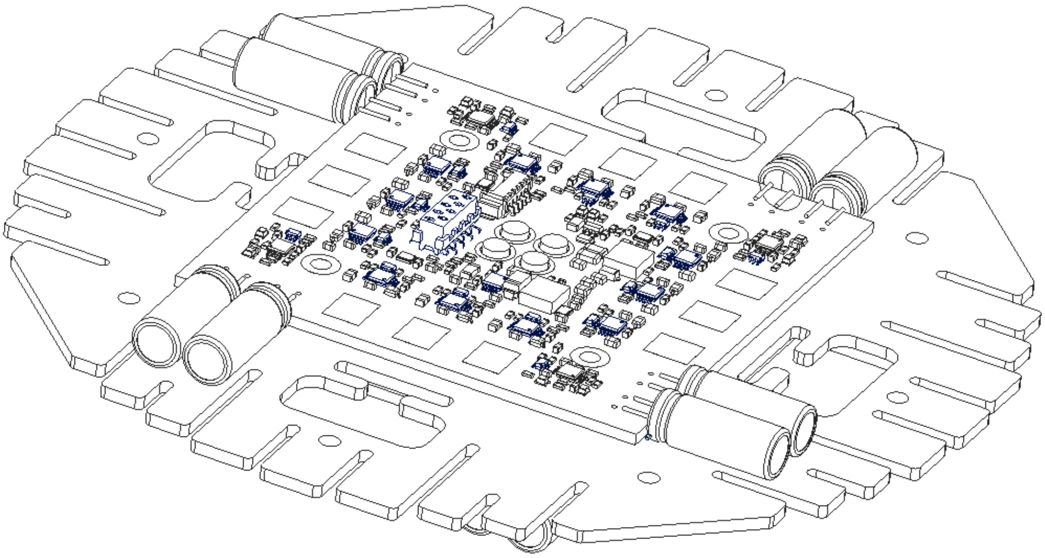
HERKULES III

User Manual

Version: 0.40

08.08.2016



1 Introduction

Thank you for purchasing the HERKULES III Multicopter ESC.

HERKULES III is the result of many years of development and continuous optimization, for professional users who cannot compromise performance, safety or reliability.

HERKULES III is a 3-phase high performance controller for Sensorless Brushless DC Motors. The Controller integrates up to four identical high performance Electronic Speed controllers (ESCs) on one PCB module. The modules can be stacked and mounted together in different variants.

This documentation describes the functions of the HERKULES III ESC and the and interfaces between the HERKULES III ESC and flight control units. It also defines operation of the an optional telemetry interface capabilites.

The latest version of Software and Documentation can be downloaded at:

www.andreasbaier.de.

Please check here for latest updates and Improvements.

Note:

For easier navigation in this document, you can use the Table of Contents feature of your Adobe Acrobat Reader.

Click on the sign as shown in the right picture and navigate with the TOC to the chapters you are interested in.

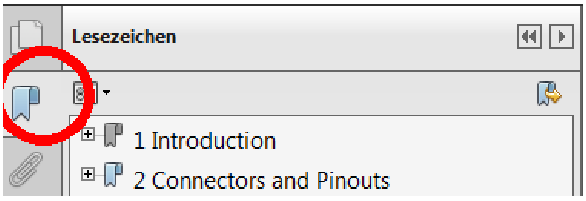


Table of Contents

List of Tables

List of Figures

1.1 Key Features

- 3-phase Sensorless brushless DC motor controller
- Control Mode based on block commutation
- Supply Voltage 9-26V (3-6S Lipo)
- 30A continuous / 60A peak motor current
- High efficiency, low-resistance powerstage (total path resistance < 3mOhm)
- Active freewheeling enabling active braking using energy recuperation
- Setpoint command by I2C and PPM
- Setpoint update rate up to 450Hz
- PWM switching frequency 8kHz-16kHz
- Programmable commutation timing 24°, 18°, 12°
- Ultrafast setpoint reaction time
- Up to 200,000 field turns per minute
- Stable and wide-band firmware
- Overload protection (overcurrent, overtemp, stall)
- Telemetry capable (e.g. Graupner HoTT, Jeti Duplex, Futaba S-BUS, standard serial and others)
- Ready to fly – fully assembled
- Compact and light weight (8-fold ESC 100x100mm at < 250g)
- 100% designed and made in Germany

1.2 HERKULES III Variants

The HERKULES III Power ESCs are available in different variants.

Max Continous Curent	Cooling Plate	Number of Motors	Ordering Number
20A	100mm	4	HKIII-QUAD-L (6s)
		6	HKIII-HEXA-L (6s)
		8	HKIII-OKTO-L (6s)
30A	150mm	4	HKIII-QUAD-XL(6s)
		6	HKIII-HEXA-XL(6s)
		8	HKIII-OKTO-XL(6s)

Table 1 - HERKULES III Variants

1.3 Important Safety Notice

To avoid unexpected motor starts please read the operating instructions very carefully. Improper wiring of the motor, battery or control wires, or set point or command line failures may result in unexpected startup or runaway conditions.

The user must always assume that such startups can happen and the user must ensure that his system is safe in all conditions. Please do all wiring and configuration work very carefully. Follow all safety procedures in the manual and work exactly as described. Never program or run tests with a flight battery connected. Use a current-limited power supply to check the basic system behavior!

The Herkules III Powerboard is delivered pre-mounted and pre-soldered with battery power wires. Never try to de-solder the power-wires from the PCBs. The high thermal conductance of the power board makes special soldering equipment necessary. Standard soldering equipment will likely destroy the electronics. Soldering is only allowed on the motor connection pads and the battery end of the flight pack power wires.

Please consider also to the absolute max ratings described in the Electrical Characteristics on page 11.

1) DO NOT CONNECT BATTERIES to THE BOARDs before having checked them for correct operation on a current-limited power supply! Never connect a battery without being sure that the installation has no short circuit. Always test the electronics for the first time, or after any programming or setup changes, on a current limited power supply (Vmax = 24V, Imax = 3A)

2) NEVER perform the first tests WITH PROPELLERS INSTALLED ON THE MOTORS! REMOVE the propellers for safety.

3) NEVER REMOVE ALL SCREWS at the SAME TIME !!! If you want to replace them, remove carefully ONE of the screws and immediately replace it with the new one!

=> There is a precision fit cooling interface plate between the PCB and cooling plate and if it is not arranged perfectly, you may create a short circuit in the electronics!

Removing of cooling plate will result in Loss of Warranty!

4) Don’t use metal screws to mount the HERKULES III Cooling plate on your frame. If using the holes on the cooling plate, use plastic screws. I case of hard landing (or crash) the frame is not damaging the HERKULES III ESC because the screws are breaking first. Only in case of mounting the inner screws (Mikrokopter dimensions) use metal screws, but be very careful not to misalign the boards!

5) NEVER Try to re-solder or remove the thick battery cables from the middle of the PCBs. You will not manage it because the thermal impedance of the total system is very high and you need special equipment to be able to heat up the boards without destroying the electronics.

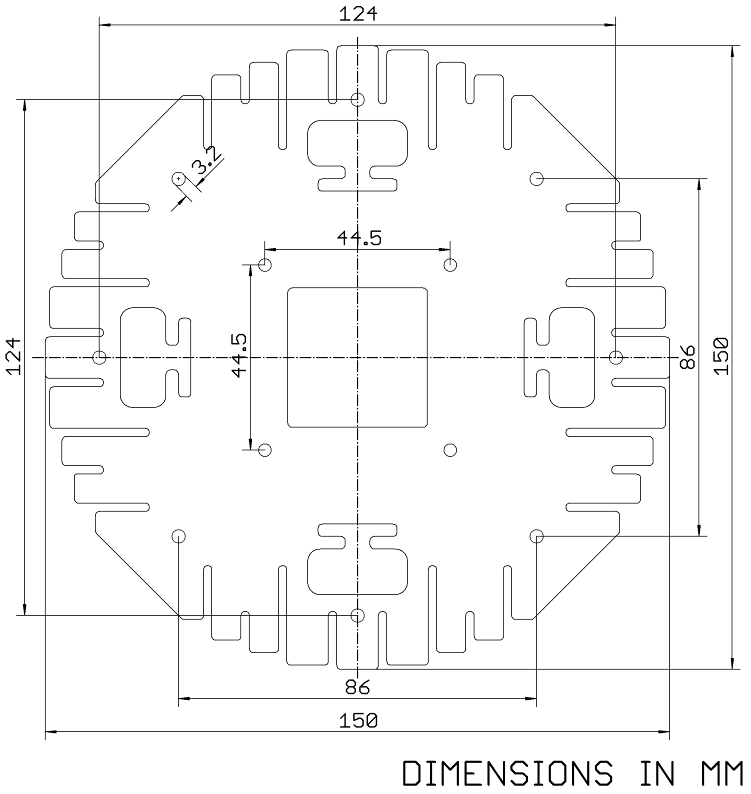
6) Only solder on the END of the wires, NEVER solder on the PCBs directly. The only permitted exception are the MOTOR wires. If they are not soldered by the factory, you can solder them but be very careful and check with magnifying glasses to make sure there are no a solder balls or wire fragments on the PCB after soldering.

7) Don’t use any protective paint or lacquer for protecting the electronics! The electronics may be destroyed by these materials. The warranty shall be voided by the use of such coatings.

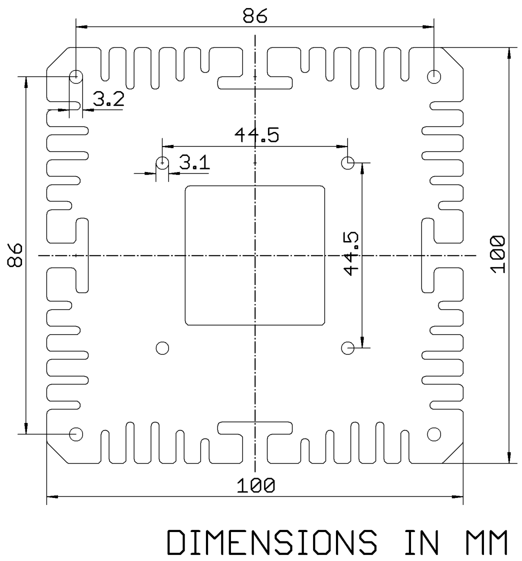
Please be aware that you are operating a 250A power system which demands complete respect and care during handling, setup and operation.

2 Dimension and Mechanics

2.1 Cooling Plate XL : 150x150mm



2.2 Cooling Plate L : 100x100m



3 Electrical Characteristics

3.1 Range of Functionality

The following parameters must not exceeded:

(all parameters are valid for 25°C ambient temperature, otherwise noted)

No.	Description	Symbol	min	typ	max	Unit	Comment
1	Ambient Working Temperature	T_amb	-20		55	°C	
2	Ambient Working Humitidy	H_amb			65	%	
3	Battery Supply Voltage	V_BAT	9		26	V	(6-cell Lipo Version)
4	Idle Current with stopped Motors	I_IDLE		150		mA	
5	Impedance of Supply Battery	R_BAT			50	mOhm	Higher impedance may cause voltage overshoot on Battery at motor decelleration => RISK of DAMAGE
6	Supply Battery Discharge Rate	C_BAT	20			C	Lower Discharge Ratge may cause voltage overshoot on Battery at motor decelleration => RISK of DAMAGE
7	Continous Average Motor Current	I_MOT_AVG_XL			30	A	XL Version only (150mm Cooling plate)
8	Peak Motor Current	I_MOT_PEAK_XL			60	A	Duration less than 20sec
9	Continous Average Motor Current	I_MOT_AVG_L			20	A	L Version only (100mm Cooling plate)
10	Peak Motor Current	I_MOT_PEAK_L			40	A	Duration less than 20sec
11	VAUX Voltage	V_AUX	8.2		8.6	V	Valid for V_BAT > 9V (Supply for Telemetry Interface)
12	VAUX Current	I_AUX			1.5	A	Valid for V_BAT > 9V Warning: Overload could lead to switch-off of all Motors!

Table 2 - Range of Functionality

3.2 Power Stage Electrical Specification

(all parameters are valid for 25°C ambient temperature, otherwise noted)

No.	Description	Symbol	min	typ	max	Unit	Comment
1	Powerstage ON Resistance	R_ON		3		mOhm	Excluding power and motor wiring
2	Motor Setpoint Duty Cycle with Active Freewheeling	PWM_AFW	15		92	%	If PPM Control Signal > T_HI_100, then PWM_AFW = 100%
3	Motor Current Duty Cycle without Active Freewheeling	PWM_noAFW	15		96	%	If PPM Control Signal > T_HI_100, then PWM_AFW = 100%
4	Motor Current Frequency	F_PWM	8		16	kHz	PWM frequency programmable
5	Motor Commutation Timing	ADV	12	18	24	deg	Phase advance programmable
6	Field Commutation Frequency	F_COM			200,000	rpm	To get Motor RPM, the commutation frequency has to be divided by number of motor pole pairs
7	Undervoltage shutdown threshold	V_UV		8.5		V	Powerstage disabled below this voltage
8	Short Circuit detection threshold	I_OC_LMT		120		A	Powerstage is disabled as long the OC event is present
9	Over voltage protection threshold	V_OV_LMT		26		V	Powerstage refuses to start if voltage is higher than this
10	Over temperature limitation threshold	T_OT_LMT		100		°C	Powerstage reduces output power to 50% of setpoint
11	Over temperature switch-off threshold	T_OT_OFF		115		°C	Powerstage is disabled as long the OT event is present

Table 3 – Power Stage Electrical Specification

3.3 I/O-Interface Signals Specification

(all parameters are valid for 25°C ambient temperature, otherwise noted)

No.	Description	Symbol	min	typ	max	Unit	Comment
	PPM1..8, I2C_SCL, I2C_SDA, RxD, TxD						
1	Level of digital signals	V_MAX			5.3	V	5V Interface
2	min HI-Level of Input Signals	V_HI_IN	2.7			V	5V Interface
3	max LO-Level of Input Signals	V_LO_IN			1.8	V	5V TTL

Table 4 - I/O-Interface Signals Specification

4 Connectors, Functions and Features

4.1 The HERKULES III Powerboard

Powerboard Block Overview

A Herkules III Hexa- or Octo Configuration consists of 2 power boards mounted top and bottom on one cooling plate.

