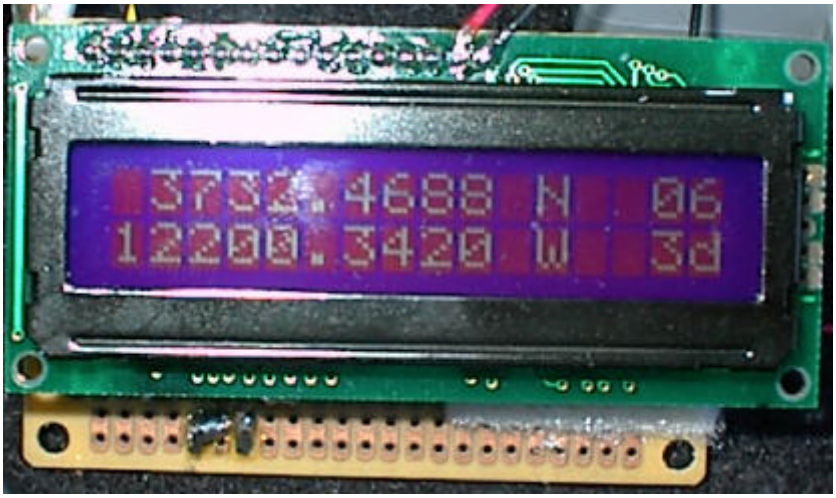


Figure 1 Prototype

GPSINIT/Tiny Track
SBARA Workshop
Feb 24, 2001

James Lee N1DDK

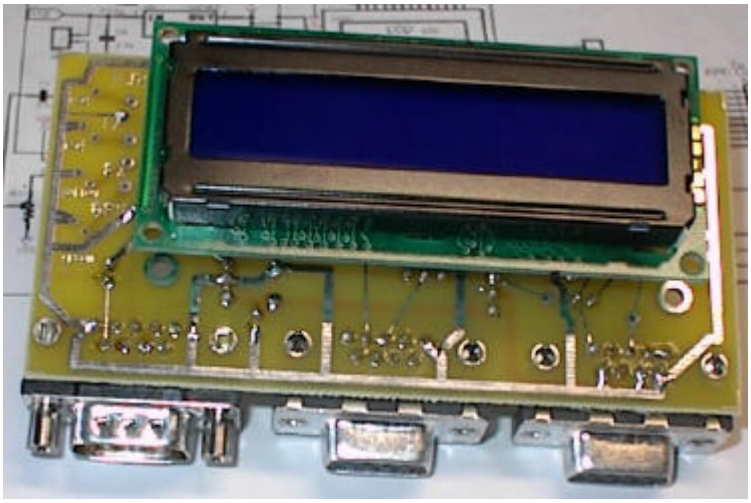


Figure 2 Workshop kit

Introduction

This unit provides Five functions:

1) It converts the el-cheepo GPS to NMEA mode.
2) It is a Transmit only TNC for APRS.
3) It simplifies and consolidates the cabling for a "Tracker box".
4) It provides a simple display for the GPS.
5) It allows the MiTac to be used with other computer programs such as Delorme Street Altas and Microsoft Streets and Trips.

By the way this will work for the MiTac GPS and any other SiRF based GPS.

How does it work?
With the help of AD6IL, we figured out how to re-initialize SiRF into NMEA mode. The software for converting the SiRF GPS into NMEA was ported to a Microchip PIC. I added display functions to check the data. A 4 line LCD displays the position and number of satellites received.

The TNC functions are provided by a second PIC. Basically the TinyTrak, see: http://www.byonics.com/ has been added to the board.

Overall cabling is simplified by using three dedicated connectors with standard connections. Once connector is for the GPS. It can provide power to the GPS, simplifying power cabling by allowing the GPS power cord to be cut off. A second connector is for an optional PC. The PC is used for initial programming, and can be used to receive the NMEA strings for use with your favorite mapping software. A third connector is for the radio connections. The connector is Kantronics compatible, so you do not need to make radio cables if you would prefer to buy cables. Internal jumpers provide for a superset of Kantronics functionality. A jumper combines the audio and PTT functions for simple connection to a HT. The Radio connector also allows for the Mike-E PTT input function or power input functions. If you provide power on the radio connector, connections are as simple as just plugging in the three cables. +7 to +15 volts can be provided either on the radio connector or via a dedicated power cable.

Acknowledgments

I would like to thank Byon Garrabrant, N6BG for the TinyTrack, Jim Wood, KE6IVA for inspiring be with the APRS workshops, Al Rendon WT6K, for coordinating the SBARA workshops, and Alan Kittrel, KE6YKK, and JP Rouland KE6UXM for their help in reviewing the board design.

