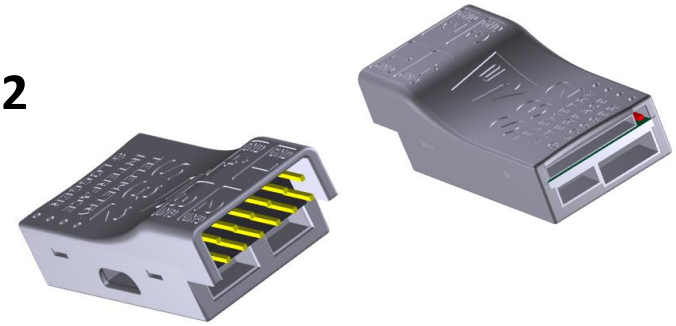
R² Prototyping **S32**



S32 (aka JLog3) is the successor of JLog2.6. S32 is telemetry gateway, logger and alarm generator with the well-known specialty to directly refer to data from "controllers" (brushless ESC), virtually sensorless. In addition, it can also use a number of own sensors.

It was already a long time vacant, now it's finally here, after all the JLog history has been re-implemented, new functionality on top of it.

In the 6 years of its existence, JLog has continued to develop itself in form of 4 models until now, partly because new communication partners (ESC, telemetry) sometimes required new interfaces. Whenever it was not possible by interface options (Who still remembers JSend, JCC, JSPEK?), a new model was due. For a long time lack of program memory and RAM was an obstacle, which is why there are for JLog 2.x more than 230 firmware images. S32 transforms the lack into an "upward open Richter scale" of available resources for further development. Compared to JLog2.6:

Program memory:	factor 16 (plus less demand on space by 32 bit ARM architecture)
Memory (RAM):	factor 48
Clock frequency:	factor 10.5 to 12 (plus more performance by 32 bit ARM)

Furthermore, new compared to JLog2.6, - hardware:

- RTC (real-time clock) with built-in backup battery, deep discharge-proof, buffering time approx. 4.5 months, automatic charging from each voltage source for S32. Time to the second, date, day of the week. Files in the log now have a time stamp. The clock provides better options for some functions. The RTC is automatically adjusted as soon as S32 is connected via USB to its terminal under Windows (soon also other OS).
- File systems: FAT12, FAT16, FAT32. Thus, there are now no longer any capacity constraints with regard to usable micro SD's, instead of max. 2GB now theoretically up to 8.8TB.
- Interfaces: 8 operational ports including 4x servo connector, plus USB (Micro B). The total of 8 servo pins are usable very flexible for the requirements of the respective configuration. The user tells "S32terminal" what he wants, the terminal tells him where and how to connect, tells S32 through USB what it has to do at runtime.
- CANbus interface to integrate the existing "C2T" for JLog-NAZA. The CANbus will hopefully also become a general data interface in R/C, resistant to "ground loops" because "differential-ended" while currently used "single-ended" data interfaces are for long already on the losing track in light of brute currents flowing in today's models.
- S32 basically only takes its operating voltage from the telemetry interfaces, on two servo ports and the SPEKTRUM XBus interface. It also measures its supply voltage at these ports, so it can replace "Ubec" (BEC voltage) if an ESC should not provide Ubec through its data interface.
- The interface to Kontronik KOSMIK and JIVE Pro is now galvanically decoupled.
- Powerful internal "buck" voltage regulator (switching) which can not easily be shocked by voltage drops on the supply voltage.