Figure 1 – HERKULES III Block Schematic – Connectors and functional blocks on one quad board

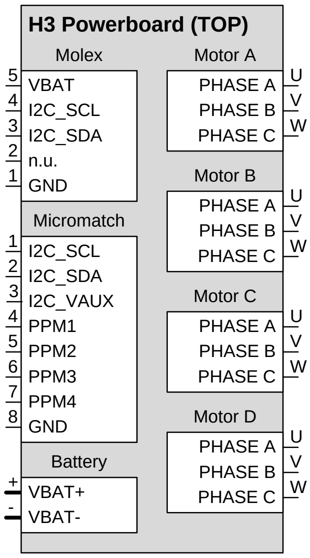
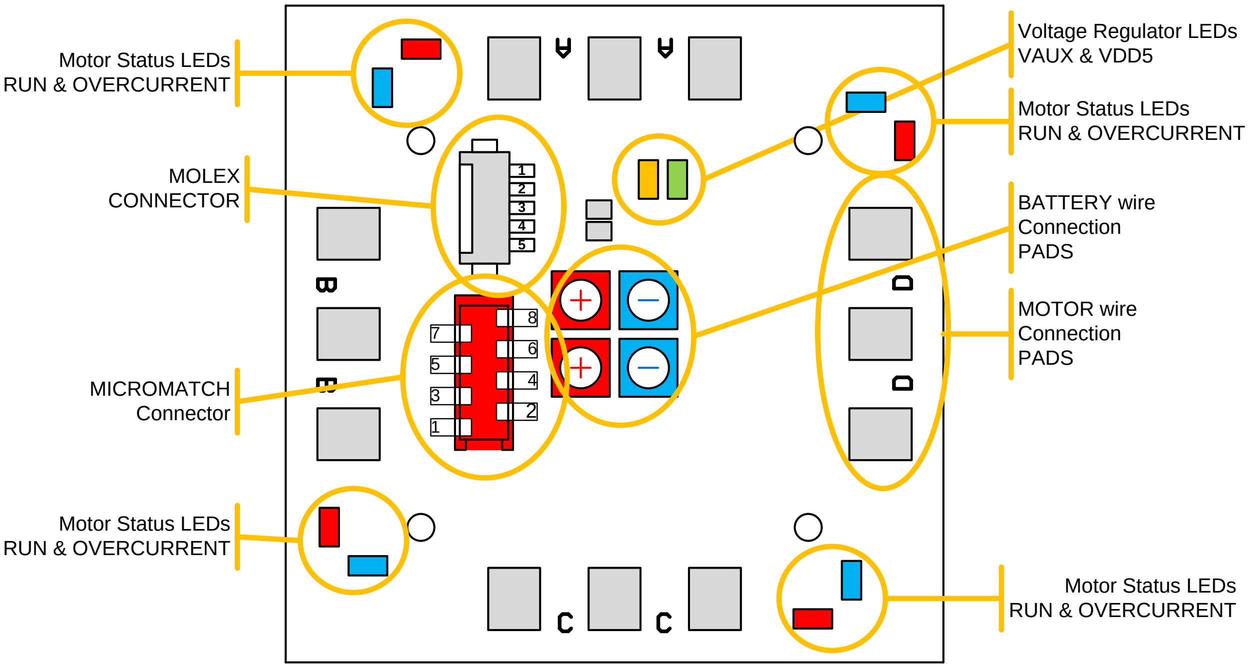


Figure 2 - HERKULES III Overview



Battery Connector

Use the 4 pre-soldered battery wires to connect two (2) identical and equally charged LiPo battery packs in **parallel** (max 6s LiPo each). So both red are connected together on the PCB and both blue wires are connected together on the PCB.

Important Note:

In case of using 2 or more battery packs in parallel, use cells of the exact same voltage, capacity and C-rate. It is important that all packs are charged equally (fully) and balanced perfectly before being connected to the power board. Failure to operate in this manner creates a high risk of destroying the battery packs and damaging the vehicle!

Always terminate the open ends of the wires with an isolated connector or isolate it with a secured shrink tube to avoid unintended short circuits to neighboring wires or other electronic parts.

Never try to re-solder the battery wires from the PCB or solder any other wires on the PCB!

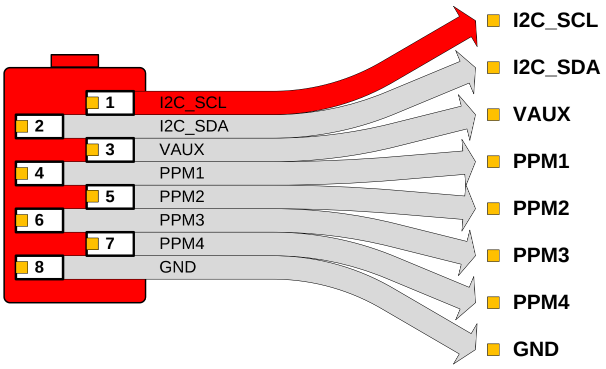
Micromatch Connector

The red 8-pin connector on the HERKULES III ESC contains the serial I2C Bus signals (I2C_SCL, I2C_SDA), the parallel PPM control signals (PPM1…4) and HERKULES **III** internal breakout voltage (VAUX).

The pins are connected to:

- PIN 1 and 2: The I2C Bus is for communication and telemetry readout,
- PIN 3: The Internal VAUX is used as supply voltage for the Telemetry interface board.
- PIN 4 to 7: The PPM1-4 Control signals are assigned to each ESC A,B,C,D of the HERKULES III quad-board and
- PIN 8: Ground

Figure 3 - HERKULES III Micromatch Connector



On Pin VAUX the HERKULES III Internal 8V/1.5A Supply Voltage is available. This can be used e.g. for the HERKULES III Telemetry Interface Board or supplying an External Flight Control.

Important Note:

Don't source more than 1.5A from this PIN! An Overload on this Voltage can lead to a complete switch-off of all Motors connected on the HERKULES III ESC. If you supply the external Flight Control with this voltage, take care that the over-all current consumption on this PIN will never exceed the Max-Ratings specified in Chapter "Electrical Characteristics" on page 11.

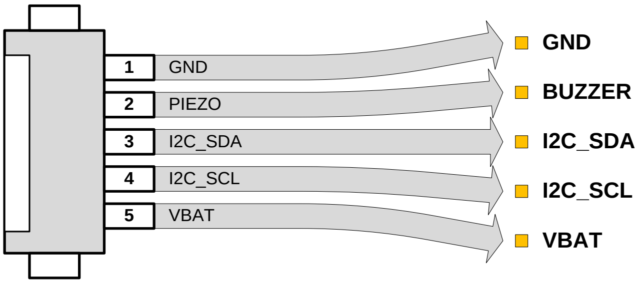
Molex Connector

The grey 5-pin Connector contains the serial I2C Bus signals (I2C_SCL, I2C_SDA), the Buzzer (not used) and the breakout Battery Voltage. The connector is **compatible with Mikrokopter Flight Control 2.1 ME**

The pins are connected to:

- PIN1: Ground
- PIN2: Buzzer Output from Flight Control (PIEZO)
- PIN3 and 4: The I2C Bus is for communication and telemetry readout,
- PIN5: The Internal VBAT or VAUX Voltage

Figure 4 - HERKULES III Molex Connector



By default, the HERKULES III Powerboard is jumpered to supply PIN5 directly with the battery voltage.

In case of using Mikrokopter, this connection is the only needed interface to flight control. There are no other Wirings necessary. For details please see the Chapter "**Setup with Mikrokopter FlightControl ME2.1 (I2C Control)**" on page 51.

Remember that the Mikrokopter Flight Control does not accept voltages higher thant 5S (18V). In case you want to drive a Mikrokopter System with more than 5S please refer to the separate description available on the internet.

Battery Power Wires

The Battery wires are already soldered by factory to the HERKULES III PCBs. The users can trim the length and solder a Battery connector on the end of these wires. It is not allowed to de-solder the wires from the PCBs because due to the high thermal impedance the electronic components might be destroyed

Motor connection Wires

The 3-phase Motor connection wires are located at the outside of the PCBs. The user has to solder carefully the motor wires or Motor connectors (e.g. gold contact female connectors).

Important Note: After soldering on the PCB check the board very carefully for solder bubbles or short circuits between electronic pins. Do the first supply tests only on a current limited power supply and REMOVE the Propellers from the motors. In case of unintended runaway of the motors there is a big risk of hazardous injury due to rotating systems.

4.2 Operation Modes and LED functions

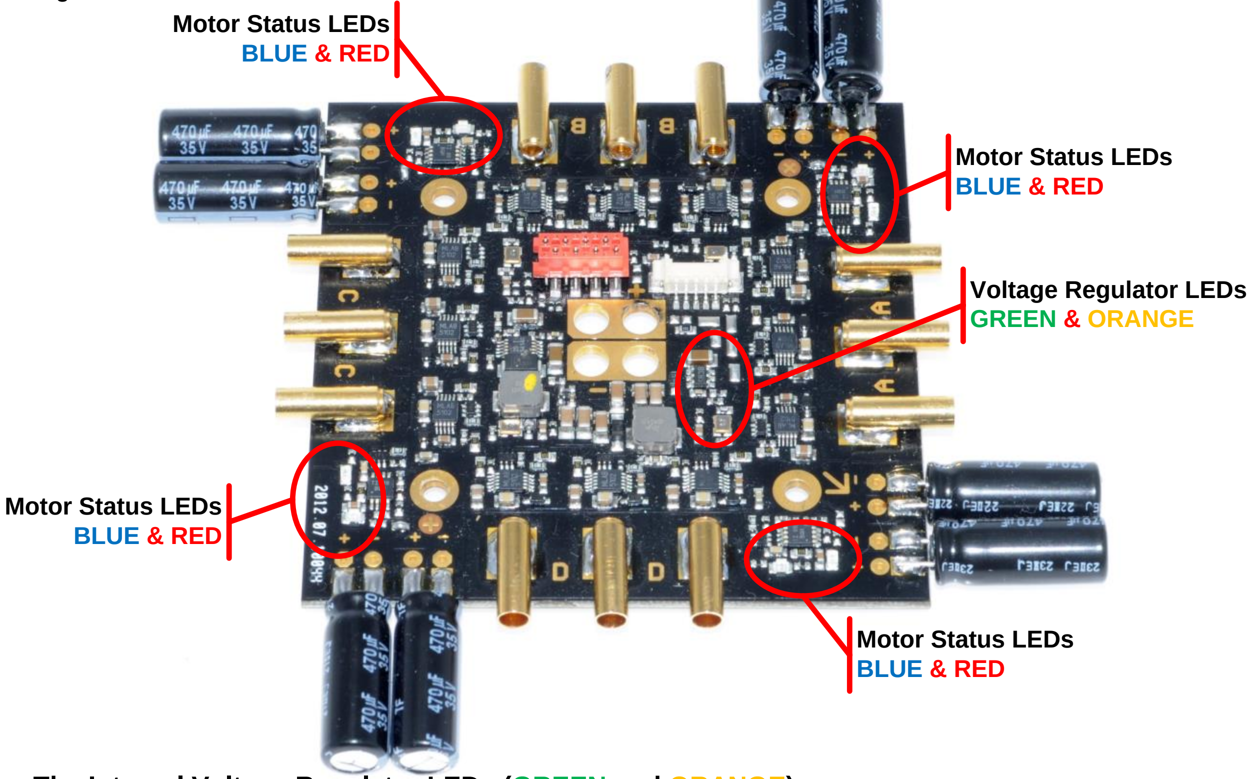
On the HERKULES III Powerboards are various LEDs indicating the status each ESC

a) The Internal Voltage Regulator LEDs (**GREEN** and **ORANGE**)

b) The ESC / Motor Status LEDs (**BLUE**)

c) The Overcurrent Protection LEDs (**RED**)

Figure 5 - Location of Status LEDs



The Internal Voltage Regulator LEDs (GREEN and ORANGE)

- The GREEN LED indicates a correct working internal 5V Voltage and
- The RED LED indicates a correct working internal 8V Voltage

If both LEDs are continuously on, the internal Power supply is working correctly.

The ESC / Motor Status LEDs (BLUE)

The blue LED is ON in case of

- The Motor is running correctly or
- The Motor controller receives valid motor setpoints

The brightness of the LEDs varies with the Motor Setpoint value. The higher the setpoint, the higher the brightness of this LED.

The Overcurrent Protection LED (RED)

A flashing RED LED indicates an overcurrent event on the Motor output which could be caused by

- A fast accelerating motor. The LED is on as soon the Motor speed increases and the ESC limits the power to avoid an overstressing of the Power MOSFETs
- A failure at the motor occurs. e.g. shorted motor wires or shorted windings in the motor itself.

A shortly flashing LED during acceleration is usually normal only in case the LEDs is continuously on and the motor has startup and running problems the system wiring and components should be checked carefully.

4.3 Protection and Diagnosis Modes

Each Individual ESC is protected against various failures. Each protection event is detected, stored and transmitted to the telemetry interface or saved on the MicroSD-card of the Telemetry interface.

The following protection mechanisms are implemented.

- Overvoltage Protection
- Overtemperature Protection
- Overcurrent Protection
- Control Signal Timeout
- Motor Stall Detection
- Setpoint Monitoring

The reaction of each ESC in case of detected Failure mode is described below:

4.3.1 Overvoltage Protection

The battery voltage is monitored and reported in the telemetry feedback data. In case it is higher than the V_OV_LMT (see Table 3 on page 12), the ESC refuses to start. This value is only checked after first power on and only in case the ESCs had not been started. If a voltage increase higher than V_OV_LMT happens during runtime, the ESCs are NOT switched off.

4.3.2 Overtemperature Protection

The temperature of each individual ESC is monitored and reported in the telemetry feedback data. The over temperature protection has two detection thresholds.

a) V_OT_LMT : Power limitation to 50%

b) V_OT_OFF : Complete ESC switch off

In case the temperature is higher than the T_OT_LMT (see Table 3 on page 12) the ESC goes to a power limitation mode. The Motors are still running but the output power of the ESCs is only 50% of the actual requested power by the setpoint. When the ESC temp falls again below the T_OT_LMT minus a hysteresis threshold, the ESC output power limitation is switched off.

In case the temperature rises further after V_OT_LMT has been activated, the individual ESC is switched off completely and is locked until the motor setpoint goes below the Motor OFF detection threshold VAL_OFF_I2C (see Table 13 on page 99) or the Motor OFF detection Time T_HI_OFF_PPM (see Table 14 on page 101).

During over temperature switch-off, the affected motor “plays” an over temperature sound.

4.3.3 Overcurrent Protection

The current in each motor phase is monitored (not the Battery current!) and in case this current goes above the I_OC_LM threshold (see Table 3 on page 12), the ESC reduces the output power until the current goes below this threshold. The ESC does NOT switch off completely and keeps on working as long as the electronics can control the motor commutation correctly.

Overcurrent events might occur

a) Dynamically : e.g. during acceleration of heavy load motors with big propellers or

b) Statically : e.g. by a short circuit in the motor windings or motor wires.

A dynamic over current event leads “only” to a slower acceleration of the motor. The influence to the overall flight behavior will not be noticeable.

A static over current event will usually cause a stall (blocking) of the motor and this is detected by the stall detection see below.

4.3.4 Motor Stall detection

The motor control algorithm monitors the commutation times of each motor phase and in case of a detected abnormality a STALL event is detected. This event is reported also to the Telemetry Interface. The ESC goes to lock mode and is only re-activated when the motor setpoint goes below the Motor OFF detection threshold VAL_OFF_I2C (see Table 13 on page 99) or the Motor OFF detection time T_HI_OFF_PPM (see Table 14 on page 101).

A Motor stall event could occur e.g. when a motor is blocked or propeller is mechanically locked or the bearings of the motor are defect.

4.3.5 Control Signal Timeout

The Motor control signal is monitored and in case of there is no control signal any more for a timeout of more than T_TO_PPM (see Table 14 on page 101) or T_TO_I2C (see Table 13 on page 99) the motor is stopped. This is mainly a safety feature in case of a broken PPM or I2C control wire. This ensures that the motor stops safely after this timeout.

