FodTrack

Satellite Tracking System

Version 4.2

User Manual

Introduction:

FodTrack is a simple, straightforward program intended to control an azimuth-elevation rotator like the Yaesu-Kenpro 5400/5500/5600, and a transceiver, for any kind of satellite orbits. Best efforts have been made to provide excellent tracking accuracy, within the limitations imposed by commonly available orbital data and without the complexities of compensating for solar and lunar periodic influences.

FodTrack continuously displays what it is doing, uses little system resources, and is so simple to use that you should not have any trouble trying to figure out what is going wrong when the silly satellites seem to go in another direction than your antenna.

Starting with version 4.0, FodTrack is a 32-bit Windows-based program. It should run well in any 32 or 64-bit Windows installation. It has a 30 year long heritage of having been a DOS/DESQview program, and is as proud of that as a program can be. The classic FodTrack screen has been preserved in the Windows version, and also the operational logic, using keyboard control and configuration via text files.

The rotator, the radio, and an optional GPS receiver, can be connected to any serial or USB ports, using various interfaces. All serial ports declared to Windows will work, regardless of whether they are directly connected, USB-connected, local or remote.

Supported rotator interfaces are the FodTrack USB rotator interface, the Yaesu GS-232 and compatibles, and the RC-2800. Each can be used in serial or USB versions, with the USB versions using USB-connected COM ports, preferably using FTDI converters. External rotator drivers can be used too. In this case, FodTrack writes the real-time rotator position to a file, from which another driver can read it.

Unfortunately support for the original, widely used parallel port FodTrack rotator interface had to be discontinued with the move to Windows, because that interface used direct hardware access to the computer's parallel port, but modern computers just don't have parallel ports, and modern Windows versions don't allow user software to write directly to any hardware. If you still have one of those parallel port rotator interfaces, and a computer that has the required parallel port, then use FodTrack 3.0, and either DOS, DOS with DESQview or DESQview X, or Windows 95, 98 or perhaps 2000.

The radio can be connected through any suitable interface that uses either a native serial port or one connected through any sort of USB adapter, again with FTDI converters being preferred. The radio can be combined with transverters for any satellite bands from 145 MHz to 24 GHz.

FodTrack supports an NMEA talker device like a GPS receiver, for automatic setting of the time and location. This is particularly useful for mobile groundstations, and for groundstations in remote areas that don't have another way to keep their computer clocks synchronized. Any serial or USB port is usable.

All three interfaces used by FodTrack are run in an uni-directional way. Rotator and radio interfaces are used for output only, and the GPS interface is used for input only. This keeps matters easy to diagnose. For example, keeping the rotator from running against its end stops is the responsibility of the rotator interface, not of FodTrack, which will of course only ever send the rotator to positions within its range.

FodTrack implements antenna flipping, so the satellites will not run your rotator against its stop. Rotator stop position can be south or north.

FodTrack is best suited for those setups where a single PC running Windows is used to run a complete groundstation. You can of course use an independent computer for FodTrack, if you prefer, and perhaps you can find ways to run FodTrack in other operating systems, through the use of emulators or virtual machines. The critical point here is that the emulator needs to properly pass through the (real or virtual) serial ports to FodTrack.

It can run under manual or automatic control. In automatic mode, a scheduler like WA2N's SatSked, my FodSked, WISP event scheduler, or something newer can provide the commands. Two modes for automation are provided: In the preferred one, FodTrack runs continuously and is controlled via a command file by the scheduler; in the alternative one, FodTrack is called for a specific satellite, and exits after one pass.

You can use FodTrack to control only your rotator, only your radio, or both. Control for AOR AR-5000, Yaesu FT-736, FT-847, FT-857, Kenwood TS-790 and several Icom radios is implemented. IK3NWV tested it on a TS-790, while IK0XBQ made the Icom test with an IC-820. PD0SBC made the AR-5000 tests, and both he and PA0AER tested it with the FT-847. Thanks to them for the beta testing. On the FT-736 I tested it, and I have been using it for 29 years with that radio. Probably it will run with other Kenwood radios, and perhaps also with other Yaesu radios, but this has not yet been tested. If someone wants to use a nonsupported radio, he can use a homemade driver for it. FodTrack writes the frequencies to a file in that case.

