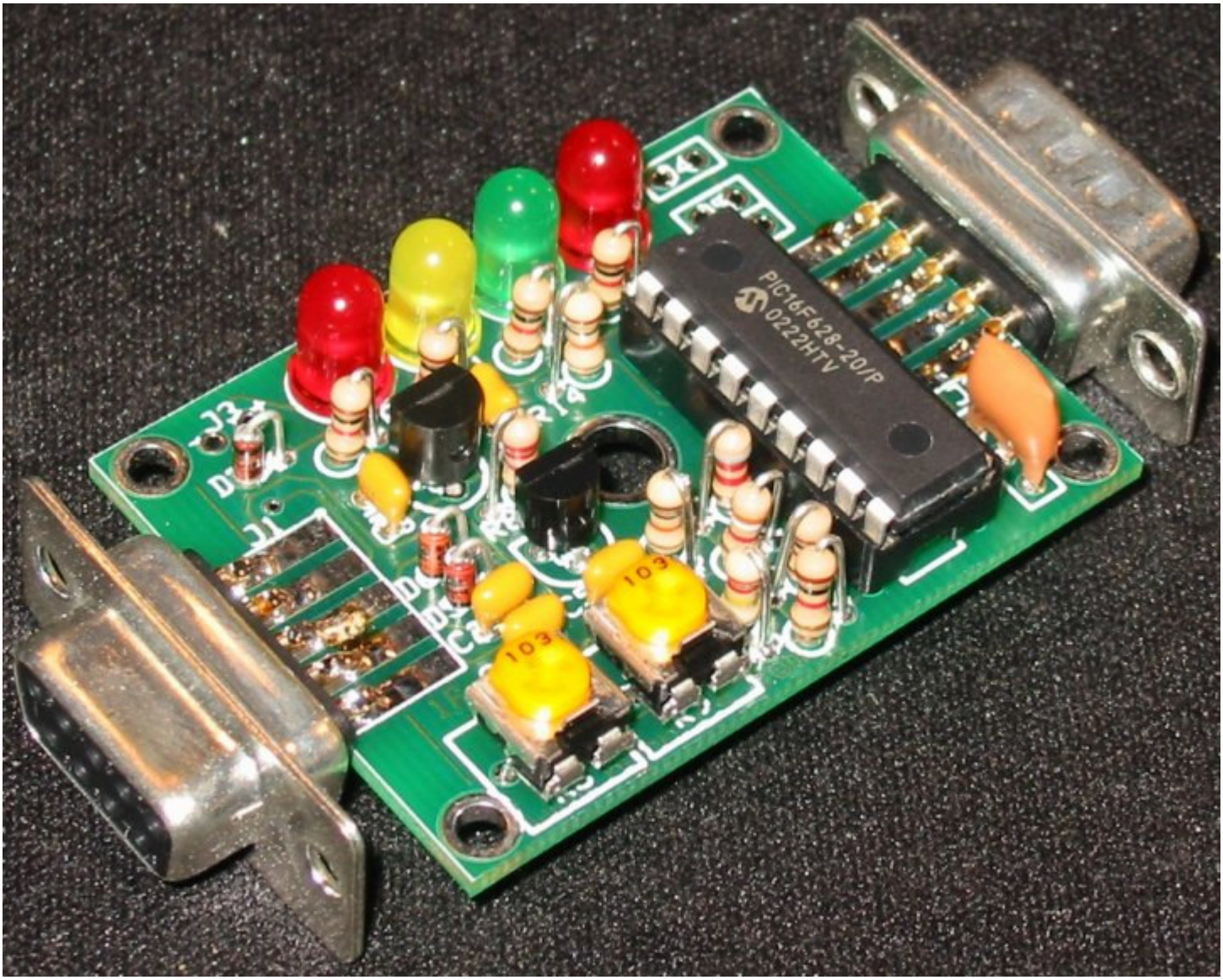
BYONICS

TinyTrak3 Owner's Manual

Version 1.14

http://www.byonics.com/tinytrak



Overview 2
Acknowledgments 2
Construction 3
Schematic 3
Parts List 3
Printed Circuit Board 4
Upgrading from an older TinyTrak 4
Case 4
Assembly Instructions 5
Assembly Completion 6
Interfacing 7
Radio – J1 7
Serial – J2 7
Power – J3 (also J1 and J2) 7
LED Control – J4 7
Primary/Secondary Switch SW 1 – J5 8
Power Switch – J6 8
Serial Power Enable – J7 8
Operations 8
Configuration Software 8
Primary / Secondary 8
Callsign 9
Digi Path 9
Symbol & Symbol Table / Overlay 9
Timing 9
Status 10
Checkbox Options 10
MIC-E Settings 11
Time Slotting 11
SmartBeaconing 12
Power Switch 12
Configure 12
Tone Test 13
Save / Load 13
Windows XP Issue 13
TinyTrak3 Adjustment 13
Troubleshooting 14
Frequently Asked Questions 15
Hints, Tricks, & Notes 16

