Barometer

Barometers measure air pressure to determine altitude. They are highly sensitive, responding to a breath of air, and are very temperature sensitive. Typically they incorporate accurate temperature sensors to provide compensation.

Betaflight uses Barometer data to help determine the altitude of the craft, relative to the take-off point, assist with altitude control in GPS Rescue.

In most cases, Betaflight should detect the barometer automatically.

Barometers are much more useful for relative altitude changes than for absolute altitude measurement. A modern barometer chip like the Infineon DPS368 can detect very small altitude changes (less than 10cm), with good noise suppression, and update those values at 40Hz.

The DPS310 chip is much better than the older BMP280, and the DPS368 is better again. Clones of both the DPS310 and the BMP280 are widely used, and these are less reliable, or have poorer temperature compensation, than the originals.

Betaflight zeroes the Barometer data soon after powering up, so that Barometer altitudes are relative to the power-up point. On arming, the zero value may be corrected to zero again.

Barometers that exposed to turbulent air, eg propwash, will return very noisy altitude values. The usual recommendation is that a baro should have soft foam over it to reduce this problem.

When the motors are activated, and the quad is on the ground, the positive pressure between the props and the ground may be interpreted as a negative altitude by the Barometer. This resolves on take-off.

NOTE

Do not paint conformal coating over Barometer chips! Doing so can cover over the port by which they measure air pressure, causing incorrect altitude readings that drift a lot with temperature change.

Confirming that your Barometer is working properly

First ensure that Barometer is enabled in the Configurator's Configuration page.

The Barometer icon at the top should then light up. This means that the firmware included support for Baro, and a suitable driver was included.

Most barometers are powered up via USB, but some require LiPo power; if the icon does not light up, take props off and check to see what happens if you connect a battery or equivalent to the LiPo connector.

If the Barometer lights up, go to the Sensors tab, and choose Barometer at the top. You should then see baro altitude readings. Lifting the quad or flight controller about 1m up and down should show a change in the altitude reading of about 1m also. There will be some noise, and lag, that's normal.

Advanced checking

The debug mode `Baro` returns:

Debug	Data
0	State (during initialization)
1	pressure in hPa absolute
2	temperature reported by baro, deg C * 100
3	altitude estimate from Baro alone in cm

These values can be logged, or displayed in the graphs in Configurator's Sensors Tab. Check that the temperature reads accurately; a value of 2145 means 21.45 deg C. If it is very wrong, the chip is probably a clone of the DPS310, and temperature compensation may be poor.

Barometer vs GPS for Altitude estimation

Both the barometer and the GPS module report altitude data.

Barometers respond more quickly than GPS modules, but can be noisier. On the other hand, unless a GPS has a good hDOP indicator, the GPS altitude estimate can vary by several meters over a flight. Modern barometer chips, eg the DPS368, are typically faster and more accurate than most GPS modules.

The `altitude_source` `DEFAULT` method 'mixes' the two sources together, or, if there is only one source, uses whatever is available. The user can alternatively force `BARO_ONLY` or `GPS_ONLY`. With these the user can decide which provides better altitude control in a GPS Rescue.

TIP

If a GPS Rescue is initiated with a return altitude of say 10m over a flat field, and the return flight observed Line of Sight, the stability of the altitude control, and the smoothness of the descent, are easily visualized. Test with Baro only vs GPS only. If both seem OK, alone, test the `DEFAULT` method. Usually `DEFAULT` delivers the best result.

When `altitude_source = DEFAULT`, the user can adjust the relative influence of Baro vs GPS on the final altitude reading.

When a GPS module is available, Betaflight gets the Horizontal Dilution of Precision (hDOP) value from the GPS. This estimates the precision of the GPS altitude estimate. We use hDOP to assign an internal `gpsTrust` value. The default value is 0.3, meaning that if Baro is available, GPS accounts for only 30% of the altitude, reading, but when the dDOP value indicates accurate GPS readings, the `gpsTrust` value can reach a maximum of 0.9.

The `altitude_prefer_baro` CLI setting further modifies the `gpsTrust` value when there is a large difference between the two readings. When `altitude_prefer_baro = 100`, the default, our trust in the GPS is unaffected if the difference between the two sensors is less than 1m, reduced by half if the difference is 2m, and reduced to 1/5th of normal when the difference is 5m. When set to 50, we need a difference of 4m before our trust in the GPS is reduced by half. When set to zero, our GPS trust estimate is not affected by the difference between the readings.

Hence, if e have both GPS and Baro, and if `altitude_source` is set to `DEFAULT`, the relative balance between Baro and GPS depends on the GPS's hDOP estimate of the accuracy of its altitude measurement, the magnitude of the difference between the two readings, and the user's `altitude_prefer_baro` setting. When `altitude_prefer_baro` is set to the default of 100, Baro readings will predominate, unless the error is low and the GPS hDOP suggests that GPS is accurate. At lower values of `altitude_prefer_baro`, GPS will have a greater relative influence.

Basically, if you've got a good Baro, especially a real Infineon DPS310 or DPS368, and probably a BMP280, you should probably accept the default `altitude_prefer_baro` value of 100, and use the `DEFAULT` mixing method.

TIP

Use the Altitude debug for a detailed evaluation of GPS vs Baro readings

Debug	Data
0	gpsTrust value
1	baroAltitude, cm
2	gpsAltitude, cm, zeroed on arming
3	Vario

Barometer Firmware Support

In Betaflight 4.5 and higher, the Config file for the board automatically includes the base `BARO` code, and the drivers for whatever Baro chips the manufacturer uses on that board, the type of interface, pins used for the interface, etc.

That's why, in most cases, the Baro should work 'out of the box'.