On the Yaesu radios only (for now), FodTrack can set subtones to key the satellites that require them.

FodTrack is portable software. It can run from anywhere, even a pendrive.

Copyright:

FodTrack is free for noncommercial use. If you want to reward me somehow, write a piece of useful software and put it in the public domain!

Disclaimer:

FodTrack is provided without any guarantee that it will actually do anything of all the nice stuff this document says. But if you find a bug, please tell me, so I can fix it for the next version.

Setting it up:

This is VERY easy. Copy the FodTrack files to a directory of your choice. You can also keep the Keplerian data file there, but if you prefer you can use a kepfile at some other place.

Edit the FODTRACK.CFG file to reflect conditions at your station. The file explains itself. If you don't understand the use of some parameter, leave it at its default value.

Edit the FODTRACK.FRC file according to the satellites, frequencies, modes and subtones you want to use. Do not include those birds for which you don't want automatic transceiver control. It's good enough to put the nominal frequencies into this file. Later you can fine-tune the frequencies from inside the program.

If you want to track a satellite on several different frequencies or modes, you can define several data blocks for that sat, differentiating them by a tilde character (~) and any designator you like, after the sat name. For example, you could define "AO-16" with the normal mode-J frequencies, plus "AO-16~S" with the frequency of the S-band beacon. In both cases the program will use the keps for AO-16.

The FODTRACK.FRC file packed with the software contains many satellites that are no longer active. I left them in as examples, and for nostalgic reasons. Feel free to delete them. New sats keep popping up, so you will surely need to edit the file to include the sats you are interested in.

You can define a "pseudo-satellite" in this file, called PARK. This will send your Icom or Kenwood radios to the parking frequencies and modes specified there. Yaesu radios return to the frequency they were on before the start of the pass, regardless of any PARK data specified here, as long as you leave the SAT switch in OFF position.

If you will run the program from somewhere else, you need a path to the FodTrack directory.

The Keplerian data file must be in 2-line format, with the satellite name appearing above each 2-line block. Title lines are no problem. They are not needed, but they do not disturb. Each line of the kepfile should end with a carriage return and a line feed character. Kepfiles downloaded from the web sometimes come without the carriage return characters! Opening these in WordPad and re-saving them fixes this. Watch the sat names in the kepfile, because sometimes the same satellite is referred to by different names, such as AO-7 and AO-07.

Satellite names shouldn't be terribly long, or they will mess up FodTrack's screen.

Running it:

Run FODTRACK.EXE. To make this easy and comfortable, you can use Windows Explorer to create a shortcut to this file and put it on the desktop. The program will come up, read its configuration file, the flipping state in which the rotator was left after the last pass, and then it will read the command file (FODTRACK.CMD). The default command file says "NONE", so the program will stay idling. That's a good time to look around the screen:

You will see a status display, which says what the program is doing. It can be idling, waiting for a satellite, tracking it, calculating AOS and LOS while guessing whether it is convenient to flip the antenna over, calibrating the rotator, or accepting your incremental tuning input. There is a nice clock, ticking away your valuable seconds; also there is a line telling you that the satellite selection is automatic, and several windows without any data in them. They will come to life when you select a satellite. There's also a small reminder for the commands you have available. Please note that the commands in the box are only available while in manual mode. During AOS-LOS calculation and GPS reading the keyboard is dead.

Now let's play a bit: Type the letter m to get the program into manual mode, then t to start tracking a satellite. The program will ask you which satellite you want. Enter its name exactly as it appears in the kepfile, otherwise FodTrack will be very unhappy with you. For example, type ISS. But FodTrack is a nice guy in some ways at least, and will forgive you for typing letters in the wrong case.

The program reads the kepfile, and tells you the age of the sat's keps in the proper window. It will also tell you whether you can get good tracking precision with those keps, or if you should get fresh ones. That's of course purely an educated guess, because how quickly a set of keps goes stale depends on various factors including atmospheric drag, which can change suddenly. And some satellites are really evil: They have their own thrusters, and suddenly modify their orbits! So, in case of doubt, get fresh keps.

