

G6GVI's guide to getting started with your Tracker



Different models

There are two US-based companies currently producing APRS Trackers:

Byonics in Las Vegas make the *TinyTrak* range;

Argent Data Systems in California make the *OpenTrackers*.

Their two product lines are very similar, and their interfaces are identical, so once you've made up suitable patch-leads for your radios and GPS, they are interchangeable.

Patch leads

There are only two interfaces to the Tracker:

The **Radio port** (9-way D-type **socket**), which connects to the transceiver;

The **Data port** (9-way D-type **plug**), which connects to the GPS or PC COM-port.

Radio port

This includes connections for DC power input, PTT, and audio in and out to/from the radio.

The Tracker includes a 5V linear regulator, so an input of at least 7.5V (anywhere between 7.5 and 15V) is required. **+Vcc is applied to pin 7, with ground on pin 6.**

The audio output from **pin 1 on the Tracker needs to be connected to the microphone input on the radio.**

The PTT output from **pin 3 on the Tracker needs to be connected to the radio's PTT line.** Note that most Trackers are supplied with an internal resistor (usually 2.2k) fitted between pins 1 and 3 (R3 in the OT, or R8 in the TT). This is designed to operate the combined PTT/mic input on most *handheld* transceivers, but **must be disconnected for radios with a separate PTT line.** Since I use my Tracker with both handie and mobile radios, I've removed the *internal* resistor, but fitted 2K2 inside the D-type plug of the patch-lead I've built for use with the handies only.

The audio (**loudspeaker**) output from the radio needs to connect to **pin 5 on the Tracker.** Note that the Tracker can't actually *decode* incoming packets, but it uses a simple *noise-detector* on the audio to determine whether or not the channel is busy. This is used to prevent the Tracker from transmitting until the channel is clear, to reduce the chance of interference.

Full wiring diagrams for connections between the Trackers and most radios are published here: <http://www.byonics.com/cables/tinytrak3.php>

Data port

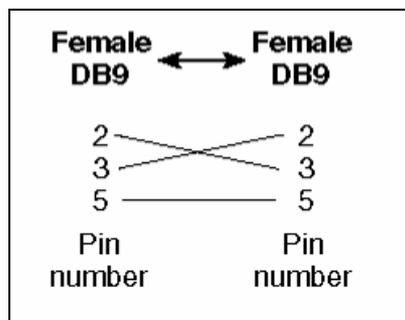
This handles bi-directional serial data at 4800 baud on an "RS-232" interface. The voltage levels are bipolar (typically between $\pm 5V$ and $\pm 9V$). See the following *Configuration* and *GPS Interface* sections for more details.

Configuration

Before it can be used on-air, the Tracker needs to be connected to a PC, to set up its configuration (once these are set, they are stored in non-volatile memory).

The PC connection is made via the serial “COM” port – a 9-way D-type. Some newer laptop PCs don’t have these, so a “USB/RS-232” COM-port adaptor will be required.

The connection to the PC must be by a “Null-modem” lead or adaptor, which has the following connections:



There are a number of different parameters which need to be set.

In some cases, the terminology is slightly different between the two Trackers:

TinyTrak	OpenTracker	Setting
Callsign		Usually add a -9 SSID suffix after your callsign <i>(this is a convention for mobile stations)</i>
Digi Path	Path	“RELAY,TRACE2-2” <i>(this governs how your signals will be digipeated by other stations)</i> ¹
Symbol		select table “/” and “>” for a car or “[” [for pedestrian ²
Auto TX Delay	TX Delay	the time between PTT enable and sending data <i>(if your radio has a slow-locking synthesizer, or transmit/receive relay, you may need to increase this to ~ 500ms)</i>
Auto Transmit Rate	TX Interval	how often do you want your beacons to be transmitted? <i>(Set a fixed interval, or use SmartBeaconing, where the rate depends on your speed)</i>
Status Text	Text	space for your own comment. <i>(keep this brief, or your packets will become too long)</i> It’s best to send the text separately after every few (3 or 5) position beacons.
Send Altitude	Altitude	if your GPS reports altitude, this can be included in your position beacon <i>(but it increases the length of the packet)</i>
Only send Valid	(Don’t) require GPS fix	prevents transmission of position packets if the GPS loses lock.
MIC-E	Compressed	encodes the position in a compressed format, to reduce the packet length
SmartBeaconing		allows you to change the rate of beacon transmissions depending on your speed, and when turning corners

¹ Note that both Configuration programs suggest a default path of “WIDE,WIDE”: this setting is used in the USA, but not in Europe.