Overview

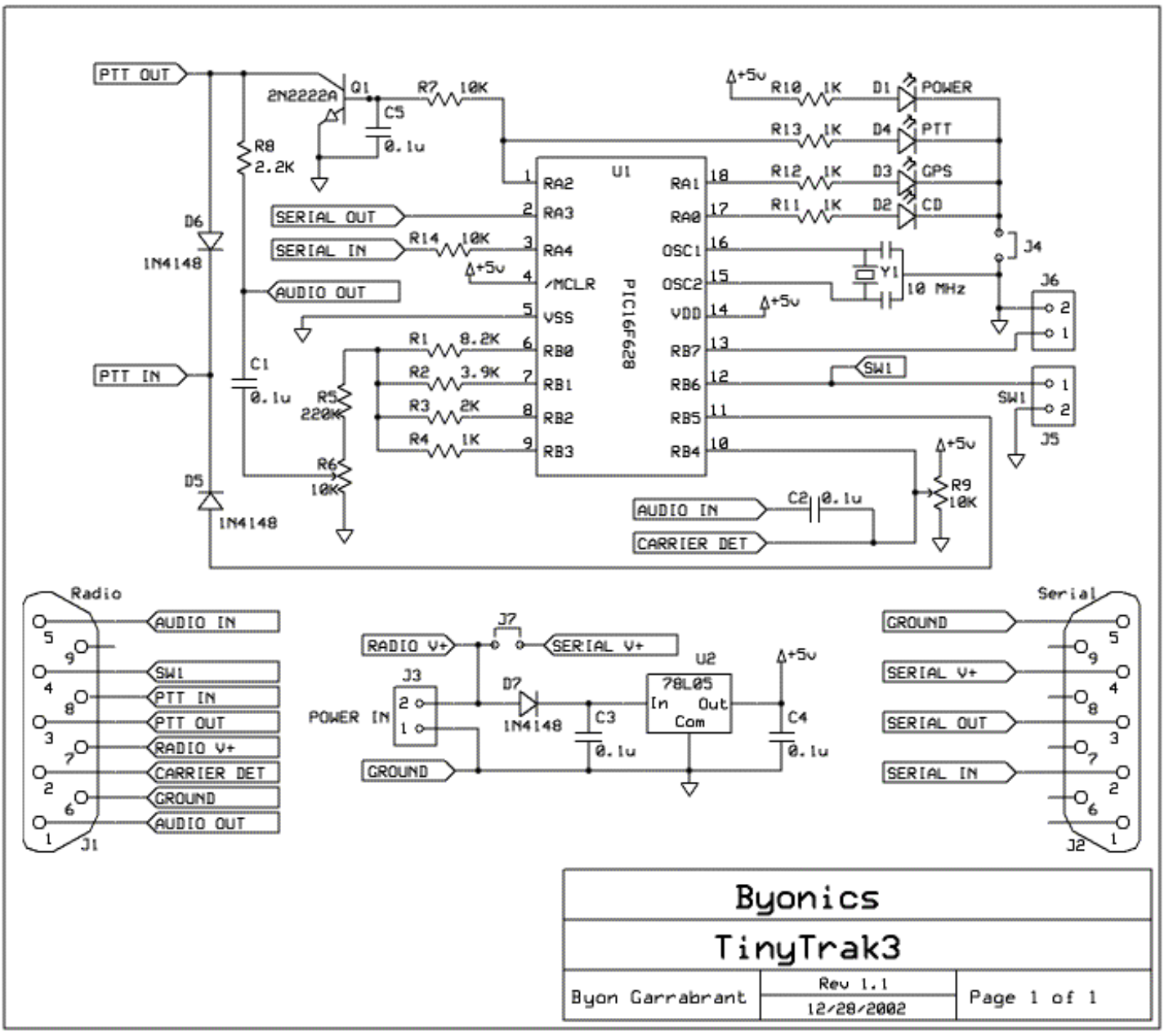
TinyTrak3 is a small, inexpensive radio controller designed to receive and broadcast position reports from a GPS receiver. It removes the need for a full TNC (terminal node controller) in an APRS tracker. When combined with an NMEA-0183 1.5 compatible GPS receiver, and a radio transmitter, TinyTrak3 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, a status message, and a timestamp. TinyTrak3 can be configured for time slotted reporting, or the dynamic reporting rate of SmartBeaconing. TinyTrak3 can conserve radio battery use by turning the radio on just before a transmission will be sent, and can be configured to transmit a 1/3 sec position burst after unkeying after voice traffic, or to send positions on command. It supports most RS-232 and TTL GPS receivers, and can even be used to broadcast a fixed position. All TinyTrak3 operating parameters are stored in the internal EEPROM, and will persist even when power is removed. TinyTrak3 was created by Byon Garrabrant, N6BG. More information can be found at the Byonics web site at http://www.byonics.com.

Acknowledgments

Thanks must go to all those who helped test TinyTrak3: Stephen H. Smith, WA8LMF; Doug Spreng, W7MCF; Jim Chandler, N0VH; Mike Connors, KI7AB; Michael Pendley, K5ATM; Tom & Beth May, KA3GYI & N3QAF; Mike Musick, N0QBF; Mark Conner, N9XTN; Gerald Peklar, DK7XE; Matt Cline, KB8WFH; Laird Stanton, N3HLL; Ken Simmons, K9TPT; Drew Wilson, N0XU; Jesse Royall, KC5LOS; Gary Crouch, KC8NAP; Jack Smith, KE4LWT; Porter Hammer, KD7OKN; Bill Cook, K5LPS; Russ Roysden, N8NPT; Rob Potvin, KB1GNC; Greg Ansley, KB4R; Nick Jones, W7AA; and Andres Vitols. In addition, thanks to the testers and users of TinyTrakII and the original TinyTrak and my wife, Lara Garrabrant, KD6AYO. Thank you to everyone for helping make TinyTrak3 all that it is.

Construction

Schematic



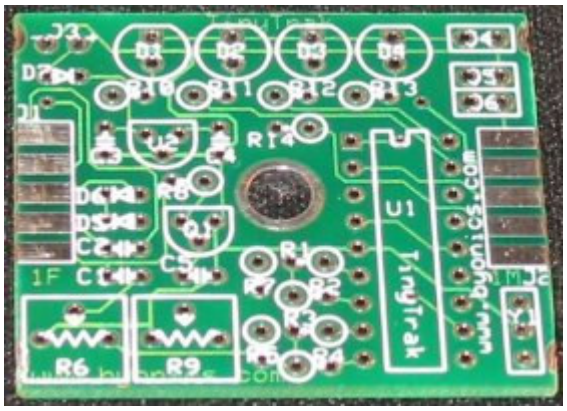
Parts List

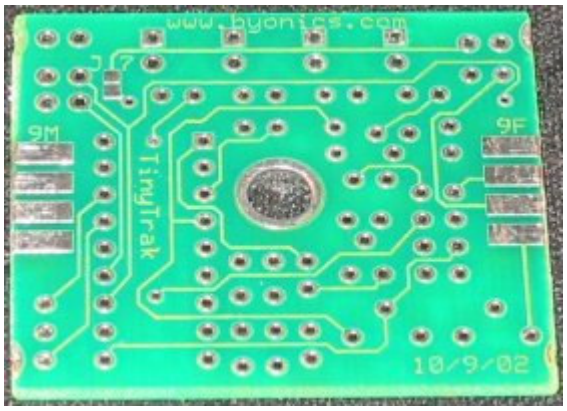
U1	PIC16F628-20/p programmed with TinyTrak3 firmware
U2	78L05 +5V voltage regulator
Y1	10 MHz ceramic resonator
Q1	2N2222A NPN transistor
R1	8.2K ohm resistor (gry-red-red)
R2	3.9K ohm resistor (org-wht-red)
R3	2K ohm resistor (red-blk-red)
R4, R10, R11, R12, R13	1K ohm resistor (brn-blk-red)
R5	220K 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, C5	0.1 uf capacitor (104)
D1, D4	Red T1-3/4 LED
D2	Yellow T1-3/4 LED
D3	Green T1-3/4 LED
D5, D6, D7	1N4148 diode
J1	Radio/Power DB9 female connector
J2	Serial/Power DB9 male connector
	18 Pin DIP socket
PCB	TinyTrak3 Printed Circuit Board