4.3.6 Control Setpoint monitoring at first power-on

At first power-on the motor control setpoint is monitored and in case it is higher than the Motor START detection threshold VAL_START_I2C (see Table 13 on page 99) or VAL_START_PPM (see Table 14 on page 101) the motor refuses to start. Only when the setpoint value comes back below the Motor Off Detection Time, the ESC is initialized and enables a motor start when it is required by the motor setpoint.

This feature avoids a motor runaway after power-on in case of the motor setpoint is unintentional high e.g. by open wires or the flight control is not working correctly.

4.4 The PPM / I2C Breakout Board

The PPM breakout board is a small interface which simplifies the connection of servo-patch-cables to the flat ribbon cable.

Note: This interface is only necessary in case of using PPM controls like DJI Wookong.It is NOT needed in case of using Mikrokopter Flight Control.

Figure 6 - PPM Breakout Board

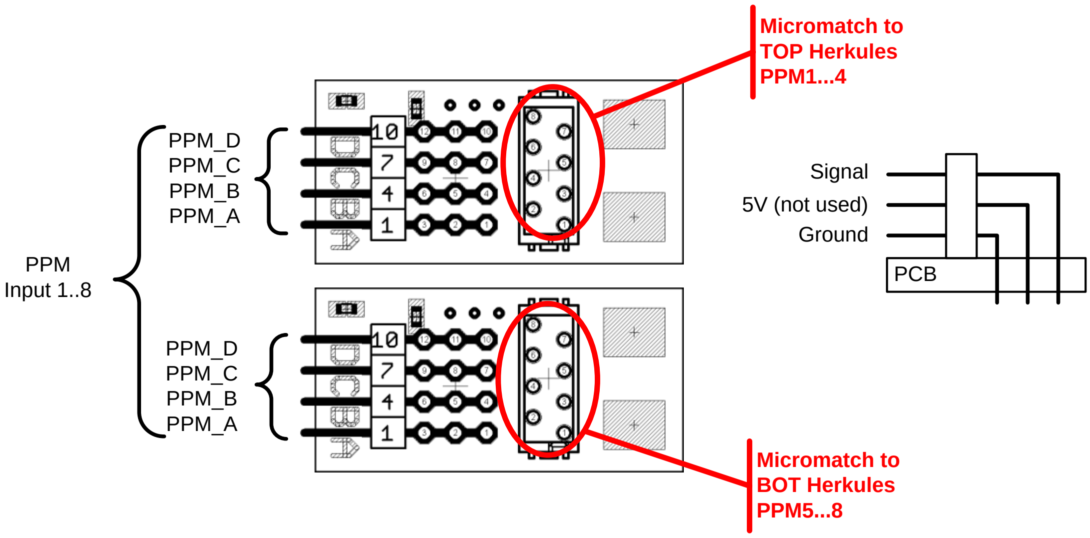
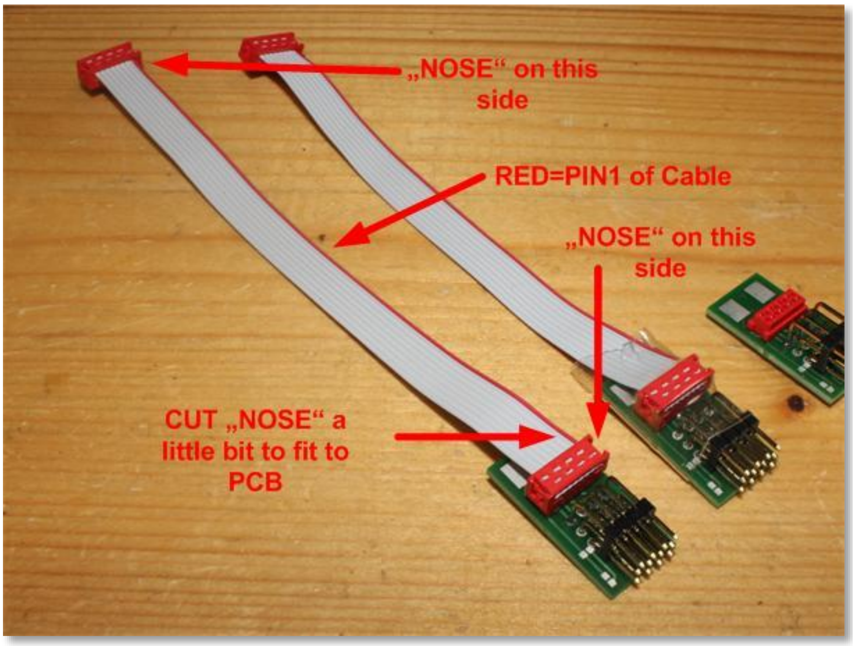


Figure 7 - PPM/I2C Breakout Board ready assembled



The connection between PPM breakout board and HERKULES III ESCs is done with a flat ribbon cable with pressed red Micromatch connectors. If the red connector does not fit correctly on the breakout board, cut the small “Nose” from the connector.

The length of the flat ribbon cables should as short as possible to reduce **in-coupling of magnetic distortions** to the control signals. The connection to the Flight Control should be done with short servo-patch cables. Please check the polarity of the cables! Black (GND) is close to the PCB (printed circuit board).

Please check the polarity of this cable according the picture above to ensure a correct wiring.

4.5 The HERKULES III Telemetry Interface

The Telemetry Interface allows the connection of PPM Flight Controls with HERKULES III and a direct connection to telemetry RC-Receivers. It can be used as an alternative to the PPM breakout board with additional features. There are two different versions of the telemetry interface available.

Functions of the Telemetry Interface:

- LEDs signaling the status of 2 HERKULES III ESCs and Datalogger
- PPM Signals 1...8. Servocable connector to FlightControls like DJI WKM
  (For connection to Flight Control please refer to the application section)
- Micromatch Connectors to 2 HERKULES III ESCs
- Telemetry Interface to HERKULES III with Datalogger
- 8V/2A breakout of the HERKULES III internal Aux Voltage
- Barometric pressure Sensor (for altitude measurement)
- And some other features more

Important Note:

This interface is only necessary in case of using PPM controls like DJI Wookong. It is NOT needed in case of using Mikrokopter Flight Control!

Firmware update:

Firmware update of the Telemetry Interface has to be done separately for the Telemetry Board and the Datalogger. The Telemetry Board (the piece with the 2 red Micromatch connectors) has to be updated with the standard AVRootloader. The Datalogger (the pieze with the SD-card connector) has to be updated by copying the new files according user manual on the SD card.

The Firmware update procedure of the Telemetry board is described in

Chapter 5.7 "Update Procedure TELEMETRY INTERFACE" on page 37.

The Firmware update procedure of the Datalogger described in

Chapter 5.8 "Update Procedure DATALOGGER" on page 42.

The details of the telemetry protocols is described in chapter

"RC Telemetry " on page 59.

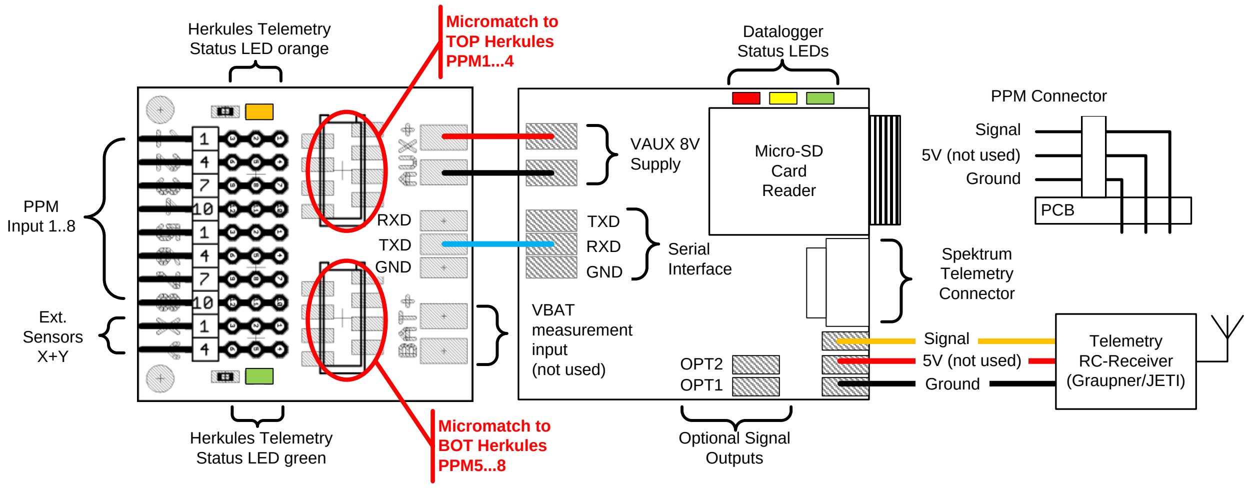
4.5.1 HERKULES TELEMETRY Interface Version 1

The Interface Version 1 is build of 2 PCBs which are mounted together.

a) The HERKULES Telemetry Interface Controller with PPM and Micromatch Connector

b) The HERKULES Datalogger Interface Controller Controller with Micro-SD card and interface connectors to different RC telemetry systems

Figure 8 - Telemetry Interface Board Version 1



4.5.2 HERKULES TELEMETRY Interface Version 2

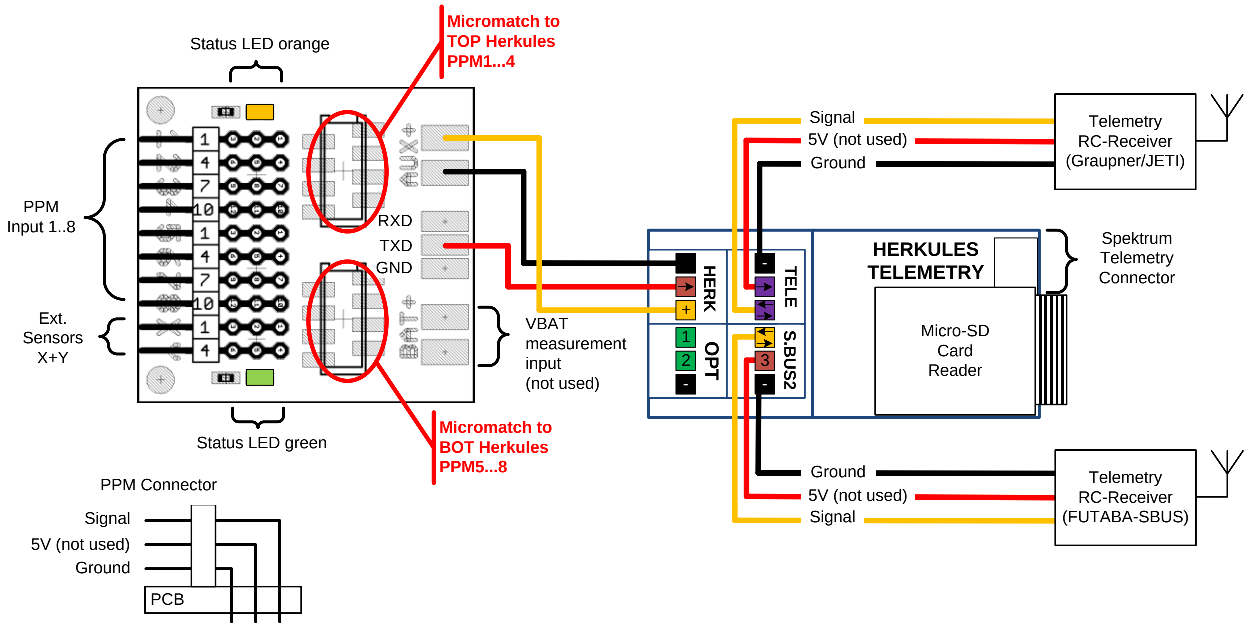
The Interface Version 2 is build of 2 parts which are connected together.

a) The **HERKULES Telemetry Interface Controller** with PPM and Micromatch Connectors

b) The **HERKULES Datalogger Interface Controller v2** with Micro-SD card and interface connectors to different RC telemetry systems.

The difference to Version 1 is that the datalogger is a separate module and can be placed better on a place where the SD card can be accessed easily for data read out and analysis.