After loading the keps, FodTrack calculates AOS and LOS times (for the novice: AOS means Acquisition Of Signal and LOS is Loss Of Signal). If you haven't disabled flipping, FodTrack also looks into its crystal ball, to see if the bird will run against your rotator stop on the next pass. If so, it will tell you that the antenna will be "flipped" over. If not, it will be on the "normal" side. When either way will work, FodTrack keeps the current flipping status. If you chose to park your antennas, through the configuration file, then the "Flipped" or "Normal" display will be overrun by the word "Parked". For geostationary sats this calculation is not done, because there is no AOS or LOS at all! Same thing for sats that will never come in range of your station, or at least not within the next 2 days, because you live at a high latitude.

After flipping determination is complete, the program will start tracking the bird. Every second the position is updated. If the AOS time is more than 2 minutes away, no data is sent to the rotator nor to the radio.

When the great moment arrives, two minutes before AOS, the program will start sending target position data to your rotator, indicating so in the rotator status display. The two-minute allowance assures that the rotator gets enough time to point at the satellite before it comes over the horizon. At this time the program will also start controlling your radio, and showing the Doppler-corrected frequencies. If the FODTRACK.FRC file does not contain data for the selected satellite, then the program will not access the radio, and will display Doppler correction in PPM (parts per million). This is useful as a guide for manual tuning of satellites. 1 PPM shift is one Hertz shift per Megahertz of nominal frequency, so it's relatively easy to manually tune the radios following the displayed PPM shifts.

You may notice that the last figure of the frequency does not change at the same time on your radio's display and on the FodTrack screen. This happens because most radios truncate the frequency, while FodTrack rounds it off. So, 435175.478 kHz would be displayed as 435175.5 by FodTrack, while most radios would display it as 435175.4.

If the rotator is not flipped, azimuth and elevation on the rotator are the real ones. If the program had to flip the antennas, azimuth is 180 degrees shifted and elevation starts backwards from 180 degrees.

If you selected a stepsize of zero degrees in the configuration file, then the program will send the rotator position to the interface at a rate of up to once every second. In this case the rotator will move in fine steps, their size being given by any dead-band in the interface and the stepsize of its D/A or A/D converter. If you selected a bigger stepsize, FodTrack will freeze the rotator until the position error is half as large as your selected stepsize. Then it will move the rotator a full step in the proper direction. If your antennas have fairly broad lobes, you can use this feature to reduce wear and noise. The recommended stepsize is about one half of your narrowest antenna's beamwidth.

When the satellite goes below the horizon, rotator output stops, the program stops controlling the radio, and calculates flipping for the next pass, then starts waiting for the bird to come up again. If you selected parking, the antennas will then be parked. Otherwise, they will stay in the position the satellite left them, until the next pass is about to start. If you have an Icom or Kenwood radio, and you specified parking frequencies and modes in the FODTRACK.FRC file, then your radio will be parked too. The FT736 instead will return to whatever frequency it was on before the start of the pass, if you leave the SAT switch in the OFF position. But there is an exception: The few satellites that use the same band for uplink and downlink need the FT-736 to run in normal mode, not sat mode! With these sats, the radio will not return to the frequency it was on.

At any time except during AOS-LOS calculation or GPS reading, you can stop tracking a bird using the s command, or quit using q. If you press a, the program will go back into automatic mode, reading the command file and doing whatever it tells.

Tuning:

While you are tracking a satellite in manual mode, you can enter the manual incremental tuning mode. FodTrack will use the bottom bar on the screen to display the theoretical frequencies at the satellite, the tuning mode selected, and a list of available commands. You can use the cursor keys to increment and decrement the frequencies. The up/down keys change the frequencies in large steps, while the left/right keys do the fine tuning. The size of these steps is configurable in the FODTRACK.CFG file.

