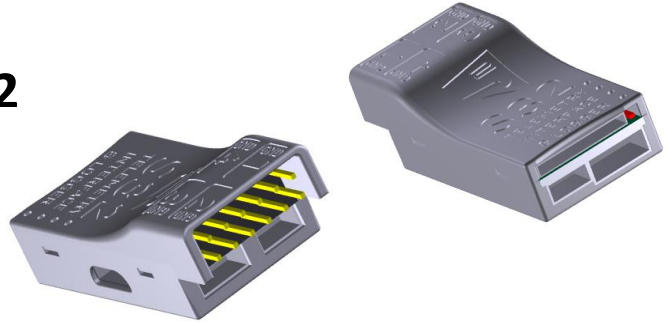




## R<sup>2</sup> Prototyping S32



S32 (aka JLog3) ist der Nachfolger von JLog2.6. S32 ist Telemetrie-Gateway, Logger und Alarmgeber mit der bekannten Spezialität, Daten aus „Reglern“ (brushless ESC) direkt zu beziehen, quasi sensorlos. Zusätzlich verwendet er bei Bedarf auch eine Reihe eigener Sensoren.

Er war bereits lange vakant, nun ist er endlich da, nachdem einfach alles der JLog Historie re-implementiert wurde, neue Funktionalität oben drauf.

In den 6 Jahren seiner Existenz hat sich JLog über bisher 4 Modelle funktionell stetig weiterentwickelt, u.a., weil neue Kommunikationspartner (ESC, Telemetrie) auch mal neue Interfaces erforderten. Immer dann, wenn man das nicht mehr durch Interface-Optionen machen konnte (Wer erinnert sich noch an JSend, JCC, JSPEK?), wurde ein neues Modell fällig. Schon lange war Mangel an Programm- und Arbeitsspeicher ein Hindernis, weshalb es für JLog 2.x am Ende über 230 Firmware-Images gibt. S32 wandelt Mangel in eine „nach oben offene Richterskala“ verfügbarer Ressourcen für weitere Entwicklung. Im Vergleich zu JLog2.6:

Programmspeicher:	Faktor 16 (plus weniger Bedarf durch 32 Bit ARM Architektur)
Arbeitsspeicher (RAM):	Faktor 48
Taktfrequenz:	Faktor 10,5 bis 12 (plus mehr Performance durch 32 Bit ARM)

Des Weiteren, neu im Vergleich zu JLog2.6, Hardware:

- RTC (Echtzeituhr) mit fix eingebauter Backup Batterie, tiefentladungsfest, Pufferzeit ca. 4,5 Monate, automatisches Laden aus jeder Spannungsquelle für S32. Uhrzeit auf die Sekunde, Datum, Wochentag. Dateien im Log haben nun einen Zeitstempel. Die Uhr verschafft bessere Möglichkeiten für einige Funktionen. Die RTC stellt sich automatisch, sobald S32 per USB mit seinem Terminal unter Windows (demnächst auch andere OS) verbunden wird.
- Dateisysteme: FAT12, FAT16, FAT32. Damit gibt es nun keine Kapazitätsbeschränkungen mehr bzgl. verwendbarer Micro-SDs, statt bisher max. 2GB nun theoretisch bis zu 8,8TB.
- Interfaces: 8 operative Ports, davon 4x Servosteckverbinder, plus USB (Micro B). Die insgesamt 8 Servo-Pins sind sehr flexibel verwendbar für die Erfordernisse der jeweiligen Konfiguration. Der Anwender sagt „S32terminal“, was er will, das Terminal sagt ihm, wo und wie was anzuschließen ist, sagt S32 via USB, was er zur Laufzeit zu tun hat damit.
- CANbus Interface, um auch den bisherigen „C2T“ für JLog-NAZA zu integrieren. Der CANbus wird hoffentlich auch mal generelles Dateninterface im Modellbau werden, resistent gegen „Masseschleifen“ weil „differential-ended“, während die bisherigen „single-ended“ Dateninterfaces schon lange auf der Verliererstraße sind angesichts brachialer Ströme, die in heutigen Modellen fließen.
- S32 nimmt seine Betriebsspannung grundsätzlich nur noch von den Telemetrie-Interfaces, auf zwei Servo-Ports und dem SPEKTRUM XBus Interface. Er misst seine Betriebsspannung an diesen Ports auch, sodass er „Ubec“ (BEC Spannung) ersetzen kann damit, wenn ein ESC kein Ubec durch sein Dateninterface liefern sollte.
- Das Interface zu Kontronik KOSMIK und JIVE Pro ist nun galvanisch entkoppelt.
- Leistungsfähiger interner „Buck“-Spannungsregler (getaktet), welchen Spannungseinbrüche auf der Versorgungsspannung nicht so leicht schocken können.

