

PLXTracker Blue (firmware v1.01)

(manual revision 18 March 2016)

MIC-E/BASE-91/PLAIN TEXT POSITION REPORTING
BUILT-IN BLUETOOTH INTERFACE
SIMPLE ALIAS-BASED APRS PACKET DIGIPEATER
WEATHER STATION SUPPORT
EXTERNAL TELEMETRY MODULE SUPPORT
DIGITAL THERMOMETER SUPPORT
INPUT VOLTAGE MEASUREMENT
SERIAL CONSOLE FOR ADMINISTRATION
EASY USB-HID PROGRAMMING

Instruction manual
Configuration software manual



Designer: Mateusz Płociński SQ3PLX
Producer: Microsat
info@microsat.com.pl



Table of Contents

1. Technical parameters.....	3
2. Device features.....	4
2.1. Mic-E/Base-91/Plain text position reporting.....	4
2.2. Built-in Bluetooth interface.....	4
2.3. Simple alias-based APRS packet digipeater.....	4
2.4. Weather station support.....	4
2.5. External telemetry module support.....	5
2.6. Digital thermometer support.....	5
3. Connectors and device status leds.....	6
3.1. Front panel.....	6
3.2. Rear panel.....	7
4. Power, radio, RS-485 connector.....	8
4.1. Audio output level regulation (digital).....	8
4.2. Audio output level regulation (potentiometers).....	8
4.3. Audio input level.....	9
5. Serial port connector.....	10
6. Connecting the weather station.....	11
6.1. LaCrosse/Technoline WS-2300, WS-2350 stations.....	11
6.2. Peet Bros Ultimeter stations.....	11
6.3. Davis Vantage stations.....	11
7. Connecting to PC.....	12
8. Description of configuration software.....	13
8.1. Reading configuration from the device.....	13
8.2. Writing configuration to the device.....	13
8.3. Firmware update.....	13
8.4. Configuration import/export.....	13
8.5. Radio Tab.....	14
8.6. Position reports Tab.....	15
8.7. Digipeater Tab.....	17
8.7.1 Digipeater / Common settings Tab.....	17
8.7.2 Digipeater / Simple aliases Tab.....	18
8.8. Beacons Tab.....	18
8.9. Telemetry reports Tab.....	20
8.9.1. Telemetry reports / Report settings.....	21
8.9.2. Telemetry reports / Analog channels.....	22
8.9.3. Telemetry reports / Digital channels.....	23
8.9.4. Telemetry reports / WXBits outputs.....	23
8.10. Weather reports Tab.....	24
8.11. External interfaces Tab.....	25
8.11.1. External interfaces / Serial ports.....	25
8.11.2. External interfaces / Bluetooth.....	26
8.11.3. External interfaces / RS-485 port.....	27
8.11.4. External interfaces / Weather station.....	28
8.11.5. External interfaces / Serial console.....	28
8.11.6. External interfaces / GPS receiver.....	29
8.11.7. External interfaces / KISS.....	30
9. Bluetooth connection with APRSDroid.....	31

1. Technical parameters

Dimensions	94x50x25 mm
Weight	70g
Supply voltage	12VDC typ. (9 - 16VDC)
Power consumption	50-80mA (depending on Bluetooth connection state)
Operation temperature	-30 °C / +70 °C
Humidity	95% max.
Power, radio, telemetry connector	DB-9 male
Serial port connector	Mini DIN 6-pin female

2. Device features

2.1. Mic-E/Base-91/Plain text position reporting

The main feature for an APRS tracker is position reporting. PLXTracker Blue allows you to select between three most common position reporting formats:

- Mic-E - uses both AX.25 destination address and information field for position report and is the most compact reporting method, giving the shortest packet length,
- Base-91 - second compressed method oriented to short report packet length,
- Plain text - in this method station position report is sent in human-readable plain text format,

You can also select between fixed time interval reporting which means that packets are sent every X number of seconds, and SmartBeaconing. SmartBeaconing is an algorithm which sends position reports with intervals dependent on speed and course, allowing for more accurate reporting especially in urban area. SmartBeaconing algorithm was first developed by Tony Arnerich KD7TA and Steve Bragg KA9MVA and now it is widely used by most APRS tracker devices.

2.2. Built-in Bluetooth interface

PLXTracker Blue integrates a built-in Bluetooth interface which can work in one of selectable modes:

- Serial console - similar to RS-232 serial console known from PLXTracker or PLXDigi, it outputs debugging information for device operation.
- KISS - allows to exchange packets with your laptop, tablet or smartphone with APRS software installed. This feature works great with APRSDroid available for Android phones and allows to plot received stations on a map. APRSDroid allows also to send and receive messages, beacons and SmartBeaconing position via PLXTracker Blue.
- GPS passthrough - if you have a GPS receiver connected to PLXTracker Blue RS-232 serial port, then you can forward NMEA stream to any other device using Bluetooth interface.

2.3. Simple alias-based APRS packet digipeater

With PLXTracker Blue you can do a simple alias-based APRS packet digipeating. Generally APRS tracker units should not be used as a full-featured digipeaters but with alias-based digipeater feature you can digipeat WIDE1-1 packets sent by other mobiles increasing their chance of reaching an I-Gate.

2.4. Weather station support

PLXTracker Blue allows you to receive weather data from compatible weather stations over RS-232 serial interface. Currently supported stations are:

- LaCrosse/Technoline WS-2300,
- LaCrosse/Technoline WS-2350,
- Peet Bros Ultimeter 100,
- Peet Bros Ultimeter 800,
- Peet Bros Ultimeter 2000,

- Peet Bros Ultimeter 2100,
- Davis Vantage Pro (with RS-232 datalogger),
- Davis Vantage Pro2 (with RS-232 datalogger),
- Davis Vantage Pro Plus (with RS-232 datalogger),
- Davis Vantage Pro2 Plus (with RS-232 datalogger),

2.5. External telemetry module support

You can use an additional telemetry module for reporting of voltages and electric current values of your installation. This is especially useful for monitoring of solar panel powered systems on remote sites. WXTelemetry module allows you to read 2 voltages, 2 currents and one temperature, then sends these values as APRS telemetry reports to RF.

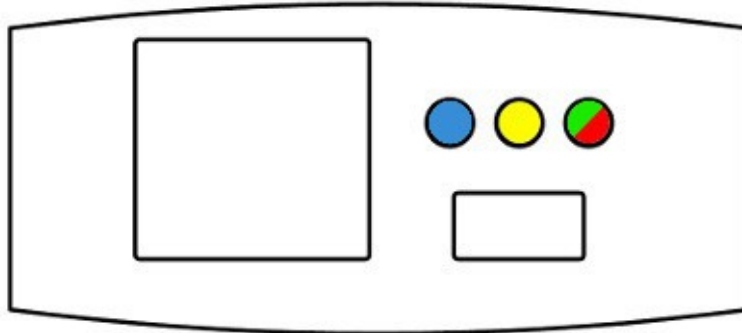
2.6. Digital thermometer support

If you do not need a telemetry module connected to RS-485 serial bus, then you can use it for direct connection of DS18B20 or DS18S20 thermometer input. Measured temperature can then be sent in beacon text or telemetry reports.

3. Connectors and device status leds

3.1. Front panel

There are 3 leds on the front panel that show the status of device operation. The following image shows the location of front panel LEDs and connectors.



Leds:

- **Blue LED** - illuminates when receiving a signal from the radio (indicates the channel is busy),
- **Yellow LED** - lights during transmission, when the device sends an APRS data packet,
- **Red/green LED** - this led color has a following meaning:
 - blinks in red, when Bluetooth is not connected,
 - blinks in green, when Bluetooth is connected.

Blinking is dependent on current device state:

- one short blink (0.1 second) per 1.6 seconds - no GPS NMEA sentence received within last second, GPS disconnected,
- two short blinks (0.1 second each) per 1.6 seconds - GPS NMEA sentence received but no position fix,
- three short blinks (0.1 second each) per 1.6 seconds - GPS NMEA sentence received with position fix,
- one long blink (1 second) per 1.6 seconds - APRS packet was properly received.

Connectors:

- **Mini DIN 6-pin** - one RS-232 serial data interface,
- **Mini USB** - a mini USB socket, used to connect to a computer for configuration,

PLXTracker Blue incorporates only one RS-232 serial interface (contrary to PLXTracker or PLXDigi), located on „RS-232 port 2” pins (port 1 is internally occupied by Bluetooth interface).

If you wish to use a weather station, serial console, etc., you will need a different set of serial cables than for PLXTracker or PLXDigi.

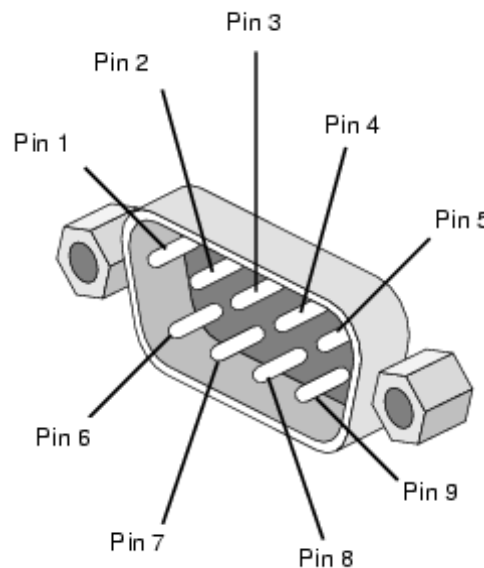
3.2. Rear panel

Connectors:

- **DB9 male connector** - used for connecting of supply voltage, radio, and RS-485 telemetry module or digital thermometer.

4. Power, radio, RS-485 connector

DB-9 male connector is used for supply voltage input, audio in, audio out, ptt, and RS-485 bus for telemetry and other external devices.



DB9 connector pins description:

- Pin 1 - PTT output, open collector, if the external PTT output is selected in device configuration, this pin is shorted to ground while broadcasting,
- Pin 2 - GND,
- Pin 3 - GND,
- Pin 4 - APRS audio input from radio to PLXTracker Blue,
- Pin 5 - GND,
- Pin 6 - RS-485 A+ positive differential input/output,
- Pin 7 - RS-485 B- negative differential input/output,
- Pin 8 - APRS audio output from the device to the microphone input of the radio. This output also acts as a PTT by shorting pin to ground through a 2k2 resistor - an option for portable radios that do not have a separate PTT input (selectable in configuration software, 2k2 resistor is built-in),
- Pin 9 - supply voltage input.

The level of input and output should be individualized in such a way that packets are decoded correctly. Audio levels can be regulated using potentiometers placed on a PCB inside of the device. Audio output can also be digitally controlled in device configuration.