There are four tuning modes: In RX mode, the receiving frequency is tuned. In TX, it's the transmission frequency (sounds obvious, doesn't it...?). In DIRECT mode both frequencies are tuned in step, which is useful for operating linear transponders that don't invert the passband, and for semiduplex sats that transmit and receive on the same frequency. Finally, INVERSE mode tunes the RX and TX frequencies in opposing sense, for inverting transponder operation like FO-29.
When entering tuning mode, it defaults to RX tuning.

The SAVE command saves the modified frequencies into the FODTRACK.FRC file. The EXIT command exits the tuning mode.

During manual incremental tuning the orbital calculation and everything else continues. The theoretical frequencies are changed as you touch the proper keys. Every second, the Doppler shift affecting these frequencies is calculated, and while the sat is above the horizon, the corrected frequencies are sent to the radio. So, although the theoretical sat frequencies can be tuned quickly, the frequencies in the radio are updated only once a second. This was done because most radios don't support a much quicker update rate through their control ports.

When you are first setting up FodTrack, you will need to fine tune the frequencies, specially the RX frequencies of PSK sats and CW beacons. You can tune them until the data is being properly decoded and the signal is in the center of the modem's working range, then SAVE them. You should not need to touch up this tuning again for considerable time. Do this adjustment while using reasonably fresh keps, the PC clock accurate to the second, and preferably near the start or the end of pass of a LEO, or near apogee for a phase-3 satellite, which is when the Doppler shift variation speed is minimal. This same kind of tuning can be done for the FSK sats, using the FM discriminator center meter, but tuning is far less critical on FSK.

The manual incremental tuning mode is also extremely useful on the analog satellites. You can tune the RX frequency to some clear spot, then tune your TX to the corresponding uplink frequency, then fine-tune it while transmitting and receiving your own echo. After doing this, you can save the frequencies, and from then on always use the INVERSE or DIRECT tuning modes, depending on the transponder type (inverse mode is by far the more common). Regardless of Doppler shift and position inside the sat's passband, your uplink and downlink will now always agree! This allows for completely hands-free operation, even on fast-drifting sats like FO-29, as long as the station you are talking to is also keeping its frequency constant at the sat! Unfortunately most stations don't.

Radio specifics:

The Kenwood TS790 is notorious for an unwelcome "feature": It will shortly mute the receiver every time it is commanded to update the frequency! So every frequency update causes one or several packets to be lost on digital sats, and a dropout on an analog sat. To reduce the impact of this, FodTrack can be configured to send a frequency update only when the frequency error is larger than a specified value. But the only real solution for this problem is to modify the radio, to eliminate this muting.

The AR-5000 scanner has the same problem.

The Yaesu FT-736 also has its quirks: It is impossible to switch bands in sat mode via the CAT control, unless you have an optional band module (or a dummy module), and even in that case the procedure is a bit tricky. If you have at least one band module or dummy, you can tell FodTrack about these good news in the CFG file. FodTrack will then take care of the bandswitching.

If you don't have such a band module or dummy, then if the radio is in mode B, FodTrack cannot put it into mode J. A simple workaround is to manually set mode J into SAT VFO A, and mode B into SAT VFO B. If you want to track a mode B sat, you then just have to press the VFO B button before FodTrack switches the CAT on, and press the VFO A button for mode J passes.

There are some special situations in which both TX and RX are on the same band. The MIR station was one example of this. The ISS packet station too. The FT-736 cannot set TX and RX to the same band in

SAT mode. FodTrack gets around this problem by setting the radio to normal (non-SAT) mode, tuning RX directly, and using the programmable offset to tune the TX frequency. No special setup is required, just put the correct frequencies into the FODTRACK.FRC file. Please be aware that in this operating configuration the FT-736 cannot use different modes for TX and RX, but this is usually not required anyway.

The FT-857 isn't full-duplex-capable. So FodTrack tries to make best use of its SPLIT mode. It sets the nominal transmit frequency and mode into one VFO, then leaves it alone during the pass and Doppler-corrects only the RX frequency in the other VFO.

If you want to use a radio that is not directly supported by FodTrack, you can write your own driver for it. In that case FodTrack will write the frequencies and modes into a file called RADIO.DAT, updating it every second during a pass. Your driver should read this file and send the data to your radio in the proper format.