Parts List

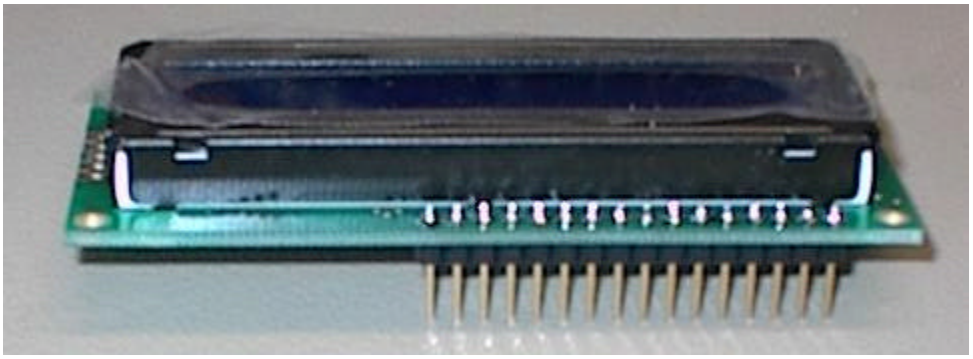
Item	Qty	References	Value	Description
1	2	U1, U2	18 pin socket	
2	6	C1,C2,C3,C4,C6,C7	0.1u	104
3	1	C5	47uf	
4	1	LCD1	LCD	2 line, 16 character, backlit
5	5	R2,R7,R8,R9,R14	1K	Brown Black Red
6	1	R4	8.2K	Grey Red Red
7	1	R5	3.9K	Orange White Red
8	1	R6	2K	Red Black Red
9	1	R10	220K	Red Red Yellow
10	1	R13	2.2K	Red Red Red
11	2	R1,R3	10K	Brown Black Orange
12	1	R15	15 ohm	Brown Green Black
13	1	Q1	2N2222	
14	2	R11,R12	trimmer-10K	
15	1	D4	1N4148	
16	1	Y1	10MHz	10.0
17	1	Y2	4MHz	4.0
18	3	P4,JP2,JP3	HRD2	
19	1	JP4	HDR3	
20	1	JP1	HDR6	
21	1	U3	UA7805	
22	1	LCD1	LCD socket	
23	1	LCD1	16 pin header	
24	2	P2,P3	DB9F	
25	1	P1	DB9M	
26	2	D3,D5	Red	
27	1	D2	Yellow	
28	1	D1	Green	
29	4	standoffs 2-56x.375	stand off	
30	8	Bolts 2-56x.25	Bolts	
31	8		Lockwasher	
32	3		Shunts	
33	1	U1	PIC16F84	
34	1	U2	tinytrack	From N6BG
35	4		Jackscrews	
36	1		board	
37	1		d9 cable	Computer Cable
38	1		DB9M	For Radio Cable
39	1		hood	For Radio Cable

Assembly Instructions

1) Check your kit for a complete set of parts.

Important NOTE;
Please read all instructions.
Some parts are placed on the back of the board.

2) Insert 18 pin sockets at U1 and U2. Pin 1 of each is indicated by a square pad. The indented end of the socket goes toward pin 1. Tack or solder in place.
3) insert the 6 0.1uf capacitors at c1,c2,c3,c4,c6,c7. Note C3 is located near the top of U1. The legend on the board looks like "C8" or "CB" since a trace runs through the "3".
4) Insert the 47uf capacitor at C5. Insure that the negative (-) lead is toward the edge of the board. Solder in place, clip leads.
5) Insert 4 1K (Brown Black Red) resistors at R14, R2, R9, R8 *be sure you do not have brown-black orange*
6) Insert R10 220K (Red Red Yellow)
7) Insert R7 1K (Brown Black Red) *be sure you do not have brown-black orange*
8) Insert R6 2K (Red Black Red)
9) Insert R5 3.9K (Orange White Red)
10) Insert R4 8.2K (Grey Red Red)
11) Solder resistors in place and clip leads.
12) Insert R13 2.2K (Red Red Red)
13) Insert R3 and R1 10K (Brown Black Orange) *be sure you do not have brown-black red*
14) Insert R15 15Ohm (Brown Green Black)
15) Solder resistors in place and clip leads.
16) Insert Q1. Solder, clip, **Verify that Q1 did not short R13.**
17) Insert the Trimmers (Three lead Yellow and Black) R11 and R12.
18) Insert D4, The banded end goes away from the PIC towards the connector. Solder and clip.
19) Insert Y1 and Y2 Three leads Carmel colored. Be careful they look similar. Insert the one labeled 10.0 at Y1 and the one labeled 4.0 at Y2. Solder.
20) Insert 2 pin headers at JP2 (HT) and JP3, (GPS PWR).
21) Optionally you could put one at P4 (+12V) and then solder wires to it if you are using external power. You can just leave the P4 holes open for now. It is up to you to decide if it will be easier to connect wires to the pins or solder to the holes if you need to use this optional connector.
22) Insert the 3 pin header at JP4 (PIN 7 Option).
23) Insert the 6 pin header at JP1 (PROG/NORM)
24) Solder headers in place
25) Insert the 7805 regulator at U3. The metal tap goes away from the near edge of the board. **NOTE: If you are going to use the gps power option, you should remotely mount the 7805 on a heatsink since the GPS take about 400 ma, and the 7805 will dissipate about 5 watts.**
26) Flip the board over and insert the 16 pin LCD connector. Solder
27) Insert the 16 pin header into the LCD module. The short end of the pins is inserted into the back of the LCD module. The header should be on the back with the long leads extending from the back. Solder.



