

Technical Guidelines and Information



Version 31.00
15 April 2023

Hardware Requirements

In order to use TotalVario, you need

- a smartphone with high quality GPS running Android 4.3 or later (5.0 or later is recommended for security and Bluetooth transmission robustness)
- and either a built-in pressure sensor with high quality, or a compatible Bluetooth mini-variometer / baro-sensor

Especially if you fly a hangglider, it will be very useful to have a smartphone with proximity sensor or ambient light sensor when running a navigation app in parallel.

Just [contact](#) me for support.

Bluetooth Sensors

Generally, the Bluetooth interface understands following protocols:

- \$PTVSOAR and \$PTV (TotalVario), \$BFX and \$BFV (Bluefly), \$XCTRC (XCTracer), \$LXWP0, \$LK8EX1, \$POV (OpenVario, S and R not supported), _PRS/_BAT (Flynet), \$PCPROBE (C-Probe), and basic PRS (raw pressure, in case the sensor can only provide this).
- The protocols \$BFX, BFV, \$XCTRC and \$PCPROBE are **device-specific**. TotalVario has special built-in support for these devices. **Therefore these protocols must not be used to send data from other device types.**
- If you build your own sensor probe for use with TotalVario, you can use the \$PTVSOAR/\$PTV protocol.
- **TotalVario works perfectly with the new [FlyingSilicon Sensor Probe](#).** This is currently being actively developed. In-flight tests have already been carried out. This probe uses BLE and high-end sensor chips, and the results are amazing.
- TotalVario works with [BlueflyVario](#) V10, V11 and V12. Newer V2* models should work as well. The **Pitot airspeed extension** is supported for V12 (and with some restrictions for V11). For reliable, precise airspeed it is strongly recommended to use Bluefly Version 12. There are different V12 firmware versions! Please make sure [that your version is the same that I used](#). The Bluefly should report version BFV 12 416. *For V11 there is currently no way to properly compensate the Pitot sensor temperature drift. Be aware that you may get wrong speed values depending on the temperature change after zeroing the Pitot.*
- TotalVario works well with the [XCTracer](#) using the \$XCTRC protocol.

Please be aware that the [Compass C-Probe](#) and several Bluetooth mini-varios available on the market should work, but have not been tested with real hardware yet.

Vane Wheel Airspeed Sensor

Depending on your smartphone model you may be able to use a [Naviter/Flytec](#) Vane Wheel Sensor for precision airspeed using an adapter cable.

In the speed and height range relevant for hanggliders and paragliders, these sensors are definitely more precise than classical differential pressure sensors and they do not need things like zeroing or temperature compensation. The tradeoff is that they are mechanically less robust (the higher the airspeed the shorter the lifetime). Due to very low friction the measured speed is equivalent to true airspeed (TAS).

This feature has been tested successfully with several different Samsung and Motorola smartphones. However there is no guarantee that it works with any other type of smartphone.



If you already own a sensor of this type or if you can get a cheap, used one, you can give it a try.

The sensor generates an analog sine signal, and the airspeed can be calculated from the frequency. It must be connected to the headset/microphone input. For this, an **adapter cable** is needed, or the original connector must be replaced.

Processing and digitalization of the analogue signal has been optimized both for power consumption and quality, but consumes some extra battery power, of course.

Important:

In this configuration, apps producing their sound on the Android media stream will stay silent! This is because Android routes most sounds (all media, phone calls, ...) forcefully to the headset, as soon as a something is plugged in. For example, you will not hear any XCSoar sounds.

When using TotalVario with a vane wheel sensor, **be sure that no other app has locked the microphone input.** Check that any voice control features, such as SVoice, Google Assistant, etc. are disabled / stopped.

The TotalVario app can use either the Android media channel or the Android alarm channel. The latter is generally routed to the external speaker on most devices. On newer smartphones you may be able to configure the audio routing in a more flexible way.

To try if this basically works with your smartphone model,

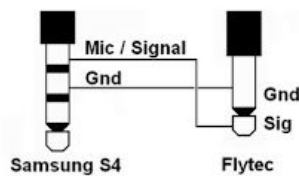
- Connect a headset to the smartphone,
- Open the **VOLUME** settings,
- Verify that TotalVario is configured to use the Android alarm channel, and
- Activate a test sound using one of the check boxes.

