TinyTrakII Owner's Manual

Version 1.00
Byon Garrabrant N6BG www.byonics.com

Overview

TinyTrakII is a small, inexpensive radio controller designed to receive and broadcast position reports from a GPS receiver. It removes the need for a full TNC (terminal node controller) in an APRS tracker. When combined with an NMEA-0183 compatible GPS receiver, and a radio transmitter, TinyTrakII will key the radio at user-defined intervals, and transmit the GPS's current position. The position can be sent as text, or in the compact binary MIC-E format, and can include position with altitude, speed, heading, and a status message. TinyTrakII can be configured for time slotted reporting, or the dynamic reporting rate or SmartBeaconing. All TinyTrakII operating parameters are stored in the internal EEPROM, and will persist even when power is removed. TinyTrak and TinyTrakII were created by Byon Garrabrant, N6BG. More information can be found at the Byonics web site at www.byonics.com.

Acknowledgments

TinyTrakII was developed with the help of many individuals who deserve recognition. To begin, TinyTrakII would not have been possible without the supporters and testers of TinyTrak: Michael Connors, KI7AB; Mike Pendley, K5ATM; Michael Lussier, K4MQF; Curt Mills, WE7U; Dave Nishikawa, KC6YFG; Harry Bloomfield, M1BYT; Tom Boyle, WA6UHB; Bert van Dalen, PA0DAL; Steve Bragg, KA9MVA; Arnold Harding, KQ6DI; Guy Beaumont, G7BWP; Dave Hargus, K6NFL; Ed Flinn, WA6YVX; Luc Bodson, ON9AAV; Rolf Bleher, DK7IN; Zachary Michael, KB9UQQ; Ken Mirabella, KM6YH; James Jefferson, KB0THN; Eric Van Norman, KF0S; Mark Ramskill, M1AJG; Ian Hamilton-Douglas, G6VND; Harry Mueller, KC5TRB; James Vincent, G1PVZ; Tommy Ellison, K4TWE; Dave Lee, W6ZL; and Lara Garrabrant , KD6AYO. Thanks go to all these people for helping make TinyTrakII all that it is.

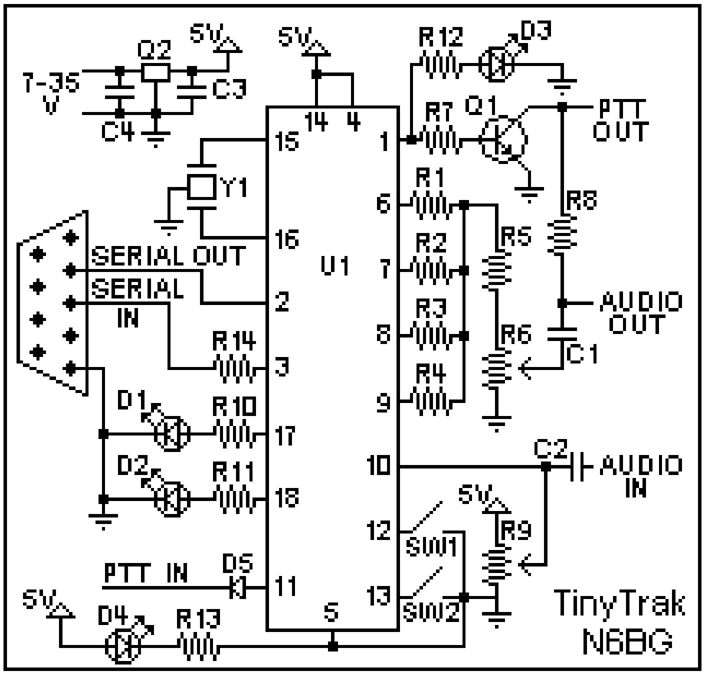
Construction

TinyTrakII runs on the same circuit as TinyTrak and uses a TinyTrak PCB. The difference is in the microcontroller. The TinyTrakII microcontroller has more memory than the TinyTrak microcontroller, which allows for the added features. TinyTrak owners can upgrade to TinyTrakII by simply replacing the TinyTrak chip with a TinyTrakII chip.

