCD contrast control. Mid-position is a good starting point when constructing the project, but it may need adjustment if the start-up message isn't visible at power-up.

A MAX232 chip connects either the Nemesis or the WBT42-D-SPP Bluetooth module to a PC via an RS-232 serial port. This chip uses the 5-V supply and internally generates a ± 10 -V supply for RS-232 communications. The board incorporates a nine-pin female serial port connector. A straight-through cable (not a crossover/null modem cable) is used to connect the project to a PC for programming and experimentation.

The power supply includes a bridge rectifier. Therefore, you can use either an AC or DC power supply module (i.e., a wall wart) without regard for the polarity of its connector. Alternatively, power may be supplied from a nonrechargeable 9-V battery or a 12-V automotive power supply. A series diode protects the circuit in case you accidentally connect the battery backwards during installation. Two three-terminal fixed-voltage regulators provide 5- and 3.3-V power for the remainder of the circuit.

The WBT42-D-SPP Bluetooth module uses a 3.3-V power supply. It can't be directly connected to the PIC or the MAX232 chip, both of which use 5-V signals. A simple resistive divider (R10 and R11) is used to attenuate the 5-V signal sent to the Bluetooth module to 3 V. A two-transistor circuit bumps up the 3-V signal to 5 V for the PIC and MAX232. Both transistors invert the signal supplied to it. The result is a pumped-up, 5-V, noninverted signal at the collector output of transistor T2.

A 3-V high signal supplied by the WBT42-D-SPP Bluetooth module is applied to the base of transistor T1 through resistor R6. This causes T1 to turn on, drawing its collector current through resistor R4 and making its collector voltage go low. A low signal supplied to the base of transistor T2 (through resistor R7) leaves T2 in the non-conductive off state, resulting in its collector being high, at 5 V. Similarly, a low signal applied to T1 results in a low signal output from T2.

The LCD's backlight is powered from the nonregulated DC power source, which eliminates the need for a heatsink on the 5-V regulator. Two series resistors are used to limit the LCD backlight's LED current. At 70 mA, it has a soft glow, and the resistors generate minimal heat. Their value can be changed, or one resistor can simply be replaced with a wire (0 Ω) to increase the backlight current and hence its brightness. For projects with extra memory, you can use the PIC's pulse width modulator to

adjust the brightness to any desired level. This is excluded from the present design.

WINTEC MODULE

The WBTV42-D-SPP Bluetooth module comes packaged in a 24-pin DIP format, making it easy to use with a prototyping board. The module includes a 2.4-GHz UHF transceiver. The radio performs frequency hopping amongst the 79 Bluetooth channels at a rate of 1,600 hops per second. The data transmission complete with compression and error correction uses Gaussian frequency shift keying. Bluetooth defines three RF power classes.

As a Class II device, this module has a typical range of up to 30 m. An antenna, complete with the required impedance matching network, is incorporated in the module. RF data transmission rates approach 1 Mbps.

Clearly, the RF hardware design alone strongly favors the incorporation of commercially available Bluetooth modules (rather than designing your own from scratch) for many applications that simply need to provide Bluetooth connectivity.

The WBTV42-D-SPP module incorporates a 32-bit RISC processor to run the UHF transceiver and to provide the Bluetooth-compatible software interface. A Bluetooth software stack is a hierarchy of layers, each performing a defined function and interfacing to the other layers in a defined manner. At the high end of the stack are profiles, each of which facilitates the performance of specific tasks. Profiles for wireless keyboards and mice, audio transmission, serial port emulation, dial-up networking, and a multitude of other functions are defined within the standard.

At the software level as well, using a commercially available Bluetooth module eliminates the need to write your own low-level Bluetooth software. This provides Bluetooth functionality and compatibility, while eliminating significant RF hardware and software design hurdles and permitting you to focus on your project's primary purpose.