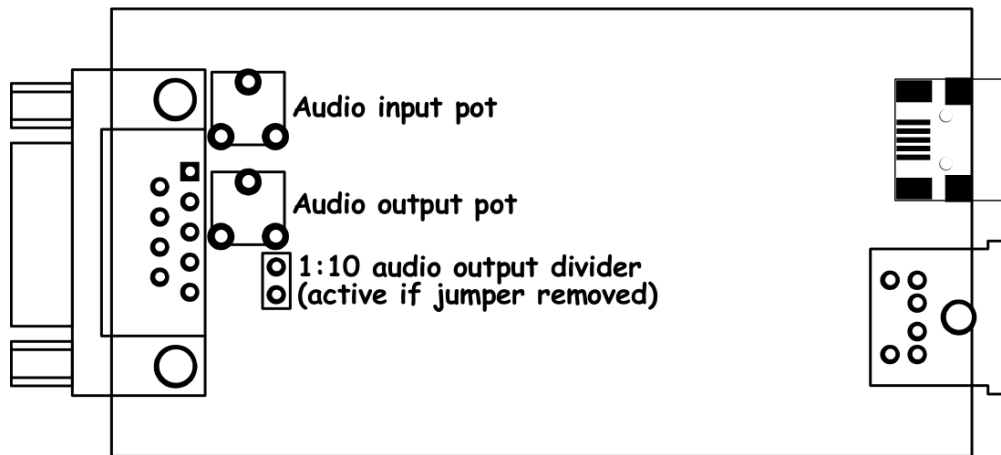
4.1. Audio output level regulation (digital)

PLXTracker Blue allows for digital control of output level in device configuration. Possible relative output levels are 100%, 50%, 25%, 12.5%, 6.25%.

4.2. Audio output level regulation (potentiometers)

Audio output level is regulated after production to some default value. The most

suitable value is dependent on the radio used, because of various input impedance and accepted input levels of different radios. If digital output level regulation is not enough, you can use a potentiometer located inside of the device. To do that, you will need to open the case and gain access to the PCB of the device. Output signal should be set to achieve the desired modulation depth at the output of the radio.



4.3. Audio input level

Audio input level should be regulated for proper packet decoding. There is no easy way to tell you how to do in properly because you will not be able to measure your signal level without an oscilloscope or similar equipment.

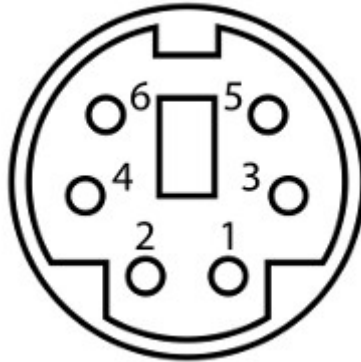
There is a 10 kOhm potentiometer on the input. The best method for first try is to set in on a neutral center position. Audio input after the potentiometer should not exceed about 400mV and should not be below 25mV. So on device input you should provide audio level of around 50-800mV (for potentiometer in neutral position).

Our tests revealed that decoding works fine for wide range of audio input levels. Packets were decoded properly even with audio input overdriven with a high audio signal from speaker outputs.

If you are using a radio with speaker output connected to PLXTracker Blue (like FT-2900), please do not set the volume too high. Excessive signal level on PLXTracker Blue input will prevent it from decoding packets from some or even from all nearby stations. Good starting volume is after 1/4 of radio volume full-scale.

5. Serial port connector

In PLXTracker Blue the MINI DIN 6pin female connector is used for single RS-232 serial port and 5V voltage output.



MINI DIN 6pin connector pins description:

- Pin 6 - not used
- Pin 4 - RS-232 Port 2 TXD,
- Pin 2 - GND,
- Pin 1 - 5V output, do not short-circuit, max. current 100mA,
- Pin 3 - hard-wired to negative voltage for WS-2300/WS-2350 weather station support,
- Pin 5 - RS-232 Port 2 RXD.

RS-232 Port 2 functions are selectable in software.

PLXTracker Blue incorporates only one RS-232 serial interface (contrary to PLXTracker or PLXDigi), located on „RS-232 port 2” pins (port 1 is internally occupied by Bluetooth interface).

If you wish to use a weather station, serial console, etc., you will need a different set of serial cables than for PLXTracker or PLXDigi.

6. Connecting the weather station

6.1. LaCrosse/Technoline WS-2300, WS-2350 stations

The only difference between WS-2300 and WS-2350 weather stations is that manufacturer has made an external USB to RS232 adapter, which is not used in case of PLXTracker Blue.

WS-23XX weather stations require a bit exotic connection method. There is no GND connection and both positive and negative 5V voltages are necessary for interfacing.

To connect WS-2300 or WS-2350 station to PLXTracker Blue you will need to use:

- "CAB02B - PLXTracker Blue serial interface cable (for WS-23XX)"

Schematic for this cable is available on our webpage.

6.2. Peet Bros Ultimeter stations

Peet Bros Ultimeter weather stations use a standard RS-232 serial port interface with TXD, RXD and GND. Other serial lines are not used. Stations must then be manually switched to "data logger" mode. According to the manufacturer's instructions, you should press CLEAR + WIND SPEED buttons for 3 seconds to activate the "data logger".

To connect Peet Bros Ultimeter station to PLXTracker Blue you will need to use:

- "CAB29B - PLXTracker Blue serial interface cable (for Davis, PeetBros)"

Schematic for this cable is available on our webpage.

6.3. Davis Vantage stations

Davis Vantage weather stations use a standard RS-232 serial port interface with TXD, RXD and GND. Fourth wire is connected to DCD, DTR, and DSR inside original DB-9 connector but it shouldn't be used on PLXTracker Blue side.

To connect PLXTracker Blue to Davis weather stations you need to use an additional RS-232 "datalogger" interface available from the manufacturer. Device part number is 06510SER.

Davis stations datalogger interface comes in two kinds: "standard" datalogger or "APRS" datalogger. According to my current knowledge, both types work fine with PLXTracker Blue (it doesn't use an "APRS" feature of datalogger but "APRS" datalogger has also the same communication options like "standard" one).

To connect Davis Vantage station to PLXTracker Blue you will need to use:

- "CAB29B - PLXTracker Blue serial interface cable (for Davis, PeetBros)"

Schematic for this cable is available on our webpage.

7. Connecting to PC

Configuration of PLXTracker Blue can be done via USB interface with PLXTracker Blue Configurator software.

PLXTracker Blue enters into configuration mode only when USB cable is detected while powering up and it is indicated by Red LED blinking.

To connect to your computer, follow these steps:

- Disconnect PLXTracker Blue DB-9 cable, device should be unpowered,
- Connect the USB cable from your computer to PLXTracker Blue, Red LED should start to blink (device is now powered from USB),
- Now you can use PLXTracker Blue Configurator for device configuration read/write or firmware update.

PLXTracker Blue device appears as USB-HID device in Windows operating system, and should be detected automatically. Therefore you don't need to install any device drivers for PLXTracker Blue.

8. Description of configuration software

In this chapter we will discuss all the Configurator application settings and describe its influence on the device operation.

8.1. Reading configuration from the device

To read the configuration from the device you must use the **“Read config”** button. After this action, the program will download current configuration values from PLXTracker Blue internal memory.

8.2. Writing configuration to the device

After entering configuration values in Configurator application, choose **“Write config”** to upload current configuration to PLXTracker Blue internal memory.

8.3. Firmware update

If you download a new PLXTracker Blue Configurator application version, you should also update device firmware to be compatible with current Configurator version.

To update firmware of the device, please click on the **“Upload firmware”** button.

Don't worry, it is not possible to “brick” or damage the device with this operation. If anything goes bad, you will simply need to try again with firmware update function.

Please remember that you should update your device with the same Configurator and firmware version numbers. It is also always necessary to upload new configuration after flashing the device with new firmware versions. The device will not start if configuration and firmware versions do not match. If your device simply blinks all the leds when powered up, it means that your configuration version does not match firmware version.

8.4. Configuration import/export

With **“Save to file”** button you can export your current configuration from PLXTracker Blue Configurator window to a file on your PC's disk drive. Similarly with **“Read from file”** button you can import configuration from a file to your PLXTracker Blue Configurator application. These functions can be used to save your device's configuration and to transfer your device's configuration to recent Configurator and firmware versions.

8.5. Radio Tab

The screenshot shows the 'PLXTracker Blue Configurator v1.01' window. At the top, there are two groups of buttons: 'Connection' (Read config, Write config, Upload firmware) and 'Configuration import/export' (Save to file, Read from file). The 'MICROSAT' logo is on the right. Below these is a tabbed interface with 'Radio' selected. The 'Radio' tab contains several configuration fields: 'Callsign' (NOCALL - 0), 'Latitude' (5223.08N), 'Longitude' (01658.14E), 'TX delay [ms]' (250), 'Quiet time [ms]' (400), 'Randomize quiet time' (+/- 64 ms), 'Output level' (100%), 'PTT' (separate output), 'Squelch level' (1 (lowest)), 'Channel busy detect' (analog signal level), and 'Consecutive packets send' (in bursts). A large empty text area is at the bottom of the configuration panel.

- **“Callsign”** - the callsign of your station with SSID extension. This field can be up to 6 characters long. The default SSID is 0 and allowed SSID's are between 0 and 15,
- **“Latitude”** - north-south static position of your radio station (not used if you are going to use GPS position),
- **“Longitude”** - east-west static position of your radio station (not used if you are going to use GPS position).

For APRS beacons latitude is encoded as “XXYY.ZZD” where XX are degrees, YY are seconds (1/60 of a degree), ZZ are hundredths of seconds (1/100), D can be N for north or S for south.

Longitude is encoded as “XXXYY.ZZD” where XXX are degrees, YY are seconds (1/60 of a degree), ZZ are hundredths of seconds (1/100), D can be E for east or W for west.

You can easily find your coordinates in APRS format on aprs.fi map, they are displayed in upper-left corner of the screen.

- **“TX delay (ms)”** - delay time in milliseconds for sending an APRS packet. It is the length of preamble sent before packet and necessary for radio receivers to synchronize with our packet. The recommended value is not less than 250ms,
- **“Quiet time (ms)”** - minimum time off after the radio channel is busy. After this time next packet can be sent from the transmit buffer,
- **“Randomize quiet time”** - this option adds some randomization to Quiet time value which changes every packet sent with a range selected in this box,
- **“PTT”** - push-to-talk output selection, we can choose between two options:
 - **“separate output”** - PTT output pin is shorted to ground while broadcasting,

- **“2k2 resistor”** - audio output is shorted to ground via 2k2 Ohm resistor, function used in some handheld radios for transmission enable,
- **“Squelch level”** - this a squelch level selection, with lower values usable signal will be detected with lower signal amplitude. Default value of 1 should be the best option,
- **“Channel busy detect”** - this option selects if channel busy is detected by input signal amplitude (**“analog signal level”**) or with proper packet start detection (**“digital detect”**). The second option allows to work with open-squelch on the radio, but it can lead to some packet collisions. The first option is default and should be used in most cases,
- **“Consecutive packet send”** - this option will make your device send all the packets from TX queue one after another without any delays (**“in bursts”**), or with TX delay, Quiet time, and channel busy detection between every packet (**“separately”**). The first one is default and used by most APRS devices,
- **„Output level”** - digital control of packet audio output signal.