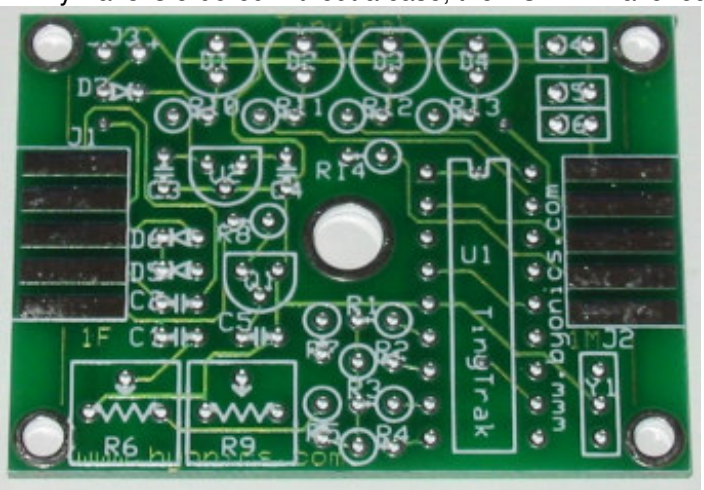
Printed Circuit Board

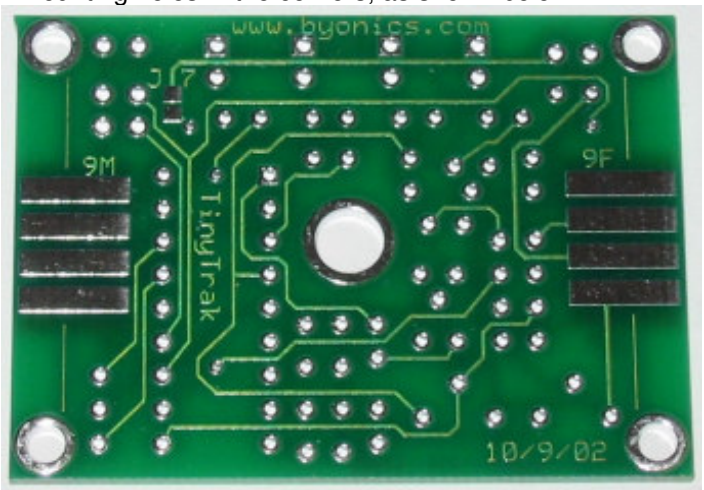
The TinyTrak3 printed circuit board (PCB) is double sided, silk-screened, and solder masked. If purchased with a case, the PCB will not have mounting holes in the corners, but will fit better in the case, similar to the images shown below.





If TinyTrak3 is ordered without a case, the PCB will have four mounting holes in the corners, as shown below.





Upgrading from an older TinyTrak

TinyTrak3 can run on the same circuit board as TinyTrak and TinyTrakII. Owners of these can upgrade to TinyTrak3 functionality by simply replacing the microcontroller with a TinyTrak3 chip, and using the TinyTrak3Config program.

Case

TinyTrak3 was designed to fit in a Serpac brand C6 plastic case (2 1/4" x 1 5/8" x 3/4"). Pre-cut cases are available from www.byonics.com. If the TinyTrak3 was ordered with a case, the PCB has alredy been sheared. If not, 0.15 inch can be cut off either end of the PCB before building, as shown above, and the DB-9 connectors will fit further inside the case, as shown in the first two images. With an unsheared PCB, the DB-9 connectors will fit more outside the case, as shown in the last image. **Note:** There are 4 flanges on the case lower center post, which should be cut off to allow the PCB to fit. The top of the PCB should be fluch with the top of the bottom mountin post. A label is also included to place on the case.





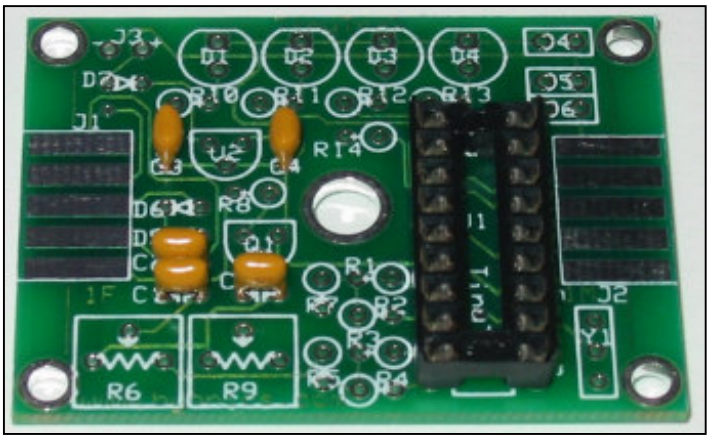


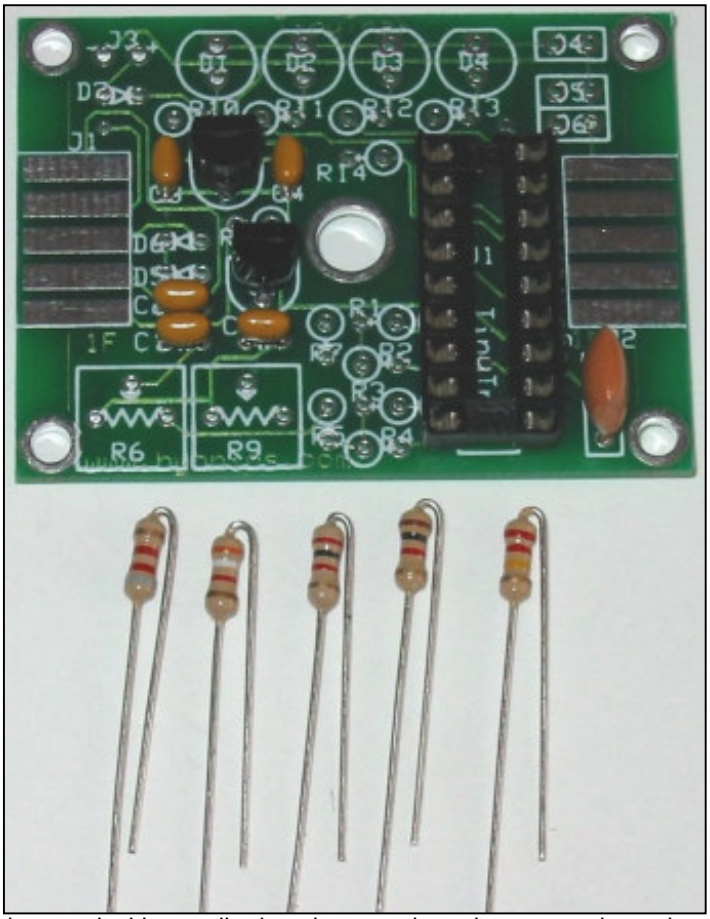
Assembly Instructions

TinyTrak3 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, and a pair of diagonal cutters. The microcontroller (U1) is static sensitive, so use standard precautions. For each item, insert on the component side (white silk-screened 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.

- ❑ Before beginning, consider how the TinyTrak3 will be mounted. If mounting with the DB-9 connectors inside the case, it will be easier to cut the PCB before inserting components. See Case section for more information.

- ❑ 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. It may help to temporarily slide J1 into position without soldering to make the socket sit flat on with the PCB. First solder just two diagonally opposite pins of the socket, and check that the socket sits flat on the PCB. Then solder the remaining socket pins. Do not insert the chip into the socket at this time.