28) Insert the two female DB9F connectors P2 and P3 into the top side of the board at the RADIO and COMPUTER connections.
29) Insert the DB9M (MALE) connector into P1. The GPS connector, Solder the Connectors.
30) Plug the LCD into the socket on the back of the board. Secure with the standoffs and bolts.
31) using the height of the LCD as a guide, Insert the two RED LEDs D3 and D5 into the back of the board. Have the tip of the LED at the same height at the front of the LCD. Insure you have the proper polarity of your LEDs be insure the flat side is toward the edge of the board.
32) Insert D2 (yellow) and D1 (green) in the same way.
33) Double check the LED's and then solder.
34) Using the standoffs, 2-56 screws, and lockwashers, bolt in the LCD.
35) Insert shunt jumpers as follows:

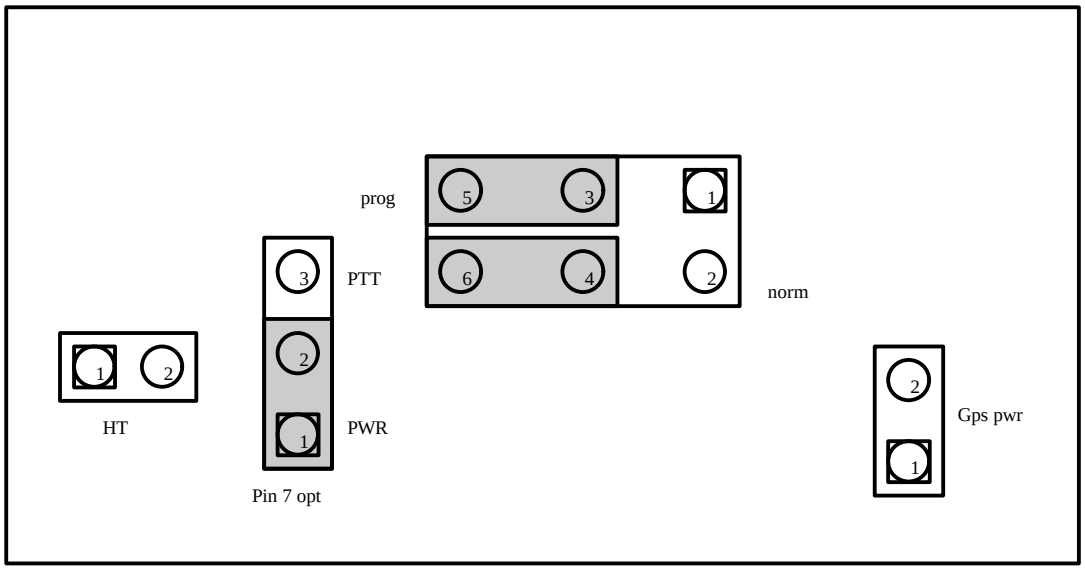


Figure 3 Jumper reference for initial programming

36) Insert U1 into its socket. Check the orientation. Pin 1 is indicated by a small dot on the chip. You may need to peel back the label.
37) Insert U2 into its socket. Check the orientation. Pin 1 is indicated by a small dot on the chip. You may need to peel back the label.
38) Screw the jackscrews into P2 and P3, the RADIO and COMPUTER connectors.
39) Use the cable to connect the computer to the computer connector.
40) Do not connect the GPS at this time.
41) Apply power.
42) Program the TinyTrack using TinyTrackConfig.exe
    Suggestions:
    Press check version first to insure the communication is working.

Set the TX delay to be over 200 ms if you are planning to us a HT for your transmitter. ICOM HT's may need a value over 300.