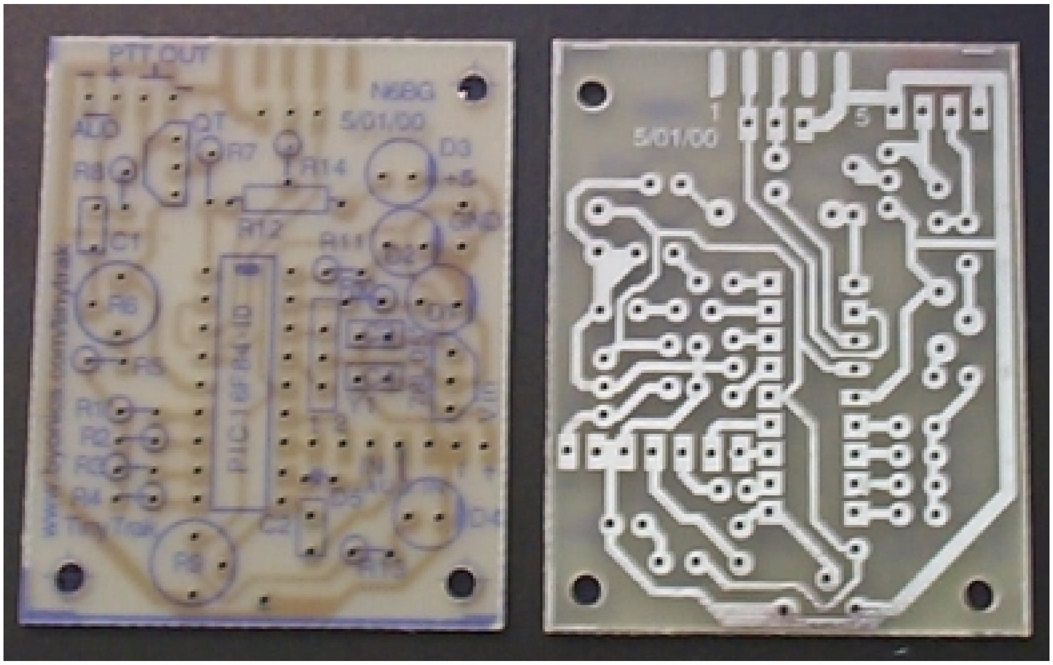
Parts List

Part	Description
U1	PIC 16CE625-20/P microcontroller (programmed with TinyTrakII firmware)
Y1	10 MHz ceramic resonator
Q1	2N2222A NPN transistor
Q2	78L05 +5V voltage regulator
R1	8.2K ohm resistor (gry-red-red)
R2	3.9K ohm resistor (org-wht-red)
R3	2K ohm resistor (red-blk-red)
R4,R10,R11, R12,R13	1K ohm resistor (brn-blk-red)
R5	220K ohm resistor (red-red-yel)
R6,R9	10K trimmer potentiometer (103)
R7,R14	10K ohm resistor (brn-blk-org)
R8	2.2K ohm resistor (red-red-red)
C1,C2,C3,C4	0.1 uf capacitor (104)
D1	Yellow T1-3/4 LED
D2	Green T1-3/4 LED
D3,D4	Red T1-3/4 LED
D5	1N4148 diode
J1	DB9 female connector
	18 pin dip socket

Schematic



Printed Circuit Board



Assembly Instructions

TinyTrakII is a fairly simple construction project that can usually be built in less than an hour. You will need a low wattage pencil-type soldering iron with a small tip, some thin solder, a pair of diagonal cutters, and a pair of needle nose pliers. The microcontroller (U1) is static sensitive, so use standard precautions. For each item, insert on the component side (blue silk-screen side), then turn the board over and solder the leads to the pads on the trace side. Be sure to only solder the correct pad, and do not let any solder touch any other pad or trace. Trim any excess leads with diagonal cutters after soldering each component. The following checklist will be useful to insure all components are properly assembled.

