

[I just hopped around a bit with the throttle stick.](#)

The throttle curve is white.

With low current (because there's no load on the motor) it naturally looks "hoppy" because KISS only provides the current in 100 mA increments. – Which is actually sufficient.

The voltage also comes in 100 mV increments.

The RPM is displayed better than stated in the KISS documentation. Not in 100 RPM increments, than only in 10 RPM increments.

The only "hop" is the temperature (in 1 °C increments), presumably due to electrostatic discharge in the AM32. Is it the temperature around the ESC's FETs? It looks that way based on the development in the log with increasing RPM and current.

This is with an SQESC 2670 on an Align MCE4028 S/L 2800 KV 6-pole brushless motor with no load on the shaft. Voltage is 12.9 V from a power supply with a regulated 2A limit, - for emulating of 3S.

"KISS"? "Keep it short and simple," "Keep it simple, stupid"? ☺

Pole count: It should also be set in AM32 config as a precaution. - Then also in the S32terminal (for S32 config) so that the actual motor speed is used. Otherwise, if needed, also the possible gear reduction to the rotating part in the S32terminal. Here, just for fun, 1:20 was used.



Well... I can't find a way to provoke it anymore: The "filter" against "data hops" in KISS packets with CRC is OK, - initially ignoring the entire packet, - later only taking 10% of the change from a data value "hopped" in 30 milliseconds, - can no longer lead to zero values. It's now completely deactivated in both firmwares from "M" to "M4" - and not active in "M5" and "M6" either, only used for analysis via special values in the log.

The actual, frequent cause of this, as experienced by the "initiating RC buddy", was apparently an excessively high RPM value because the AM32 didn't use the number of poles configured in its settings for motor RPM calculation in KISS. This, of course, also causes larger "hops" - the higher the number of poles on the motor (14 poles in the "RC buddy's" case). Haha, he was using a different AM32 than the SQESC 2670 here.

Now, in my "pathologically self-critical view of reality," I've also packed [more sensors into time-consuming HoTT](#). The screenshot shows it without the accidentally activated warning on the

"CVS16+PID" button. ☺ 

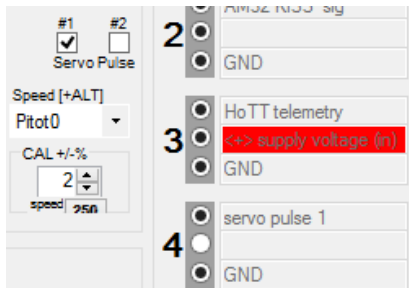
[Another test with "throttle stick hopping": Everything's OK.](#)

May 7, 11:55 PM Add-on, resulting in changes to all 4 firmwares:

M, M2 1.86 M3, M4 1.87 M5 1.86 Tester M6 1.87 Tester

AM32 doesn't provide (via KISS) a throttle input "Throttle", nor does it provide instantaneous "throttle" on the motor, "PWM." Having "Throttle" in the S32 log improves the view. This might also be desired in telemetry, even when the throttle stick is in the hand. 😊

S32 also offers two "Servo Pulses" as sensors, via the port selected for this purpose by S32terminal.



Here, Pulse #1 is "misused" for Throttle Pulse Measurement (parallel to the ESC signal, GND already connected, taken from the receiver's line).

"Pulse #1" appears in the log as "Pulse1:length," but in microseconds. This might "take some getting used to". 😊 It is then converted to "throttle" 0-100%, with 1000 μ s for 0% and 2000 μ s for 100%. It occupies the "throttle" field in the S32 for its logging and for many telemetry functions.

"Throttle" is not controlled by the "AM32" ESC selected in the setup (via KISS).

Using "PWM," even if not used by AM32, would be misleading because there would be no signal from the ESC.

Regarding "Throttle," there's only a slight possible deviation in the ESC's interpretation: At 0% it might still detect a pulse length a few microseconds above 1000 microseconds, while at 100% it might detect a pulse length slightly below 2000 microseconds.