The SPP in the Bluetooth module emulates a virtual high-speed wireless serial port. Data rates up to 128 kbps are supported. With a Bluetooth module incorporated within each device, a connection is set up through the software, totally eliminating connectors and cables.

The WBTV42-D-SPP module uses a serial port (UART) to connect to its host processor (in this case the Nemesis or a PC). The serial port's data rate is programmable from 9,600 bps to 921.6 kbps.

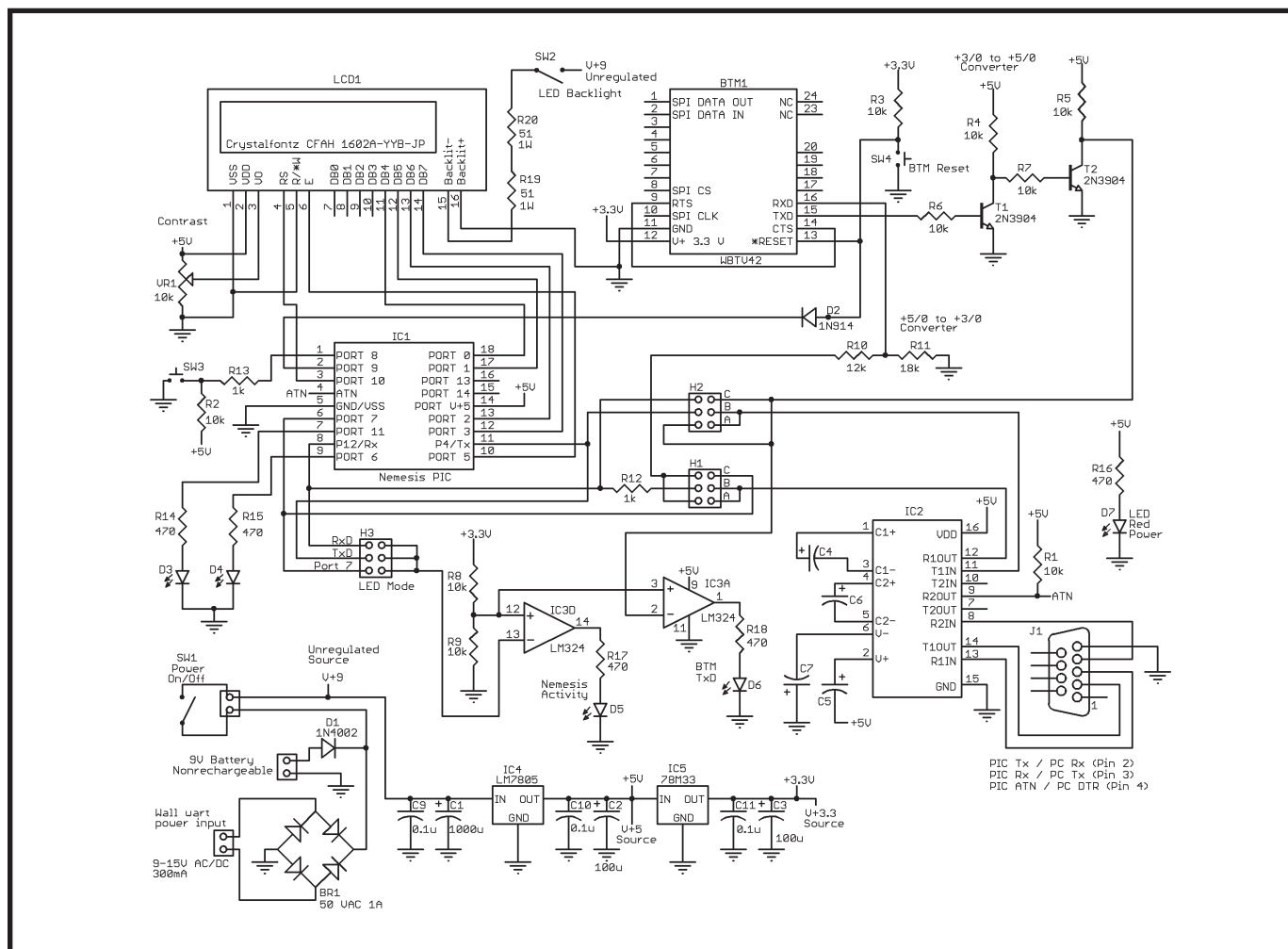


Figure 2—The schematic shows a BASIC-programmed PIC using a WBTV42-D-SPP Bluetooth module to connect to a Bluetooth GPS receiver and display its data on an LCD. RS-232 communications is used to test the WBTV42-D-SPP module and program the PIC. Two transistors bump the module's 3-V signal to 5 V for the remainder of the circuit.

Commands are sent and received using ASCII text. By connecting the Bluetooth module to a PC, you can type in commands and read responses directly on the PC's display. You can easily connect to the Bluetooth GPS using HyperTerminal. This enables you to determine its unique Bluetooth address and watch the GPS data packets stream by on the PC's display. Manually entering the commands is a good way to learn the Bluetooth connection process, which is then automated by the PIC.

Although there are more than 25 commands that may be sent to the WBT42-D-SPP module, only a few are needed to connect to a Bluetooth GPS and initiate the sending of its data stream. All commands sent to the WBT42-D-SPP module begin with AT+ZV, which is followed by a space, the command itself, any arguments, and a final carriage return (0D hex). The module responds with AT-ZV, a space, its response, a final carriage return, and line feed (0D, 0A hex). All Bluetooth devices have a unique 48-bit address, which will vary from what's shown in Listing 1.

After issuing a Discovery command, all discoverable Bluetooth devices in

the vicinity reply with their Bluetooth addresses, their names, and the profiles (services) they support. You then bond to the desired device (in this case the Bluetooth GPS receiver) selected by entering its Bluetooth address and a four-character PIN code. After they're bonded, the SPP connection is initiated. When a successful connection is formed, the Bluetooth module automatically switches from Command mode to Bypass mode and bidirectional data flow is initiated. The PIN code for my GPS unit is 0000. Using the last four characters of the device's unique Bluetooth address is another common scheme.