- ❑ Install capacitors C1 – C5. (0.1uf, 104). Direction does not matter. If the leads need to be formed to 0.1 inch spacing, be careful not to bend too close to the yellow ceramic material. Bend the leads apart slightly after insertion to prevent them from falling out when the PCB is upside-down for soldering.

- ❑ Install transistor Q1 (2N2222A), which provides Push-To-Talk (PTT) for the radio transmitter. Be sure to orient flat side to match the flat side shown on the silk-screen pattern. Q1 is just above C5.

- ❑ Install voltage regulator U2 (78L05). Be sure to orient flat side to match the flat side shown on the silk-screen pattern.

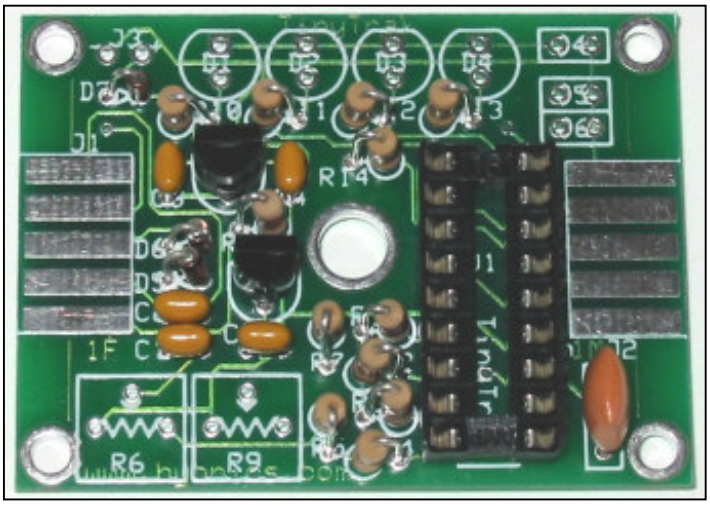
- ❑ Install ceramic resonator Y1 (10 Mhz), which provides the clock oscillation for the chip. Direction does not matter.

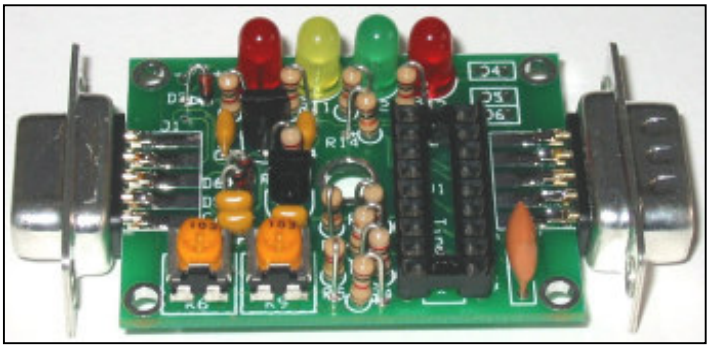
- ❑ Install resistors R1 (8.2K gry-red-red), R2 (3.9K, org-wht-red), R3 (2K, red-blk-red), R4 (1K, brn-blk-red), & R5 (220K, red-red-yel) which create the 4–bit digital to analog resistor ladder. For each resistor, bend one of the leads 180 degrees at the bulb of the resistor so that both leads are parallel and 0.1 inch apart. See the image to the right. Install vertically on the board. Polarity does not matter. Bend the leads apart once inserted to hold in place.

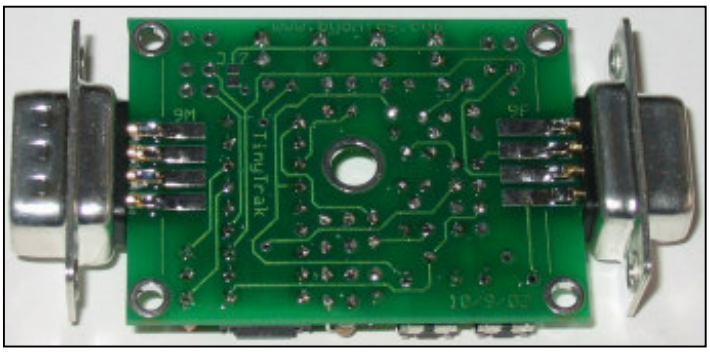
- ❑ Install resistors R7 and R14 (10K, brn-blk-org) using resistor instructions above.

- ❑ Install resistor R8. (2.2K, red-red-red). Use instructions for resistors above. Note: If TinyTrak3 will be used with a radio that does not key via current through the microphone 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 resistors R10 – R13 (1K, brn-blk-red). These limit the current flowing into LEDs D1 – D4.
- Install glass diodes D5 & D6. (1N4148) The black stripes should be nearest J1. Bend like resistors.
- Install glass diode D7. (1N4148) The black stripe should be nearest LED D1. Bend like resistors.
- Install potentiometer R6, (10K, 103) which adjusts audio output level.
- Install potentiometer R9, (10K, 103) which adjusts carrier detect sensitivity.
- Install Power Display LED D1 (red). This LED will light when device is powered. Be sure to align the flat side of all LEDs with the shape on the silk-screen, nearest the top edge of the PCB. The short lead on an LED is nearest the flat side.
- Install Carrier Detect LED D2 (yellow). This LED will light constant when audio is detected from the radio receiver, and flash during the quiet time before transmitting.
- Install GPS State LED D3 (green). 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 D4 (red). This LED will light when the radio is keyed via PTT.
- Install female DB-9 radio connector J1. This is the connector with the holes. Wedge the board in between the two rows of solder cups, with the 5 pins aligned with the 5 pads on the PCB. The connector must be completely against the PCB edge. Solder just one pin first, and insure the connector is on straight. Then solder on both sides of the PCB, all nine cups to the pads.





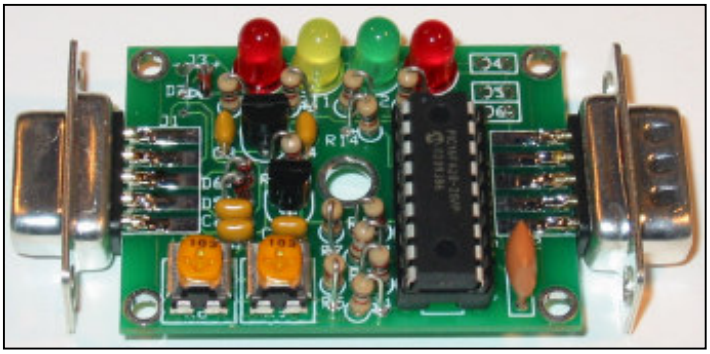


- Install male DB-9 serial connector J2 as you did for J1. This is the connector with the pins.