Using converters:

When using converters for the bands not covered by your radio, there are several aspects to be considered. The most common converters ADD a fixed frequency to your radio's output, in order to reach the operating frequency. In this case, all that FodTrack must do is calculate the Doppler shift for the operating frequency, then subtract the converter's local oscillator frequency and send the result to the radio.

On the other hand, some converters use a higher oscillator frequency and SUBTRACT your radio's frequency to reach the desired frequency. In this case, the tracking on the radio must be inverted, and also the MODE (if it is SSB) must be inverted! FodTrack takes care of these issues, so you must simply do the following to use converters:

In the .CFG file, for each band you will operate through a converter you must specify the converter's local oscillator frequency. After this, you can set up the .FRC file with the real frequencies and modes. As simple as that! FodTrack will display the real frequencies during tracking and tuning, while sending the converted ones to the radio.

Easy, isn't it?

I hope this scheme is flexible enough to accomodate most (or hopefully all) needs. If you have some suggestion, don't hesitate to contact me.

The FodTrack Rotator Interface:

Let's weep some tears. The original parallel port FodTrack rotator interface, designed in 1994, became a de-facto standard over the years, directly supported not only by FodTrack but also by many other programs, and used by many people around the world. But this interface used a parallel printer port, and used it in a non-standard way. So it could not use any printer drivers, but instead FodTrack took direct control of the parallel port hardware. This doesn't work on parallel ports connected via USB adapters, and of course most modern PCs no longer have parallel printer ports! On top of that, while direct hardware access by user programs was a normal way to do things when FodTrack was created, newer Windows versions (and also other operating systems) do their very best efforts to absolutely keep user programs from directly controlling any hardware. I could not find a solution to this problem, and finally decided to put the original FodTrack rotator interface into retirement, and switch over to USB. May the good old parallel port rotator interface rest in peace! It must have tracked millions of passes, in stations all around the world.

To replace it, I decided to use a very compact and straightforward protocol for a new USB/serial rotator interface. The software will write command strings of 6 bytes to the port. The first byte is a G character, which stands for “go” and serves for sync. Then come two bytes for azimuth and two for elevation, followed by a line feed character as end signal. The azimuth is expressed as ten bits for 0-360 or 0-450 degrees, depending on the rotator model, and elevation as 10 bits for 0 to 180 degrees.

In calibration mode, the G is replaced by a C.

This format lends itself very well to make a very simple interface using a PIC that has a 10-bit A/D converter. The control loop is closed in the PIC. If an SMD version of the PIC is used, it might be possible to build the entire interface small enough to fit inside the DIN connector that plugs into the rotator control box. A USB FTDI cable, cheaply available on eBay, does the USB-serial conversion. This is a very compact, simple, inexpensive, and even elegant USB-rotator interface.

The schematic of the FodTrack USB rotator interface is in the file USB-ROTATOR.PNG. It's much simpler than the original parallel port interface, but since it's microcontroller-based, the builder has to load the firmware into the PIC, which can be done either before or after installing the PIC in the circuit. The firmware is provided both in compiled version, in the file USB-ROTATOR.HEX, and as source code, in USB-ROTATOR.GCB.

The hex file can be directly read and burned into the PIC by any common PIC programming software, which is usually free. A simple programming hardware is needed, which can be built very easily, or bought cheaply. Search the web for PIC programmers, if you don't have one already.

You don't really need the source code to build the interface, but it could be useful if you want to make changes, such as using a different PIC, or adding features. It compiles in Great Cow Basic, which is free too.

All components of this interface are available both in SMD and through-hole versions, so you are free to use the style you prefer.

The interface continuously listens to commands coming from the PC, using the hardware UART of the PIC, and if any arrives, it checks whether the position sent is really new, separately in azimuth and elevation. If it is, it initiates motion in the required direction. When the desired position is reached or overshot, it stops. During normal tracking, it does not actively hold on to this position, so it doesn't interfere with direct manual control of the rotator, while the computer isn't sending position updates. During calibration instead it does continuously hold the position, with a small dead range, to allow easily setting the adjustments on the rotator box.