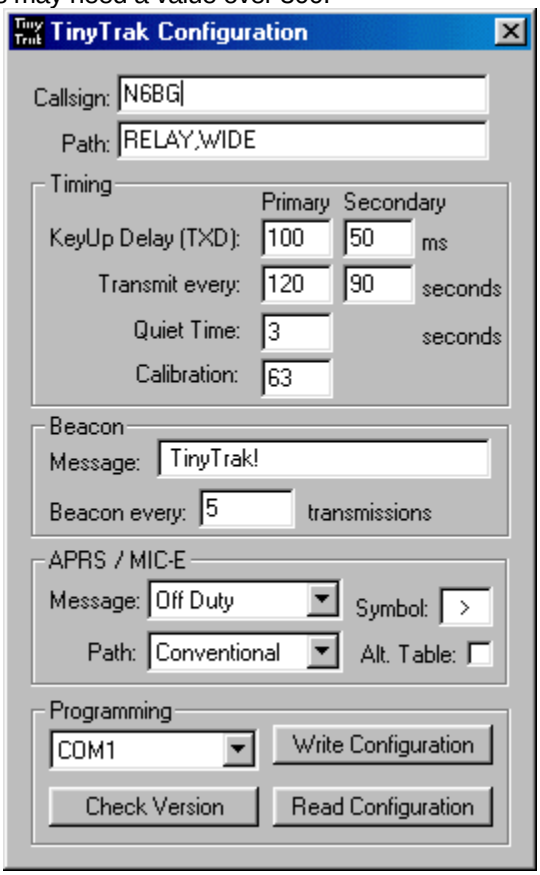


Figure 4 TinyTrack Configuration Utility

43) Power down and reset jumpers as follows:

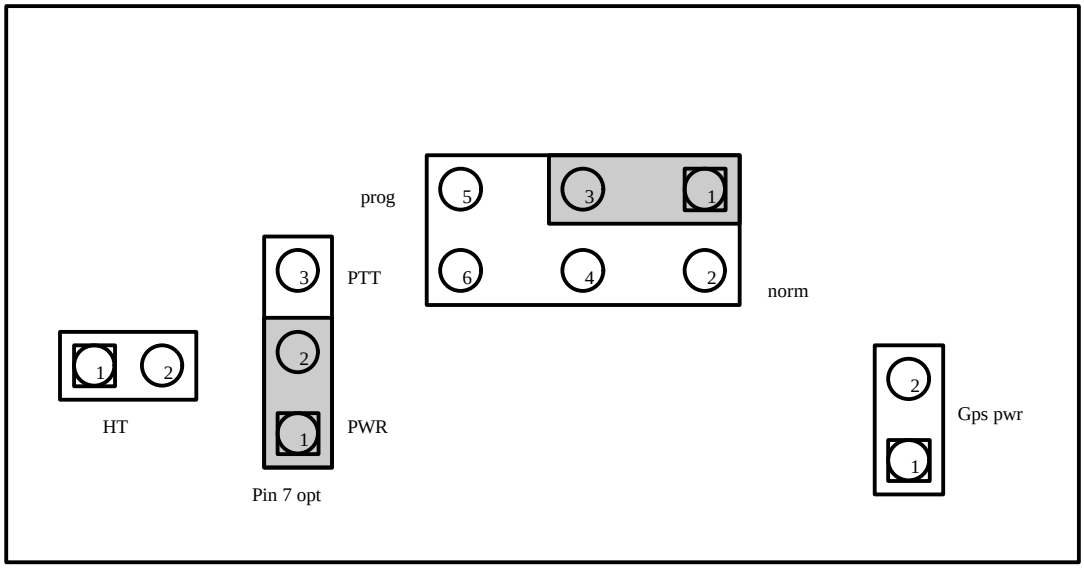


Figure 5 Jumper setting for normal use

44) The supplied connector and hood are for creating your own Radio cable.

Reference

Connectors

RADIO

This connection is compatible with the Kantronics KPC3. You can use a Kantronics cable or make your own.

Pin	Connection
1	Audio out, Connect to mic input on radio
2	No Connection
3	PTT out
4	Audio in, Connect to speaker output from Radio
5	Audio in, Duplicate connection
6	Ground
7	Optional Connection see below, PTT in or Power in.
8	Ground
9	Ground

Pin 7 is jumper select able for one of two functions. JP4 can have no jumper if neither option is used. It is recommended to simplify cabling that ping 7 be used for Power in (+12 Volts). Optionally this input could be used for MIC-E style PTT in.