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. Use an ohmmeter to be sure power (socket pin 14) and ground (socket pin 5) are not shorted, and have about 3K ohms across them. You can then also provide power to the board (see below), and confirm +5 volts between socket pins 14 and 5. Pin 1 is nearest R14, pin 9 is nearest R4, pin 10 is nearest Y1, and pin 18 is nearest J6. If the board looks ready, complete the assembly as follows.

- Insert programmed microprocessor U1. The chip is static sensitive, so ground yourself by touching a large metal object before touching the chip. 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 the LEDs). An improperly inserted chip may become permanently damaged. When powered, the green and yellow LEDs should flash 3 times, which means the TinyTrak3 is running.



Interfacing

The following are the interface connections for the TinyTrak3, which are needed before operation.

Radio – J1

Female DB-9 connector J1 is used to interface TinyTrak3 to a radio transceiver. It is compatible with the radio connector on Kantronics TNC, such as the KPC-3. Connect AUDIO OUT (J1 pin 1) to the radio's mic input. If the transmitter transmits when the microphone input is grounded (most handheld (HT) radios do, except the Kenwood brand), resistor R8 must be installed, but PTT OUT (J1 pin 3) will not need to be connected to the transmitter. For all other transmitters, PTT OUT (pin 3) will be needed, and should be connected to the transmitter's PTT input. PTT OUT is grounded when the transmitter is to be keyed. To prevent transmissions over other stations, connect the receiver's squelched audio out (earphone) jack to the AUDIO IN (pin 5). Also connect J1 GROUND (pin 6) to the radio's ground. Note: TinyTrak3 cannot decode the incoming packet data. Instead, it just watches for audio energy (voice, data, or static) to hold off transmitting over other stations. Refer to the transceiver's manual for more information, and look for a section on installing a terminal-node controller (TNC) for packet operation, as TinyTrak3 is interfaced in a similar manner. J1 can also be used to supply or receive TinyTrak3's power, via pin 7 and pin 6.

J1 also provides a PTT INPUT (pin 8) to allow TinyTrak3 to transmit a data burst after the microphone is unkeyed after voice traffic. This input should be grounded when PTT is pressed, and floating when PTT is released. This input is not needed for normal operation. J1 also can optionally interface to a radio's carrier detect output via the CARRIER DET input (pin 2). To use this option, CARRIER DET should be grounded when the channel is busy. If the radio asserts CD output high when the channel is busy, use the Invert CD option in the configuration program. In addition, SW1 is available on J1 via pin 4. Some example radio interface diagrams are available at http://www.byonics.com/tinytrak/wiring.php and at http://www.packetradio.com/tnc2rad.htm.

Serial – J2

TinyTrak3 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. The male DB-9 J2 serial connector is used for this purpose. J2 pin 3 is used to transfer serial data from the TinyTrak3 to the computer. J2 pin 2 is used to transfer serial data from the computer or GPS to the TinyTrak3. J2 pin 5 is serial ground. Both a gender-changer (female-to-female) AND a null-modem adapter will be needed to interface the computer to TinyTrak3. A null-modem adapter swaps pins 2 and 3, and connects pin 5. Use a DB-9 serial extension cable if it is difficult to connect the gender-changer, null-modem adapter, and TinyTrak3 directory to the computer 9-pin serial port. Due to the TTL voltage levels used, some laptop computers may not be able to communicate with the TinyTrak3.

If using a GPS that plugs directly into a computer serial port, that GPS can be plugged directly into TinyTrak3's J2 serial connector. 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 female DB-9 with GPS serial data out wired to pin 2, and ground to pin 5. GPS serial input is not used. J2 can also be used to supply or receive TinyTrak3's power. If this is desired, short J7 on the back of the PCB by dropping a bead of solder across the two square pads. After this, J2 pin 4 will be connected to J3 and J1 pin 7. 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.

Power – J3 (also J1 and J2)

TinyTrak3 must be powered with an external source of 7-35 volts DC, such as a 9-volt battery, or a 12-volt cigarette lighter plug. TinyTrak3 is NOT powered via a computer's serial port. Power can be applied via J1, J2, or J3. J1 is the most common way to power the TinyTrak3. To use J1, connect pin 7 to positive voltage and pin 6 to ground. To use J2, connect pin 4 to positive voltage and pin 5 to ground. Also, short J7 on the back of the PCB by dropping a bead of solder across the two square pads. To use J3, apply positive voltage to the plus (+) hole, nearest the LEDs, and ground to the minus (-) hole. Only one of the three jacks J1, J2, and J3 should be used to supply power to the TinyTrak3. If power is applied via J2 or J3, the same power will be available on J1 to power a radio. If power is supplied via J1 or J3, the same power can be retrieved via J2 to power a GPS, if J7 is shorted. Be sure not to draw more current than your supply can handle. 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.

LED Control – J4

This optional jumper is normally bridged on the PCB, but if desired, can be cut, and replaced with a true jumper. This will allow easy disconnection of the LEDs if current saving is important. If desired, cut the trace between the two J4 holes, and insert a 1x2 header.

Primary/Secondary Switch SW 1 – J5

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 Carried Detect LED (D2) will blink rapidly for a short time, and then a transmission will be sent. SW1 is also available on J1 pin 4. This switch input is optional, and can be left unconnected.

Power Switch – J6

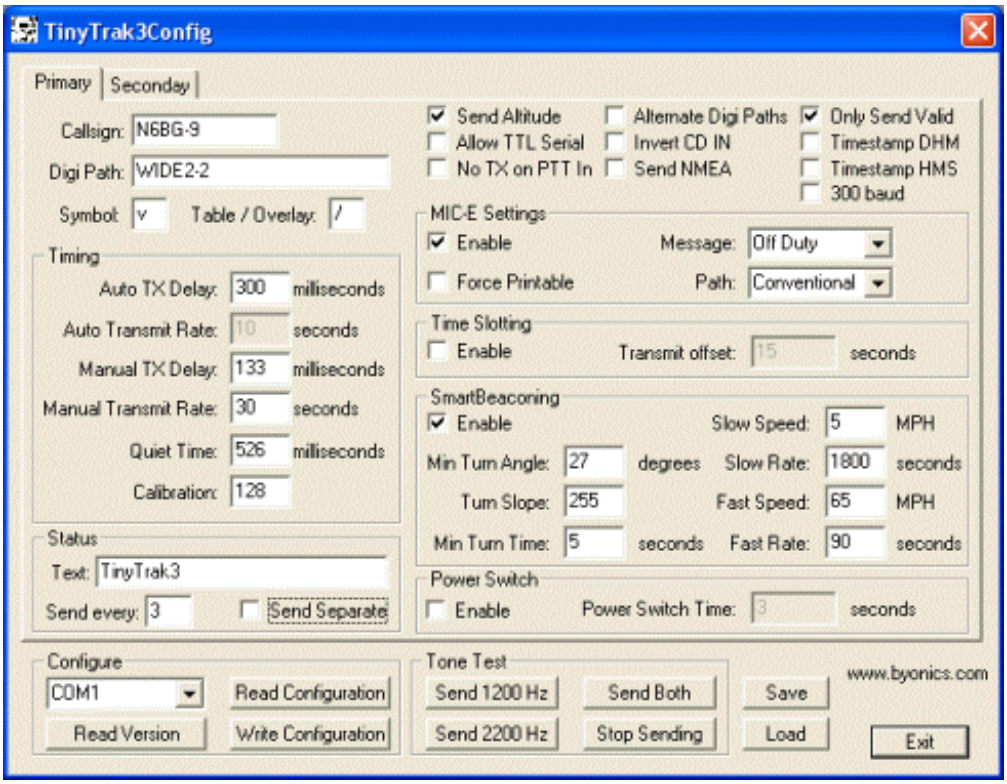
J6 can be used to control power to the radio and/or GPS. TinyTrak3 can be configured to assert J6 with 5V just before a transmission is sent. After the transmission, J6 will drop back to 0 volts until the next transmission is ready. Be sure to choose a relay that can handle the power requirements of the radio and GPS. A Radio Shack 5VDC Reed Relay (275-232) can be used to control power to a radio that draws less than 1 amp. J6 can only supply 25ma. This output is optional, and can be left unconnected. See the Power Switch section in the Configuration Software.