- ❑ Install the 18-pin IC socket for U1. Be sure to align the notched end of the socket with the marked end on the silk screen. Do not insert the chip into the socket at this time.
- ❑ Install ceramic resonator Y1, which provides the clock oscillation for the chip. Direction does not matter.
- ❑ Install resistors R1 – R5 which create the 4–bit digital to analog resistor ladder. For each resistor, bend one of the leads 180 degrees at the bulb of the resistor so that both leads are parallel and 0.1 inch apart. Install vertically on the board. Polarity does not matter. Bend the leads apart once inserted to hold in place. See parts list for color coding and identification.
- ❑ Install potentiometer R6, which adjusts audio output level.
- ❑ Install capacitor C1, which smoothes and de-biases the audio output. Polarity does not matter. Leads may need to be bent to fit holes.
- ❑ Install resistors R7 and R14. Use instructions for resistors above.
- ❑ Install transistor Q1, which provides Push-To-Talk (PTT) for the radio transmitter. Be sure to orient flat side to match the long flat side shown on the silk-screen pattern (labeled QT on some boards).
- ❑ Install female DB-9 serial connector J1. Wedge the board in between the two rows of solder cups, with the 5 pin side aligned with the 5 pads on the PCB. Solder all five cups to the pads.
- ❑ Install voltage regulator Q2. Be sure to orient flat side to match the long flat side shown on the silk-screen pattern. (Note: If TinyTrakII will be supplied with 5V directly, Q2, C3, and C4 will not be needed)
- ❑ Install bypass capacitors C3 & C4. Use instructions for capacitors above.
- ❑ Install resistor R8. Use instructions for resistors above. Note: If TinyTrakII will be used with a radio which does not key via current through the mic line, such as mobile radios and Kenwood handhelds, R8 will not be needed. If R8 is used, you should not wire PTT OUT to the radio.
- ❑ Install potentiometer R9, which adjusts carrier detect sensitivity.
- ❑ Install capacitor C2. Use instructions for capacitors above.
- ❑ Install glass diode D5. The black strip should be furthest from the PIC. (Note: Although is will cause no problem to include, D5 is not needed for TinyTrakII, as it does not support MIC-E style burst after voice.)
- ❑ Install Carrier Detect LED D1 (yellow) and resistor R10. This LED will light constant when audio is detected from the radio receiver, and flash during the quiet time before transmitting. Be sure to align the flat side of all LEDs with the stripes on the silk-screen.
- ❑ Install GPS State LED D2 (green) and resistor R11. This LED will light constant when receiving good (locked) GPS data, and flash when receiving bad (unlocked) GPS data. It will be off when no GPS data is received.
- ❑ Install PTT LED D3 (red) and resistor R12. This LED will light when the radio is keyed via PTT.
- ❑ Install Power Display LED D4 (red) and resistor R13. This LED will light when device is powered.

Assembly Completion

After all components have been installed, inspect the solder side of the board for poor or cold solder joints. All pads should be shiny and smooth. Inspect for any undesired solder bridges. If the board looks ready,

- ❑ Insert programmed microprocessor U1. The rows of pins may need to be bent slightly. Be sure to align the notch on the chip with the notch on the socket, as well as the notch on the silk-screen (nearest connector J1). An improperly inserted chip may be permanently damaged.

Interfacing

There are a few interface points for the TinyTrakII PCB which must be connected before operation.

Power

Power can be applied via the Vin + and – points. The voltage regulator can handle 7-35 volts. At 12 volts, current draw is 6.6ma + 3ma for each LED lit. If desired, the 4 1K LED resistors can be replaced with 10K resistors, which will cause the LED to only draw about 0.5 ma each. If Q2, C3, & C4 were NOT installed on the board, 5V power can be applied via the +5 and GND points near the LEDs. Only 5V can be applied at this point.