The other differences are made up by the firmware of S32, - as mentioned, now always only one image for everything. Just as examples:

- Futaba telemetry (S.Bus2): S32 presents itself as 6 sensors (previously 8 by JLog) to the transmitter in a more intuitive use of the displays of these sensors for its data. Registering at the transmitter is now quite easy - because S32 automatically enters the registration mode, because its sensors can be re-registered at any time, because it no longer insists that all its sensors have been registered. In addition, the obtained slot assignments of the registration are now part of the configuration of S32, so S32 terminal shows them, you can push them from one S32 to another without real registration (if it matches the registration status in the transmitter). -- *Since a transmitter T10J obviously has problems with the registration process, because at least the transmitter T18SZ has a bug in the region of the sensor display "ESC" (now "Roxxy ESC") used above, S32 a) can alternatively function as 10 sensors, "ESC" replaced by "CUR + RPM + TEMP", b) sensors can also be registered manually in the S32terminal (and in the transmitter).*
- KOSMIK/JIVE Pro: S32 emulates to another "TelMe" as JLog. He no longer has to struggle with the familiar firmware bugs of these ESCs which require workarounds for years from JLog and users, - he also gets more data items than JLog.
- JETI Telemetry: Now besides JETI v.1 and EX also EXbus -- *For large models with more than one ESC, ie more than one S32, the sensor ID can be changed for parallel operation of several S32s. Since JETI telemetry is limited to a maximum of 31 usable sensor displays, S32 offers an alternative, limited compilation of sensor displays for multi-ESCs.*
- Castle Creations ESC's: Essentially trouble-free software interface to these ESC's
- HobbyWing V4 ESC 60A: This ESC does not have its own current sensor. The current sensor option HiTec C200 is now also supported in combination with this ESC (in addition to V3 ESC's).
- Former features, which had to leave JLog from lack of space, came back, e.g. connection capability of a SM GPS-Logger for ground speed (+ altitude), also simultaneously with the pitot tube, here Prandtl probe SM #2560 for air speed. Also voltage measurement on the pins of the S32, max. +3,2V (9.6 or 64V with external resistors) - as well as servo pulse measurement.
- An analogue (SM #2220, 2221) and digital (SM #2820 or DS18B20) temperature sensors remained, of course - an external speed sensor of any design (also as brushless RPM sensor) and up to two alarm lines of the S32, one of them also pulsed , also modulated "flash" or "Morse".
- Of course there is again the "data bus" for the data interface to R2 HV²BEC, R2 CVS16 16S Cell Voltage Monitor and more.
- The "BID" for battery tagging also came back, a Robbe invention. The BID is connected to S32's data bus. Its application has been completely re-implemented in S32. It now also uses an own data set on the BID, separately from the data set of the charger. Alarm thresholds for S32 and CVS16 are thus individually adjustable for each battery, and S32 also uses time stamps of its RTC to prevent someone from inadvertently using an insufficiently charged battery. The corresponding setup saves S32 also on the BID. The operation of a charger with a BID is not affected by S32. S32 knows the charger but the charger does not know anything about S32.
- S32 generally logs, no enabler is required.

S32 is configured through its USB interface by "S32terminal" (what is "JLC" for JLog) as application. If there is a firmware update, the server nn the Internet offers it to the terminal. You press "Update" and that's it. The terminal itself also updates automatically. S32terminal obeys the requirement to use the online manual only as a supplement, otherwise, S32terminal itself is the instruction.

Updating the firmware of the S32 by a file on the SD card, as with JLog, is no longer supported (but for occasionally updating of the bootloader). This is done via USB and the terminal. You can also have a setup of S32terminal stored in a file. On the one hand, you can manage your setups (each setup has

a selectable name besides the filename), - you can also, as with JLog, let S32 learn another setup at startup from a file on the SD card.

The setup of a BID battery tag for S32's specific use is performed by the user at the S32 terminal. S32 serves as a gateway, terminal--USB--S32--I^2C(data bus)--BID.

Despite much more hardware, S32 is only marginally larger than JLog2.6 (3mm longer). This was achieved by a 6-layer PCB and using SMD components of the smallest size (01005). Its housing is now no longer glued, but clipped. Labels are no longer used, the housing is no longer transparent, but anthracite with engravings and holes for the 4 LEDs. The ports are simply numbered 1 to 8, - the terminal provides necessary explanations.

The egg-laying-wool-milk-sow grew again. ☺

<u>At a glance</u>

(This is the momentary state of March 2017. Firmware and terminal are constantly being further developed - while the firmware of JLog 2.0, 2.5, 2.6 are no longer subject to development, but the maintenance in case of bugs or changes, forced by attachable third-party devices, if feasible within the hardware limits of JLog.)