SYSTEM SETUP

Assembling the circuit board is a straightforward process. The circuit board layout's silkscreen layer, which is available for download, illustrates the location and orientation of the various components. The initial set-up process involves three steps. The Bluetooth module's default communications rate is 115,200 bps. By setting the jumpers on headers 1 and 2 to the A position, the module is connected to a PC through the board's serial port. You may then test the module and reset its default

data rate to 9,600 bps, which matches that of the Nemesis. Refer to the "Bluetooth Test Bed Circuit Board" sidebar for more information.

Placing the jumpers in the B position connects the Nemesis to the PC, thereby enabling you to download the program to it. Be sure to change the Bluetooth address in the BASIC code to match that of your Bluetooth GPS receiver.

Finally, moving the jumpers to the C position connects the Nemesis to the WBT42-D-SPP module. Be sure to turn off the system prior to repositioning the jumpers.

The board's third header is primarily for experimenting with the Nemesis microcontroller. It connects LED D5 to the Nemesis's port 7, which is used to transmit commands to the WBT42-D-SPP module or the Nemesis's transmit or receive UART pins (monitoring the uploading and downloading of data between the PIC and a PC).

LED D6 is another hardware monitor. It flashes with data transmissions from the WBT42-D-SPP Bluetooth module. When it's SPP-connected to the GPS receiver, the LED flickers with each data packet that is received and sent to the Nemesis for analysis.

Bluetooth Test Bed Circuit Board

The original prototype for this project was constructed on a solderless breadboard. Subsequently, I designed a small two-layer circuit board to make the project portable and to enable others to experiment with Bluetooth. I decided to use ExpressPCB to manufacture the boards because of its software, pricing, rapid turnaround, and my prior experience with its service.