Serial Power Enable – J7

To bridge serial power J2 pin 4 to TinyTrak3 power J3 and J1 pin 7, this jumper on the bottom of the PCB should be shorted by dropping a bead of solder across the two square pads. This will allow the GPS to receive power from the TinyTrak3 power source, or let the TinyTrak3 be powered from the serial connector. If serial power is not needed, leave this jumper open.

Operations

Configuration Software



To set user options, such as call signs and transmit rates, use the TinyTrak3Config.EXE program, available from www.byonics.com. First, power TinyTrak3 via an external battery or other power source and connect it to a computer serial port. The computer serial port wil NOT power the TinyTrak3. Be sure to use a gender-changer, and a null-modem adapter. Launch TinyTrak3Config.EXE, and select the connected serial port. Click "Version" to confirm the config program can communicate with TinyTrak3. 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, simply try again. Detailed information about each configuration parameter is included below.

Primary / Secondary

These tabs select whether the 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 be an amateur radio call sign such as N6BG, or a tactical call such as BUS. 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 compliance with FCC identification requirements. As with normal packet radio, an optional SSID between 1 and 15 can be included, such as N6BG-2 or BUS-15.

Digi Path

This optional path will allow the transmission to be repeated by digipeaters. It should be entered in the form of call signs or aliases with optional SSIDs, separated by commas, such as RELAY,WIDE,WIDE. WIDEn-n format is also supported, such as WIDE3-3. More call signs in the digipeater path will limit the space available for the status beacon.

Symbol & Symbol Table / Overlay

These settings set 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 character. Use / for the primary symbol table, use \ for the alternate symbol table, and use a number (0-9) or a letter (A-Z) for an overlay on 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, and http://www.jarviscomputer.com/jim/aprs-symbols .

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

Timing

Auto TX Delay

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.

Auto Transmit Rate

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

Manual TX Delay

This sets the delay in milliseconds after the transmitter is keyed, until the data begins when a transmission is manually triggered via PTT IN. It should be set to a value less than the Auto TX Delay when using PTT IN for data burst after voice. If PTT IN is not used, the setting doesn't matter.

Manual Transmit Rate

This setting controls how often, in seconds, a manual position transmission can occur. If the PTT IN line is grounded, and it has been at least this long since the last transmission, TinyTrak3 will transmit. CD input is ignored when checking for manual transmissions.

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

Text

TinyTrak3 can send a status text / comment message after periodic location transmissions. This setting sets the beacon text. A fixed position can be entered here in the form "!3612.34N/11512.34W-". This message will be sent, even if no GPS is connected. The number of digipeaters in the path limits the length of the beacon message. 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. The length of the text is also reduced by 2 or 4 characters if TimeSlotting is enabled, and 10 or 20 characters if SmartBeaconing is used.

Status Beacon 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 transmission.

Send Separate

If this option is selected, the status text will be sent as a separate packet, right after a position packet, rather than being appended to the end of a position packet. This may allow improved status decoding by web sites such as www.findu.com.

Checkbox Options

Send Altitude

Clearing this setting will disable the TinyTrak3 from transmitting altitude information. If this option is enabled, and the TinyTrak3 receives a $GPGGA message from the GPS, altitude information will be sent. This option defaults to enabled.

Alternate Digi Paths

This option makes TinyTrak3 alternate between the primary callsign and digipeater path and secondary callsign and digipeater path with each transmission. It can be useful to use two different digi paths when the best digi path is not known.

Only Send Valid

Setting this option will disable TinyTrak3 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 TinyTrak3. 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.

Allow TTL Serial

If this option is enabled, TinyTrak3 will check the serial input level at power-up to determine if it is connected to a RS-232 or TTL level GPS. If not using a TTL level GPS, this option should be left disabled. If enabled, and TinyTrak3 detects a TTL GPS, it will flash the yellow and green 3 additional times at powerup. This will invert the polarity of the GPS input signals.

Invert CD IN

If this option is enabled, the sense on the CD input will be inverted. This is used when connecting a radio's carrier detect output out to J1 pin 2.

Timestamp DHM

If this option is enabled, TinyTrak3 will send a timestamp with the day, hour, and minute with all APRS format packets. MIC-E format does not support timestamps, so if MIC-E output is enabled, this option is ignored.

No PTT Out while PTT In

If this option is enabled, PTT Out will not be asserted when PTT In is detected. This can be used to enable the use of a single wire for PTT Out and PTT In. Replace D6 with a shorting jumper, and check this box, and TinyTrak3 can monitor the PTT Out line for PTT In.

Send NMEA

If this option is enabled, packet data will be sent out in the form of a NMEA –0183 1.5 $GPRMC sentence, rather than APRS text format. The last two fields will be left empty. In this mode two additonal digits for latitude and longitude will be sent, giving much higher resolution. Altitude will not be available when sending NMEA. An example NMEA packet looks like this: $GPRMC,201050,A,3610.9912,N,11516.4034,W,000,005,240799,,*0B

Timestamp HMS

If this option is enabled, TinyTrak3 will send a timestamp with the hour, minute, and seconds with all APRS format packets. MIC-E format does not support timestamps, so if MIC-E output is enabled, this option is ignored.

300 Baud

If this option is enabled, TinyTrak3 will send data suited for HF transmissions. Data will be sent at 300 baud, rather than at 1200 baud, and the tones will be 1600 Hz and 1800 Hz rather than 1200 Hz and 2200 Hz. The Tone Test buttons below will send the new tones when in 300 Baud mode. The 30-meter HF/FSK APRS frequency pair in the United States is 10.149.200 / 10.149.400 MHz. To transmit the correct frequency pair with the TinyTrak's 1600 / 1800 audio tone pair, set your transceiver to 10.150.00 LSB or to 10.147.60 USB.