Serial

TinyTrakII must be connected to a computer for configuration of call sign and other operating parameters, and then connected to a GPS to receive position data. J1 pin 2 is used to transfer serial data from the TinyTrakII to the computer. J1 pin 3 is used to transfer serial data from the computer or GPS to the TinyTrakII. J1 pin 5 is serial ground. The TinyTrakII can be plugged directly into a computer 9-pin serial port, either with or without a serial extension cable. Due to the voltage levels used, some laptop computers may not be able to communicate with the TinyTrakII. If the GPS to be used normally can plug directly into a computer's serial port, both a gender-changer (male-to-male) AND a null-modem adapter will be needed to interface the GPS to TinyTrakII. If the GPS does not connect directly into a computer's serial port, an interface will need to be built. The GPS should have a male DB-9 with GPS serial data out wired to pin 3, and ground to pin 5. GPS serial input is not required. The GPS must output the $GPRMC sentence or the $GPGGA sentence or both at 4800 baud N81. The $GPRMC sentence provides position, speed and direction. The $GPGGA sentence provides position and altitude.

Transmitter

To transmit the positional packet beacon, a transmitter must be connected to TinyTrakII via the AUD OUT and optionally PTT OUT points. If the transmitter transmits (PTT) when the microphone input is grounded (most handheld (HT) radios, except the Kenwood brand), resistor R8 must be installed, but PTT OUT will not need to be connected to the transmitter. For all other transmitters, PTT OUT will be needed. Connect AUD OUT, Ground, and PTT OUT if needed to the transmitter microphone / PTT input. Refer to the transmitter's manual for more information, and look for a section on installing a terminal-node controller (TNC) for packet operation.

Receiver

To prevent transmissions over other stations, TinyTrakII should be interfaced to a radio receiver. Connect the receiver's audio out (earphone) jack to the AUD IN and Ground points near R10. Note TinyTrakII cannot decode the incoming packet data. Instead, it just watches for audio energy (voice, data, or static) to hold off transmitting over other stations.

Mic Input

The PTT IN interface is not used with TinyTrakII.

Switch 1

This switch input will select the primary or secondary operating parameters. When left floating, or at 5 volts, the primary parameters are used. When grounded, secondary parameters are used. Whenever this switch is changed, the timers are reset, the CD LED will blink rapidly for a short time, and then a transmission will be sent..

Switch 2

This switch input is not currently used.