Die weiteren Unterschiede macht die Firmware des S32 aus, - wie gesagt, nun immer nur ein Image für alles. Nur beispielhaft genannt:

- Futaba Telemetrie (S.Bus2): S32 stellt sich als 6 Sensoren dar (zuvor 8 durch JLog) gegenüber dem Sender, in einer intuitiveren Verwendung der Displays dieser Sensoren für seine Daten. Das Registrieren am Sender ist nun ganz einfach, - weil S32 automatisch in den Registrierungsmodus geht, weil seine Sensoren jederzeit neu registriert werden können, weil er nicht mehr darauf besteht, dass alle seine Sensoren registriert wurden. Außerdem sind erhaltene Slot-Zuweisungen der Registrierung nun Bestandteil der Konfiguration von S32, sein S32terminal zeigt sie an, man kann sie von einem S32 auf einen anderen schieben ohne echtes Registrieren (sofern es zum Registrierungsstatus im Sender passt). – *Da ein Sender T10J offenbar grundsätzlich Probleme mit dem Registrierungsprozess hat, da zumindest der Sender T18SZ einen Bug im Bereich des oben verwendeten Sensor Displays „ESC“ hat (nun „Roxxy ESC“), kann S32 a) alternativ als 10 Sensoren arbeiten, wobei sich „ESC“ jeweils durch „CUR+RPM+TEMP“ ersetzt, können b) Sensoren auch manuell registriert werden im S32terminal, dazu auch im Sender.*
- KOSMIK/JIVE Pro: S32 emuliert an anderes „TelMe“ als JLog. Er hat dadurch nicht mehr mit den bekannten Firmwarebugs dieser ESCs zu kämpfen, die seit Jahren Workarounds abverlangen von JLog und Anwender, - er bekommt auch mehr Datenelemente als JLog.
- JETI Telemetrie: Nun neben JETI v.1 und EX auch EXbus. -- *Für Großmodelle mit mehr als einem ESC, also mehr als einem S32, kann die Sensor ID verändert werden für Parallelbetrieb mehrerer S32. Da JETI Telemetrie auf maximal 31 nutzbare Sensor Displays begrenzt ist, bietet S32 für Multi-ESCs eine alternative, limitierte Zusammenstellung von Sensor Displays.*
- Castle Creations ESCs: Wesentlich störfreieres Software-Interface zu diesen ESCs
- HobbyWing V4 ESC 60A: Dieser ESC hat keinen eigenen Stromsensor. Die Stromsensor Option HiTec C200 wird nun auch in Kombination mit diesem ESC unterstützt (neben V3 ESCs).
- Ehemalige Features, die JLog aus Platzmangel verlassen mussten, kamen zurück, z.B. Anschlussfähigkeit eines SM GPS-Logger für Ground Speed (+ Höhe), auch gleichzeitig mit Staurohr (Pitot, hier Prandtl Sonde) SM #2560 für Air Speed. Ebenso auch Spannungsmessung an den Pins des S32, max. +3,2V (9,6 oder 64V mit externen Widerständen) - sowie auch Servopulsmessung.
- Eigene analoge (SM #2220, 2221) und digitale (SM #2820 bzw. DS18B20) Temperatursensoren blieben natürlich, - auch ein externer Drehzahlsensor jedweder Ausführung (auch als Brushless RPM Sensor) und bis zu zwei Alarmleitungen des S32, eine davon auch gepulst, auch als „Flash“ und „Morse“.
- Natürlich gibt es wieder den „Datenbus“ für das Dateninterface des R2 HV<sup>2</sup>BEC, für R2 CVS16 16S Cell Voltage Monitor und mehr.
- Auch der „BID“ für Battery Tagging kam zurück, eine Erfindung von Robbe. Der BID schließt sich an S32's Datenbus an. Seine Applikation wurde völlig neu implementiert in S32. Der verwendet jetzt neben dem Datensatz des Ladegeräts auch einen eigenen auf dem BID. Alarmschwellen für S32 und CVS16 sind dadurch pro Batterie individuell, automatisch einstellbar, - außerdem versucht S32 mittels eines eigenen Datensatzes und von Zeitstempeln seiner RTC zu verhindern, dass jemand versehentlich eine nicht gut geladene Batterie verwendet. Das dazugehörige Setup speichert S32 auch auf dem BID. Das Funktionieren eines Laders mit einem von S32 beschriebenen BID wird nicht tangiert. S32 weiß vom Charger, der Charger aber nichts von S32.
- S32 loggt generell, kein Enabler mehr erforderlich.