Figure 9 - Telemetry Interface Board Version 2



4.6 The HERKULES III Precharger Module

Description here Missing

Figure 10 - The Precharger Module

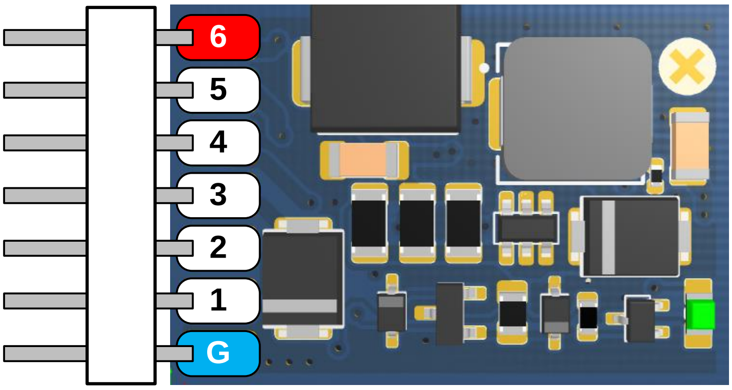


Figure 11 - Precharger Connection with 6s LiPo Battery

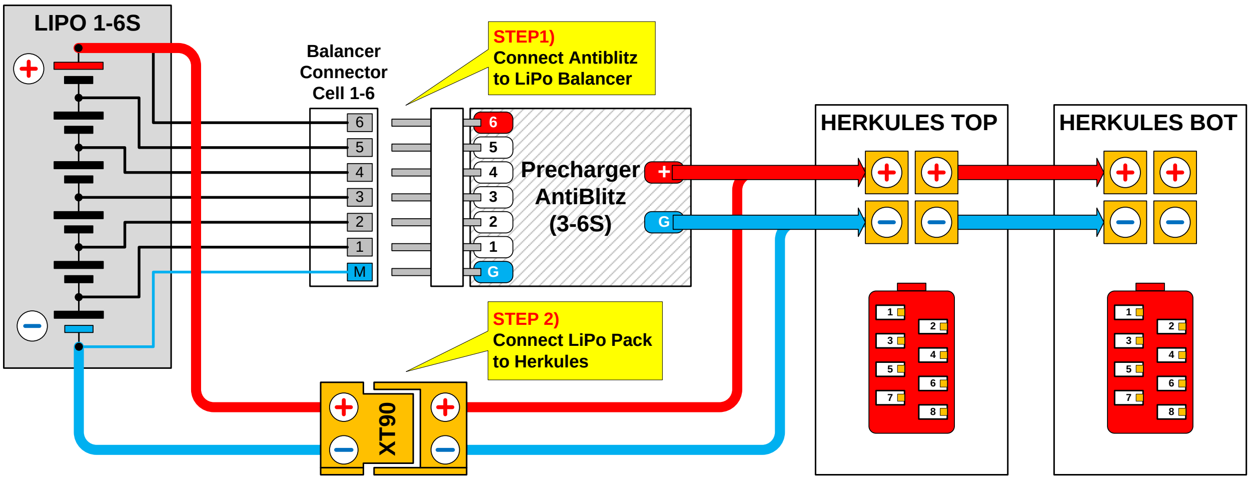


Figure 12 – Precharger Connection for 12S Batteries with single pin connectors

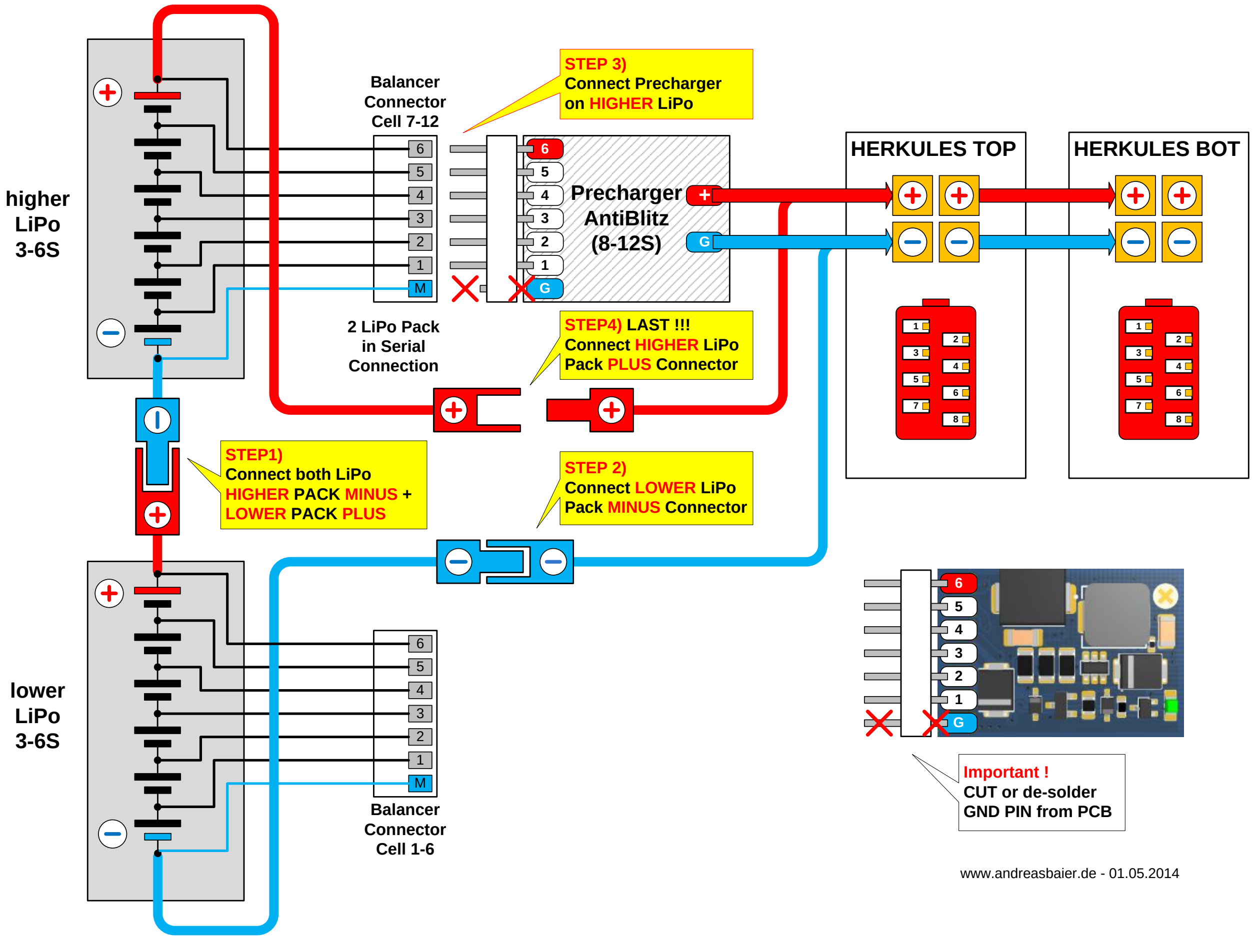


Figure 13 – Precharger Connection for 12S Batteries with double pin connectors

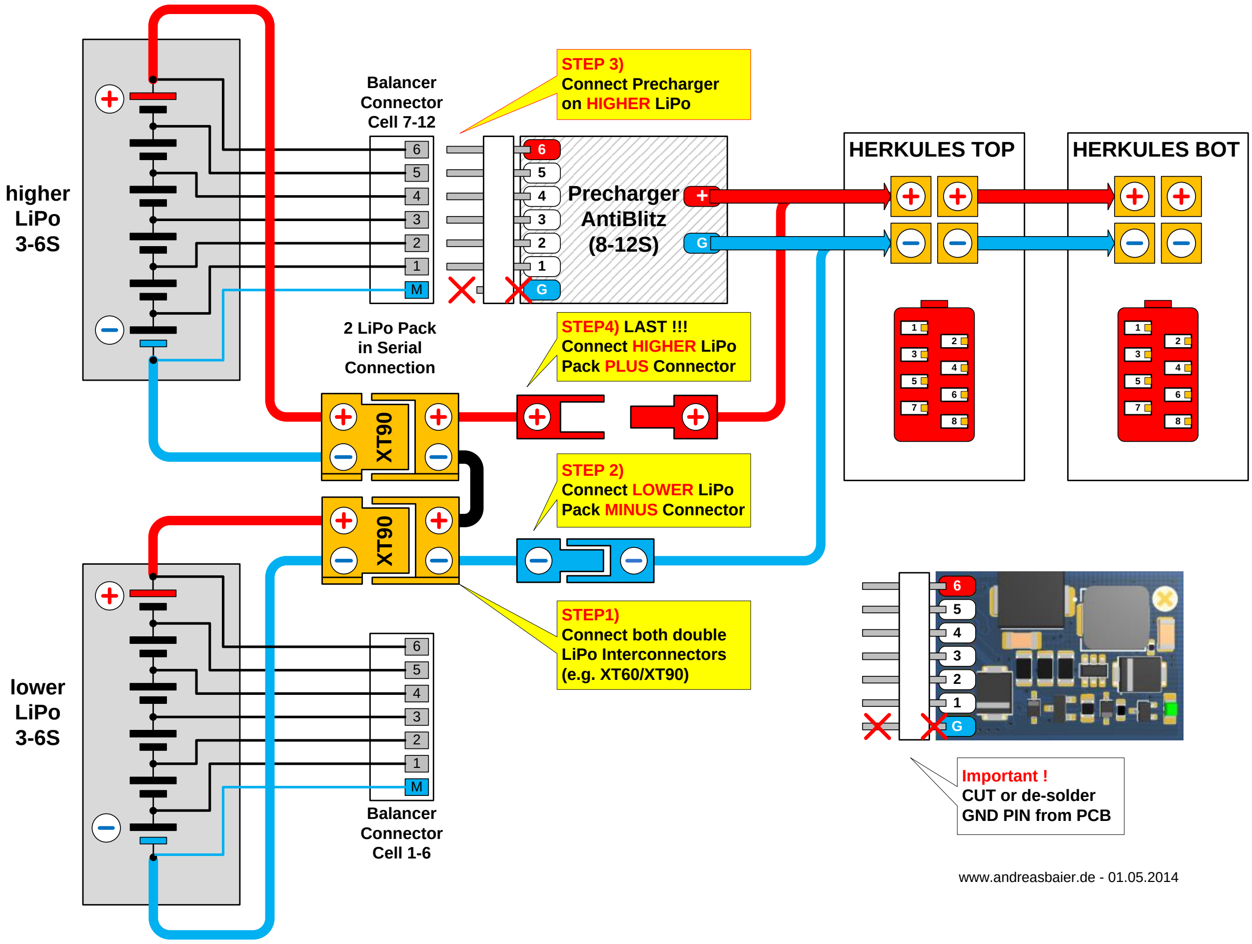
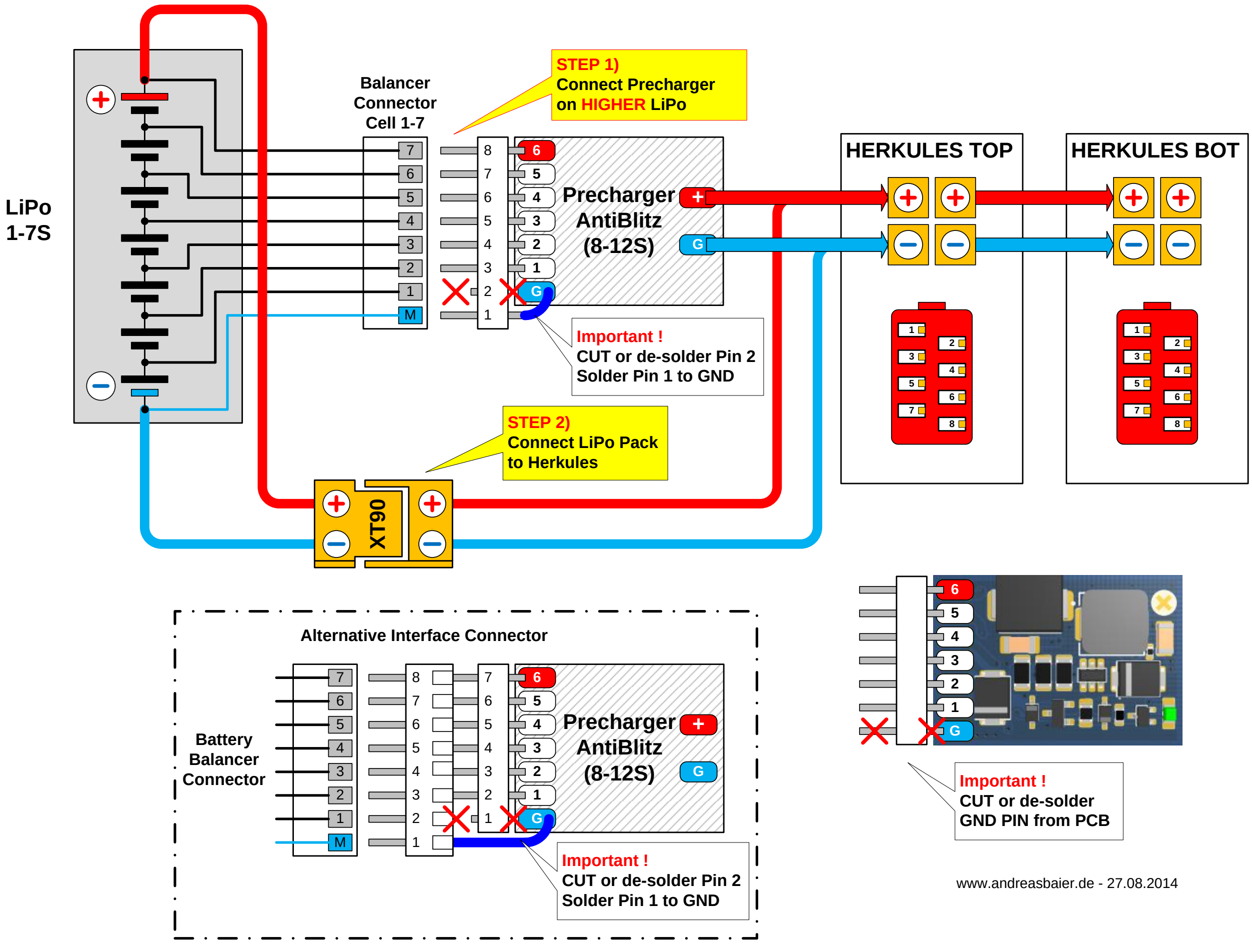


Figure 14 – Precharger Connection with 7s and double pin connector



5 Software Update and Programming

This chapter describes the programming and firmware update of the HERKULES III ESC power boards and HERKULES III Telemetry Interface..

The latest firmware can be downloaded at www.andreasbaier.de.

If you are asked, please register, click on the confirmation link you received by email and login with your account. You will find the latest Firmware and User Manual in the download section.

5.1 Programmable Features

- **Firmware update** of HERKULES III Powerboard and Telemetry Interface
- Setpoint Control Mode **PPM** (e.g. DJI WKM) or **I2C** (e.g. Mikrokopter)
- Telemetry **ON (PPMT, I2CT)** or **OFF (PPM, I2C)**
- Programming of **I2C-Addresses (M01..M16)** to each individual HERKULES ESC
- Active Freewheeling **ON (AFW1)** or **OFF (AFW0)**
- PWM switching frequency **(FREQ)** 16kHz or 8kHz
- Commutation Phase Advance **(ADV)** 12°, 18° or 24°

Factory Settings at delivery:

The HERKULES III ESCs are delivered with a default configuration which is a good compromise for most of the currently available brushless motors for multicopters. There should be little reason to modify these settings. Modify the settings only if you know exactly that your motor performs better or you have troubles with commutation stability.

The effects of the different control modes are described in chapter

“Selecting the Firmware and Control Modes” on page 28.

Most of the common multicopter motors should run well with this setting.

- **PPM** : PPM control mode active, no Telemetry
- **AFW1** : Active Freewheeling switched ON
- **kHz16** : PWM switching frequency = 16kHz
- **ADV18** : Phase advance / Timing = 18 degrees
- **I2C-ADDRESSES** : Default I2C mapping see Figure 15 on page 28
- **CH4 (CH6) Version** : CH4 is default, CH6 has additional information. See 5.3.8 on page 30.

If the user wants to change any of these parameters, the appropriate firmware files have to be flashed into each microcontroller of the powerboard and the optional Telemetry Interface.

5.2 Hardware Versions and Revisions

As HERKULES III hardware and software is continuously being improved, each hardware revision works only with its dedicated firmware. By using coded boot signs it is only possible to program the matching firmware to the hardware.

Available Hardware Revisions:

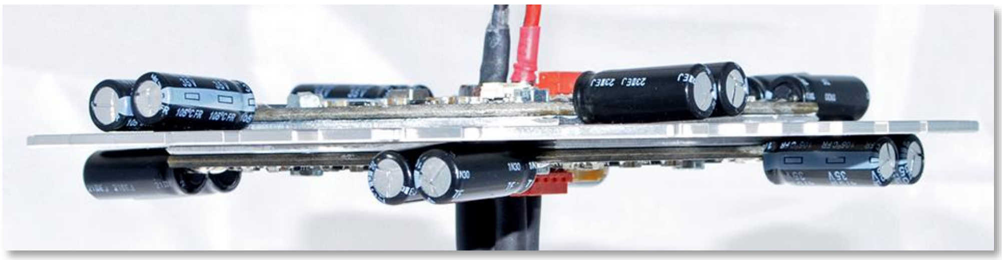
<table>
<tr><th>Hardware Revision</th><th>Booloader Signs</th><th></th></tr>
<tr><td>HERKULES III v2</td><td>BLDCHK3v2_A…H</td><td></td></tr>
<tr><td>HERKULES III v3</td><td rowspan="2">BLDCHK3v3_A…H</td><td rowspan="2">Same Bootloader for v3 and v4</td></tr>
<tr><td>HERKULES III v4</td></tr>
<tr><td>HERKULES III v5</td><td>BLDCHK3v5a_A…H</td><td>HERKULES III 12S Variant</td></tr>
</table>

5.3 Selecting the Firmware and Control Modes

Depending on the use-case and type of flight control, the user has to choose from the attached programming files the correct one with the best fitting features for his system setup. Depending on motor, battery voltage, propeller and type of flight control, the Herkules III must be programmed with the right files.