The interface can also update positions before the rotator has even reached the previous one, inverting the motion direction if needed. It does not need to complete one move before accepting and executing a new command, as some other interfaces do.

If FodTrack doesn't get access to the port specified in the configuration file, because the port isn't present (USB not plugged in!) or because there is some other program using the port, it will simply turn off rotator output, warn about this on screen, and go ahead.

The calibration function provided ib Fodtrack allows you to manually send position commands to the interface, which is helpful in setting up and testing the system. There is no calibration proper in the interface, though. It simply expects to see 0 to 5 Volt coming from the rotator, for each axis. The interface uses a 78L05 regulator as a reference, so the 5V full scale value should be quite accurate. This should precisely cover the full range of motion of each rotator, and it's the rotator's duty to provide the correct voltage. The Yaesu rotators have the necessary adjustments on the back of their power supply/control boxes.

The sequence for properly setting up a Yaesu G-5400, 5500 or 5600 series rotator is:

- Correctly set up the Fodtrack configuration file.

- Wire up the rotator with its control box, and provisionally install the antennas, or if you prefer, a broom or anything that allows you to clearly see where it's pointing. Don't connect the interface yet.

- With the rotator box turned off, carefully adjust the mechanical zeros of both meters.

- Turn on the rotator box, and use its buttons to rotate left and down, all the way to the left end of the scales.

- Go to your rotator, and mechanically rotate/align it such that the antennas are exactly horizontal, and point exactly in the correct azimuth, according to your rotator type: South for the G-5400 and G-5600, north for the G-5500. Use landmarks to check the azimuth, rather than a compass, because a compass suffers from magnetic declination plus unknown local deviations. After correct setting, tighten down all bolts so that nothing can slip.

- Use the rotator control box buttons to rotate azimuth one full turn to exactly the same position, in case of the G-5400 and G-5600, and to one turn plus exactly a quarter turn, for the G-5500. The real pointing of the antennas or the broom is what counts at this point, not the rotator box meters.

- Adjust the "full scale" potentiometers on the back of the rotator control box to make both meters point exactly at full scale. These two adjustments should nor require touching up for many years, so you may want to place adhesive tape over their holes, to prevent disadjusting them by mistake.

- Plug in the interface, fire up FodTrack, go to manual, calibration. Send the rotator to full scale. With a G-5400 or G-5600 rotator, that's an azimuth of 179°, because 180° would be the left end of the scale! With a 5500 instead, it should be 450°. The full scale elevation is 180° with all rotators.

- Carefully adjust the "out voltage" adjustments on the back of the rotator controller box so that both meters exactly and just barely point at the end of their scales. The interface will make the rotator follow the adjustments changes you make.

- Now check if all is right: Use the calibration function to send the rotator to various positions, checking that the antennas (or the broom!) go to the specified positions, and that the meters agree with them too. If the interface can drive the axes beyond the end of scale, into the end stops or limit switches, or if it fails to reach the end of the scale, you can very slightly touch up the "out voltage" adjustments, and try again.

Be kind and understanding with your rotator. Don't ask it to do what it cannot! Position feedback in the Yaesu rotators uses a potentiometer as sensor, and some of the circuitry in the rotator box isn't perfectly linear. So you might see some nonlinearity, but the performance should still be plenty good enough for the antennas hams typically use, which rarely have a beamwidth much narrower than 20°.

Using the GS-232, RC-2800 and compatibles:

Instead of a FodTrack rotator interface, you may use the Yaesu GS-232 or the RC-2800, or any of several other interfaces that suppport the protocol of one of these. This allows using FodTrack with ready made hardware, for those who don't want to build anything. But of course the cost is much higher! MM5ISS beta-tested one of them, CE4JDM tested another. Both worked well.

To use one of these interfaces, you simply set up the configuration file accordingly (details are commented in that file), and connect the interface to the selected serial port. FodTrack uses only the Wxxx yyy command of the GS-232, and the A and E commands of the RC-2800. It does not expect any answer.