8.6. Position reports Tab

- **“Position report mode”** - selects position report mode,
 - **“SmartBeaconing”** - algorithm which sends position reports with intervals dependent on speed and course, allowing for more accurate reporting especially in urban area,
 - **“Fixed time interval”** - packets are sent every with a fixed time interval,
- **“SmartBeaconing”** is an algorithm invented by Tony Arnerich KD7TA and Steve Bragg KA9MVA,
 - **“Min Turn Angle [deg]”** - this is a minimum heading change which causes an immediate position report packet send,

- **“Turn Slope”** - this value is a coefficient used to modify heading change necessary for immediate position report packet send.

These two values are used in the following formula: $\text{turn_threshold} = \text{min_turn_angle} + \text{turn_slope}/\text{speed}$. Default value for min_turn_angle is 27 degrees, turn slope is 410. So if your speed is 10 km/h, packet will be sent immediately if $\text{turn_threshold} = 27 + 410/10 = 68$ degrees. If your speed is 80 km/h, packet will be sent immediately if $\text{turn_threshold} = 27 + 410/80 = 32$ degrees. Now we see that packets will be sent with lower heading changes for fast speed and with higher heading changes for low speeds. This is called “corner pegging”.

- **“Min Turn time [s]”** - this value limits a minimal time between two position packets sent even if above turn_threshold formula is met,
- **“Slow Speed [km/h]”** - if speed is below that value, packets are sent with interval of “Slow Period [s]”, corner pegging algorithm is turned off,
- **“Slow Period [s]”** - time between position packets when speed is below “Slow Speed [km/h],
- **“Fast Speed [km/h]”** - if speed is above that value, packets are sent with interval of “Fast Period [s]”, but corner pegging algorithm is tuned on too,
- **“Fast Period [s]”** - time between position packets when speed is below “Fast Speed [km/h],
- **“Fixed time interval”**,
 - **“Position report interval [s]”** - time between packets in “Fixed time interval” report mode,
- **“Position encoding protocol”** - this option defines a format of APRS position report packets,
 - **“Mic-E”** - uses both AX.25 destination address and information field for position report and is the most compact reporting method, giving the shortest packet length,
 - **“Base-91”** - second method oriented to shortest report packet length,
 - **“Plain text”** - in this method station position report is sent in human-readable plain text format,
- **“Report encoding settings”**,
 - **“Symbol code”** - APRS packet symbol code,
 - **“Symbol table”** - APRS packet symbol table,
 - **“Mic-E message”** - you can select one of predefined Mic-E messages,
 - **“Path”** - there you can define up to 3 APRS position packet path components,
 - **“Status text”** - this is a status text attached at the end of position packet,
 - **“Time”** - selects position report time format,
 - **“Send last position if fix lost”** - if enabled, device will send last known position in fixed intervals on GPS fix lost,
 - **“Send interval on fix lost (s)”** - fixed interval for packets sent on GPS fix lost if enabled,
 - **“Course/Speed”** - enables or disables course/speed reports,
 - **“Altitude”** - enables or disables altitude reports.

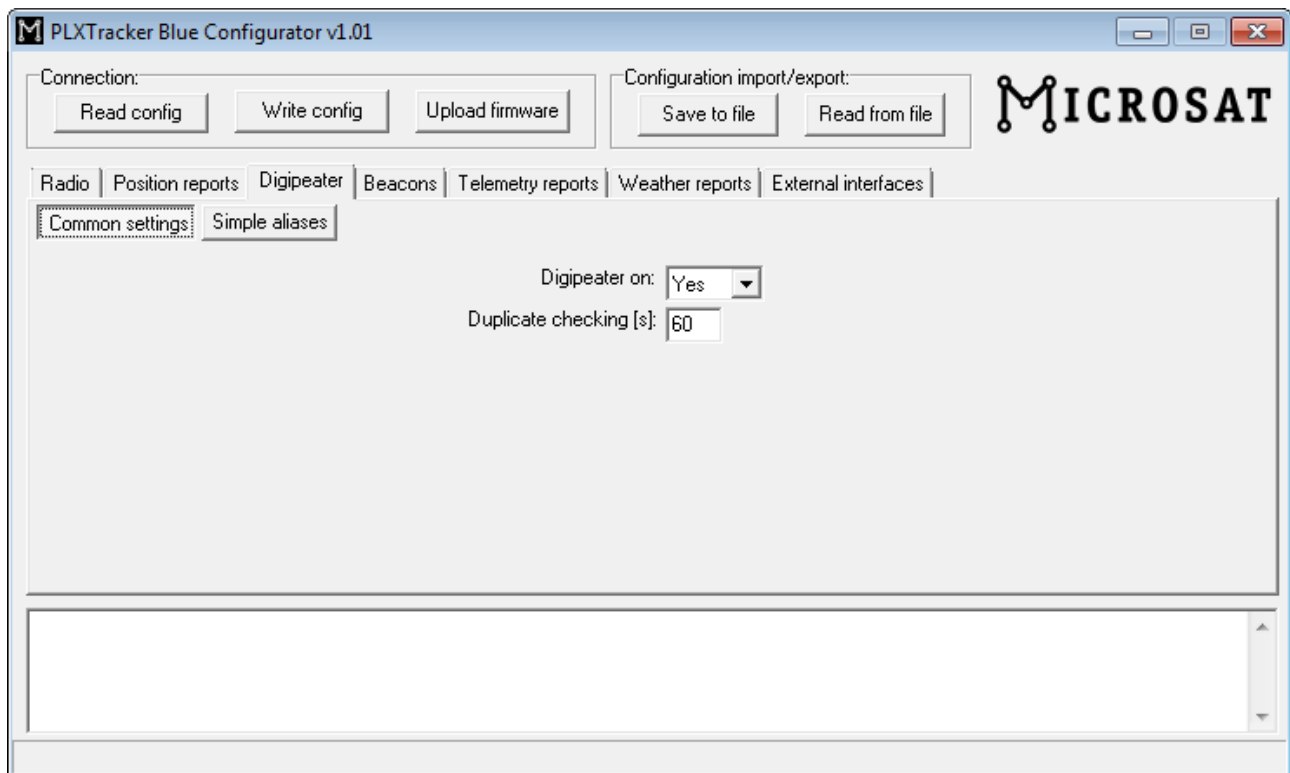
If you are new to APRS or if you are running the device for the first time, you can leave

all the options as default. By default, the device will send SmartBeaconing packets in Mic-E format with frequency dependent on your moving speed and course (if GPS is connected and valid position fix is being received).

8.7. Digipeater Tab

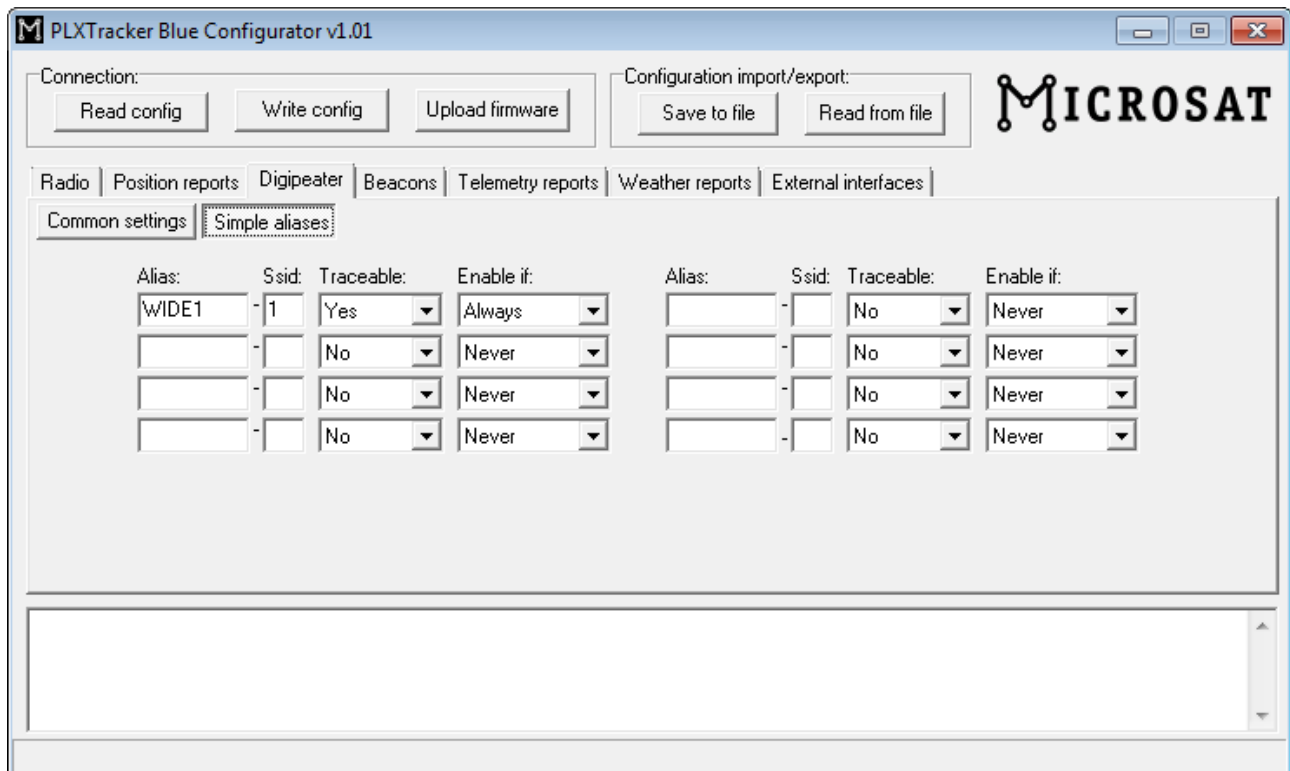
Digipeater Tab contains several options for simple alias-based digipeater feature of PLXTracker Blue.

8.7.1 Digipeater / Common settings Tab



- **“Digipeater on”** - Enables or disables a function of RF APRS packets digipeating,
- **“Duplicate checking (s)”** - Time in seconds after which the next packet will be forwarded, if it has the same content as the previous one. This function allows you to ignore stations, which are flooding the network and also to ignore packets which were heard a while before and are heard again after being digipeated by other stations.

8.7.2 Digipeater / Simple aliases Tab



- **“Alias”**, **“SSID”** - if packet path contains this string and SSID, it will be digipeated,
- **“Traceable”** - if Yes, PLXTracker Blue will add its own callsign to the path, if No, it will not do that,
- **“Enable if”** - Disables (Never) or enables (always) current row of digipeater settings.

8.8. Beacons Tab

This Tab contains settings related to device beacon packets. If you are going to use GPS position reports configured in Position reports Tab, then you can leave this Tab unchanged.