Operations

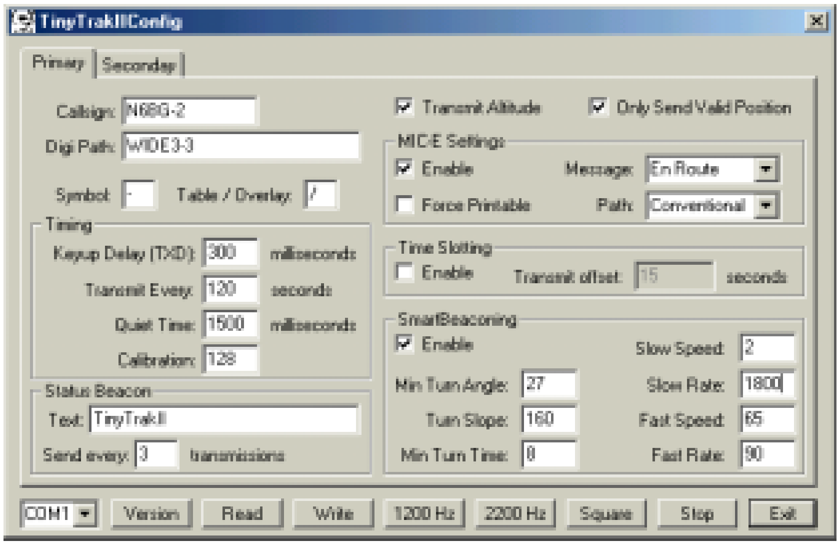
TinyTrakII Adjustment

There are only a few adjustments required for proper operations of TinyTrakII. First, the transmit audio level should be adjusted at R6 for proper deviation. Use the transmit tones buttons (1200 Hz/2200 Hz/Square) in the configuration software to cause the TinyTrakII to transmit while adjusting. You can listen on a separate receiver, and start the potentiometer at maximum drive. Lower the drive level until there is a noticeable change in the receiver. Overdriving the transmitter is a common cause of failure to decode. TinyTrakII was designed for hand held radios. Some mobile radio require more audio drive than TinyTrakII puts out. If audio levels are too low, even with the R6 pot set to maximum, consider replacing the 220K R5 with a 100K resistor. This should allow for about double the audio range.

The next adjustment point is R11, which is used to adjust the carrier detect base level. This should be adjusted such that Carrier Detect LED is usually off, but turns on when the radio opens squelch. Radio receiver volume should be set to maximum, and the radio can NOT have a constantly open squelch.

The last adjustment necessary is the calibration value. This adjusts timing slightly to compensate for ceramic resonator error. Try a few different values while monitoring decoding on a TNC, and set to the value which gives the best decoding.

Configuration Software



TinyTrakIIConfig

Primary | Secondary

Callsign: N6BG-2

Digi Path: WIDE3-3

Symbol: - Table / Overlay: /

Timing
- Keyup Delay (TXD): 300 milliseconds
- Transmit Every: 120 seconds
- Quiet Time: 1500 milliseconds
- Calibration: 128

Status Beacon
- Text: TinyTrakII
- Send every: 3 transmissions

☑ Transmit Altitude ☑ Only Send Valid Position

MIC-E Settings
- ☑ Enable Message: En Route
- ☐ Force Printable Path: Conventional

Time Slotting
- ☐ Enable Transmit offset: 15 seconds

SmartBeaconing
- ☑ Enable Slow Speed: 2
- Min Turn Angle: 27 Slow Rate: 1800
- Turn Slope: 160 Fast Speed: 65
- Min Turn Time: 8 Fast Rate: 90

To set user options, such as call signs and transmit rates, use the TinyTrakIIConfig.EXE program, available from www.byonics.com. First, connect TinyTrakII to a computer serial port. Launch TinyTrakIIConfig.EXE, and select the connected serial port. Click "Version" to confirm the TINYTRAKIICONFIG program can communicate with TinyTrakII. If a firmware version number is reported, the software and circuit are communicating. If not, re-check all previous steps. Click Read to upload the current configuration, edit all fields for desired options, and click Write to download options to the chip. Reading the Configuration is always performed twice and compared in the program to insure correct data. Writing the configuration is always followed by a read and compared to insure correct data. A dialog will notify the user when the read or write is complete, and if it was successful. A read or write operation may fail if the unit is transmitting. If this happens, simple try again. Detailed information about each configuration parameter is included below.

Primary / Secondary

These tabs select whether to software is viewing and setting the primary or secondary parameters. All parameters can be adjusted independently in the two configurations. The primary settings will be used if the SW1 input is left floating or connected to 5 volts. The secondary settings will be used if SW1 input is grounded.

Callsign

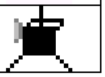
This is the identification of the transmitting station. It can each be an amateur radio call sign such as N6BG, or a tactical call such as SHUTLE. No more than 6 characters (excluding SSID) may be used. If a tactical call is used, an amateur radio call sign should be included in the status beacon for FCC identification requirement compliance. As with normal packet radio, an optional SSID between 1 and 15 can be included, such as N6BG-2 or SHUTLE-15.

Digi Path

This optional path will allow the transmission to be repeated by digipeaters. It should be entered in the form or call signs or aliases with optional SSIDs, separated by commas, such as RELAY,WIDE,WIDE . The more callsigns in the digipeater path will limit the space available for the status beacon.