S2 wird konfiguriert durch sein USB Interface mit „S32terminal“ („JLC“ genannt bei JLog) als Anwendung. Gibt es ein Firmwareupdate, bietet der Server im Internet es dem Terminal an. Man drückt „Update“, und das war's. Auch das Terminal selbst aktualisiert sich automatisch. S32terminal gehorcht der Maßgabe, das Online Manual nur noch als Ergänzung benutzen zu können, ansonsten ist S32terminal selbst die Anleitung.

Das Updaten der Firmware des S32 durch eine Datei auf der SD-Karte, wie bei JLog, wird nun nicht mehr unterstützt (jedoch zum gelegentlichen Updaten des Bootloaders). Das erfolgt rein via USB und das Terminal. Man kann aber ein Setup von S32terminal auch in einer Datei ablegen lassen. Zum einen kann man seine Setups damit verwalten (jedes Setup hat einen wählbaren Namen neben dem Dateinamen), - man kann auch, wie zuvor mit JLog, S32 beim Starten ein anderes Setup lernen lassen aus einer Datei auf der SD-Karte.

Das Setup eines BID Battery Tag für S32's spezifische Verwendung erfolgt durch den Anwender am S32terminal. S32 dient dabei als Gateway, Terminal—USB—S32—I<sup>2</sup>C(Datenbus)—BID.

Trotz noch mal deutlich mehr Hardware ist S32 nur unwesentlich größer als JLog2.6 (3mm länger). Das gelang durch eine 6-Layer Platine und Verwenden von SMD Komponenten der kleinsten Größe (01005). Sein Gehäuse ist nun nicht mehr geklebt, sondern geklippt. Label werden nicht mehr verwendet, das Gehäuse ist nicht mehr durchsichtig, sondern anthrazit mit Gravuren und Löchern für die 4 LEDs. Die Ports sind einfach nur nummeriert 1 bis 8, - notwendige Erklärungen liefert das Terminal.

Die Eierlegendewollmilchsau wurde wieder erwachsener. ☺

### Im Überblick

*(Das ist der Momentanzustand März 2017. Firmware und Terminal werden ständig weiterentwickelt, - während die Firmwares von JLog 2.0, 2.5, 2.6 keiner Weiterentwicklung mehr unterliegen, wohl aber der Pflege im Falle von Bugs oder Änderungen, die anschließbare Fremdgeräte erzwingen, sofern machbar im Rahmen der Hardware des JLog.)*

Unterstützte ESCs (bzw. FBL) mit Dateninterface:

- JIVE (alt) und Heli-JIVE
- KOSMIK, JIVE Pro
- Castle Creations ICE/EDGE (alle, die Castle Link Live haben)
- YGE T Serie
- Graupner/SJ BLC+T
- JETI MEZON
- Hifei KingKong III Pro
- HobbyWing Platinum V3 mit HiTec C200 Hall Core Stromsensor als Option
- HobbyWing Platinum V4 Modell 60A mit HiTec C200 Hall Core Stromsensor als Option
- Scorpion Tribunus
- AC-3Xv3
- DJI NAZA M *momentan nur mit Telemetrie JETI EX, EXbus und HoTTv4*
- Herkules III Multi-ESC *(momentan noch nicht)*  
*NAZA/HERK wurden noch nicht vollständig portiert von JLog2.6*