COMPUTER

The computer connection is used for programming, or for monitoring your position. This is not a full TNC, so you only see your position information. The output at this connector is MNEA, so you can use any mapping software you like, such as Delorme Street Atlas or Microsoft Streets and trips.

Pin	Connection
1	No Connection
2	TXD
3	RXD
4	No Connection
5	Ground
6	No Connection
7	No Connection
8	No Connection
9	No Connection

The Jumper set at JP1 is used to select the data connections to this connector. In the programming position, the TinyTrackConfig program is run. In the Normal position, NMEA data is output.

GPS

This is the connection to the MiTac GPS. This connector can optionally provide power to the GPS, so the power cord to the GPS can be left unconnected or cut off.

Pin	Connection
1	No Connection
2	RXD
3	TXD
4	No Connection
5	Ground
6	No Connection
7	No Connection
8	No Connection
9	Optional power to GPS

The Jumper at JP3 can be closed to have the GPS powered from this connector.

P4

This is the optional power connector. Connect +7 to +15 volts for power, if you are not powering the unit through the RADIO connector.

Jumpers

JP1 6 Pin Header (prog/norm) . This jumper selects *normal* or *programming* mode for the computer connector.
Connect 1-3 for normal mode, and 3-5 and 4-6 for Programming mode.

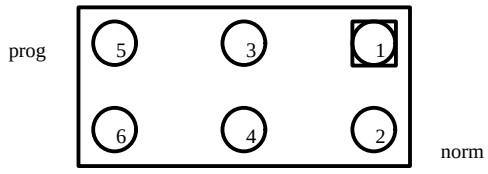


Figure 6 JP1

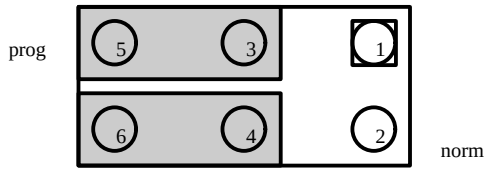


Figure 7 JP1 in Programming mode

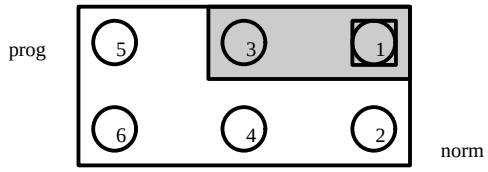


Figure 8 JP1 In Normal Mode

JP2 2 Pin Header (HT). This jumper combined the audio out and the PTT out for many HT's. Close this jumper for a HT, leave it open for a Mobile Rig.

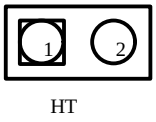


Figure 9 JP2

JP3 2 pin header (gps pwr). This jumper allows the Mitac GPS +5 line to be supplied from the tracker. Close this jumper if you wish to eliminate the need for the additional GPS power cable.

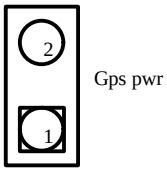


Figure 10 JP3

JP4 3Pin Header (pin 7 opt). This jumper selects the function of pin 7 of the radio connector. Connect 1-2 for power in; 2-3 for PTT in, leave open if unuesd.

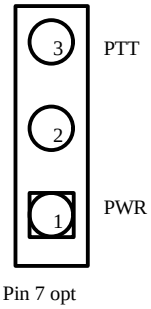


Figure 11 JP4

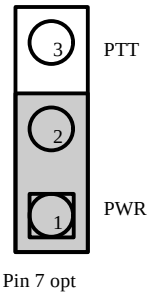


Figure 12 JP4 When using Power from the radio

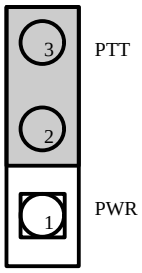


Figure 13 JP4 When used as Mic-E

TinyTrack

This section contains excerpts from the TinyTrack Documentation

Overview

TinyTrak 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, TinyTrak will key the radio at user-defined intervals, and transmit the GPS's current position.