Symbol & Symbol Table / Overlay

These settings sets the symbol most APRS programs will display when this tracker's position is received. The symbol table / overlay will modify the display of the symbol by switching to the APRS Alternate Symbol Table, or adding an overlay char. Use / for the primary symbol table, use \ for the alternate symbol table, and use a number (1-9) for an overlay of the symbols that allow it. The table on the right shows some symbol examples. Other symbol characters can be found in the APRS documentation, as well as APRS receiving programs.

Symbol	Table/Overlay	Icon
>	/	
j	/	
<	/	
[	/	
S	\	

Timing

The timing parameters affect when the transmissions occur.

KeyUp Delay (TXD)

This sets the delay in milliseconds after the transmitter is keyed, until the data begins. It is similar to the TXD setting in most TNCs. A value of 200 ms would be equal to 1/5 second. Maximum delay is about 1700 milliseconds.

Transmit Every

This setting controls how often, in seconds, a position transmission will occur. Valid range is between 1 seconds and 65535 seconds (18.2 hours).

Quiet Time

This setting controls the approximate delay in milliseconds that must occur after the receiver squelches, before a transmission will occur. It can keep transmissions from occurring on a busy channel. During the quiet time, the CD LED will flash quickly.

Calibration

This setting is included to help compensate for inaccuracies in the ceramic resonator. By adjusting the value, packet rates can be sped up or slowed down. A value or 128 represents no correction. Experiment with different values to see which has the best results on a receiving TNC.

Status Beacon Text

TinyTrakII can send a text beacon message after periodic location transmissions. This setting sets the beacon text. The length of the beacon message is limited by the number of digipeaters in the path. If the primary and secondary callsign and digipeaters are identical, and if the primary and secondary status beacon texts are identical, that will allow even more room for the status beacon text. For example, if the primary and secondary callsigns and status beacon texts match, and neither use a digipeater, the status text can be 89 characters (including spaces). If the callsigns or digipeaters are different between primary and secondary, and 2 digipeaters are used in each, the identical beacons can be 51 characters long, or each beacon can be 25 chars long.

Send Every

This setting controls how often a status beacon is sent, in units of normal position transmissions. A setting of 1 would send the beacon with every position transmission, a setting of 2 would beacon with every other position.

Transmit Altitude

Clearing this setting will disable the TinyTrakII from transmitting altitude information. If this box is checked, and the TinyTrak receives a $GPGGA message from the GPS, altitude information will be sent.

Only Send Valid Position

Setting this option will disable TinyTrakII from sending a position transmission when the GPS data is invalid. GPS data will be invalid before the receiver has locked on, if the receiver loses view of the satellites, or if the GPS is disconnected from TinyTrakII. Status beacons will continue to be sent even when the position is invalid. If sending invalid positions is allowed, and MIC -E format is selected, the INVALID flag will be sent in the MIC-E data.

MIC-E Settings

These following four settings configure the MIC -E format. This format is not readable text, but instead a compressed binary form. It should look something like this:

```
        N6BG-9>S8PRPY,RELAY,WIDE:'2+"!r,j/]"4K}
```

This compressed form contains position, speed, and bearing and altitude. More information about this format can be found in the APRS documentation at TAPR’s web page (www.tapr.org).

MIC-E Enable

This setting enables the MIC -E format. Is the MIC -E format is not selected, APRS text format will be used.

MIC-E Force Printable

If this setting is selected, the MIC-E format data will be slightly modified to maintain printable characters. Sending non-printable characters has been found to corrupt MIC-E data in some receivers in the past. When the data is modified, some rounding will occur. For example, a bearing of 2 degrees is will be converted to a bearing of 4 degrees.

MIC-E Message

This setting selects one of the 8 pre-assigned MIC-E messages. Caution should be used when selecting this setting, as beaconing with the “Emergency!” setting will alert most APRS receiving users to your emergency.

MIC-E Path

This setting selects of the 16 pre-assigned MIC-E paths. In order to use the path entered above, this should be set to “Conventional”.