Unterstützte Telemetriesysteme:

- Multiplex MSBv1, MSBv2
- JETI v1 (JETIbox), EX, EXbus + Text Displays
- Graupner/SJ HoTTv4 alle 5 Sensoren (ESC, GAM/GEM, EAM, VARIO, GPS) + Text Displays
- Futaba (S.Bus2) 6 Sensoren: ESC TEMP TEMP TEMP ESC CUR  
oder 10: CUR-RPM-TEMP TEMP TEMP TEMP CUR-RPM-TEMP CUR
- Horizon SPEKTRUM Xbus (TM1000, Empfänger ARnnnnT)
- Horizon SPEKTRUM „SRXL“ (momentan nur mit Empfänger SPM4649T)

- HiTec
- JR DMSS
- FrSky S.Port

Textboxen und Live Stream:

- JETIbox [Mini] (rein Text) und Profibox (Text + EX)
- Graupner/SJ HoTT SmartBox, - Text Mode, GAM/GEM auch binär (S32 als HoTT Tx Modul)
- SM Unidisplay
- OpenFormat Live Stream für LogView und DataExplorer

S32-eigene Sensoren:

- analoger Temperatursensor auf Basis Microchip TC1046 (SM #2220, 2221)
- analog: PT1000
- digitale Temperatursensoren (bis zu 3) auf Basis Dallas DS18B20 (SM #2820)
- Drehzahlsensor für externe Drehzahl, - jedweder, optisch, magnetisch (Hall), brushless
- 1 bis 2 Voltmetereingänge, - direkt oder mittels CVS16 (16 Eingänge)
- 1 bis zwei Pulsmeter für Servopulsängen
- 1 bis 2 Alarmleitungen zur Ansteuerung von Alarmgebern
- Prandtl Sonde (Staurohr) SM #2560 für Air Speed
- SM GPS-Logger für Ground Speed und barometrische Höhe
- Datenbus: Monitoring des R2 HV<sup>2</sup>BEC: BEC Spannung u. Strom, Temperatur, Eingangsspannung
- Datenbus: R2 CVS16 16S Cell Voltage Monitor (oder 16 Spannungseingänge 0 bis +73V)
- BID Battery Tag

Nahezu jeder Wert (auch Imot) kann mit Alarmschwellen belegt werden, auch logische und abgeleitete Alarme wie für Stall Speed Warning. Alarme gehen in jeweils möglichem Umfang und Form in die Telemetrie und werden geloggt.

S32 loggt im OpenFormat für LogView und DataExplorer, - max. 65535 Log-Dateien. Weitere Informationen befinden sich als von LogView ignorierte Zeilen im aktuellen Logfile. Informationen zu Setup und Status von S32 befinden sich im Header einer Log-Datei, - weniger für den Anwender, denn für den Support.

Log: Je nach Setup in bis zu 4 Kanälen (Record-Typen)

(S32-NAZA/HERK verwenden eigene Record-Strukturen, mit NAZA auch NMEA für Google Earth.)

- Kanal 1 (Basic): 48 Werte und Alarme
- Kanal 2 (Ergänzung): 12 Werte
- Kanal 3: 16 Servo-Kanaldaten von Futaba S.Bus, JETI EXbus oder SPEKTRUM Empfängern mit „bidirektionalem SRXL“
- Kanal 4 (CVS16): 32 Werte

### Lieferumfang

- S32
- KOSMIK / JIVE Pro Anschlusskabel (kleinerer Stecker an S32 als an JLog)
- Servokabel für Telemetrie außer SPEKTRUM Xbus (Horizon SPMA 9580 benötigt)

USB Kabel (Micro Typ B auf Typ A, wie für Smartphones), SD-Karte und SD Card Reader sind nicht enthalten. Wie erwähnt, bzgl. der SD-Karte wird jede Kapazität unterstützt.