How to make an adapter cable

Be aware that there are different wiring schemes and internal circuits for different models from different manufacturers - even if the same connector type is used!

Smartphones like the Samsung S, A or Note typically have 3.5 mm 4-conductor (TRRS) connectors with the newer connection scheme (L,R,GND,MIC).

For example, you can build an adapter from two extension cables. You need one with a **female TS**, and one with a **male TRRS** (please do not misinterpret the diagram below). Just cut the extension cables as needed and solder up the two pieces. Be sure to connect Gnd to Gnd and Signal to Microphone as shown in the circuit diagram. In any case I recommend to use gold plated connectors.



This should be done carefully and by someone experienced in electronics. Later, if everything is working well, and if the speed sensor is going to be used only with the smartphone, you can completely replace the original connector.

After the sensor is connected, use the function **[Check Vane Wheel Sensor]** in the TotalVario **Configuration** menu to verify its correct operation. If that doesn't work (no signal), try swapping the two wires of the IQ sensor cable. Some smartphones also don't work at all with only the microphone input connected, they need a connection with minimal load on the two rings/pins for the headset. Again someone experienced in electronics can easily handle this by adding two small resistors.

BlueflyVario Temperature Compensation

This chapter refers to the usage of the **Bluefly Vario V12 with Pitot airspeed extension**. If you want to use this optional extension, please check first the [Bluefly Vario blog](#). For V11 there is no way to properly compensate the residual temperature drift of the Pitot sensor, so you can only rely on the manufacturer's pre-compensation.

For obvious reasons, NO LIABILITY is assumed for consequential damages, injuries or loss of life resulting from the procedure described here!

There are different V12 firmware versions! Please make sure that your V12 firmware is up-to-date. I use **Bluefly Vario_12.M16** (BlueflyVario_Bluetooth_USB_12.416.hex).

The Pitot extension uses the well known differential pressure sensor 4525DO-DS5AI001DP. It is produced by TE Connectivity / Measurement Specialities, and it can also be obtained from [Mouser](#).

Unfortunately some of these ICs (at least the ones I got) seem to have a residual temperature drift which cannot be ignored at the low speeds we are flying with (see the note at the end of this section). **The drift curve is basically different for each single sensor coming from the factory.** For this reason, TotalVario has a built-in temperature drift analyses and compensation feature.

Before you can reliably use the Pitot sensor, run the residual temperature drift analyses. This is needed even if you fly at rather constant air temperature (about the same altitude), because the DP sensor will start warming up during operation, and because the pod (whatever type) will warm up significantly in sunlight and during operation as well. If you are lucky, you may notice that your sensor chip is already sufficiently compensated by the manufacturer and the drift remains within a few Pa.

During the analyses, the residual temperature drift of the DP sensor is measured and stored.

You will need a deep freezer and a modern oven with precise electronic temperature control. It must be possible to limit the temperature reliably. Never exceed 50 °C. Possibly you can use a wall heater instead.

Like smartphones, the Bluefly Vario uses a stock LiPo battery. **Be aware that these batteries are generally sensitive to temperature and the battery lifetime may be shortened by wrong treatment.** NEVER use a hot oven or anything else that you cannot touch with your hands!

It is important that you put the Bluefly Vario with the connected Pitot into a suitable box (or pod), not only to protect the pressure ports, but also to avoid too rapid temperature changes.

Preparations:

- Charge the Bluefly Vario and the smartphone to 100%.

- Run the app and connect the Bluefly Vario.
- The connection must be stable even when you put the box with the Bluefly Vario into the oven or into the deep freezer.
- The box must be placed in the middle of the oven, away from the walls or heater.

Now start a quick test run. Open the page **[Pitot Temperature Compensation]** and push **[Start]**. The sensor will be zeroed once again and the measurement will begin.

On top you can see essential data. The current and average differential pressure are displayed in a graph by a moving dot and a line. At the end, the graph will show the temperature drift over the measured range.

If you wait a minute, probably the pressure indicator dot will start to 'draw' the graph because the sensor is warming up slowly. Try to put the Bluefly Vario into the freezer and into the oven, each only for a minute or so, and observe how the graph is drawn. If this works well, the compensation process can start. Push **[Abort]** if you want to run the full compensation later.

We need to run a complete temperature cycle over the full range, starting with a properly zeroed DP sensor at stable, constant room temperature. We need to cool the sensor slowly down to about -5°C , then warm it slowly up back to room temperature, and finally warm it slowly up to the maximum temperature.