- **“Dest”** - destination field of a beacon packet. It would be the best to leave it default which will allow your device to be recognized as PLXTracker Blue (and added to working stations callsign list on my webpage),
- **“Path 1”, „Path 2”** - APRS path, if you leave these fields empty, beacon will be sent without path,
- **“RF[min]“** - beacon transmit rate to RF (expressed in minutes), beacon will not be sent if this field is set to 0,
- **„Enable if”** - beacon can be disabled (Never), enabled (Always), enabled only if device is receiving a valid position fix from GPS (GPS fixed), or enabled only if device is not receiving a valid position fix from GPS (GPS not fixed).
- **“Message“** - APRS packet information field content. Beacon will not be sent if this field is left empty.

To properly generate a message field structure, please read the information on the APRS packet format.

Example beacon packet:

```
</gpsdate>z<gpslat>/<gpslong>>PLXTracker U=<volt>V.
```

- **/** - APRS data ID,
- **<gpsdate>z** - GPS time and date in the format DDHHMM (day/hour/minute),
- **<gpslat>/<gpslong>** - station latitude and longitude obtained from GPS,
- **„/”** - together with **“>”** sign between **<gpslat>** and **<gpslong>** defines a station symbol (House in this example), please refer to the table of symbols at <http://aprs.pl/ikony.htm>,
- **PLXTracker U=<volt>V** - beacon information field, **<volt>** will be replaced with

input voltage reading.

As you can see in the above example, PLXTracker Blue allows to use some special strings, which are automatically replaced with corresponding values:

- `<statlat>` - inserts latitude defined on "Radio" Tab.
- `<statlong>` - inserts longitude defined on "Radio" Tab.
- `<gpsdate>` - inserts time received from GPS in format: DDHHMM.
- `<gpslat>` - inserts latitude received from GPS.
- `<gpslong>` - inserts longitude received from GPS.
- `<volt>` - inserts measured input voltage,
- `<temp>` - inserts temperature from thermometer connected to RS-485 bus (Celsius),
- `<tempf>` - inserts temperature from thermometer connected to RS-485 bus (Fahrenheit),

After inserting corresponding values example beacon packet will look for example like that:

```
/100214z5221.60N/01653.63E>PLXTracker U=12.1V
```

Important note: Sometimes I see that users try to replace special `<gpsdate>`, `<gpslat>`, `<gpslong>`, `<volt>` strings with direct values. You don't need to do that. If you set for example "`<gpsdate>`" string in your beacon, current date will be placed instead of this string. The same for "`<gpslat>`", "`<gpslong>`", "`<volt>`", etc.

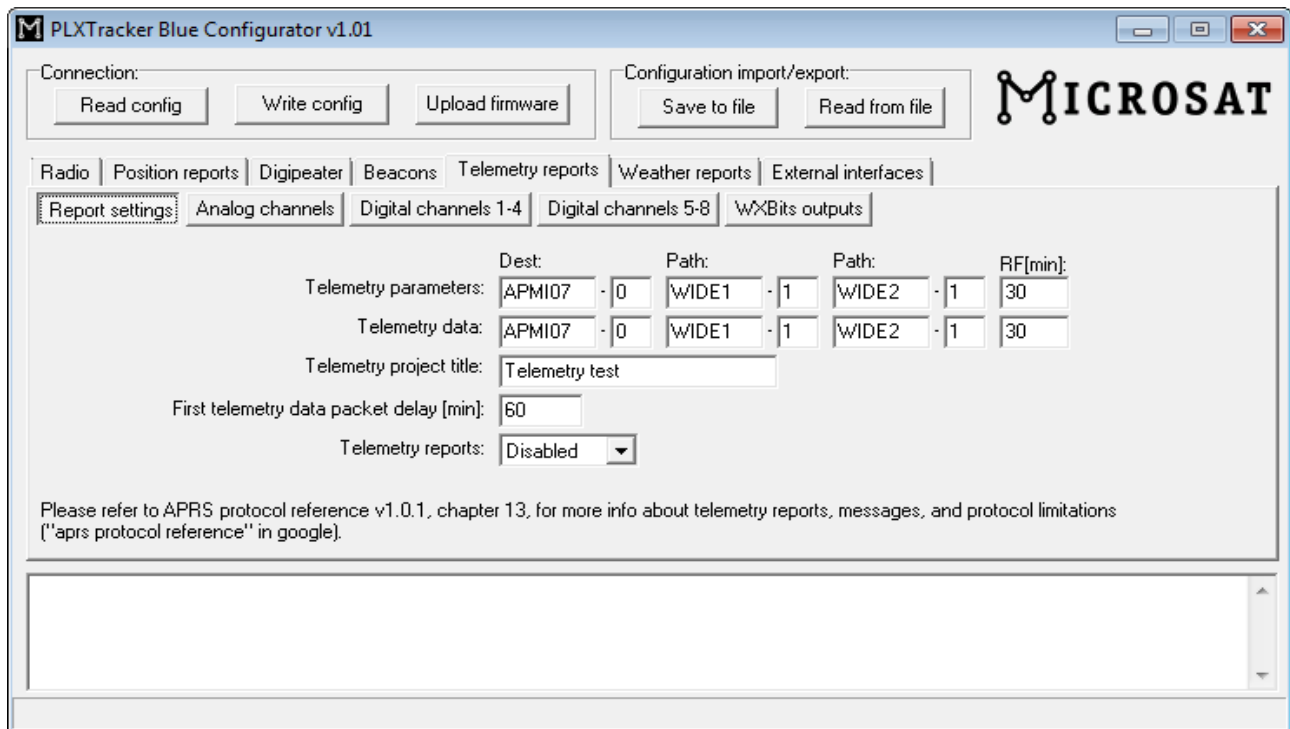
If you are new to APRS or if you are running the device for the first time, you can leave all the options as default. By default, the device will send standard beacon packet with time, latitude, longitude and current voltage reading to RF every 30 minutes (if GPS is connected and valid position fix is being received).

8.9. Telemetry reports Tab

This Tab contains all settings related to telemetry reports feature present in APRS protocol. Telemetry reports allow to send various data to APRS, which then can be plotted on graphs by WWW services connected to APRS/APRS-IS network (like well-known aprs.fi service).

You can find more useful information about telemetry reports format in APRS specification: <http://aprs.org/doc/APRS101.PDF>, page 68.

8.9.1. Telemetry reports / Report settings



PLXTracker Blue Configurator v1.01

Connection:

Configuration import/export:

MICROSAT

Radio | Position reports | Digipeater | Beacons | Telemetry reports | Weather reports | External interfaces

Report settings | Analog channels | Digital channels 1-4 | Digital channels 5-8 | WXBits outputs

Telemetry parameters: Dest: APMI07 - 0 Path: WIDE1 - 1 Path: WIDE2 - 1 RF[min]: 30

Telemetry data: Dest: APMI07 - 0 Path: WIDE1 - 1 Path: WIDE2 - 1 IS[min]: 30

Telemetry project title: Telemetry test

First telemetry data packet delay [min]: 60

Telemetry reports: Disabled

Please refer to APRS protocol reference v1.0.1, chapter 13, for more info about telemetry reports, messages, and protocol limitations ("aprs protocol reference" in google).

- **“Telemetry parameters”** - this group of fields defines settings of PARM, UNIT, EQNS, and BITS parameter packets which describe the telemetry data sent by the device,
 - **“Dest”** - destination field of a telemetry parameters packet.
 - **“Path”** - APRS path (default value of WIDE2-2 will allow your device to be digipeated by most nearby digipeaters), if you leave this field empty, packet will be sent without path,
 - **“RF[min]”** - telemetry parameters packet transmit rate to RF (expressed in minutes), packet will not be sent if this field is set to 0,
 - **“IS[min]”** - telemetry parameters packet transmit rate to APRS-IS (expressed in minutes), packet will not be sent if this field is set to 0,
- **“Telemetry data”** - this group of fields defines settings telemetry data reports,
 - **“Dest”** - destination field of a telemetry data packet.
 - **“Path”** - APRS path (default value of WIDE2-2 will allow your device to be digipeated by most nearby digipeaters), if you leave this field empty, packet will be sent without path,
 - **“RF[min]”** - telemetry data packet transmit rate to RF (expressed in minutes), packet will not be sent if this field is set to 0,
 - **“IS[min]”** - telemetry data packet transmit rate to APRS-IS (expressed in minutes), packet will not be sent if this field is set to 0,
- **“Telemetry project title”** - this is a project name associated with the telemetry station, sent in BITS message,
- **“First telemetry data packet delay [min]”** - delay for the first telemetry data packet, it allows to gather data (especially for 10-min and 60-min report) after device power-up.
- **“Telemetry reports”** - if disabled, no telemetry parameters or telemetry data

messages will be sent,

8.9.2. Telemetry reports / Analog channels

This Tab defines 5 analog channels available in APRS telemetry reports. You can select many different sources for channels data, including external WXTelemetry module data (M), other external devices data (E) and device internal data (I).

	Name:	Units:	Source:	Coeff A:	Coeff B:	Coeff C:	1 min average:	
Analog 1:	Vin	Volt	(I) Input voltage	0	0.075	0	Off	Default parameters
Analog 2:	A2	None	Not used	0	0	0	Off	Default parameters
Analog 3:	A3	None	Not used	0	0	0	Off	Default parameters
Analog 4:	A4	None	Not used	0	0	0	Off	Default parameters
Analog 5:	A5	None	Not used	0	0	0	Off	Default parameters

Note: types of telemetry sources are: (M) - from WXTelemetry, (E) - from external devices, (I) - device internal sources.

- **“Name”** - short name for the channel, length is limited by APRS standard and is different for different channel numbers,
- **“Units”** - short name for units of data for the channel, length is limited by APRS standard and is different for different channel numbers,
- **“Source”** - most important field, specifies what data will be send for particular channel, (M) sources are associated with WXTelemetry module and (I) sources are associated with device's internal statistics and measurements,
- **“Coeff A”, “Coeff B”, “Coeff C”** - these coefficients tell recipients of the telemetry reports how to decode analog value in following form:
$$\text{value} = A * \text{data} * \text{data} + B * \text{data} + C,$$
- **“1 min average”** - only functional for external telemetry module data (M), defines if reported values are averaged or not,
- **“Default parameters”** - you should click this button after you change data source for the channel. It loads default coefficient values and name/units for a channel.

Important note: If you change “Source” field for a channel, you also need to change “Name”, “Units”, and “Coeffs” fields. Otherwise everyone else who receive your data (including web services like aprs.fi) will not be able do decode your reports properly. Every source has its own coefficients, names and units. With USB Configurator you can get default settings by clicking on “Default parameters” button. When changing configuration via WWW you need to set these fields manually (you can always write

them down from USB Configurator application).

8.9.3. Telemetry reports / Digital channels

This Tab defines 8 digital channels available in APRS telemetry reports. These tabs should be used when WXBits module is connected.