Time Slotting

If enabled, the offset setting sets the time from the start of the hour that a transmission will occur. After that transmission, transmissions will continue at the Transmit Every rate until re-syncing at the next start of the hour. This option is useful to preset transmission times for multiple trackers. TinyTrakII will only synchronize with the GPS time at the start of the hour, so there may be a delay of up to an hour before the TinyTrak begins transmitting in its assigned slot. TinyTrakII will transmit at the Transmit Every rate, even if is has not yet synced. If the Transmit Every rate evenly divides into 600 seconds, TinyTrakII will synchronize at every 10 minutes. If the Transmit Every rate evenly divides into 60 seconds, TinyTrakII will synchronize at every minute. Time slotting and SmartBeaconing are mutually exclusive.

SmartBeaconing

These options setup SmartBeaconing. SmartBeaconing is an algorithm created by Steve Bragg for adjusting the transmit rate by the speed and heading changes on the tracker. As the tracker moves faster, the transmit rate will increase linearly. SmartBeaconing also uses CornerPegging to cause transmissions to occur when the tracker turns corners. The SmartBeaconing algorithm works like this:

```
IF (speed < slow_speed)                        // "Stop" threshold
        beacon_rate = slow_rate;
ELSE {                           // We're moving; adjust beacon rate to speed, and peg corners
       turn_threshold = min_turn_angle + turn_slope / mph;    // adjust turn threshold according to speed
       IF (heading_change_since_beacon > turn_threshold) AND (secs_since_beacon > min_turn_time)
              CornerPeg = TRUE;  // Corner pegging
       IF (speed > fast_speed)     // Adjust beacon rate according to speed
              beacon_rate = fast_rate;
       ELSE
              beacon_rate = fast_rate * fast_speed / speed;
}
if ((secs_since_beacon > beacon_rate) OR CornerPeg)
        // ... send beacon
```

More information on SmartBeaconing can be found at www.hamhud.net . SmartBeaconing and time slotting are mutually exclusive.

SmartBeaconing Min Turn Angle, Turn Slope, and Min Turn Time

These settings set CornerPegging parameters for the minimum turn angle (degrees), turn slope (MPH-degrees), and minimum turn time (seconds). Turns less than the minimum turn angle since the last transmission will not cause a transmission. The lower the turn slope, the quicker TinyTrakII will transmit due to turning.

SmartBeaconing Slow Speed, Slow Rate

This setting sets the SmartBeaconing slow speed (MPH) and rate (seconds). Speeds slower than the slow speed will cause TinyTrakII to beacon at the Slow Rate.

SmartBeaconing Fast Speed, Fast Rate

This setting sets the SmartBeaconing fast speed (MPH) and rate (seconds). Speeds faster than the fast speed will cause TinyTrakII to beacon at the Fast Rate.

Com Port

This setting selects which communication port the TinyTrakII is connected to during configuration. Currently only the first four serial ports are supported.

Version Button

This button will get the current firmware version from the connected TinyTrakII. This is a useful, simple test to confirm the computer and TinyTrak can communicate.

Read Button

This button will upload all configuration settings from the connected TinyTrakII into the setting fields above. This step is useful to change a single setting on the TinyTrakII configuration.

Write Button

This button will download all configuration settings entered above into the TinyTrakII. The settings are stored in non-volatile memory, so they will persist when power is removed from TinyTrakII.

1200 Hz Button / 2200 Hz Button / Square Button

These buttons will cause TinyTrak to transmit a 1200 Hz tone, a 2200 Hz tone, or alternate between the two (square) on a connected radio. This is useful for setting transmission levels, and measuring audio frequencies when adjusting calibration.

TinyTrakII Testing & Troubleshooting

After setting the desired configuration options, TinyTrakII should be connected to a radio, GPS, and power supply, and will be ready for use. When power is applied, the carrier detect and valid LEDs should flash three times to show proper firmware operation and proper LED wiring. If carrier is not detected, PTT LED D3 should light for about a second. Power LED D4 should always be lit.