The best results are obtained if the temperature changes slowly, because for each temperature step multiple sensor values are read and averaged. Therefore,

- NEVER put the Bluefly Vario directly from the fridge into the oven or vice versa !!
- NEVER cool the Bluefly Vario down to less than -5°C !!
- NEVER warm the Bluefly Vario up to more than 45°C !!

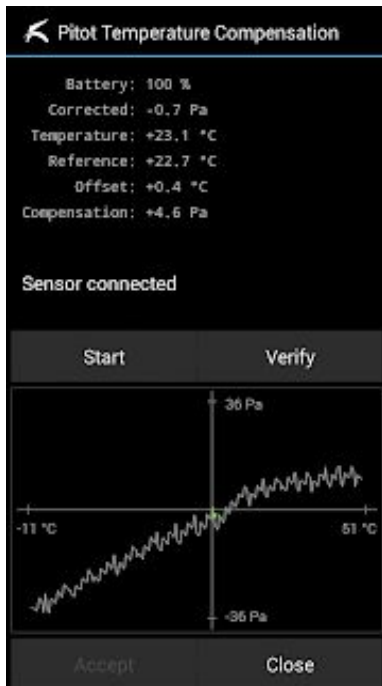
Start with putting the box with the Bluefly Vario into the deep freezer. At the latest when -5°C are reached, get the box out of the fridge. Do NOT open the box. Wait until temperature is close to room temperature again. Observe how the compensation curve is drawn.

The same procedure is then done for the upper temperature range in the oven. Start with the oven switched off! The slower the sensor temperature increases, the better.

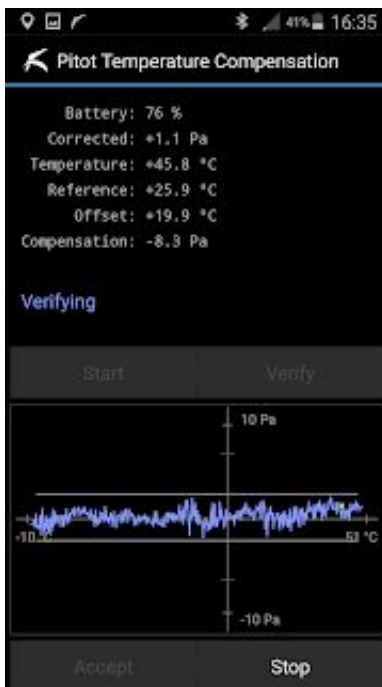
At the latest when 45°C are reached, you should see a nice, complete compensation curve. Push **[Accept]** to store it forever. That's it.

You can repeat the compensation run at any time, but it is normally needed only once for a specific sensor chip. So far I did not observe any aging effects (over 48 months).

The final compensation will look similar to the one I found. The overall average slope may differ. Note that I even went from -10 to $+50^{\circ}$. Fortunately the Bluefly Vario, the DP sensor and the LiPo did survive.



In case you have some doubt whether the drift compensation is good, you can use the **[Verify]** button. Just start and observe the graph while the temperature changes. Again, a curve for the residual drift will be drawn, but it should be very small, similar as my result:



Why is it so complicated? - a short explanation

Pitot sensors measure the differential pressure between the static air pressure and the dynamic air pressure caused by the air hitting the Pitot tube in flight. Especially at low speeds, reliable measurement is very difficult, even with modern membrane based precision sensor chips, due to the extremely small differences in the order of magnitude of a few Pascal. In addition the square root dependency between speed and pressure is most critical at low speeds (unfortunately physics works against us here).

TotalVario PTVSOAR/PTV Protocol Specification

PTVSOAR Message

In general, the protocol uses an NMEA-like message structure with \$PTVSOAR as the protocol identifier, followed by a comma (,).

The payload consists of a series of subsequent data fields. The order of the fields can be arbitrary. Each field consists of a unique *tag* (usually 3 letters) followed by a *value*. Each *tag* and each *value* is followed by a comma (,) except the last one.

```
$PTVSOAR,<tag>,<value>,<tag>,<value>,<tag>,<value>,...*<cksum>
```

The values can be plain ASCII, signed integer numbers, or signed floating point numbers, with a period (.) as the decimal separator. The values must not contain any of the special characters (\$ * ,).