Some of these interfaces allow software calibration, through a setup program. This software calibration interacts with the "out voltage" adjustments on the back of the rotator box, so I suggest to follow the calibration instructions of your interface. FodTrack can still be used to send calibration values, if required.

Using external rotator drivers:

Setting the rotator interface parameter to 3, FodTrack will write real-time rotator position data to a disk file named ROT.DAT. The file contains a single line in the format AZxxx ELyyy. You can write your own driver to read this file and send the position to whatever rotator interface you fancy to use.

And if you use a non-Yaesu rotator, or you build your own rotator, you are probably knowledgable enough to not need much further help!

Radio Interface:

Some radios (the FT-847 is one example) can accept a computer's RS232 output directly. Some modern radios accept USB directly, but it's likely that they can't understand any of the radio command languages FodTrack knows... Most older radios instead have serial ports that operate at TTL levels, so they need some interface to convert RS232 to TTL for use with old computers and COM ports, or USB-to-serial converters that deliver TTL level outputs. Some radios use TRUE polarity, others need the signal inverted. Check the docs of the radio.

The file USB-FT736.png contains the schematic of a very simple USB radio interface. It uses an FTDI

USB to serial converter cable, and a simple open-collector driver, to make sure that no problem happens with any combination of powering up and down the radio and the USB chip. It uses two transistors, because the FT-736, just like the Icom radios of that vintage, require true polarity. The Kenwood radios of that time use inverted logic, so the interface can be adapted for them by removing one of the inverter stages.

That FTDI cable can of course be identical to the one used by the rotator interface. The FTDI chips remember the COM port number they get assigned by Windows, so there is absolutely no problem with using several of them at the same time, and shifting them between USB ports. Instead the USB-serial converter chips of some other brands might be a little more confusing in this regard, but they can be used too. It should go without saying that if you use an USB-serial converter that isn't natively supported by Windows, you need to install the proper driver for it.

The very few components required by this radio interface can easily be built into the radio connector.

You can also use the original RS232 interfaces sold by the radio makers, either directly on serial ports or through USB-RS232 adapters, but they are costly. You can also buy a ready-made USB control cable for your specific radio, but that's also somewhat expensive.

I'm still including schematics of the old RS-232 adapters I used in my station, back when I had a computer with 9 serial and 3 parallel ports. RS232Y.PCX is the version for Yaesu, which I initially used for my FT-736. RS232K.PCX is for Kenwood radios. I used that one for my TS-450, and I hope it will run also with the TS-790, but I have not tested this. These interfaces can be used with USB-RS-232 adapters too. And RS232SIM.PCX is the old ultra-simple radio interface: Just three components! This circuit does not provide a "reply" connection for the radio, and it can be used only with radios that have an input with pull-up, and use inverted signals. It works perfectly well with FodTrack and the Yaesu FT-736R, and will probably also work with Icom radios (using another plug, of course...). If you still have RS-232 serial ports, this is the easiest way to get going. With modern computers, of course use the USB interface.

GPS support:

If you have a GPS receiver or another device that can output NMEA-0183 datagrams, you can connect it to a serial port of your PC, or to a USB-serial adapter. There are even some GPS receivers that connect directly by USB, having a built-in USB-serial converter. You can then use FodTrack to keep the PC's clock accurately set. Optionally you may enable a function that gets your location from the GPS. This is very attractive for maritime mobile sat stations, and for anyone who moves his station around a lot! In the sat heydays of the 1990s, there was at least one maritime mobile sat station sailing around the globe and using FodTrack with a GPS! And perhaps some more that I didn't get to know about. But even if you operate a typical fixed station in a place where you don't have internet available to use an NTP service, the time-setting feature alone may warrant the purchase of a suitable GPS receiver. In most fixed stations, however, internet is available, and then it's pointless to use a GPS connected in this way, because NTP over the internet does a better job of keeping the computer clock on time.

What the GPS feature in FodTrack does is this:

- Whenever the program is sent to a satellite, or if you press the "G" key, FodTrack will read the specified port and wait for up to 6 seconds for the arrival of two consecutive datagrams of the same type, bearing different time stamps. It will then assume that the latter time stamp is reasonably fresh, and will set the PC clock to this time plus a small offset which is there to compensate for the time interval between the instant the GPS fixed the position, and the moment the datagram actually arrived at the PC. You can specify this offset in the CFG file. The proper value depends on your specific GPS receiver, but will typically be about 1 to 3 seconds, and there will be some jitter.