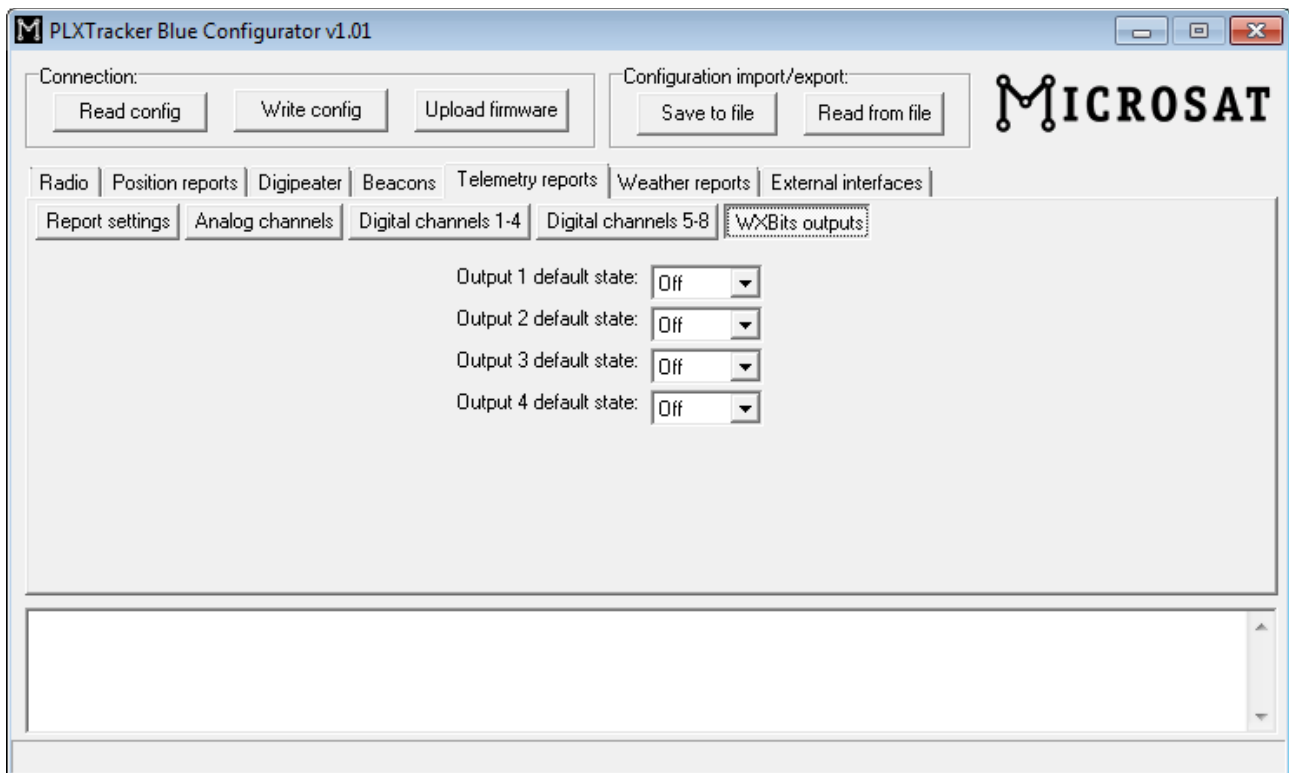
	Name:	Label	Source	Inverted:	
Digital 1:	01	0n	(M) Output 1	No	Default parameters
Digital 2:	02	0n	(M) Output 2	No	Default parameters
Digital 3:	03	0n	(M) Output 3	No	Default parameters
Digital 4:	04	0n	(M) Output 4	No	Default parameters

- **“Name”** - short name for the channel, length is limited by APRS standard and is different for different channel numbers,
- **“Label”** - short label for the channel, length is limited by APRS standard and is different for different channel numbers,
- **“Source”** - most important field, specifies what data will be send for particular channel, (M) sources are associated with WXBits module,
- **“Inverted”** - defines if channels should be reported as inverted in BITS telemetry message,
- **“Default parameters”** - you should click this button after you change data source for the channel. It loads default coefficient values and name/units for a channel.

Important note: If you change “Source” field for a channel, you also need to change “Name”, “Label”, and “Inverted” fields. Otherwise everyone else who receive your data (including web services like aprs.fi) will not be able do decode your reports properly.

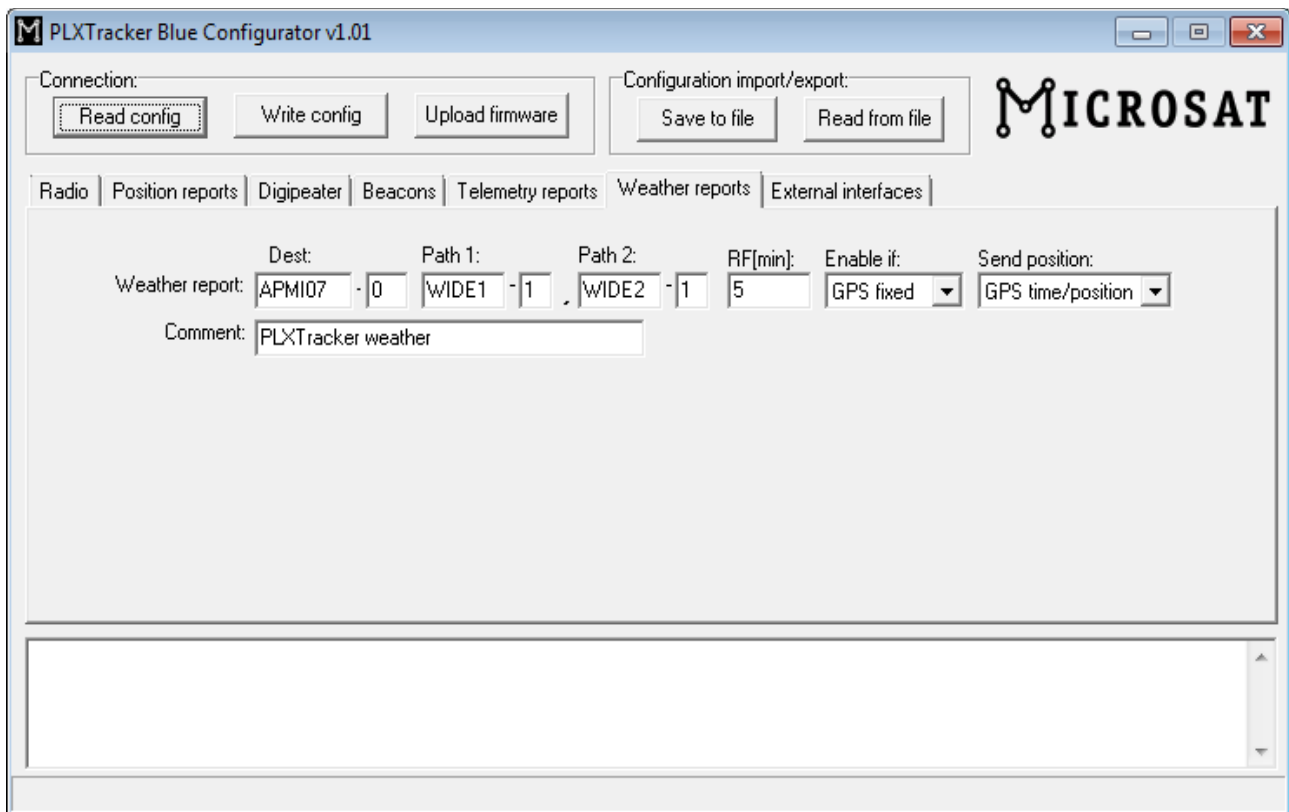
8.9.4. Telemetry reports / WXBits outputs

This Tab defines what should be the default states for WXBits output channels. After power-up PLXTracker Blue will constantly try to set outputs to the defined states, unless changed by the user.



8.10. Weather reports Tab

This Tab contains all settings related to weather station reports. If you are not going to use weather station, you can leave it unchanged.

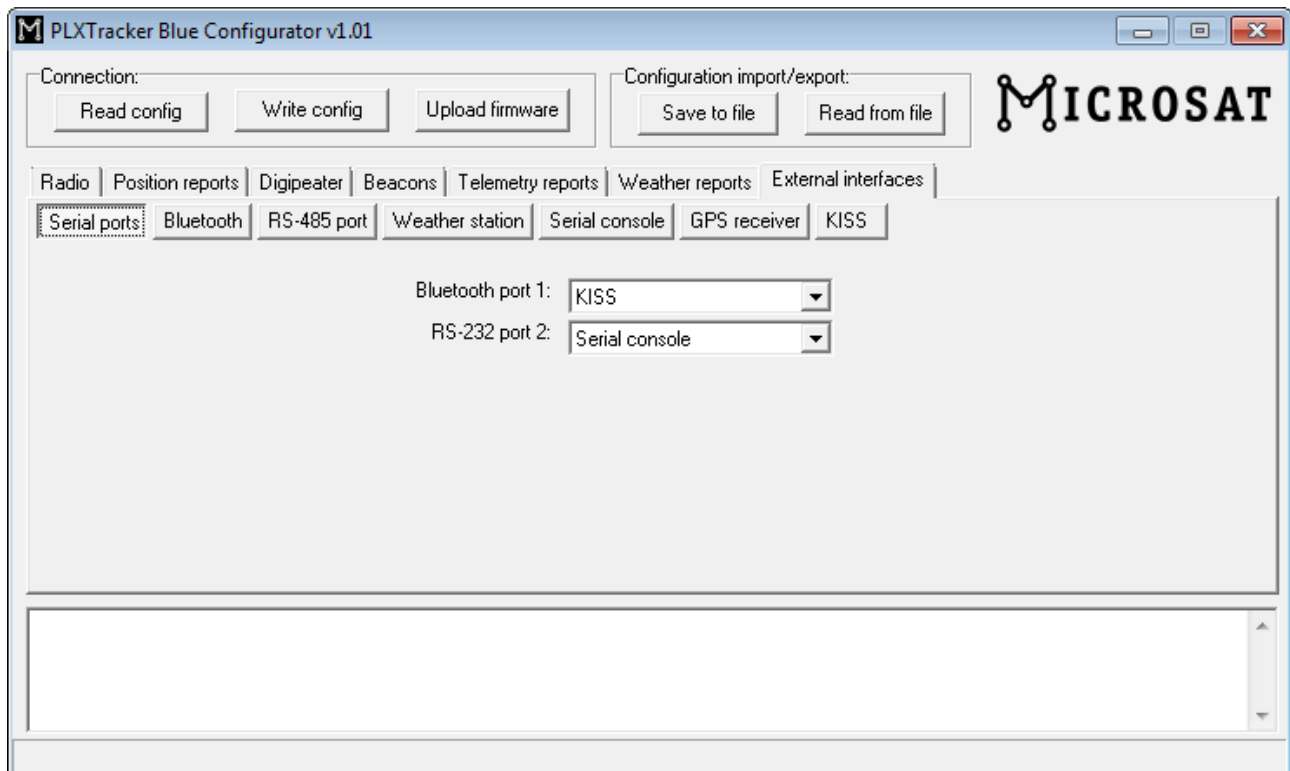


- **“Dest”** - destination field of a weather report packet.
- **“Path1”, “Path 2”** - up to 2 components of APRS path, if you leave this field empty, packet will be sent without path,
- **“RF[min]”** - packet transmit rate to RF (expressed in minutes), packet will not be sent if this field is set to 0,
- **“Enable if”** - beacon enable condition:
 - **“Never”** - beacon disabled,
 - **“Always”** - beacon enabled,
 - **“GPS fixed”** - beacon enabled if GPS fixed,
- **“Send position”** - selects between dynamic position from GPS and static position from “Latitude”, “Longitude” fields in “Radio” Tab,
- **“Comment”** - weather report comment.