MIC-E Settings

The 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, bearing and altitude. The MIC-E standard does not allow timestamps. Since it is shorter than other packets, it is more likely to be received without error. More information about this format can be found in the APRS documentation at TAPR's web page (http://www.tapr.org).

MIC-E Enable

This setting enables the MIC-E format. If the MIC-E format is not selected, APRS text format will be used. This option defaults to enabled. When using the APRS text format, packets more human readable, and are addressed to "APT310" which signifies a APRS tracker type TinyTrak3, version 1.0.

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 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 one of the 16 pre-assigned MIC-E paths. In order to use the digi path referenced above, this should be set to "Conventional".

Time Slotting

If enabled, the transmit offset setting sets the time from the start of the hour that a transmission will occur. After that transmission, transmissions will continue at the Auto Transmit rate until re-syncing at the next start of the hour. This option is useful to preset transmission times for multiple trackers. The Transmit Offset value must be less than the Auto Transmit rate. TinyTrak3 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 TinyTrak3 begins transmitting in its assigned slot. TinyTrak3 will transmit at the Auto Transmit rate, even if it has not yet synced. If the Auto Transmit rate evenly divides into 600 seconds, TinyTrak3 will synchronize at every 10 minutes. If the Auto Transmit rate evenly divides into 60 seconds, TinyTrak3 will synchronize at every minute. Time slotting and SmartBeaconing are mutually exclusive. It is not important to sync every minute, because TinyTrak3 timing doesn't drift very much over an hour.

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. In order to use SmartBeaconing, the GPS must send speed and heading information with the GPRMC sentence. 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 / speed;    // 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 http://www.hamhud.net . SmartBeaconing and time slotting are mutually exclusive: The Auto Transmit rate is ignored when SmartBeaconing is enabled.

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 TinyTrak3 will transmit due to turning. Both Min Turn Time and Min Turn Angle must be less than 256.

SmartBeaconing Slow Speed, Slow Rate

This setting sets the SmartBeaconing slow speed (MPH) and rate (seconds). Speeds slower than the slow speed will cause TinyTrak3 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 TinyTrak3 to beacon at the Fast Rate. Fast Speed must be set higher than Slow Speed.

Power Switch

When enabled, this setting will enable Power Control J6 at the given time before a transmission will occur. If Only Send Valid is enabled above, and the Power Switch is also enabled, and GPS data is not valid at the scheduled transmission time, power will remain on until the next transmission to give the GPS a chance to lock on. If this happens, the Power Switch time should be increased. See the Power Switch section above for more information.

Configure

Com Port

This setting selects which communication port the TinyTrak3 is connected to during configuration. Only the first eight serial ports (COM1 – COM8) are supported.

Read Version Button

This button will get the current firmware version from the connected TinyTrak3. This is a useful, simple test to confirm the computer and TinyTrak3 can communicate.

Read Configuration Button

This button will upload all configuration settings from the connected TinyTrak3, and populate the settings field in the software interface. This step is useful to change a single setting on the TinyTrak3 configuration.

Write Configuration Button

This button will download all configuration settings entered in the software interface into the TinyTrak3. The settings are stored in non-volatile memory, so they will persist when power is removed from TinyTrak3.

Tone Test

These buttons will cause TinyTrak3 to transmit a 1200 Hz tone, a 2200 Hz tone, or alternate between the two on a connected radio. When in 300 baud mode, the tones sent are 1600 Hz and 1800 Hz. This is useful for setting transmission levels, and measuring audio frequencies when adjusting calibration.

Save / Load

These buttons will allow saving and loading of all current TinyTrak3Config parameters to and from a file. The file format is the binary data stored in the TinyTrak3 on-chip EEPROM.

Windows XP Issue

Some users have reported a problem using the configuration software under Windows XP. The problem involves entries in the config dialog that contain a single character or digit being changed or reset to 0 when writing or changing between Primary and Secondary configurations. If you experience this problem, the work-around is to make sure no edit box entry field contains a single char or number by entering characters twice in the Symbol and Symbol Table fields (>> instead of >), and add leading zeros in the numeric fields (03 instead of 3).

TinyTrak3 Adjustment

There are only a few adjustments required for proper operation of TinyTrak3. First, the transmit audio level should be adjusted at R6 for proper deviation. Use the transmit tones buttons (1200 Hz/2200 Hz/Send Both) in the configuration software to cause the TinyTrak3 to transmit while adjusting. When in 300 baud mode, the tones sent are 1600 Hz and 1800 Hz. 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. TinyTrak3 was designed for hand held radios. Some mobile radios require more audio drive than TinyTrak3 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 or shorting jumper. This should allow for about double the audio range.

The next adjustment point is R9, 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. As R9 is rotated from one extreme to the other, U1 pin 10 should swing from 0 volts to 5 volts.

The last adjustment is the calibration value. This adjusts timing slightly to compensate for ceramic resonator error. Try a few different values while monitoring the decoding on a TNC, and set to the value which gives the best decoding. Usually the default 128 works best.

After setting the desired configuration options, TinyTrak3 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.