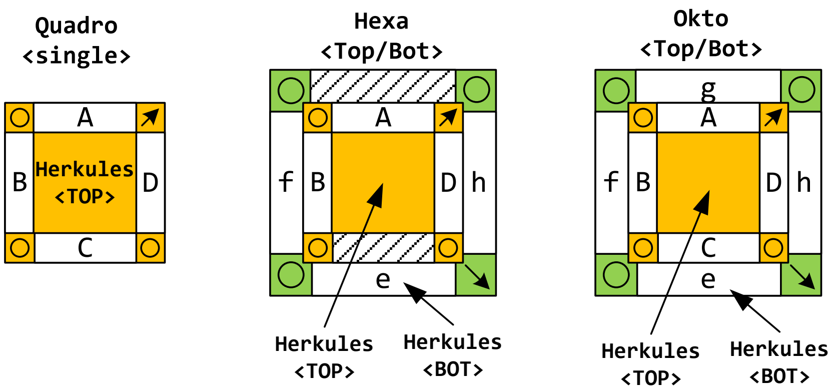
Bootloader Mapping

Each ESC on the HERKULES III powerboard has its unique Bootloader Address (Bootsign).
The motor channels are numbered with A,B,C,D on each quad PCB. If the 2 PCBs are mounted on a single cooling plate TOP and BOT like shown in the picture below, even the TOP and BOT PCBs have different Bootsigns.



As the bottom HERKULES III is mounted "flipped" across the B-D axis (A-C exchanged) the motor addresses have to be set accordingly. The top board has the bootsigns written in superscript "**A,B,C,D**" and the bottom board has the bootsigns written in subscript "**e,f,g,h**" (which are linked on the physical "A,B,C,D" ESCs on the bottom board).

Figure 15 - Bootloader Signs in different Mounting Options



QUAD		
NAME	Bootloader Bootsign	Default I2C-Address
A	BLDCHK..._A	01
B	BLDCHK..._B	02
C	BLDCHK..._C	03
D	BLDCHK..._D	04

HEXA			
Quad ESC	NAME	Bootloader Bootsign	Default I2C-Address
TOP	A	BLDCHK..._A	01
	B	BLDCHK..._B	02
	C	not used	#
	D	BLDCHK..._D	04
BOT	E	BLDCHK..._E	05
	F	BLDCHK..._F	06
	G	not used	#
	H	BLDCHK..._H	08

OKTO			
Quad ESC	NAME	Bootloader Bootsign	Default I2C-Address
TOP	A	BLDCHK..._A	01
	B	BLDCHK..._B	02
	C	BLDCHK..._C	03
	D	BLDCHK..._D	04
BOT	E	BLDCHK..._E	05
	F	BLDCHK..._F	06
	G	BLDCHK..._G	07
	H	BLDCHK..._H	08

Note: If you can't connect the BOT Board with E,F,G,H then use A,B,C,D also on the BOT. The first delivered HERKULES Octo-Versions had the same Bootsigns for TOP and BOT.

5.3.1 Control Mode (PPM or I2C)

Depending on the needs of the used Flight Control, the right control protocol must be selected.

- **PPM: Pulse-Pause- or Pulse-Position Modulation**. Is a standard RC control protocol for Servo interfaces. Most common standard in RC systems.
  Parallel Control of each ESC individually via a separate Control line or Servo connector.
  The Motor Setpoint is modulated with the pulse width of the control signals.

  Two variants of this protocols can be found in the programming files:
  a) **PPM**: standard protocol **without Telemetry** feedback
  b) **PPMT**: PPM protocol **with Telemetry feedback** via I2C Bus

  Specification details on PPM-Mode see in chapter
  “PPM-Mode : Setpoint Write via PPM and Data Read via I2C” on page 101.

- **I2C: Inter-Integrated-Circuit or IIC-Bus.** This is a standard in consumer electronics, originally invented by PHILIPS. Benefit is the less wiring effort and the control of up to 128 Devices on one single bus. Serial Control of all ESCs via two common BUS wires (SDA=Data and SCL=Clock) and also the transmission of Telemetry data from the ESCs is possible.

  Two variants of this protocols can be found in the programming files:
  a) **I2C**: (setpoint control **without Telemetry** feedback
  b) **I2CT**: (setpoint control **with Telemetry** feedback via I2C bus

  Specification details on I2C-Mode see in chapter
  “I2C-Mode : Setpoint Write and Data Read via I2C” on page 99.

5.3.2 Active Freewheeling (AFW)

This is a method to dramatically reduce the power loss of the power MOSFETs. Another positive is that the ESC can decelerate the motor actively by directing current back to the battery (regenerative braking). With AFW ON, the motor reaction time to setpoint changes is reduced and the multicopter flight control loop gets faster and more stable.

By default, AFW is enabled (**AFW1**). The user can disable AFW by using the **AFW0** files but this is **NOT recommended!** The ESCs will run hotter and the risk that the ESCs will switch off due to over temperature increases.

5.3.3 PWM switching frequency (kHz)

The average current in the motor phases is set by switching the voltage at the motor phases on and off with a certain ratio ON to OFF. The higher the ON-Time and the lower the OFF-Time, the higher the power in the motor. The frequency of this ON-and-OFF sequence can be changed with the value **KHz.**

The higher the frequency, the smoother the motor runs in low-load condition. The default value is 16kHz and should only be changed if the user knows exactly what he is doing.

5.3.4 Commutation Phase Advance (ADV)

This value defines the angle of the motor timing. Standard is ADV18 and should fit to most of the motors.

It’s a good compromise between commutation stability and efficiency. Some high pole motors (e.g. > 22-poles) might require a higher value like ADV24. In case of commutation problems at high RPMs use this value.

Low pole Motors (e.g. < 14poles) might have a slightly higher efficiency with ADV12. But the risk of commutation errors increases with this value. Change the timing only when you are sure what you do!

5.3.5 I2C-Address

The I2C Addresses are especially relevant in Mikrokopter application. Each ESC has to have its unique address so that the flight control can write the set points and read the telemetry data from the ESCs.

In case of using PPM flight controls like DJI WKM, HERKULES III uses a mixed-mode of controlling the ESCs via parallel PPM signals and reading the telemetry data from the ESCs via the serial I2C bus.

Each ESC has to have its individual I2C-address only in case of using Mikrokopter or the external HERKULES III Telemetry Interface. For “simple” PPM control mode only, ESCs don’t need an I2C Address and can have the same PPM firmware on all ESCs.

The user can select any of 16 I2C addresses. The detailed explanation of the available address-range is described in “**Chapter I2C Address Range**” on page 98.

Important Note:

Each I2C address may only be used once in a complete system. Do not use the same I2C address on more than one ESC channel. The ESCs will not respond to the flight controls´ commands properly and this will lead to a crash of the multicopter!

5.3.6 Unused ESCs

In case not all ESCs shall be used on the HERKULES III Powerboards, e.g. using an Octo-Board driving only six 6 motors for a Hexacopter, then two ESCs are unused.

To avoid interaction of these unused channels with the used ones, especially when using I2C control and telemetry feedback, you have to program the unused channels with a dummy firmware.

Please program the unused channels with the firmware settings:

PPM / AFW1 / kHz16 / ADV18

Please note to use the **PPM** (no Telemetry) and **NOT the PPMT** (with Telemetry) firmware.

Otherwise the unused channels will influence the I2C communication in I2C mode and Telemetry mode.

5.3.7 Changing Motor Rotational Direction

To change the spin direction of a motor, simply swap any two of the three motor wires. It does not matter which ones you select but exchange only two.

HERKULES III does not support changing motor rotational direction by software!

5.3.8 CH4 or CH6 Version

Default for Telemetry Data is the CH4 version. With this, CH1…CH4 are used

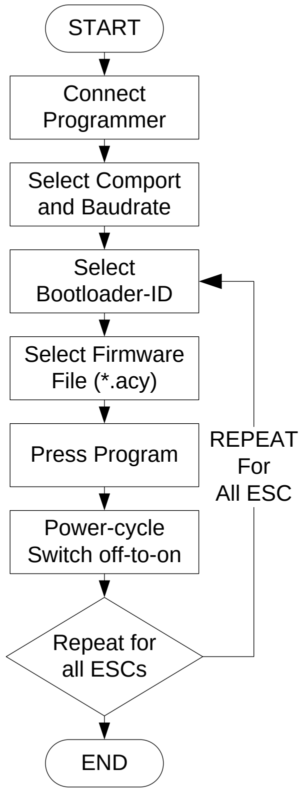
With the CH6 version there is additional Telemetry data information in CH5 and CH6 available. Here are the single cell data Voltage fields “re-used” with Single ESC RPM and Single ESC VBAT measurement). For details please refer to the protocol description in Chapter 8.4 on page 94.

5.4 General Firmware Update Procedure

The general software update process is the same for the HERKULES III Power boards and the Telemetry Interface. The boot loaders are responding to the programming tool only in a time-frame of 100msec after power-on-reset. This requires a power-cycle (off-to-on) at each programming sequence.

Programming Steps

1) Disconnect any Flight Control from the Herkules
2) Connect Programming Interface to Herkules III
3) Start Programming Tool
4) Select Comport and Baud rate
5) Select Bootloader-ID of target ESC
6) Select Software update files
7) Press “Program”
8) Do Power cycle (switch On and Off)
9) Repeat Step 1) to 6) for each ESC



Important Note:

Please disconnect any Flight Control from the communication interface (Micromatch or Molex Connector) to avoid any disturb of the programming sequence!

Firmware Folder Structure

Download and unzip the file and you will find the following structure:

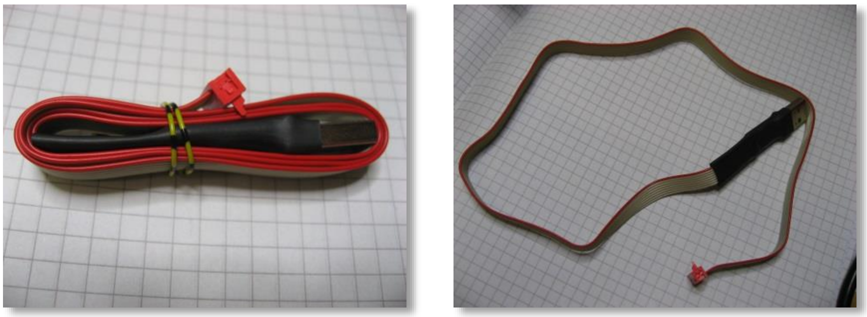
Table 5 - Content of the Firmware Update Folder

01_Herkules_ESC	Update files for the ESCs. The firmware files for selecting AFW, kHz, ADV and I2C Addresses are arranged in subfolders
02_Telemetry_Interface	Update files for the Telemetry Interface Board.
03_Data_Logger	Update files for the Datalogger Interface Board
04_LogView	LogView Setting Files with OpenFormat definition to display HERKULES III Telemetry Data
AVRootloader.dev Typ: DEV-Datei	Bootloader Device File. Don’t change or delete this file
AVRootloader.exe Typ: Anwendung	HERKULES III Programming Tool “AVRootloader.exe”. Use this application to update the firmware.
AVRootloader.ini Typ: INI-Datei	Bootloader INI File. Don’t change or delete this file
VersionInfo.txt	Version and change information of the current firmware.

5.5 The Programming Adapter

To program the HERKULES III ESCs a simple USB- to-Serial converter is needed which delivers the signals RxD, TxD and GND. The fastest and easiest way is to use the original HERKULES III USB-to-HERKULES III ProgTool. This converter has a fitting Micromatch connector for the HERKULES III Boards and it can be used directly to program and update the firmware or all components.

Figure 16 – HERKULES III Programming Adapter



Self-made programming tool

As the programming interface is a standard serial any self-made programming tool could be used. For this the I2C-pins have to be mapped to RxD and TxD.

- Connect PIN I2C_SCL to USB_RxD
- Connect Pin I2C_SDA to USB_TxD
- Connect GND to the serial interface

Figure 17 – HERKULES III Programming Tool with FTDI USB to TTL Adapter

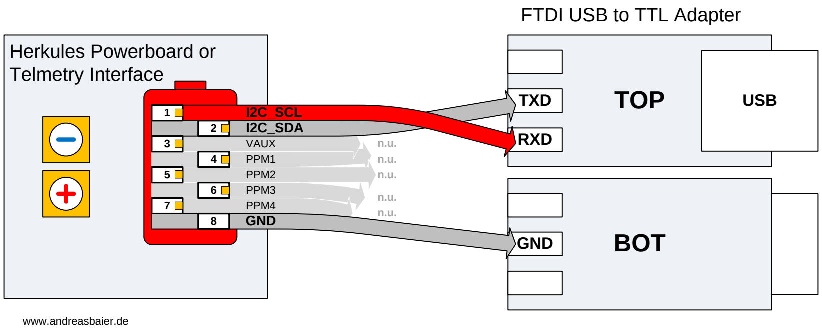
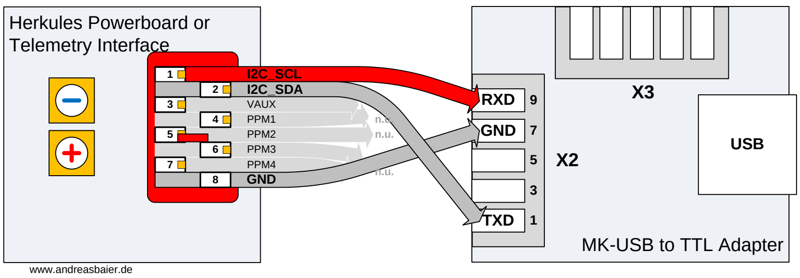


Figure 18 – HERKULES III Programming Tool with MK-USB Adapter



The programming adapter can be connected directly on the HERKULES III Powerboard or on any of the Telemetry Interface Micromatch connectors. Both Connectors on the Telemetry Interface are equally usable for programming the whole system. Remember: The supply of the Telemetry Interface boards is done from the HERKULES III Powerboard, therefore must be always one connection via flat ribbon cable to any of the HERKULES III Powerboard connected.

Driver installation for the Programming Adapter

Follow the download link on the website at www.andreasbaier.de and load the latest driver for the FTDI programming tool from www.ftdichip.com.