- If the position feature is enabled, then FodTrack will also set the new geographic coordinates, and it will write the datagram to a file called FODTRACK.GPS, in order to allow other programs to make use of this data.

- During the first and last minute of each day the GPS access is inhibited. I did this because the NMEA datagrams sent by most low-cost GPS receivers do not provide the date. So, by inhibiting GPS timesetting near the date switch I hope to avoid setting today's date with tomorrow's time!

FodTrack understands the GLL, GGA and RMC datagrams. This should provide compatibility with almost any GPS receiver.

I first tested the GPS function many years ago using my Magellan Trailblazer, later with a Motorola, and I have heard of several people using it with Garmin units. And there is no reason why a modern U-blox or similar GPS shouldn't work.

Automation:

There are two different ways for automatic tracking of multiple satellites.

The preferred mode is this:

Configure your scheduling program in such a way that it writes a FODTRACK.CMD file into the FodTrack directory, containing the name of the satellite to be tracked, exactly as it appears in the kepfile. This CMD file should be written two minutes before the start of the pass, to give the rotator enough time to preset the antennas. The easiest way to write these command files is simply to copy them from somewhere elso into the FodTrack directory, using a COPY command inside the batch file you run at the start of a pass. For example, you can have a FODTRACK.CMD file in your Falconsat-3 directory, containing the text "Falconsat-3" (without the quotes, of course...). Into the batch file you run at the start, you put the command

COPY C:\SAT\FALCONSAT-3\FODTRACK.CMD C:\SAT\FODTRACK\FODTRACK.CMD

After the pass, the scheduling system should write a CMD file saying NONE, so FodTrack stops.
Now you just run FodTrack. When your satellite scheduler decides to run a pass, it calls the batch file, which copies the proper FODTRACK.CMD into the FODTRACK directory. When FodTrack reads the file within the next second, it starts tracking the specified sat. The same batch file starts your communication software, etc. After the pass the CMD file contents is changed by the scheduling system to "NONE", and FodTrack goes to idle mode, waiting for the next pass.

FodTrack checks the CMD file continuously, so it is mandatory to have disk caching enabled, but that's standard nowadays, and hard disks anyway are going the way of the dinosaurs, replaced by SSDs.

The alternative mode for automation is simpler:

It consists just in configuring FodTrack for automatic exit after a pass (in the CFG file), and having the scheduler call it before each pass specifying the satellite on the command line. For example, you use the command

FODTRACK FALCONSAT-3

in the batch file. FodTrack will start up, wait for the specified sat, track it until the end of the pass, and then quit. The command file is not read in this mode.

If you switch to manual mode after starting FodTrack with a sat on the command line, it will then work in the usual way for that mode. If you stop tracking and then switch into automatic mode, it starts reading the command file. But if you switch directly from command line mode into automatic mode, the program will abort, because it considers the pass finished.

If you use WISP, you can tell the event scheduler in GSC to do the command file business via batch files, as described above. There is also a program available, called FOD-INIT and written by CN8HB, which does the glueing job between WISP and FodTrack.

In case of trouble:

If you have any trouble with FodTrack, tell me, so I can fix the bugs for the next version. Also I will do my best to act upon any suggestion for improvement, if reasonable and possible to implement.

During the heydays of the Pacsat and UoSat digital satellites, in the 1990s, the best way to contact me was right there, on those sats. With these digital sats having unfortunately joined the dinosaurs and hard disks in extinction, you can instead resort to e-mail. Look up my current address on my web site, and if that doesn't work, make a web search for my name or callsign.

Of course you are welcome to visit my hobby homepage, not only for looking up my address! There are several electronic projects, my ham history, and lots of text and photos about my other hobbies. The address is:

http://ludens.cl

A last word:

I'm not a professional programmer, so you are allowed to smile about my program and me...