Acknowledgments

TinyTrak was developed with the help of many individuals who deserve recognition. Randy Holland (KO6KC), Dave Lee (W6ZL), Ken Mirabella (KM6YH), Marty Mitchell (N6ZAV), Luc Bodson (ON9AAV), Walter Crauwels (ON4BCB), Vesa Kauppinen (OH4EA), Barbaros Asuroglu (TA2CBA), Klaus Hirschelmann (DJ7OO), Mårten Persson (SM7SYX), Sylvain Mercier (VE2SIL), James Gutshall (N7VHF), Anders Richardsson (SM7WGG), Allan Gibbs (G7GFU), James Gorr (N3TOY), David Inkster (ZL2BLI), Ron Graham (VK4BRG), Robert Stessel (K1WXY), Jacob Tennant (KB8QIR), Hans Meijer (PA1PG), Mike Palmer (K8LG), Alfons Wittoeck (ON4AWT), Wes Johnston (KD4RDB), Peter Mulder (PE1IEE), David Andersen (K0RX), Ernie Howard (W8EH), Bert van Dalen (PA0DAL), Ed Newman (VK4JEN), Curt Mills (WE7U), Brian DeYoung (KE4HOR), Harry Bloomfield (M1BYT), Ulrich Stolz (DJ9XB), Johan Hansson (SM0TSC), Fred Reimers (KF9GX) of FAR Circuits, and Lara Garrabrant (KD6AYO). Thanks go to all these people for helping make TinyTrak all that it is.
Note: The combined GPSINIT/TinyTrack board was done by James Lee N1DDK.

Programming the chip

The TinyTrak kit is shipped with a programmed microcontroller, so programming the chip is not necessary if you have this kit. If you do not already have a programmed TinyTrak microcontroller, you will need to burn the firmware into a blank chip. You will need a copy of the TinyTrak firmware (TINYTRAK.HEX), a Microchip PIC16F84-10/P microcontroller, and the necessary hardware and software to program the firmware into the chip. The latest firmware can be found at http://www.byonics.com/tinytrak. The PIC chip need not be blank, since it is an EEPROM, and can be erased during programming. There are several options for programming equipment, including a PicStart Plus and Microchip's MPLAB software, a LidiPipo programmer or TAPR PIC-E and PicProg or PIX software, or a Tait-sytle programmer and corresponding software. Follow the specific instructions included with the programmer to program the chip.

Firmware Revision Notes

Version 1.0 was the first official release of TinyTrak.

Interfacing

Serial

TinyTrak 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. This is accomplished on the gpsinit/tinytrack board through the JP1 setting. Due to the voltage levels used, some laptop computers may not be able to communicate with the TinyTrak.

Transmitter

To transmit the positional packet beacon, a transmitter must be connected to TinyTrak 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), JP2 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

Connect the receiver's audio out (earphone) jack to the AUD IN and Ground points.When sending position beacons after unkeying on a voice channel is desired, the radio microphone input should be

interfaced to pin7, and the PTT in option should be selected on JP4. The line should be grounded when PTT is active, and floating at other times.

Switch 1

This switch input will select the alternate timing settings when grounded. It should be left floating, or at 5V at other times. This could be useful to change transmission rates when a car's engine is not running.

Switch 2

This switch input is not currently used.

Operations

Configuring User Options

To set user options, such as call signs and transmit rates, use the TinyTrakConfig.EXE. First, connect TinyTrak to a computer serial port. Launch TinyTrakConfig.EXE, and select the connected serial port. Click "Check Version" to confirm the TINYTRAKCONFIG program can configure the TinyTrak options. If a firmware version number is reported, the software and circuit are communicating. If not, re-check all previous steps. Click Read Configuration to upload the current configuration, edit all fields for desired options, and click Write Configuration 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. If either process fails, it will be retried. A dialog will notify the user when the read or write is complete, and if it was successful. Detailed information about each configuration parameter is included below.

Callsign

This is the identification of the transmitting station. It can each be an amateur radio call sign such as KD6BCH, or a tactical call such as SHUTLE. No more than 6 characters (excluding SSID) may be used. If a tactical call is used, a amateur radio call sign should be included in the beacon for FCC identification requirement compliance. As with normal packet radio, an optional SSID between 1 and 15 can be included, such as KD6BCH-9 or SHUTLE-15.

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 number of call signs in the path is limited by the length of the beacon message.

Timing

The timing parameters affect when and how the transmissions occur. There are two sets of timing parameters, primary and secondary. Primary parameters are used for normal transmissions. Secondary parameters are used for manual, after-voice transmissions (MIC-E style), and for the alternate transmissions triggered with switch 1.

KeyUp Delay (TXD)

This sets the delay in milliseconds after the transmitter is keyed, until the data begins. During the key-up delay, the AX.25 flag byte (0x7E) is sent. It is similar to the TXD setting in most TNCs. A value of 100 ms would be equal to 1/10 second.

Transmit Every

This setting controls how often, in seconds, a position transmission will occur, rounded to 10 seconds. Valid range is between 10 seconds and 2550 seconds (42.5 minutes).

Quiet Time

This setting controls the delay in seconds that must occur after the receiver squelches, before a transmission will occur. It can keep transmissions from occurring on a busy channel.

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 63 represents no correction. Experiment with different values to see which has the best results on a receiving TNC.