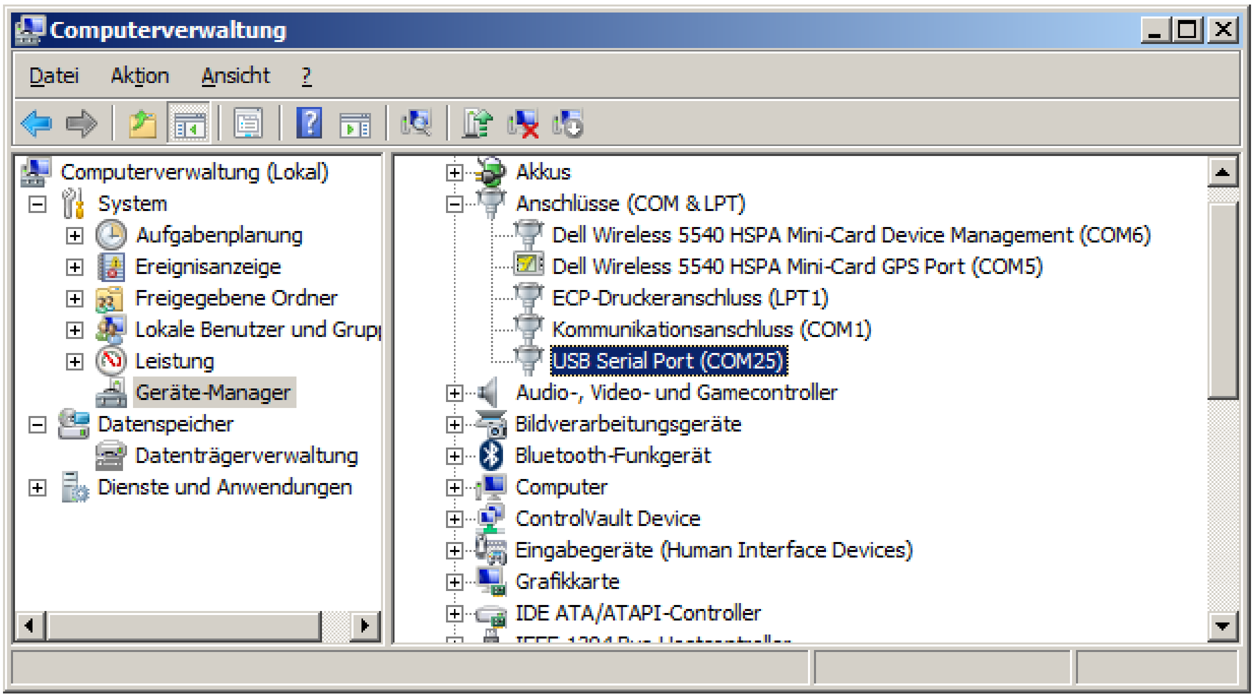
Download to Virtual Comport Driver “VCP” and install the driver.

Then connect the programming adapter to your PC. The device should be detected by your System.

Now find out the **Virtual Comport Number** of your programming adapter.

Open the Device Manager (Windows XP, 7) go to “Device Manager / Geräte Manager” and check the Number behind the “**USB Serial Port (COMxx)**”.

Figure 19 - Finding out the ComPort of Programming Adapter



Note this number and use it when you are asked in the HERKULES ProgTool.

5.6 Update Procedure HERKULES III Powerboards

Important Note:

Please disconnect any Flight Control from the communication interface (Micromatch or Molex Connector) to avoid any disturbance of the programming sequence!

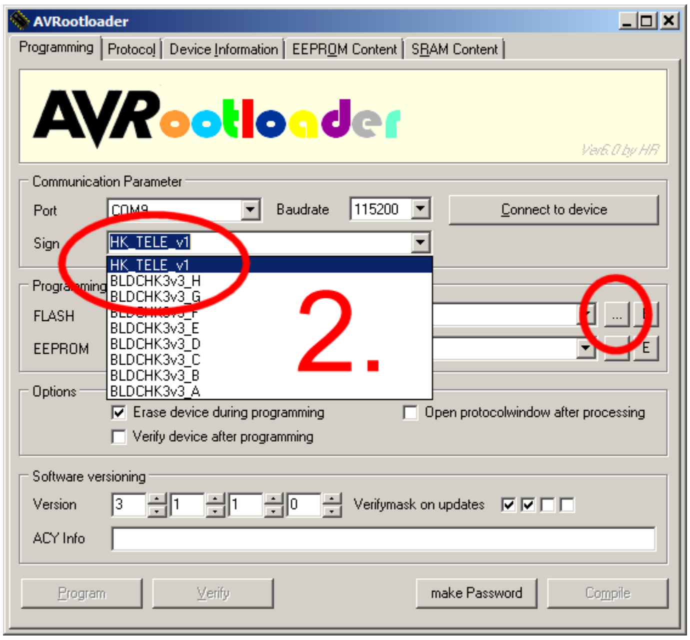
Please connect the USB programming adapter to either the Herkules Powerboard connectors or on one of the connectors on the Telemetry interface.

From the HERKULES III ProgTool Folder, start the application “AVRootloader.exe”.

Step 1 - Selecting the Comport

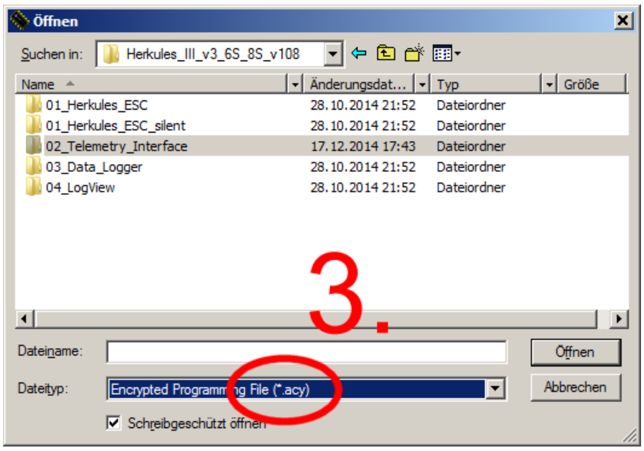
First select the correct Comport of the USB-to-HERKULES III - ProgTool. The Port must not be set to “AUTO”. You have to set there the real Comport which is used by the ProgTool.

Step 2 - Select the bootsign of the target ESC



Select the one of the target ESC bootloaders in “Sign” which are ending with M01 to M04 or A to F.

Step 3 – Selecting the right file type

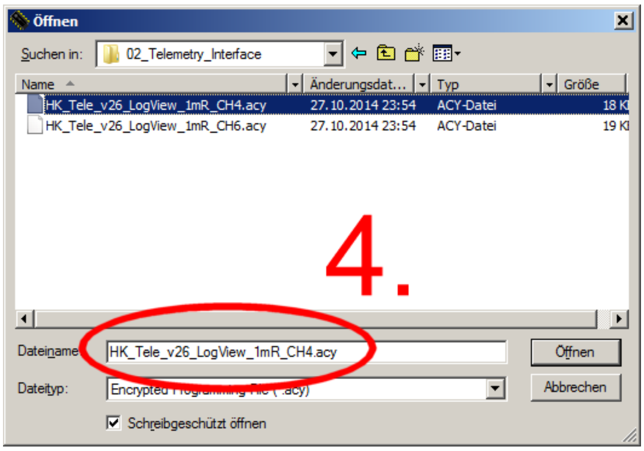


Now click on the “3 dots button” on the right and chose in the popping-up file selection winde the file type to “Encrypted Programming File (*.acy)”

Important Note:

Please check carefully that you are in the right sub folder! If you have an older version of the ProgTool already installed on your PC, the tool may remember the path to the previous software. Please do here “the long way” and browse from the root of your PC to the place where you have stored the latest HERKULES III ProgTool.

Step 4 – Selecting the desired files



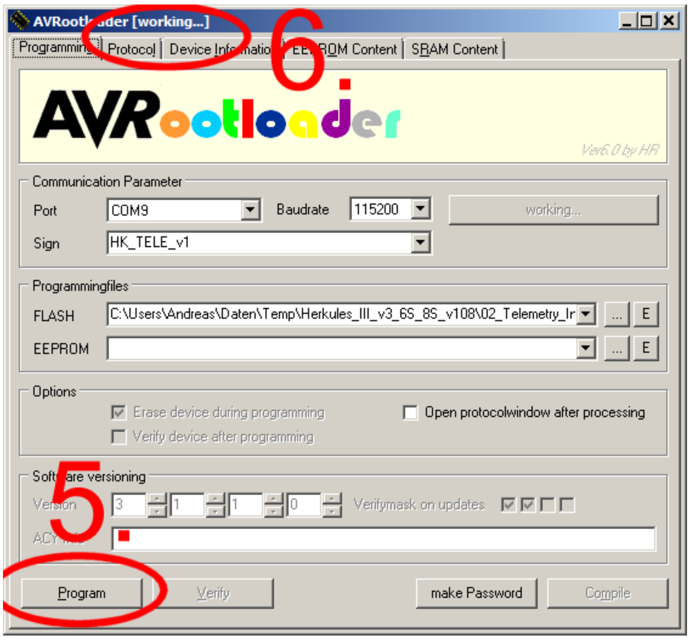
Browse to the subfolders and find the intended “acy” file with the whished setting.

Remember:
The Version “CH4” is standard.

The Version “CH6” is for extended Telemetry data output in the datalog file.

Instead of Single cell voltage the RPM of each individual motor is stored.

Step 5 and 6



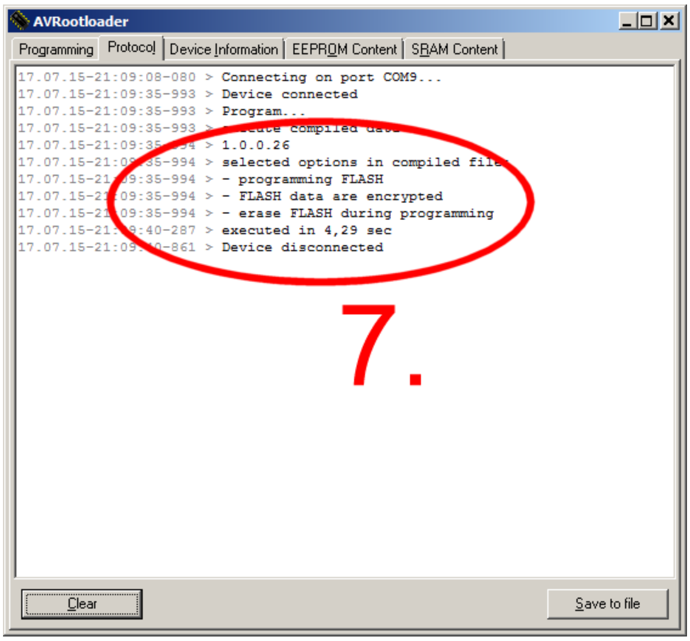
Connect now the USB-to-Herkules Programming Cable with the HERKULES III and do the following steps

1.) Switch-OFF the power supply

2.) Click button "Program" The ProgTool is now waiting for a power-on-reset for 100msec after power-on

3.) Switch-ON the power supply and the programming starts

Step 7 – Check the programming



If the programming was successful you can see in the "Protocol" window. It should look like this.

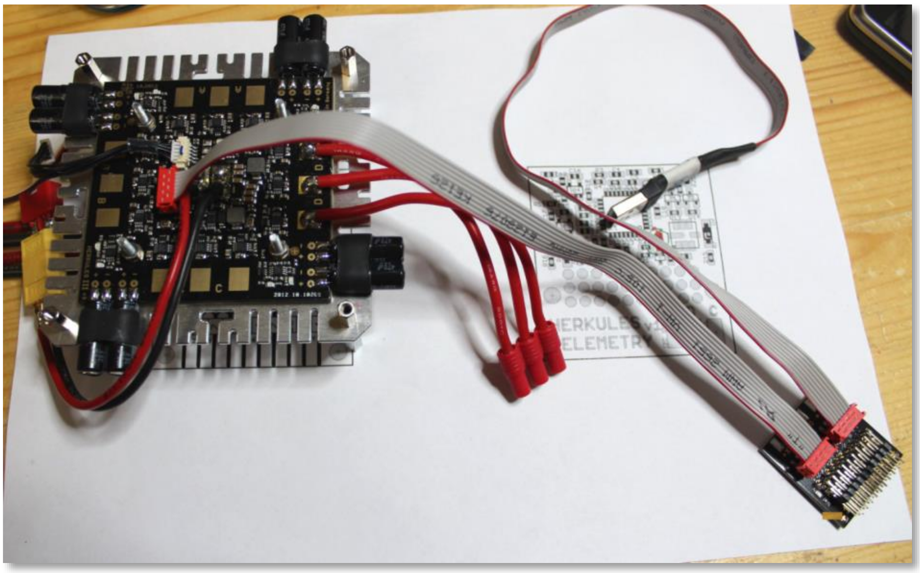
5.7 Update Procedure TELEMETRY INTERFACE

From the HERKULES III ProgTool Folder, start the application “AVRootloader.exe”.

The programming sequence for the Telemetry Interface is similar to the Powerboard's.

The only difference is the boot sign which has to be selected in the programming tool.

Figure 20 - Programming via Telemetry Interface



When the USB-to-HERKULES III ProgTool is connected to the Telemetry Interface, both HERKULES III Powerboard’s can be accessed and programmed without direct connection of the programmer to the HERKULES Boards.

The programming adapter can be connected on any of the HERKULES telemetry interface micromatch connectors. Both Connectors on the Telemetry Interface are equally usable for programming the whole system. Remember: The supply of the Telemetry Interface boards is done from the HERKULES III powerboards, therefore it must be always one connection via flat ribbon cable to any of the HERKULES III powerboards.

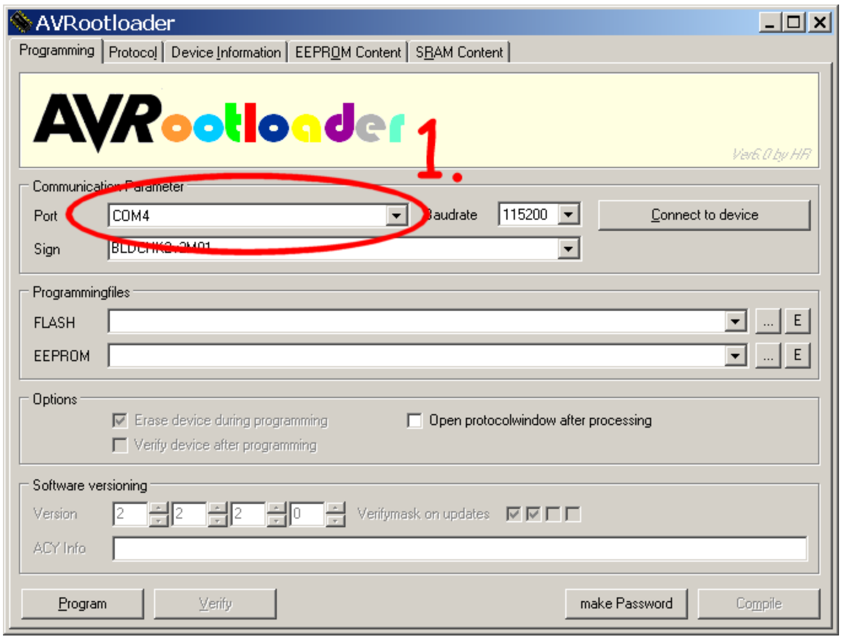
In case of programming the HERKULES III Boards via the Telemetry Interface, it might be that the Bootloaders do not respond. In this case, connect the programmer directly to the HERKULES III boards again and redo the programming.

Important Note:

Please disconnect any Flight Control from the communication interface (Micromatch or Molex Connector) to avoid any disturbance of the programming sequence!