If a radio is connected, it should transmit periodically and a packet burst should be heard on a receiving radio. When the receiver's squelch is opened, the carrier detect LED should light, and all transmissions will be delayed. If a radio receiver and TNC is available, packets should be able to be monitored.

If the TinyTrakConfig.EXE program cannot communicate with the TinyTrakII, it may be possible to test the serial link by running a terminal program, such as HyperTerm, configuring for 4800 baud, N81. When TinyTrakII is first powered up, it will send its version to the terminal program. Then by sending ESC V from the terminal program, TinyTrakII should again respond with its version. This should test the ability to communicate serially with TinyTrak. If serial communications are still failing, monitor the PIC pin 3 with a meter or an oscilloscope to check for incoming serial data, or try a different computer.

If TinyTrakII keys a transmitter, sends a packet burst, but then fails to un-key the transmitter, it may be due to local RF energy. Lowering power output, or moving the transmitting antenna further away from the TinyTrakII can resolve this. If it doesn't, place a 0.1uf capacitor across the base and emitter of PTT transistor Q1 (ground and center).

If the radio keys up upon connection to TinyTrakII, even before power is applied to TinyTrakII, it could be that the radio does not key via current through the mic line (most mobile and Kenwood radios), and resistor R8 should be removed.

Hints, Tricks, & Notes

- The TinyTrakII uses inverted TTL levels (0 & 5 volts) for serial communication, rather than true RS-232 levels (-12 & +12 volts). This may cause communication problems with some GPS receivers and computers, especially laptops. A RS-232 level converter, such as a MAX232, may be used, but the TTL levels will need to be un-inverted, with transistors or a TTL inverter chip.
- TinyTrakII is configured as a DCE with a female DB-9 connector to allow simple connection to a PC for programming. If a DTE configuration with a male DB-9 is desired, this can be accomplished by placing a DB-9 male connector of the PCB edge, but shifted one pin over, so the old DB-9 female pin 2 pad mates with the new DB-9 male pin 3, the old pin 3 mates with the new pin 2, and the new DB-9 male pin 5 is not mated with any pad, but is instead just past the old pin 1. Then, jumper the new DB-9 pin 5 to ground. This configuration should allow simple connection to a DCE GPS, but will require a gender changer and NULL modem to connect to and configure with a PC.
- If higher current is desired from the TinyTrakII's voltage regulator in order to provide 5 volts to a GPS, the supplied 100ma 78L05 can be replaced with a 1 amp 7805. A heat sink may also be required.
- TinyTrakII's carrier detects circuit detects all audio energy, not just packet audio tones. Therefore, it will not transmit over received voice audio. Be sure NOT to run an open squelch, as this will prevent all transmissions. The receiver should be set for maximum volume.
- If carrier detect, MIC-E style PTT input, SW1, or SW2 are not desired, the components normally connected to these pins can be left out. Unconnected microcontroller pins will float to the default correct behavior.
- For the 4-bit digital to analog resistor ladder, it has been determined that neither high precision resistors (1%) nor perfect powers of 2 resistors (1K, 2K, 4K, 8K) are required.
- TinyTrakII can be mounted in a case with a GPS, with a single DB-9 jack for PC configuring. To do this, wire the GPS serial output through a 10K resistor before connecting to the solder side of pin 3 of TinyTrakII's DB-9 connector. This way, with no PC connected, the GPS serial data will feed into the TinyTrakII, but when the PC is connected, the PC will over-power the GPS during configuration.

- If you enter a small Transmit Every value, you may find it difficult to write new config settings, since the TinyTrakII will be too busy transmitting. To resolve this, turn the R9 pot such that the CD LED turns on. This will prevent TinyTrakII from transmitting, and all the new settings to be written.
- TinyTrakII was designed for hand held radios. Some mobile radio require more audio drive than TinyTrakII puts out. If audio levels are too low, even with the R6 pot set to maximum, consider replacing the 220K R5 with a 100K resistor. This should allow for about double the audio range.