As usual, a message can (and should) be terminated with an asterisk (*) followed by the hexadecimal NMEA checksum. If the checksum is not present, the verification is skipped and the data is accepted without verification.

Messages must be separated by line breaks (\n).

Following tags are defined. Others are reserved for future extensions (button/event identifier etc).

MNA	Manufacturer name, ASCII, except [\$*,]
MMO	Device Model, ASCII, except [\$*,]
MSN	Device serial number, ASCII, except [\$*,]
OAT	Outside air temperature in deg C
OAH	Outside air humidity in percent RH
PRS	Static pressure in hPa
PIT	Pitot differential pressure in Pa
VOL	Battery voltage in V
PCT	Battery capacity in percent
CHG	Battery charging flag (1=charging)
VAR	Vario in m/s
TEV	TE Vario in m/s

Example:

```
$PTVSOAR,OAT,21.4,OAH,42.42,PRS,1013.25,PIT,88.456,PCT,50,VAR,1.234
```

More fields can be defined in the future. MNA, MMO, MSN, and VOL are so far (Beta-29) ignored.

If VAR or TEV is sent, the internal calculation of the vario value from PRS is disabled and the vario value is taken as it is. If TEV is sent (VAR is ignored), the internal TEV calculation is skipped and the total energy vario value is taken as it is.

If PRS is provided and VAR / TEV are not present, TotalVario activates the internal option to tune the Kalman filter for vario calculation.

For processing real-time data such as PRS, PIT, VAR, TEV, it is strongly recommended to send records at a constant rate of 10/sec or more. It makes little sense to use a rate higher than 20/sec, although this also works.

PTV Short Message

If the Bluetooth/BLE connection is too slow for whatever reason, the alternative short form \$PTV can be used to send the essential data with less overhead.

As usual, a message can (and should) be terminated with an asterisk (*) followed by the hexadecimal NMEA checksum. If the checksum is not present, the verification is skipped and the data is accepted without verification.

Messages must be separated by line breaks (\n).

The format is defined as

```
$PTV,<pitot>,<baro>,<temp>,<humid>,<batt>,<chg>*<cksum>
```

<pitot> : differential pressure in Pa

<baro> : static pressure hPa

<temp> : air temperature in C

<humid> : relative air humidity in %

<batt> : battery capacity in %

<chg> : 1=charging, 2=not charging

<cksum> : NMEA checksum (2 hex digits)

Technical Data

Operating Time

Typical operation time is in the order of magnitude of 1.5 to 2 hours per 1000mAh true battery capacity depending on the smartphone model and the configured features.

Note that the battery capacity is specified at room temperature, it is usually less at lower temperatures due to the LiPo batteries used. You can expect about 25% less at 5°C (just a rule of thumb).

On my Note 3 smartphone, TotalVario runs about 6 hours in flight until the 15% warning appears (tested with new 3400mAh battery, temperature > 15°C, Bluefly connected, No Screensaver, Brightness 100%, Volume 100%, Network off, without secondary app).

The time depends also very much on display brightness, size, resolution, on vario volume, usage of live tracking / network, etc. Using the screen saver, the operating time can be significantly extended.

For longer flights, or at low temperatures, you may need a power bank. However with a built-in battery of 6000 mAh or more (Moto G9, Samsung M51) you can reach 10 hours flight time. This is also the operating time of the BlueflyVario with standard battery (buzzer off).

Tested Hardware

Smartphone	Android	Baro Sensor	Sensors/Notes
Samsung S4	4.3	ok, 15-20cm	temp, humidity
Samsung Note3	5.0	good, 10-15 cm	temp, humidity
Samsung S7	8.0	slower, 15-20 cm	baro is prefiltered
Samsung Tab A	9.0	none	no cover sensor
Moto G7	9.0	none	(1,2)
Moto G8	10.0	none	(1,2)
Moto G9	10.0	none	(1,2)
Samsung M51	10.0	none	(1)
Samsung M51	11.0	none	(1)
Samsung M51	12.0	none	(1)
Xiaomi Mi10T	10.0	none	(3)

- (1) Cover sensor inactive when alternate app is running
- (2) Flytec vane wheel may need 2 additional resistors (speaker L/R to GND)
- (3) Tested with Android and also with Lineage OS 17.1

Logger Storage Consumption

IGC: 150 kByte/h at 1 record/sec

GPX: 500 kByte/h at 1 record/sec

FDR: 600 kByte/h