8.11. External interfaces Tab

This Tab is related to settings of all external devices and interfaces which may be connected to PLXTracker Blue.

8.11.1. External interfaces / Serial ports

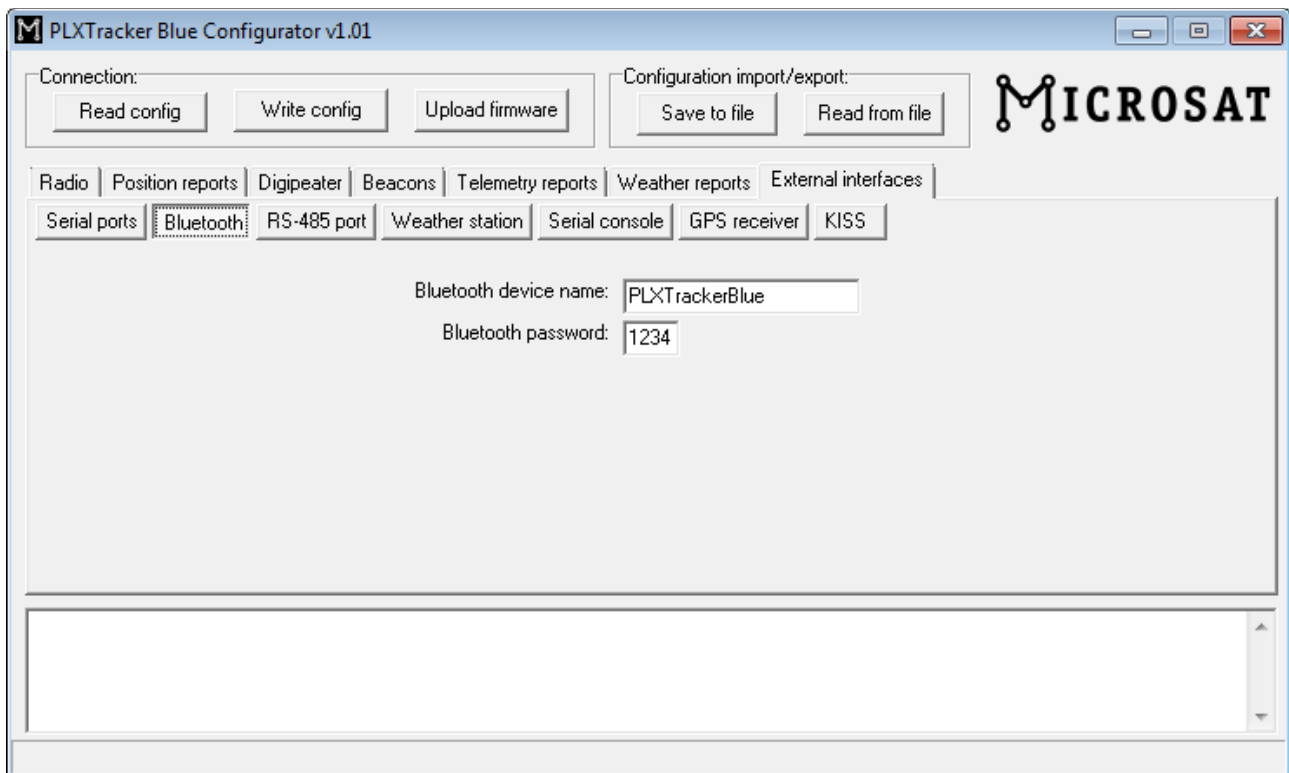


- „Bluetooth port 1” - selection of Bluetooth port function,
- „RS-232 port 2” - selection of RS-232 port function.

PLXTracker Blue incorporates only one RS-232 serial interface (contrary to PLXTracker or PLXDigi), located on „RS-232 port 2” pins (port 1 is internally occupied by Bluetooth interface).

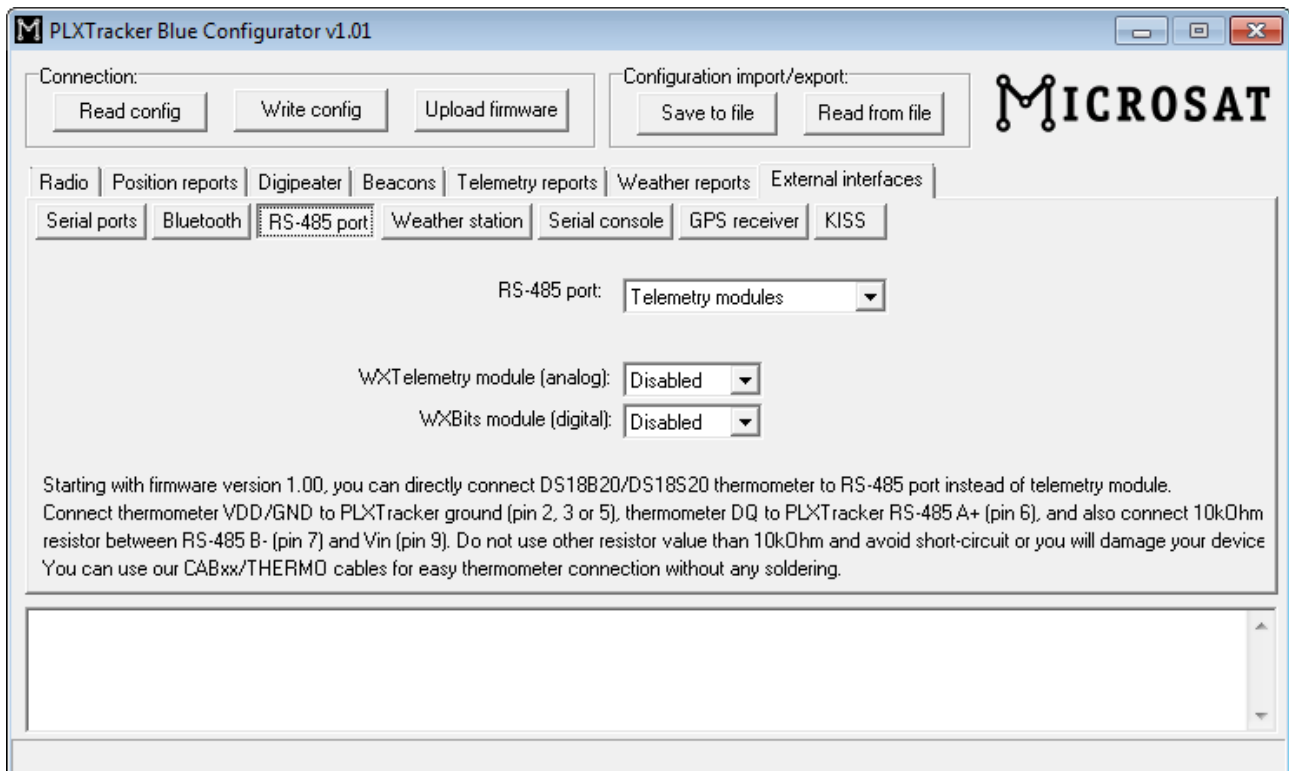
If you wish to use a weather station, serial console, etc., you will need a different set of serial cables than for PLXTracker or PLXDigi.

8.11.2. External interfaces / Bluetooth



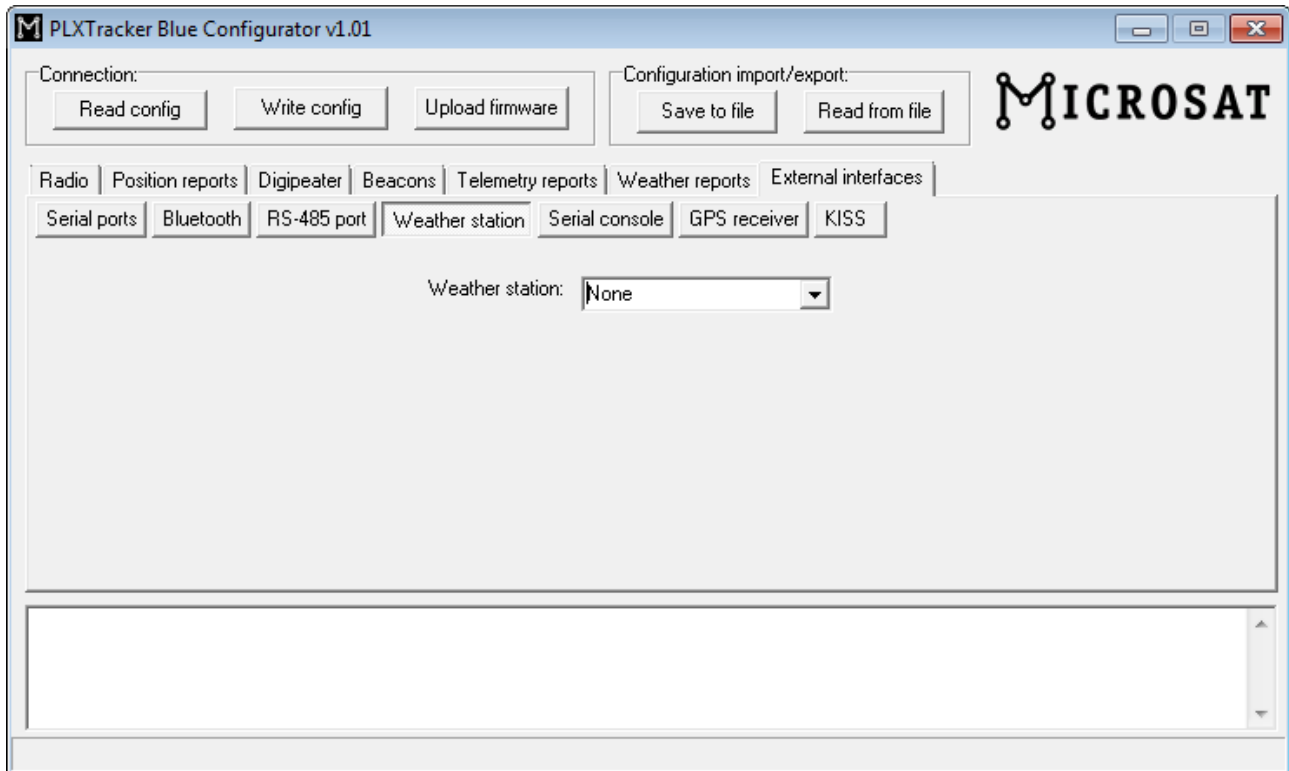
- „**Bluetooth device name**” - Bluetooth name which will be displayed on your laptop/smartphone/tablet when searching for nearby Bluetooth devices,
- „**Bluetooth password**” - Bluetooth password which needs to be entered while pairing with PLXTracker Blue Bluetooth interface.