The circuit board design is posted on the *Circuit Cellar* FTP site (NemBTMV3.pcb). Go to the ExpressPCB web site to download its software to view, print, modify, and order the board.

To connect the WBT42-D-SPP Bluetooth module to a PC, set the two mode configuration header jumpers to position A. Establish serial communications with the module via HyperTerminal. In HyperTerminal, you must first name your new connection. Ignore the country, area code, and phone number, but set the "Connect using" option to the serial port to which the project is connected. This is apt to be Com 1, 2, 3, or 4. Set the communications parameters to 115,200 bps, 8 data bits, no parity, and 1 stop bit. Set "Flow

control" to "None," and then apply these settings. Under File/Properties/Settings/ASCII Setup, enable both "Send line ends with line feeds" and "Echo typed characters locally."

The start-up message in Figure 1 should appear when you turn on the system. After entering this sequence of commands, you can reboot so the new data rate takes effect. Be sure to reset the data rate in HyperTerminal if you want to continue using it to exercise the WBT42-D-SPP module.

BTM:	AT-ZV -CommandMode- AT-ZV BDAAddress 000F7010001	'Power up message 'This is the Wintec module's BT address
User:	AT+ZV Reset	'Send a software Reset command to the BT module
BTM:	AT-ZV ResetPending AT-ZV -CommandMode- AT-ZV BDAAddress 000F7010001	'Reset response
User:	AT+ZV ChangeDefaultBaud 9600	'Send Set default baud rate command
BTM:	AT-ZV Baudrate Changed	'Response

Figure 1—This command sequence, which is used to communicate with the Nemesis MCU, resets the Wintec Bluetooth module's default data rate from 115,200 to 9,600 bps. The change takes effect when the module is rebooted.

LEDs D3 and D4 are controlled by the Nemesis's software. Currently, D3 indicates a successful Bluetooth connection, while D4 flashes when a desired data packet is recognized in the datastream sent by the GPS receiver.

Your Bluetooth GPS's unique address must be inserted in the program downloaded to the Nemesis microcontroller. With tighter coding, or more abundant memory, you could easily scroll through a list of

discovered Bluetooth devices and select the device to which you desire to be connected.

BLUETOOTH BLUES?

The 1,200-page Bluetooth standard can be intimidating. This project, however, demonstrates the ease with which you can incorporate Bluetooth technology in even the simplest projects. Only a few instructions are required to discover, bond, and connect devices using the SPP.

The WBT42-D-SPP module simplifies the process of experimenting with BT technology. The Nemesis microcontroller provides a low-cost, BASIC-programmed core. As a fully functional example and test bed, this project will eliminate your Bluetooth blues and put a smile on your face as you watch the GPS datastream pass. 📍

Jay Carter (carter@docjc.us) earned B.S.E.E. and M.S.E.E. degrees from Purdue University. He also holds an M.D. degree from Northeastern Ohio Universities College of Medicine. When he isn't practicing emergency medicine in an ER, you'll find him flying or tinkering in electronics.

PROJECT FILES

To download the code and additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2006/190.

RESOURCES

P. Bennett, NMEA-0183 and GPS information, <http://vancouver-webpages.com/peter/>.

NMEA-0183 Format and information, <http://gpsd.berlios.de/NMEA.txt>.

M. G. Simpson, "Athena, Perseus, and Nemesis Manual," Kronos Robotics and Electronics, 2003.

Trimble tutorial on GPS, Trimble Navigation Limited, www.trimble.com.

Wintec Industries, "Bluetooth Module: WBT42," 2005, <http://wintecind.com/oem/download/bluetooth.pdf>.

SOURCES

CFAH1602A-YYB-JP LCD
Crystalfontz America, Inc.
www.crystalfontz.com

Nemesis Microcontroller
Kronos Robotics
www.kronosrobotics.com

PIC16F88 Microcontroller
Microchip Technology, Inc.
www.microchip.com

WBT42-D-SPP Bluetooth module
Wintec Industries, Inc.
www.wintecind.com