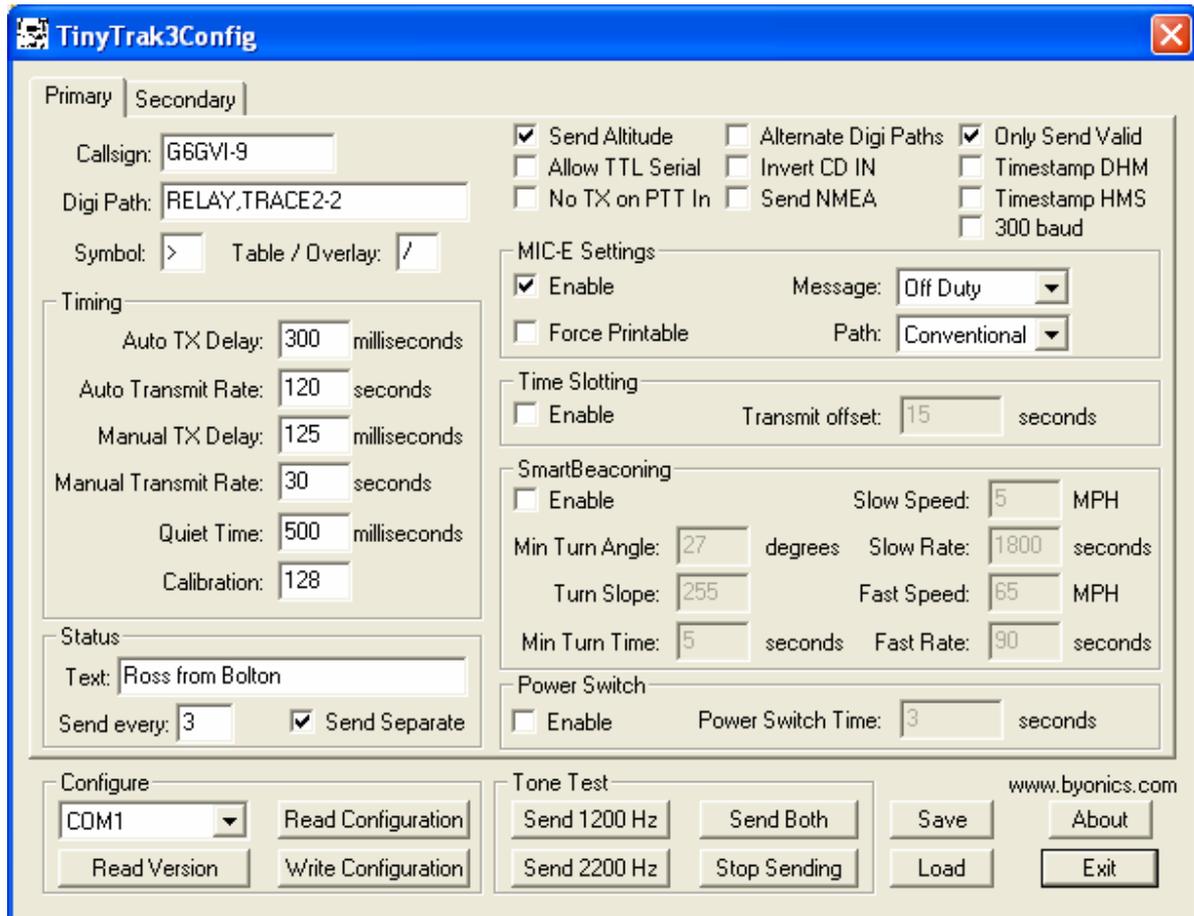
² A full listing of symbols may be found here: http://wa8lmf.net/miscinfo/APRS_Symbol_Chart.pdf

Once all the parameters are set to your preferences, **Write** the configuration to the unit, and also **Save** it to a file on your hard-disk.

Note that both Trackers support two different configurations (or “profiles”), selected by an internal jumper. I’ve brought this out to a switch on the case, so that I can quickly select between the two in the field.

TinyTrak

The *TinyTrak* configuration window looks like this:



The MIC-E “Message” field is akin to the “Position Comment” in the *Kenwood D-radios*:

Off Duty (default)	Committed
En Route	Special ¹
In Service	Priority ¹
Returning	Emergency! ²

¹ Selecting these comments will highlight your station on all other APRS computer displays.

² Select this comment only when absolutely necessary. You will activate alarms in all monitoring APRS stations.