Beacon

TinyTrak can send a text beacon message after periodic location transmissions. The following two fields control the beaconing.

Message

This setting sets the beacon text. The length of the beacon message is limited by the number of digipeaters in the path.

Beacon Every

This setting controls how often a 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.

APRS/MIC-E

The following four settings set special APRS settings which were introduced by the Tucson Amateur Packet Radio (TAPR) Mic-Encoded (MIC-E). For additional information about these settings, refer to TAPR's web page at www.tapr.org.

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.

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".

Symbol

This setting sets the symbol most APRS programs with display when this beacon's position is received. A symbol setting of '>' will display a car, 'k' will display a truck, 'v' will display a van. Other symbol characters can be found in the APRS documentation.

Alternate Table

This setting will modify the meaning of the Symbol selected above to reference icons from the APRS Alternate Symbol Table. For normal use, it should not be checked. Refer to APRS documentation for more information on this setting.

Programming

This section controls communication with the TinyTrak, and will allow the setting selected above to be configured into the TinyTrak.

Com Port

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

Check Version

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

Write Configuration

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

Read Configuration

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

TinyTrak Adjustment

There are only a few adjustments required for proper operations of TinyTrak. The transmit audio level should be adjusted at R11 for proper deviation. You can listen on a separate receiver, and start the resistor at maximum drive. When transmitting, lower the drive until there is a noticeable change in the receiver. It may help to temporarily set a long transmit delay (TXD) during this step in order to have more time to adjust the level. Overdriving the transmitter is a common cause of failure to decode. The other adjustment point is R12, 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.