8.11.3. External interfaces / RS-485 port



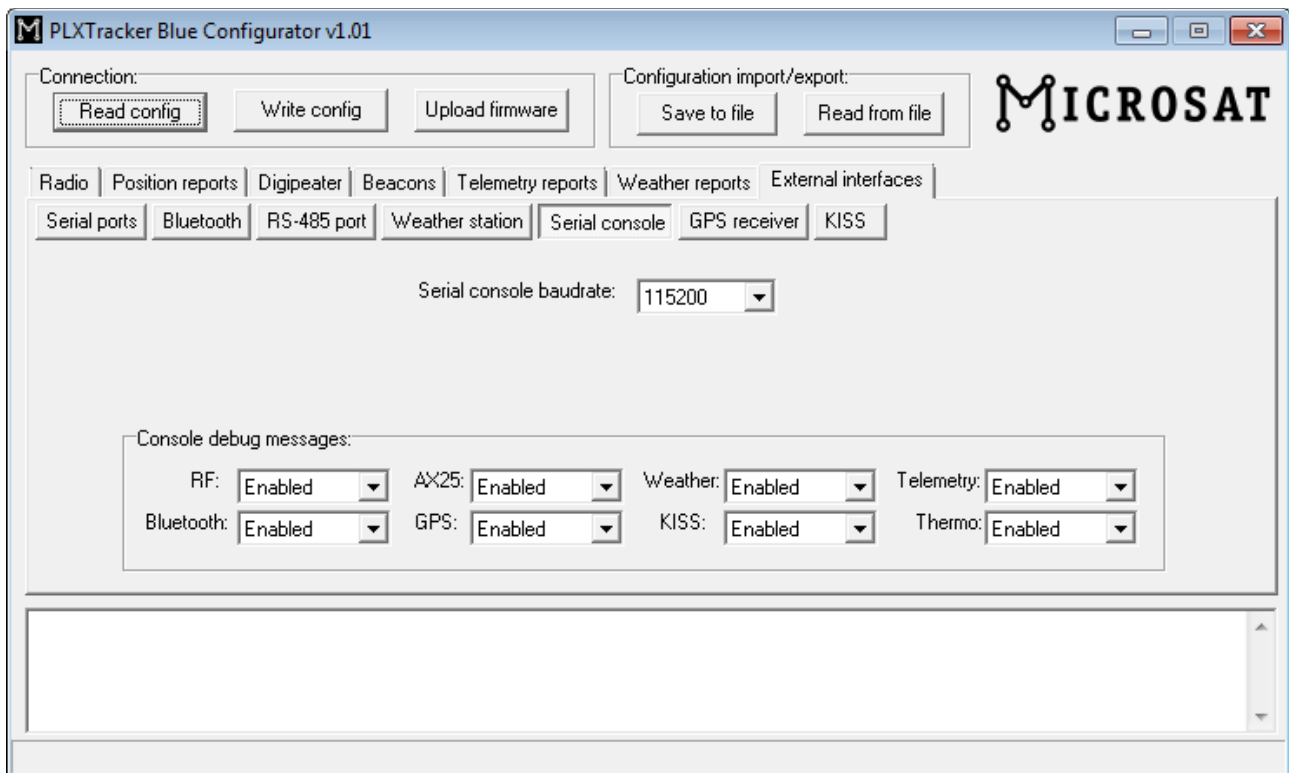
- „**RS-485 port**” - selects between telemetry modules or digital thermometer connected to RS-485 bus,
- “**WXTelemetry module (analog)**” - enables or disables WXTelemetry module support,
- “**WXBits module (digital)**” - enables or disables WXBits module support,

8.11.4. External interfaces / Weather station



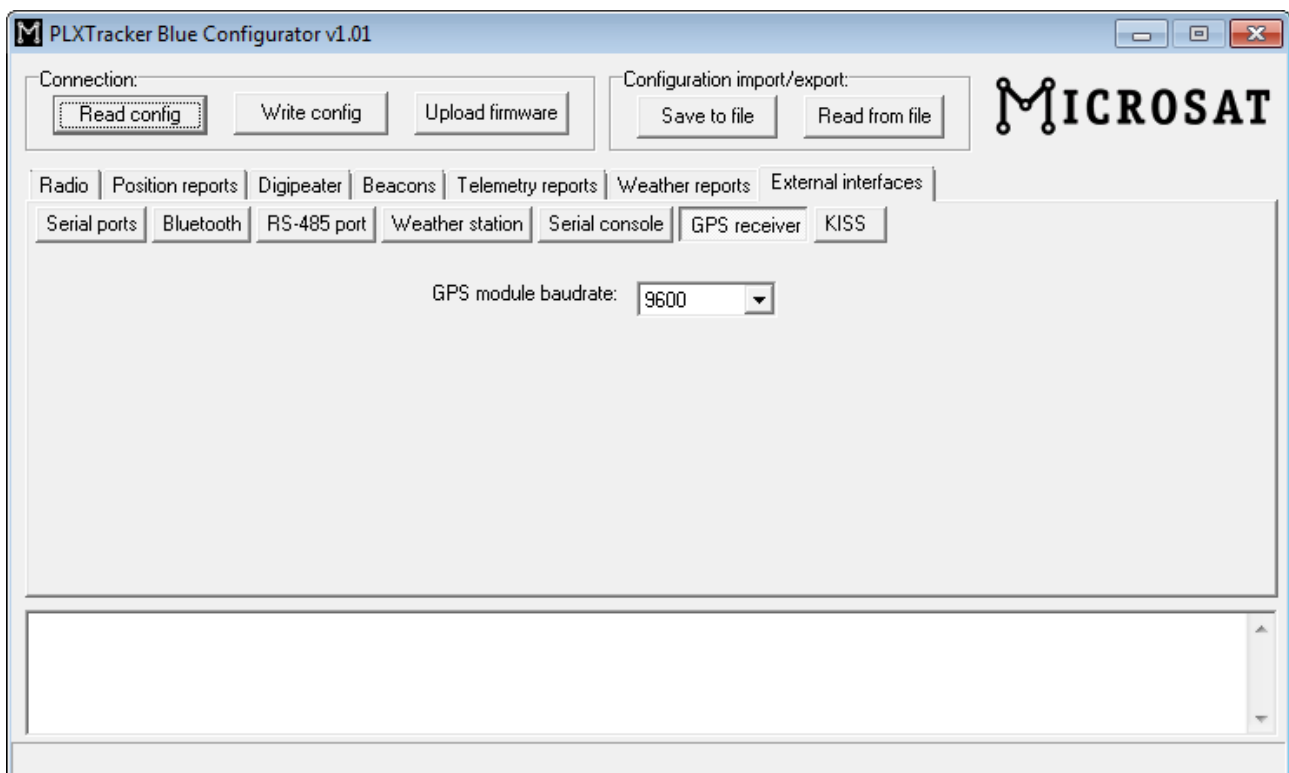
- “**Weather station**” - selection of weather station model connected to PLXTracker Blue serial port.

8.11.5. External interfaces / Serial console



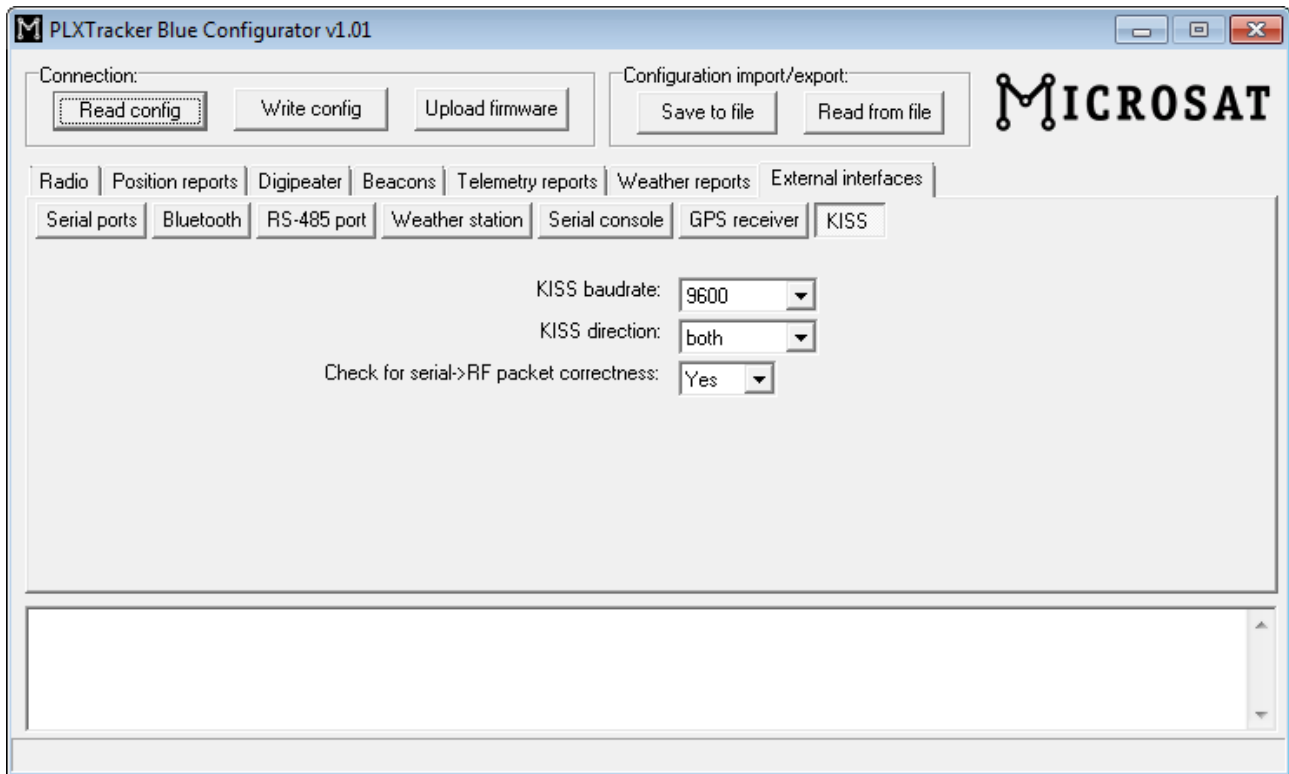
- „Serial console baudrate” - selection of serial console communication baudrate,
- „Console debug messages” - enables or disables various debug messages printed via serial console.

8.11.6. External interfaces / GPS receiver



- **“GPS module baudrate”** - selection of GPS receiver communication baudrate.