OpenTracker

The *OpenTracker* configuration window looks like this:

The screenshot shows the OpenTracker configuration window for Profile 1. The window title is "Profile 1" and "Profile 2" is also visible. The firmware build number is 53969. The configuration is as follows:

- Call sign:** G6GVI-5
- Baud rate:** 1200 Baud (selected), 300 Baud
- Path:** RELAY.TRACE2-2
- Symbol Table:** /
- Symbol Code:** []
- Temp. Adjust:** 0 °C
- Quiet Time:** 16
- Text:** Ross, home QTH Bolton
- In Comment:** Status (selected)
- Every:** 3 Transmissions
- Transmission Control:**
 - TX Interval:** 200 Seconds (greyed-out)
 - Use SmartBeaconing (Settings button)
 - Use PTT Input
 - Enable timeslotting
 - Timeslot:** 0
- Position:**
 - GPS, Fixed
 - N:** []
 - W:** []
 - Don't require GPS fix
- Counter:**
 - Enable
 - Reset on Transmit
 - 250** debounce (mSec)
- Power Control:**
 - Enable
 - Active Low
 - 3** delay (seconds)
- Reporting Options:**
 - Altitude, DAO
 - Course/Speed
 - Time, DHM, HMS
 - GPS Quality
 - Temperature
 - Voltage
 - Compressed
 - Telemetry every []
- Radio Interface:**
 - TX Audio Level:** min [] max []
 - 300** TX Delay (mSec)
 - Don't transmit if supply < **0.00** Volts
 - Invert Carrier Detect, Software DCD
 - Suppress PTT out on PTT in

Buttons: Load Firmware From (File, Web), Profile Switching, Tuning/Diagnostics.

Since the *SmartBeaconing* mode has been selected, the “TX Interval” has been greyed-out. (the SmartBeacon settings are displayed in a separate window, once the “Settings” button is pressed).

GPS interface

The Tracker is designed to read the output from the GPS unit in *NMEA 0183 format, at 4800 baud only*. The simpler GPS units normally default to this, but the “smarter” ones may need to be set up in their Interface menu.

If you have a serial lead which connects your GPS to a PC COM port, this will also be suitable to connect to the Tracker. Otherwise, all that’s required is to connect the GPS “**TX Data**” output to **pin 2** on the Tracker, and the **signal ground to pin 5**.

The newer versions of the Trackers (*TinyTrak3+* and *OpenTracker1x*) can also supply a regulated 5V DC output to power up an external GPS receiver: this is available on pin 4, once an internal jumper has been fitted. Otherwise, the GPS should be powered off its own independent supply.

In order for the Tracker to work, it must be connected to a GPS receiver producing *live* data, so the receiver must have good visibility of the sky. Both Trackers provide an LED indication that the GPS data is good: the *TinyTrak* has its own green light, whereas the *OpenTracker* flashes its multi-function light once per second.

Setting audio levels

The Tracker's audio output level is controlled by a pre-set potentiometer on the PCB. This must be set for the correct modulation (FM deviation) level on the transceiver in use.

To do this, set the Tracker to transmit every 10 seconds, and whilst running the transmitter into a dummy-load, monitor its output on another receiver. It's easy enough to set the level "by ear", but if you wish to do it visually, try connecting the receiver to a PC sound-card, and use the "Sine Wave" window in the "Sound Card Tuning Aid" of the *AGW Packet Engine* program.

The only other adjustment is to set the threshold of the noise detector: make sure that the LED indicates when the receiver's squelch is opened, and that it goes out when the channel is quiet.

On-air testing

Once everything is set up, have a look at the Tracker's output off-air on your home-station APRS receiver. If you don't have one, or know any local stations who can check this for you, try taking your Tracker system up to a local high-spot, and running it for half an hour or so. Then go back home and have a look at an on-line APRS server, such as:

<http://www.db0anf.de/app/aprs/stations>

<http://aprs.he.fi/?c=raw>

Enter your callsign-SSID, and see if any of your packets have made it onto the network.

Incidentally, if you find that your indicated position is about 100 yards out, check the "Map Datum" setting on your GPS: it should be set to "WGS84", rather than "OSGB" for APRS work.

Burst-after-voice (PTT mode)

This mode is used when voice and APRS are used on a single shared channel. The APRS burst is triggered by the *release* of the microphone's PTT, and sounds a bit like a modulated *pip-tone*.

In order to use it, the microphone's PTT must be connected to pin 8 on the Tracker, and different settings used in the configuration. For more details, see this web-page:

<http://homepage.ntlworld.com/jon.mossman1/digimodes/burst.html>

Ross G6GVI
Version 4, July 2007