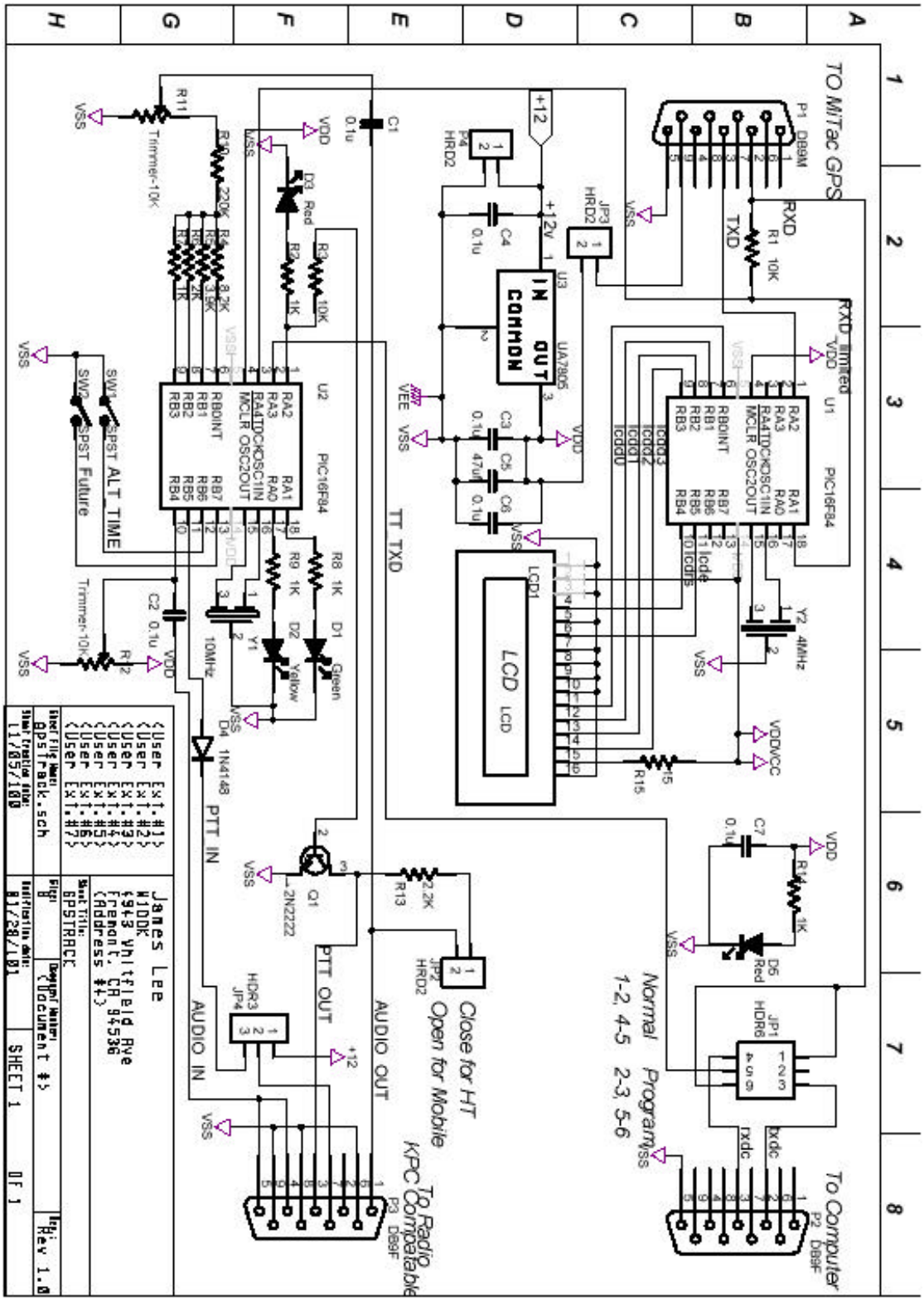
Testing

After setting the desired configuration options, TinyTrak 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 LED should flash three times to show proper firmware operation. 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. The data is not sent in human readable form, but instead in compressed MIC-E form. It should look something like this:
N6BG-9>S8PRPY,RELAY,WIDE:'2+"!r,j/]"4K}
This compressed form contains position, speed, and bearing. More information about this format can be found in the APRS documentation.

Hints, Tricks, Notes, & Troubleshooting

- The TinyTrak 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.
- Currently, the only NMEA-0183 GPS string recognized by TinyTrak is $GPRMC. Any GPS to be connected to TinyTrak must be configured to output this string, if possible. GPS data must be sent at 4800 baud, N81.
- If the TinyTrakConfig.EXE program cannot communicate with the TinyTrak, it may be possible to test the serial link by running a terminal program, such as HyperTerm, configuring for 4800 baud, N81, and sending ESC followed by 'V'. These two keys should cause TinyTrak to respond back with the firmware version number. If serial communications are still failing, monitor the PIC pin 3 with a meter or an oscilloscope to check for incoming serial data.
- TinyTrak was designed for hand held radios. Some mobile radio require more audio drive than TinyTrak puts out. If audio levels are too low, even with the R6 pot set to maximum, consider replacing the 220K R10 with a 100K resistor. This should allow for about double the audio range.
- If TinyTrak 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 TinyTrak can resolve this. If it doesn't, place a 0.1uf capacitor across the base and emitter of PTT transistor Q1 (ground and center).
- On powerup, LEDs D1 and D2 should flash twice, then the radio should send a test transmission, if carrier is not detected, and so LED D3 should light for about a second. LED D5 should always be lit. If the LEDs light this way, it will confirm that the LEDs are wired properly, and that the firmware is correct and running.
- If the radio keys up upon connection to TinyTrak, even before power is applied to TinyTrak, it could be that the radio does not key via current through the mic line (most mobile and Kenwood radios), and jumper JP2 should be removed.
- TinyTrak'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 an 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.

Schematic


TO MiTac GPS
P1 DB9M
RXD
TXD
R1 10K
RXD_limited
U1
PIC16F84
RA2
RA3
RA4TOCKOSC1IN
MCLR OSC2OUT
RBOINT
RB1
RB2
RB3
RB7
RB6
RB5
RB4
RA1
RA0
Y2 4MHz
LCD
LCD1
VDDVCC
R15
C7 0.1u
R14 1K
D5 Red
Normal Programs
1-2, 4-5 2-3, 5-6
JP1 HDR6
P2 DB9F
To Computer
txdc
rxdc
JP2 HDR2
Close for HT
Open for Mobile
KPC To Radio Compatable
P3 DB9F
AUDIO OUT
AUDIO IN
PTT OUT
PTT IN
HDR3 JP4
Q1 2N2222
R13 2.2K
C1 0.1u
C2 0.1u
C3 0.1u
C4 0.1u
C5 47uF
C6 0.1u
+12
+12v
VDD
VSS
VEE
U3 UA7805
IN COMMON OUT
JP3 HRD2
P4 HRD2
R11 Trimmer-10K
R12 Trimmer-10K
R10 220K
R5 8.2K
R6 3.9K
R7 2K
R8 1K
R4 1K
R3 10K
R2 1K
D3 Red
D2 Yellow
D1 Green
D4 1N4148
R9 1K
Y1 10MHz
U2 PIC16F84
SW1 SPST ALT TIME
SW2 SPST Future
TT_TXD
James Lee
N1DDK
6943 Whitfield Ave
Fremont, CA 94536
(Address #2)
GPSTRACK
gpstrack.sch
11/05/100
11/28/101
SHEET 1
OF 1
Rev 1.0

Board Layout