The transmitter, in conjunction with the receiver, could also use the signal slightly differently, ranging from a pulse length slightly above 1000 to slightly below 2000 microseconds.

Example in a test here, with a Futaba T14SG transmitter and an R7008SB receiver:

0% 1012.5 μ s (-> +1.25%) , 100% 1910 μ s (-9%=> 91% <-)

Of course, at least theoretically, JLog S32 could also contribute to some timing inaccuracies. This depends on the magnitude and accuracy of its clock frequency when used with timers in the microcontroller. - The same applies, of course, to the transmitter and receiver.

Finally, regarding in the ESC. Its information about its perception of "throttle" is therefore the most accurate statement – but this is not provided by AM32.

However, only "PWM" can truly be in effect on the motor – including its commutation, if/because it's a brushless motor. AM32 doesn't provide this information either. Guessing "PWM" is completely impossible, logically – only the ESC has the "vision" as the "actor" at the motor.

For a technical explanation, in case it's helpful:

"PWM," "Pulse-width modulation": Its initial purpose is essentially an "on-off switching," for example for the brightness of a light bulb.



Applied in this way by the ESC to the motor, - with some additional "interruption" in the commutation of a brushless motor.

As a "servo pulse" or "throttle," it also has a clock frequency for the ON/OFF signal, usually 50 Hz. So, it's "only" an input to a servo or ESC, not a true ON/OFF signal with an effective value.

Tested again, without ESC power (more encouraging 😊), with a Futaba Tx T14SG and Rx R7008SB:

Throttle stick at zero: 1089 μ s, so Delta = +8.9% for 0% throttle

Throttle stick at center: 1500 μ s, so Delta = zero for 50% throttle

Throttle stick at full throttle: 1918 μ s, so Delta = -8.2% for 100% throttle

I probably didn't push the stick far enough down to zero. (Regarding only +1.25% as above)

I was "brutal" about the maximum setting. 😊

May 9, 12:07 AM: All 4 **firmwares** in M to M6 have been **corrected** by the throttle derivation:

If the pulse length is up to 90 microseconds above 1000, then "throttle" is corrected to 0%.

If the pulse length is up to 90 microseconds below 2000, then "throttle" is corrected to 100%.

... [Now a bit more "human" in the log](#) 😊 *Throttle is represented by a white curve.*

Voltage drops again due to the 2A limit on the power supply.

One more note:

"Pulse#1" is always measured as soon as S32 is started.

However, the conversion to "throttle" only occurs during the evaluation of each KISS data packet from the AM32. The throttle will not be filled if the S32 is not yet connected to the AM32 (AM32 green, S32 flashing green).

Again a note:

The high versions shown in the S32 terminal, in M, M3, M5 and M6, are not actually necessary for flashing of S32 with firmware of a potentially equal or lower version than the one the S32 reports. Enabling the "Update Anyway" button will also suffice.

alternative function

Update Anyway

This would render two "M" versions unusable—good for me for temporary test versions. I'm just too "lazy" right now to update the other two PDFs accordingly. The fact that this "add-on" is only available in one language is also due to my "current laziness". 😊 (Now no more 😊😊)

After further testing by the "RC buddy" (and hopefully others), version 1.86 will become the new standard version, replacing the current 1.85.

It remains to be clarified whether the SQESC 2670 is truly a general issue with the voltage via KISS (almost only half the voltage), or if it's a bug specific to the AM32 at my end.

If it is indeed a general issue with the 2670, the "sloth" needs to "force" themselves to add a correction button to S32terminal instead of continuing to offer special firmware version 1.87.

Well, too bad... Open Source for ESC AM32 with "telemetry" (data protocol) [KISS](#) isn't open enough for the necessary information.

Throttle and "PWM" for monitoring user actions and the ESC's reactive behavior.

May 11th: Just tested on the SQESC 2670: Set the number of poles in its configuration to 2, instead of the previous actual 6. No difference in RPM via KISS. – Commutation doesn't seem to be affected. Used in PWM? - Still somewhat unclear, only guessable: Ignoring pole count for RPM only via KISS, a bug.