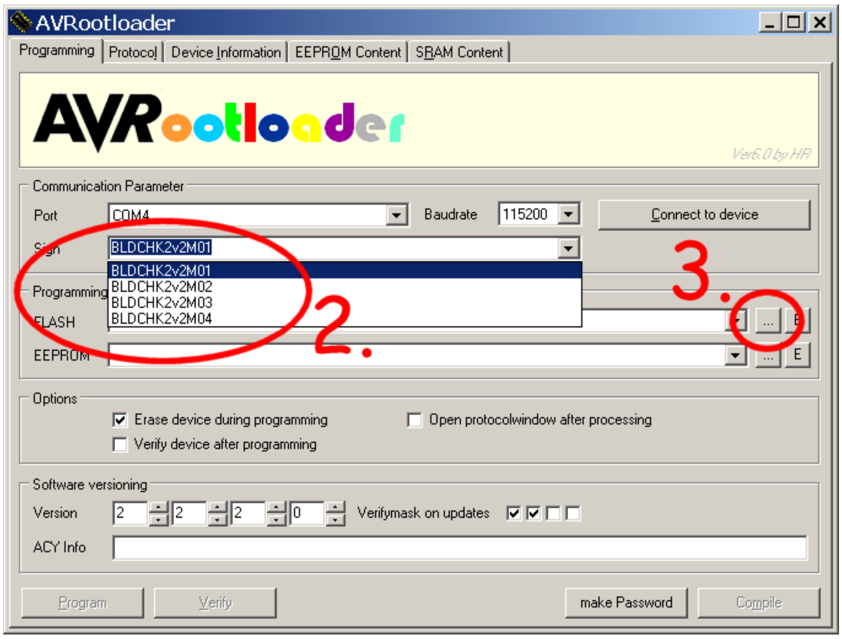
From the HERKULES III ProgTool Folder, start the application “AVRootloader.exe”.

Step 1 - Selecting the Comport



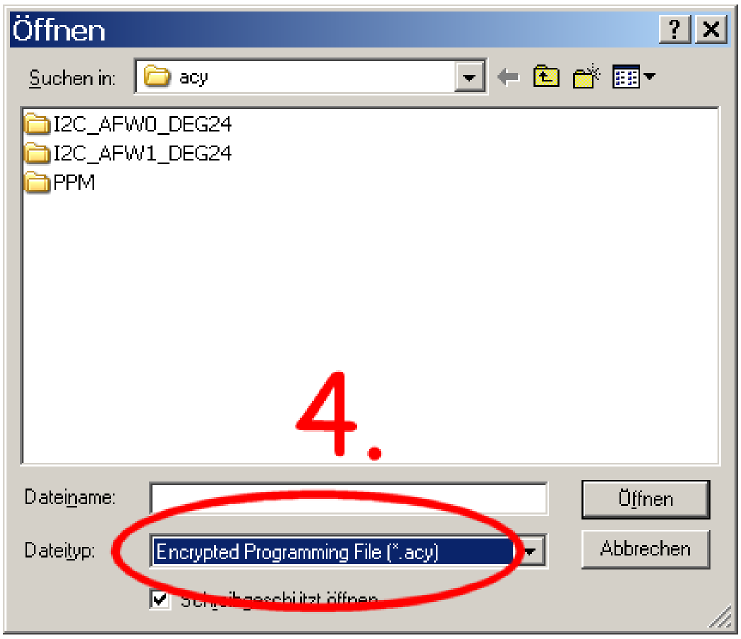
First select the correct Comport of the USB-to-HERKULES III - ProgTool. The Port must not be set to “AUTO”. You have to set there the real Comport which is used by the ProgTool.

Step 2 - Select the bootsign of the target ESC



Select the one of the target ESC bootloaders in “Sign” which are ending with M01 to M04 or A to F.

Step 3 and 4 – Selecting the right file type

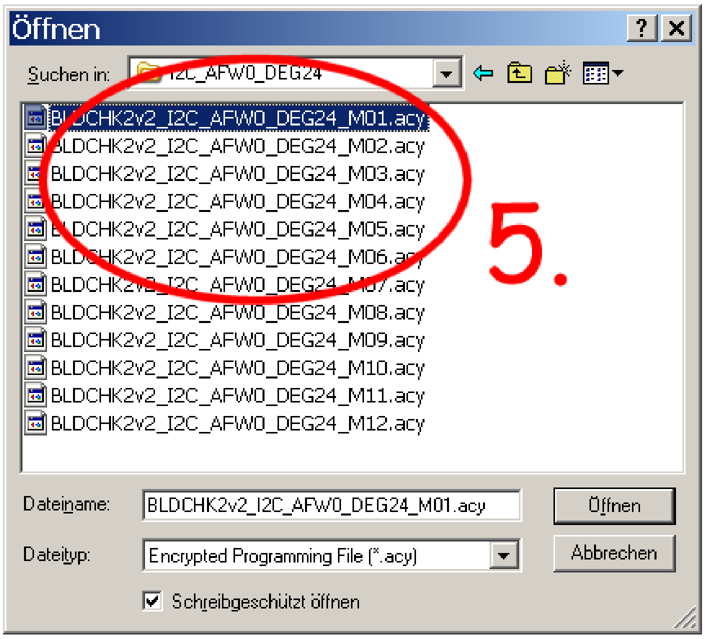


Now click on the “3 dots button” on the right and chose in the popping-up file selection winde the file type to “Encrypted Programming File (*.acy)”

Important Note:

Please check carefully that you are in the right sub folder! If you have an older version of the ProgTool already installed on your PC, the tool may remember the path to the previous software. Please do here “the long way” and browse from the root of your PC to the place where you have stored the latest HERKULES III ProgTool.

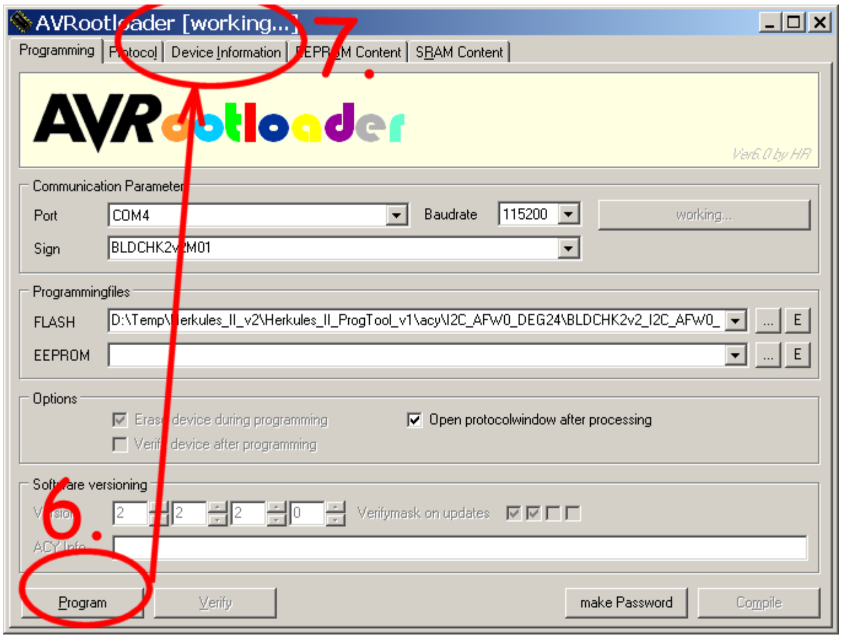
Step 5 – Selecting the desired files and the desired I2C-Address



Browse to the subfolders and find the intended “acy” file with the whished setting.

The file ending with 01 to 16 are the dedicated I2C-addresses which have to be arranged to each ESC

Step 6 and 7 – Programming



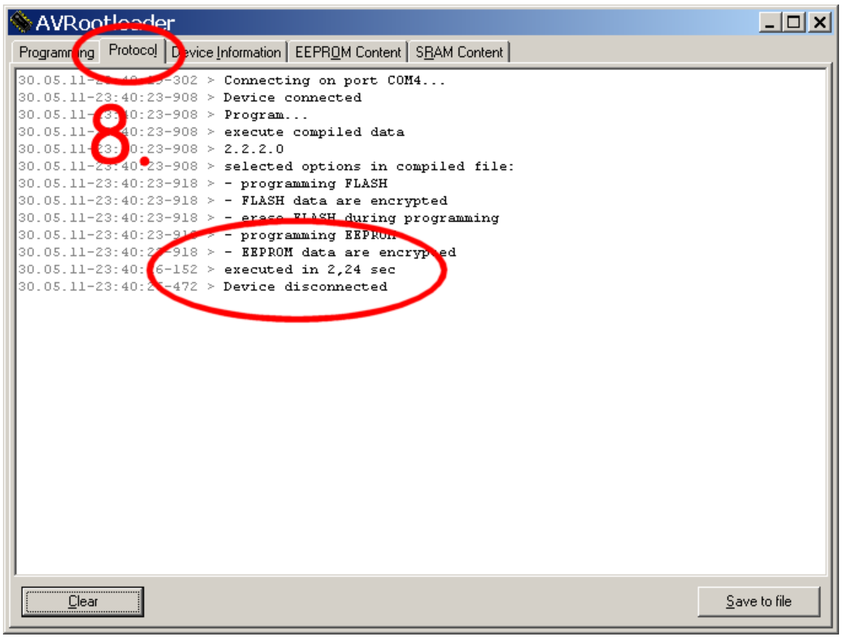
Connect now the USB-to-Herkules Programming Cable with the HERKULES III and do the following steps

1.) Switch-OFF the power supply

2.) Click button "Program" The ProgTool is now waiting for a power-on-reset for 100msec after power-on

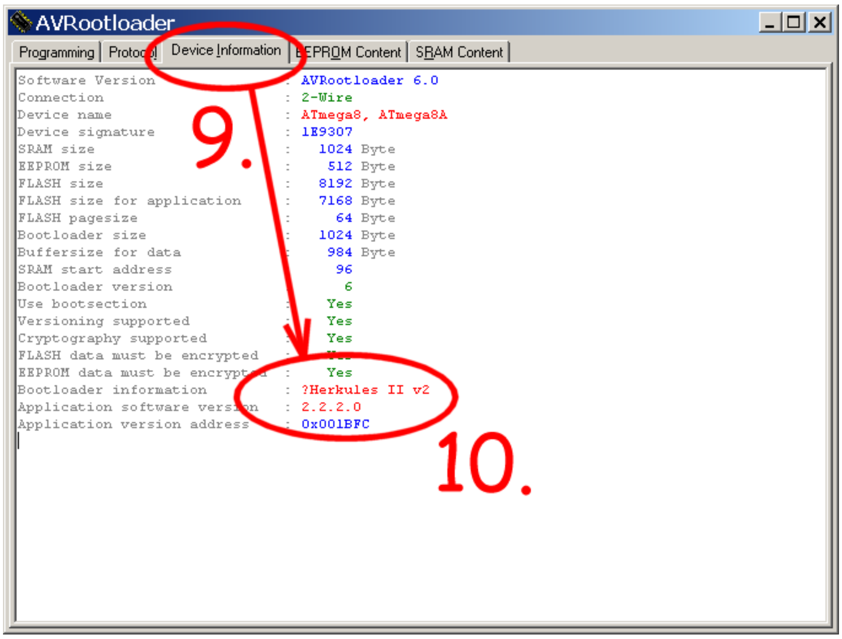
3.) Switch-ON the power supply and the programming starts

Step 8 – Check the programming



If the programming was successful you can see in the "Protocol" window. It should look like this.

Step 9 and 10 – Checking of device-information



Click on “Device Information” and check the Bootloader Information

5.8 Update Procedure DATALOGGER

The Datalogger is able to emulate lots of different RC-Telemetry Systems.

Supported Vendors

- **Standard Serial** Protocol for data live view with data logging software UNILOG
- **GRAUPNER HoTT (released)**
- **JETI DUPLEX (+EX) (released)**
- **FUTABA S.BUS (released)**
- **MULTIPLEX M-LINK (released)**
- **SPEKTRUM (released)**
- **JR (released)**
- **HITEC (released)**
- **FRYSKY (released)**

The Micro-SD-card in the Datalogger must contain special files to enable a certain Telemetry System.

Important Note:

The file system on the SD card must mandatory FAT16 and FAT32 is NOT supported. Therefore the data storage is limited to 2GB max. I t might be that SD cards from different manufactures are not supported. In case you are unsure, please use the original SD card included. You can order also the original replacement card at your supplier.

Table 6 - Content of the Datalogger MicroSD-Card

d510-xx	Datalogging Folder. All generated logfiles from the data logger will be stored in this folder
Firmwares	Firmware updates for different Telemetry Systems are stored here. Copy the content of the subfolder to the Root folder
.setup.hti Typ: HTI-Datei	System internal file. Don't delete or change this file!
HTI25_HOTT_192.123_textafterCH4.bin Typ: VLC media file (.bin)	Firmware File which should be programmed to the Telemetry Dataloger.
hticonf.txt	User Configuration File. Edit this file to setup Alarms and Displayed values (see below)
Readme.txt	

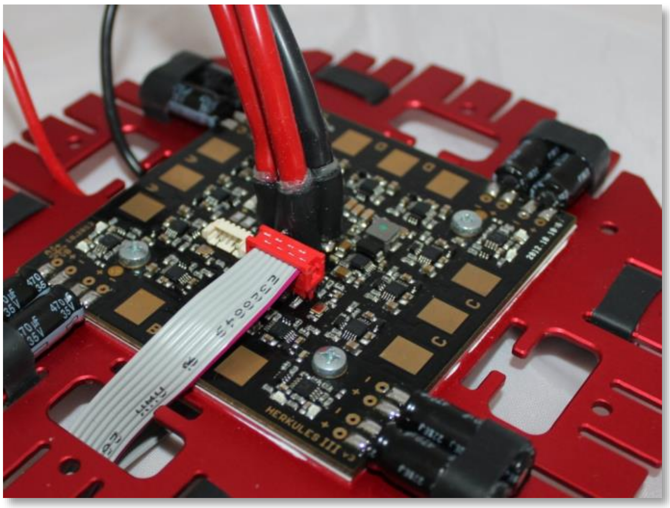
The setup sequence of the HERKULES III Datalogger is different for each telemetry system. Please refer on chapter "RC Telemetry " on page 59 for detailed information.

6 Application Examples

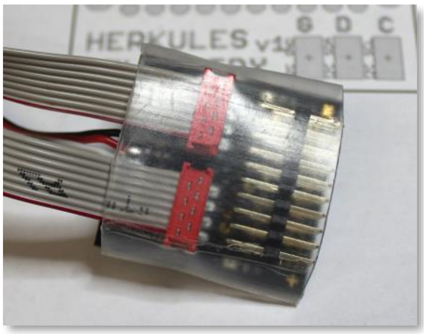
6.1 Setup with DJI Wookong (PPM Control)

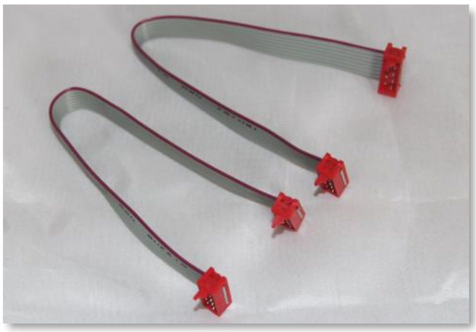
6.1.1 Pictures

Figure 21 - Wiring of Bot-Powerboard with flat ribbon cable



Remember to put the Telemetry Interface into the attached shrink-tube to protect it from short circuits and to secure fix the Molex connectors to the board.