8.11.7. External interfaces / KISS

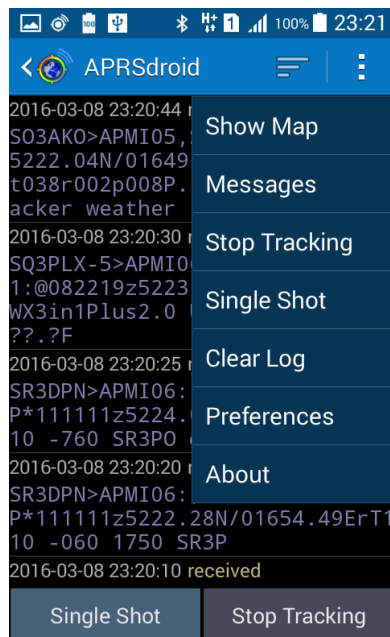


- **“KISS baudrate”** - KISS packet communication baudrate setting,
- **“KISS direction”** - selects KISS direction:
 - **“RF->serial”** - only output of packets received from RF,
 - **“both”** - output of packets received from RF and forwarding of packets from serial port to RF,
- **“Check for serial->RF packet correctness”** - if enabled, then it tries to prevent forwarding of corrupted or malformed packets from serial port to RF.

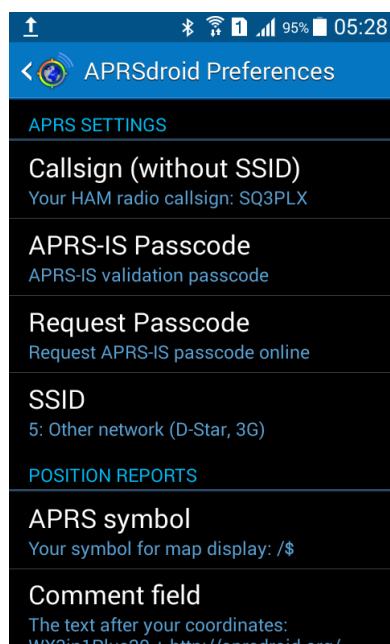
9. Bluetooth connection with APRSDroid

In this chapter I will show you how to pair PLXTracker Blue Bluetooth interface with APRSDroid application running on a smartphone. My testing platform is Samsung Galaxy Grand Neo GT-i9060 hosting Android Kitkat 4.4.4.

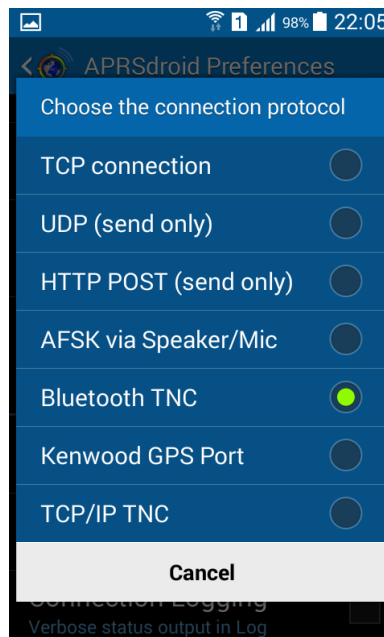
After running APRSDroid we need to get into application preferences. To do it we need to press on three dots in upper-right corner of the screen. On my phone the same result occurs after pressing for few seconds a button in lower-left corner of smartphone surface. A menu will open:



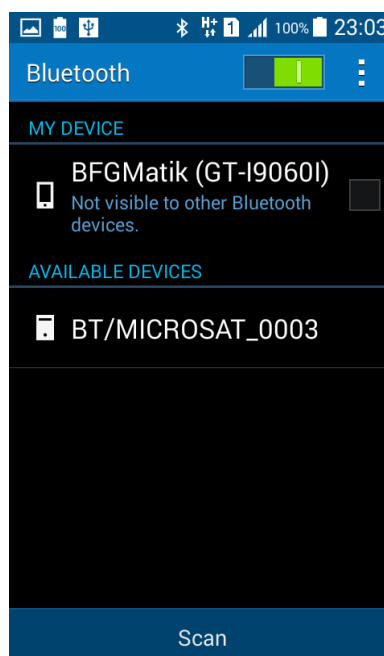
Next we need to press on „Preferences” to see something like this:



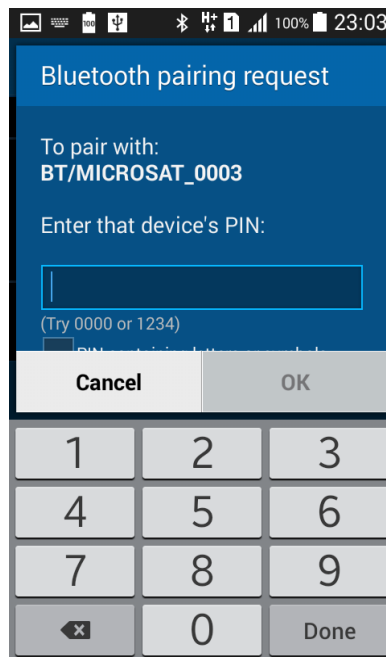
Here you can set your Callsign, SSID, APRS symbol and Comment field of your position beacons. Now scroll down to „Connection protocol” field and press it:



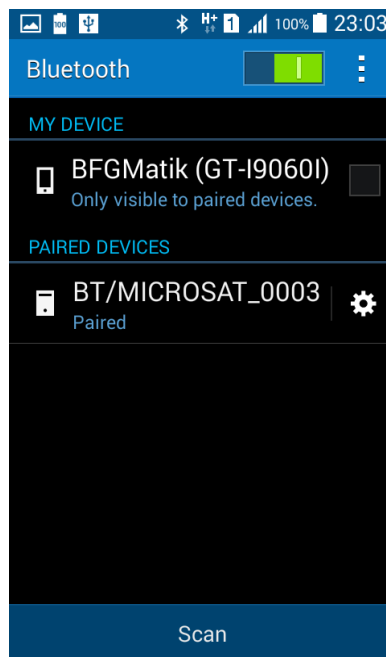
You need to select „Bluetooth TNC” option in „Connection protocol” setting. After making a selection you will get back to Preferences menu. Now click on „Bluetooth settings” option. Enable Bluetooth if necessary and you should see a list of available Bluetooth devices. With PLXTracker Blue you will see „PLXTrackerBlue” name instead of „BT/MICROSAT_0003”.



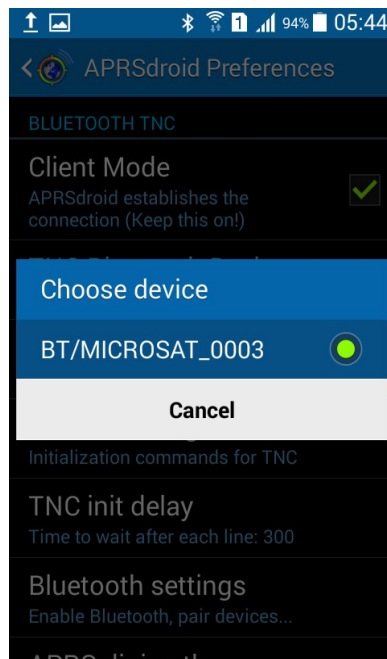
Now click on discovered device to pair with it. You will be asked for device's PIN (default: 1234 if you didn't change it in PLXTracker Blue configuration).



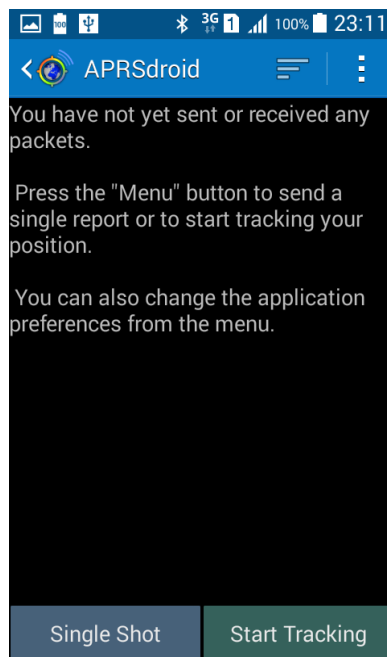
After successful pairing you should see the device as paired:



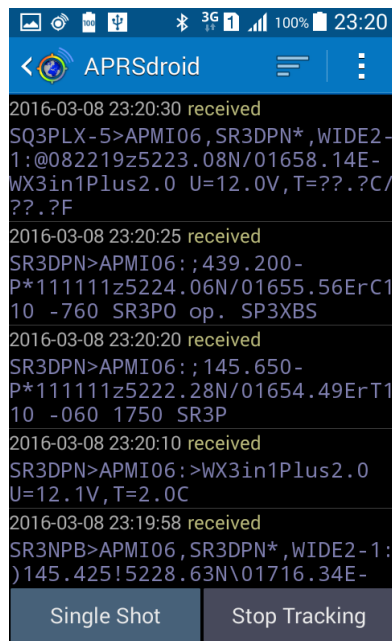
Now get back to Preferences, press on „Connection Preferences” and then on „TNC Bluetooth device” to select your paired device as preferred Bluetooth TNC (again you will see PLXTrackerBlue instead of „BT/MICROSAT_0003”):



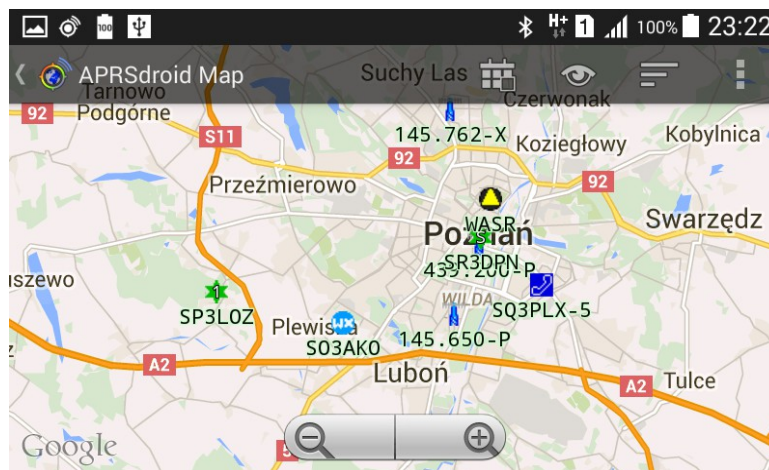
Now you can leave the menu and get back to the main window of APRSDroid application:



Press on „Start Tracking” and you should get a confirmation that connection was opened correctly. Your PLXTracker red/green led should start to blink in green. After some time you should start to see received packets:



You can also open a map to see received stations plotted on Google Maps:



And that is all about connecting your PLXTracker Blue to APRSdroid. With APRSdroid you can also send and receive messages from other APRS users or use your smartphone's built-in GPS to make a SmartBeaconing tracker without additional external GPS receiver. Good luck in revealing other APRSdroid features.