Troubleshooting

Problem	Possible Solutions
The TinyTrak3Config.exe program cannot communicate with the TinyTrak3. The config program reports "TinyTrak3 not found".	Be sure to use a null-modem adapter. Be sure TinyTrak3 is powered via an external power supply, such as a 9V battery. Be sure to use the correct configuration program, TinyTrak3Config.exe. Test serial by running a terminal program, such as HyperTerm at 4800 baud N81. When TinyTrak3 is first powered up, it will send a version message. If it doesn't, serial output is not working. If it does, send ESC V, which will make it send the version again. If it doesn't, serial input is not working. Try a different computer.
The TinyTrak3Config program reports "Cannot open COM port"	Be sure to use the correct COM port. Be sure no other programs are also currently using the COM port.
The transmitter keys up, even when no power is applied to TinyTrak3	Perhaps the radio doesn't key via current through the microphone line. If so, resistor R8 should be removed. Check for shorts or opens on the solder side of the PCB. Check the radio manual. Perhaps a different value of R8 is needed.
The green GPS status LED flashes, but never goes solid.	The GPS is not getting a lock on the satellites. Make sure the GPS antenna has a clear view of the sky. Set the GPS to NMEA-0183 1.5 at 4800 baud.
The power LED D1 is not lit.	Check for 5V across LED D1. Check for shorts or opens on the solder side of the PCB. Be sure the LEDs are inserted in the proper direction. Confirm the battery is fully charged, and delivering 7 – 35 volts.
The green and yellow LEDs do not flash 3 times at powerup.	Be sure the LEDs are inserted in the proper direction. Check for shorts or opens on the solder side of the PCB. Check for 5 volts across socket pins 5 and 14 with the PIC removed.
TinyTrak3 keys the transmitter, sends a packet, but fails to un-key the transmitter.	TinyTrak3 is probably receiving too much local RF energy. Lower transmitter power, move the transmitting antenna further away, or try shielding the TinyTrak3 in a metal case. Try re-configuring the settings.
Sent packets sound distorted on a receiver.	Be sure the calibration value is around 128. Be sure there is not too much RF energy near the TinyTrak3.
Sent packets are not being decoded by a receiver (TNC, digipeater, etc)	Make sure your TXD time is high enough. Start it high (500 ms) and lower it as long as it keeps working. 250ms – 300ms is a good setting. Make sure the transmitted audio level is correct. It should be near, or just below, other transmitters received. Adjust with R6 Try slightly different calibration values. Try re-configuring the settings.
The yellow CD LED always blinks about once per second.	Serial data may be creating noise on the received audio line. Try moving the cables around, shortening and separating them, or switching to shielded cables.
No audio is heard on a receiver.	Check for shorts or opens on the solder side of the PCB. Raise transmit audio level with R6. If R6 doesn't give enough range, try lowering the value of R5, or even shorting it.
TinyTrak3 is stuck in transmit mode, or transmitting too often, and it cannot be re-configured.	Rotate R9 to the extreme so the TinyTrak3 will think there is a carrier. Then cycle power, and it should not transmit. It should then be able to be configured.
The green GPS LED is off with a GPS connected.	Be sure the LEDs are inserted in the proper direction. Be sure the GPS has power and is set to output NMEA-0183 1.5 at 4800 baud. Be sure there is not a null-modem adapter in line with the GPS. Check for shorts or opens on the solder side of the PCB. The GPS serial levels may not be readable by TinyTrak3. To test, connect the GPS to a computer running a terminal program at 4800. Don't forget the null-modem and gender-changer. Capture some of the data. Then connect the TinyTrak3 to the computer, and send the data back. If it reads the data and lights the LED, the data is valid, but the levels are not compatible.
TinyTrak3 never transmits.	Be sure the yellow LED is off. Adjust R9 if necessary. Check for shorts or opens on the solder side of the PCB. Be sure the cable is built correctly. Grounding PTT should make the radio transmit.

Frequently Asked Questions

Is the firmware .HEX file available so I can burn my own chips?
No. The only way to get a TinyTrak3 chip is preprogrammed from www.byonics.com.

Can you add feature XXX to TinyTrak3?
I don't know. Ask at tinytrak@byonics.com. Many of the features of TinyTrak3 came as requests from users. However, only features that are desired by a reasonable number of users are likely to be added. Code space, data space, and programming effort required must also be considered.

Can I get the source to TinyTrak3 so I can add my own features?
No, the source to TinyTrak3 is not public. However, consider Son of TinyTrak (SOTT), available at the TinyTrak Yahoo! Group (http://groups.yahoo.com/group/TinyTrak/). It is an open source project that runs on the TinyTrak3 hardware. Also look at the PIC-E programs available at ftp://ftp.tapr.org/picsig/software/.

What language is TinyTrak written in?
All TinyTrak programs were written in PIC assembly.

Can the TinyTrak3 implement a true DCD, output received packet data to the serial port, or be a digipeater?
No. That would require a modem chip to decode the actual packet tones.

Can I hook a weather station to the TinyTrak3 to send APRS weather data?
No. TinyTrak3 only receives GPS NMEA data. However, WXTrak will interface to weather stations and send the data. Details are at www.byonics.com

Is there a surface mount TinyTrak3 available?
Yes, It is now available at the TinyTrak website.

Can I replace the ceramic resonator Y1 with a crystal?
Sure. You should add two small capacitors (about 6 pf) from each lead to ground, also.

TinyTrak3 seems to have a problem sending speeds greater than 100 MPH.
TinyTrak3 sends speeds greater than 100 without a problem. However, there appears to be a bug in the Kenwood D7A radios decoding MIC-E format packets with a speed greater than 100 MPH.

I want to sometimes use TinyTrak3 with an HT, and sometimes with a mobile radio. What should I do about R8?
Leave R8 off the board, and then add it back into the cables for the radios that need it (like HTs). All R8 does is go between the PTT OUT line and the AUDIO OUT line, so it is very easy to add into a cable. Then, for the mobile, run the PTT line separate.

Is the TinyTrak3Config program available for DOS, Palm, Linux, Mac, etc?
There are currently versions for Windows, Mac OS and Palm OS. The Windows version can be found at http://www.byonics.com/tinytrak/tinytrak3.zip. The Mac OS version, by Mark Dieter, N2BLI, can be found at http://mdco.net/irving/tinytrak.html. The Palm OS version, by Axel Fischer, DL3OBP, can be found at http://www.afischer-online.de/pilot/ttconfig/index.html. The Windows version should run under Linux with WINE. If you would like to receive information on writing your own config program, contact tinytrak@byonics.com.

What is APT311?
In some output mode, the desination callsign of the send packet is set to APT311. This is just used for indentification. It signifies that the packet was send from a APRS (AP) TinyTrak3 (T3) version 1.10 (11).

I have other questions. Where do I go?
There is a TinyTrak mailing list at Yahoo! Groups. (http://groups.yahoo.com/group/TinyTrak/). There are over 500 members who have probably already asked and answered your question. Check out the message archive. The TinyTrak3 Links web page (http://www.byonics.com/tinytrak/links.html) has useful information. Questions can also be addressed to tinytrak@byonics.com

Hints, Tricks, & Notes

- TinyTrak3 purchased already built will include R8. If R8 is not needed, it doesn't need to be completely removed. It can be clipped with diagonal cutters at the top of the bend in the resistor. Just separate the two ends so they do not connect.
- There are 4 flanges on the lower center mounting post of the TinyTrak3 case. These need to be trimmed off so the TinyTrak PCB top is flush with the top of the post.
- The TinyTrak3 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. An RS-232 level converter, such as a MAX232, may be used, but the Allow TTL serial should be enabled.
- TinyTrak3'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.
- TinyTrak3 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 2 of TinyTrak3's DB-9 connector. This way, with no PC connected, the GPS serial data will feed into the TinyTrak3, 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 configuration settings, since the TinyTrak3 will be too busy transmitting to monitor serial. To resolve this, turn the R9 pot such that the CD LED turns on. This will prevent TinyTrak3 from transmitting, and all the new settings can then be written.
- TinyTrak3 was designed for hand held radios. Some mobile radios require more audio drive than TinyTrak3 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. R5 can even be replaced with a jumper (0 ohm resistor) for maximum audio.
- TinyTrak3 can be affected by local RF energy. To avoid this, run the transmitter at low power and keep TinyTrak3 away from the transmitting antenna. It may also help to use shielded cables and case, and toroid filters or ferrite beads on all wire going to TinyTrak3.
- Some GPS receivers may send too much data for the TinyTrak to decode properly. If this is the case, try limiting the sending to once every 2 seconds, and/or only send out GPRMC and GPGGA sentences.