Supported ESC‘s (or FBL) with data interface:

- JIVE (old) and Heli-JIVE
- KOSMIK, JIVE Pro
- Castle Creations ICE/EDGE (all which have Castle Link Live)
- YGE T series
- Graupner/SJ BLC+T
- JETI MEZON
- Hifei KingKong III Pro
- HobbyWing Platinum V3 with HiTec C200 Hall core current sensor as an option
- HobbyWing Platinum V4 model 60A with HiTec C200 Hall core current sensor as option
- Scorpion Tribunus
- AC-3Xv3
- DJI NAZA M *(currently only with telemetry JETI EX, EXbus and HoTTv4)*
- Herkules III Multi-ESC *(not yet available)*
  NAZA/HERK have not yet been fully ported from JLog2.6 yet

Supported telemetry systems:

- Multiplex MSBv1, MSBv2
- JETI v1 (JETIbox), EX, EXbus + text displays
- Graupner/SJ HoTTv4 all 5 sensors (ESC, GAM/GEM, EAM, VARIO, GPS) + text displays

- Futaba (S.Bus2) 6 sensors: ESC TEMP TEMP TEMP ESC CUR
  or 10: CUR-RPM-TEMP TEMP TEMP TEMP CUR-RPM-TEMP CUR
- Horizon SPEKTRUM Xbus (TM1000, receiver ARnnnnT)
- Horizon SPEKTRUM „SRXL“ (currently only with receiver SPM4649T)
- HiTec
- JR DMSS
- FrSky S.Port

Textboxes and Livestream:

- JETIbox [Mini] (pure text) and Profibox (Text + EX)
- Graupner/SJ HoTT SmartBox, - text mode, GAM/GEM also binary (S32 as HoTT Tx module)
- SM Unidisplay
- OpenFormat live stream for LogView and DataExplorer

S32 own sensors:

- analogue temperature sensor based on Microchip TC1046 (SM #2220, 2221)
- digital temperature sensors (up to 3) based on Dallas DS18B20 (SM #2820)
- analogue: PT1000
- speed sensor for external speed, - any, optical, magnetic (Hall), brushless
- 1 to 2 voltmeter inputs, - directly or via CVS16 (16 inputs)
- 1 to 2 pulse meters for servo pulse lengths
- 1 to 2 alarm lines for controlling alarm devices
- Prandtl probe (pitot tube) SM #2560 for air speed
- SM GPS-Logger for ground speed and barometric altitude
- Data bus: Monitoring of R2 HV²BEC: BEC voltage and current, temperature, input voltage
- Data bus: R2 CVS16 16S Cell Voltage Monitor (or 16 voltage inputs 0 to +73V)
- BID battery tag

Almost every value (also Imot) can be assigned with alarm thresholds, also logical and derived alarms as for a stall speed warning. Alarms are given into the telemetry as possible and are logged.

S32 logs in OpenFormat for LogView and DataExplorer, - max 65535 log files. For additional information LogView-ignored rows are provided in the log file. Information on the setup and status of S32 can be found in the header of a log file – less intended for the user as for the support.

Log: Depending on the setup in up to 4 channels (record types)
(S32-NAZA/HERK use their own record structures, with NAZA also NMEA for Google Earth.)

- Channel 1 (basic): 48 items, values and alarms
- Channel 2 (supplement): 12 items

- Channel 3: 16 servo channel data from Futaba S.Bus or JETI EXbus receivers
- Channel 4 (CVS16): 32 items including CVS-specific warnings

Delivery

- S32
- KOSMIK/JIVE Pro connection harness (smaller connector on S32 than on JLog)
- servo line for telemetry except of SPEKTRUM (Horizon SPMA 9580 required)

USB cable (micro type B on type A as for smartphones), SD card and SD card reader are not included. As mentioned, with respect to the SD card, each capacity is supported.