If the board's config file has no Baro support, and you want to connect an external Barometer, in Betaflight 4.5 you must manually include the `BARO` code and the needed driver support into your firmware build.

This is achieved, for say a DPS310 baro, by entering `BARO BARO_DPS310` in the `Custom Defines` part of Configurator's Firmware Flasher tab. The word `BARO` enables the base Barometer code, and the text `BARO_DPS310` includes the `DPS_310` driver code. For a BMP280, the string to use would be `BARO BARO_BMP280`.

In 4.4 and earlier, all Baro code and all drivers were included in all firmware builds.

Barometer Bus Connections

Barometers are nearly always connected with an i2c interface, though some use SPI. The board's config file should define the interface/bus details for Baro, and the pins used for the interface.

A flight controller may have one or two i2c buses, or 'devices'. The config file sets the `baro_i2c_device` setting for Baro automatically.

Boards that have no defined `baro_i2c_device` will have that value set to 0. When an external Baro is manually connected on an i2c interface, the user must configure the `baro_i2c_device` setting, manually, to either bus 1 or 2, depending which bus the Baro is connected to.

Barometer Hardware Support

As mentioned above, in Betaflight 4.5 or higher, the drivers for the barometer chips used on the board are defined in the config file. Drivers for additional or different barometer chips must be manually added using the `Custom Defines` part of Configurator's Firmware Flasher tab, as above.

Each barometer chip has a default i2c 'address'. Some Barometers can support alternate addresses, which sometimes confuses automatic detection.

By default, Betaflight's `baro_i2c_address` setting is 0, and Betaflight will search for supported barometers by their default address, and connect to whatever baro is found. Note that in 4.5 and above, only the baro drivers that were built into the firmware will be supported.

Barometer	default i2c address	notes
BMP085	119	same address as BMP180
MS5611	119	uncommon
BMP280	118	alternate is 119, which is the default address of the BMP180
QMP6988	112	uncommon
DPS310	118	alternate is 119 (which is suggested as default in the spec sheet)
SPL06	118	clone of BMP280, alternate is 119 so that it can appear like a BMP180
BMP388	118	same address as BMP280 and SPL06
2SMPB_02B	112	uncommon
LPS22DF	93	uncommon

If the `baro_i2c_address` is set to custom value, Betaflight will only search for barometers which respond on the specified i2c address.

The `baro_hardware` CLI command defaults to `AUTO`, meaning that Betaflight will attempt to automatically identify the correct Baro chip. Although many share the same i2c address, there should only be one baro present on any one i2c address at a time.

In some cases, it can be necessary to force Betaflight to *only* use the driver code for a specific chip, e.g. when you know for sure that a particular chip is being used. For example `set baro_hardware = DPS310` will force Betaflight to use the driver for the DPS310 (obviously that driver must be included in the firmware in 4.5).

Betaflight's Github lists all supported barometers and their driver code, which includes the i2c address we search on, etc.

Note that nearly all barometer chips have the same i2c address. If it's not 118, the most common alternative option is (rarely) 119.

Barometer Not Identified

If, with the default `baro_i2c_address = 0` setting, or automatic barometer detection, a supported Baro is not identified, the possible problems and solutions include:

Problem	Solution
Baro was not enabled in Configuration tab	Check that the switch is 'on' for Baro
baro chip is only powered from LiPo	test with power on the LiPo input
The `baro_i2c_device` or bus value is incorrect	it should be 1 or 2, and must match the physical bus/pins to which the barometer chip is soldered
Driver code for that barometer was not included in the firmware	extra drivers can be added using the `Custom Defines` part of Configurator's Firmware Flasher tab
the barometer chip address has been customised	try manually configuring the `baro_i2c_address` from the list above

Problem	Solution
multiple barometer chip hardware drivers exist on the same i2c address and the wrong one is detected	if you know for sure which barometer is on your board, set `baro_hardware` to that barometer
the barometer chip is broken, or wired up incorrectly	check the wiring / replace the chip

CLI Command Summary

- Barometer related CLI variables

Variable	Range	Description
`baro_bustype`	`NONE`, `I2C`, `SPI`	Specifies a type of bus a selected barometer device is connected.
`baro_i2c_device`	`0` ~ Max I2C bus ordinate for MCU type (1 origin, same as `x` in `I2Cx` expression in `target.h`)	Specifies a bus ordinate of the I2C bus the device is connected when `baro_bustype` is `I2C`. `0` means "none".
`baro_i2c_address`	`0` ~ `119` (0x77)	Specifies an I2C address of the device in 7-bit representation. `0` is a special value to specify "*driver default address*". Values `1` ~ `7` are invalid and behavior is unpredictable.
`baro_spi_device`	`0` ~ Max SPI bus ordinate for MCU type (1 origin, same as `x` in `SPIx` expression in `target.h`)	Specifies a bus ordinate of the SPI bus the device is connected when `baro_bustype` is `SPI`. `0` means "none".
`baro_hardware`	`NONE`, `AUTO`, `BMP280`, `MS5611`, `BMP085`	`NONE` = Barometer support disabled. `AUTO` = Firmware will determine device to use under

Variable	Range	Description
		pre-defined rule. `BMP280`, `MS5611` and `BMP085` = Explicitly specifies device to use.

- If the device is SPI connected, CS (Chip Select) pin can be specified with the `resource` CLI command

Resource name	Description
`BARO_CS`	Specifies CS (Chip Select) pin to enable the SPI connected barometer device.

Previous Documentation

See /docs/wiki/guides/archive/barometer-configuration for older documentation about the initial Baro code developments, which took place around 3.2.

Youtube

Joshua Bardwell

Ivan Efimov

Links

Oscar Liang

VitroidFPV

Community

Discord

IntoFPV

Feeds

RSS

Atom

JSON