The next pictures show the overview of wiring all parts together.

Figure 22 - Wiring of Top-Powerboard with Telemetry Interface and Receiver

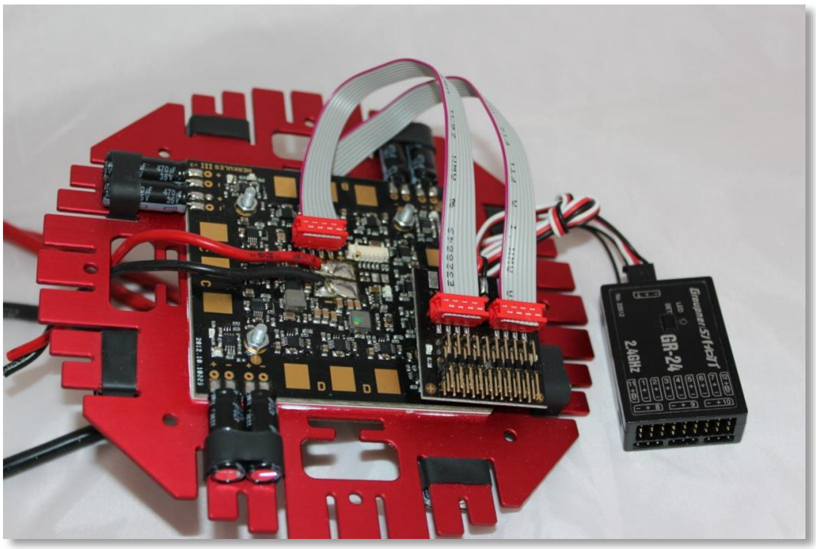
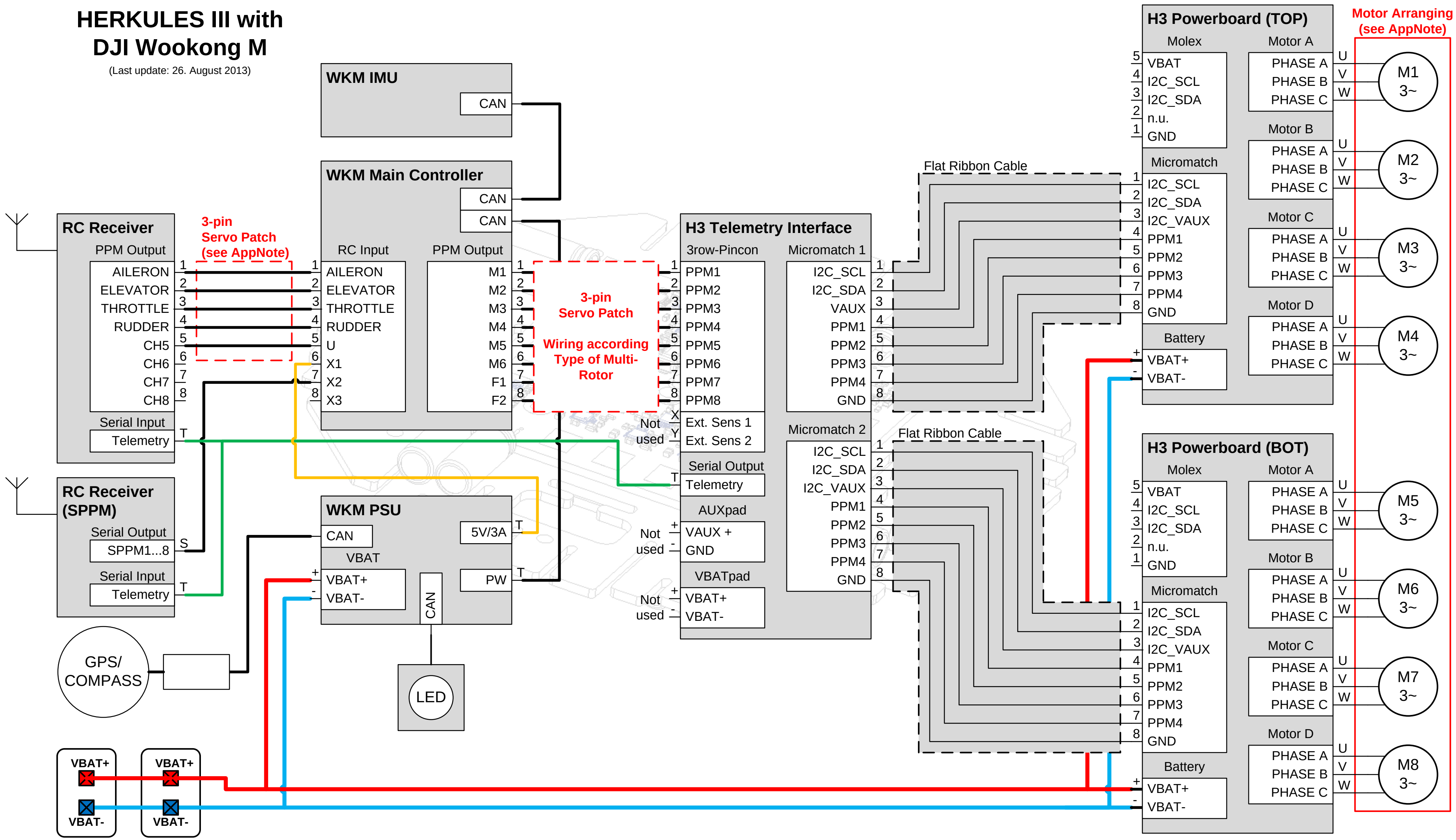


Figure 23 – **Example Connection Scheme : HERKULES III with DJI-Wookong M**



Important Note:

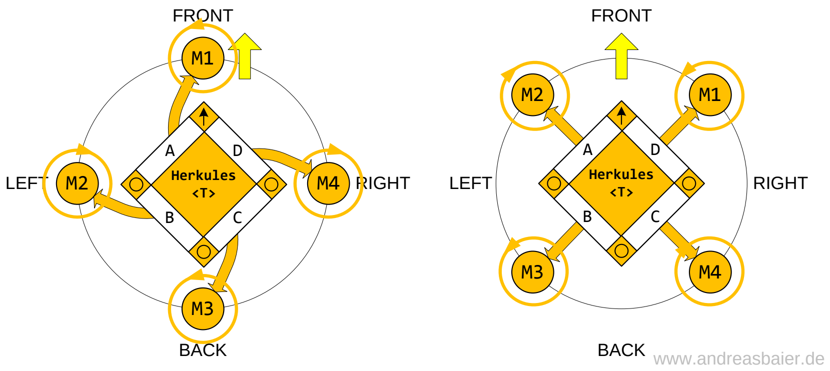
Flight Controls like DJI WKM deliver PPM control signals to ESCs. The motor channel outputs have to be "hardwired". The wiring is usually done with 3-pin servo patch cables.

The software I2C addresses have nothing to do with the flight control and are only for telemetry readout using the Herkules III Telemetry interface.

It is useful to assign the I2C motor addresses in the same order as the real motor numbering. This simplifies the linking of diagnosis data to the real motors..

6.1.2 Quadcopter "+" and "X" with DJI-Wookong

Figure 24 – Mapping Table: Quadcopter with DJI-Wookong



Herkules Quad-ESC TOP +

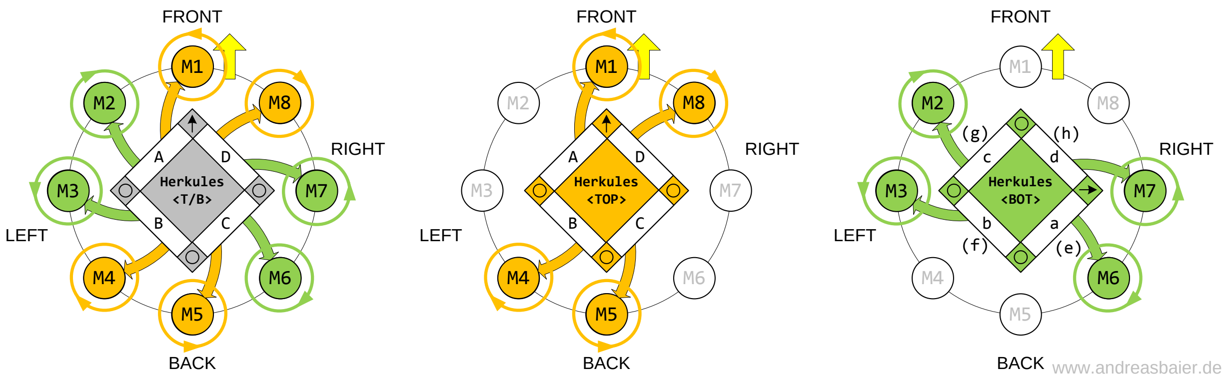
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front	M01
B	BLDCHK…_B	Left	M02
C	BLDCHK…_C	Back	M03
D	BLDCHK…_D	Right	M04

Herkules Quad-ESC TOP X

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front-Left	M02
B	BLDCHK…_B	Back-Left	M03
C	BLDCHK…_C	Back-Right	M04
D	BLDCHK…_D	Front-Right	M01

6.1.3 Octocopter "Flat +" with DJI-Wookong – Variant 1

Figure 25 – Mapping Table: Octocopter Flat + - DJI Wookong - Variant 1



Herkules Quad-ESC TOP

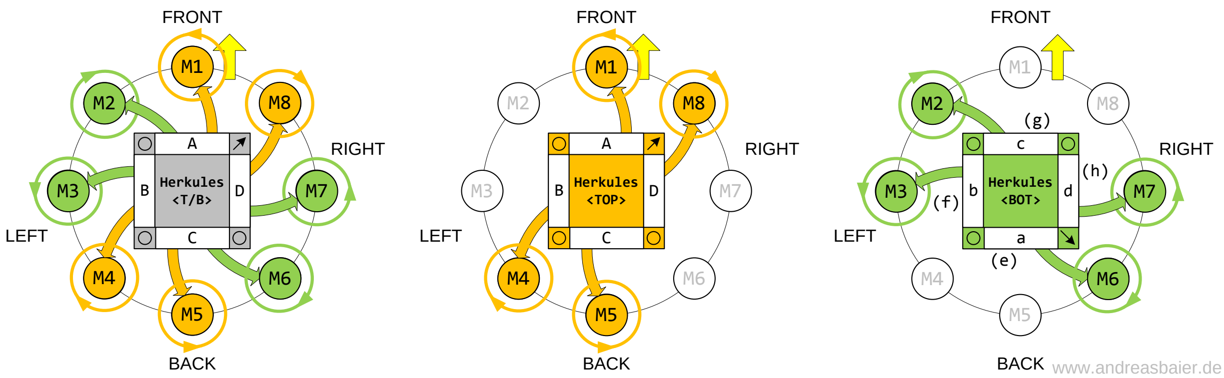
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front	M01
B	BLDCHK..._B	Back-Left	M04
C	BLDCHK..._C	Back	M05
D	BLDCHK..._D	Front-Right	M08

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M06
B (f)	BLDCHK..._F	Left	M03
C (g)	BLDCHK..._G	Front-Left	M02
D (h)	BLDCHK..._H	Right	M07

6.1.4 Octocopter "Flat +" with DJI-Wookong – Variant 2

Figure 26 – Mapping Table: Octocopter Flat + - DJI Wookong - Variant 2



Herkules Quad-ESC TOP

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front	M01
B	BLDCHK..._B	Back-Left	M04
C	BLDCHK..._C	Back	M05
D	BLDCHK..._D	Front-Right	M08

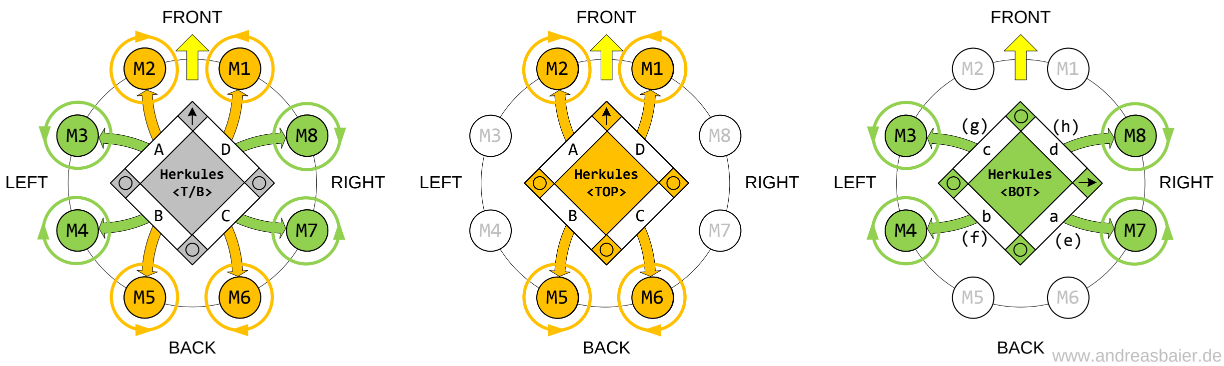
Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M06
B (f)	BLDCHK..._F	Left	M03
C (g)	BLDCHK..._G	Front-Left	M02
D (h)	BLDCHK..._H	Right	M07

6.1.5 Octocopter "Flat-X" with DJI-Wookong – Variant 1

This example shows the address-assignment for a Flat-X Octocopter configuration with DJI Wookong. The motors are distributed in such a way that if in case of a complete failure quad-ESC (top or bot) the multicopter is still controllable and can land in a controlled way. It is important that always two left-turning and two right-turning motors are linked to one HERKULES III quad-ESC.

Figure 27 - Mapping Table: Octocopter - Flat X - DJI-Wookong - Variant 1



Herkules Quad-ESC TOP

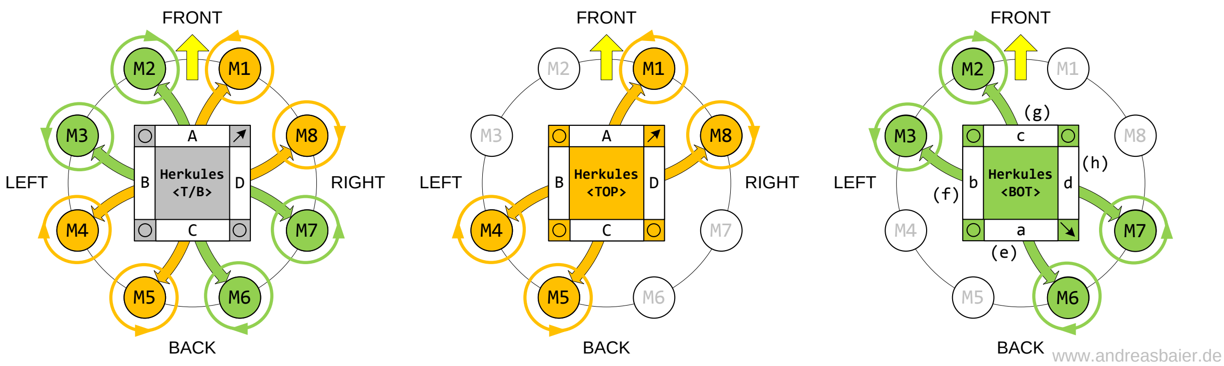
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front	M02
B	BLDCHK…_B	Back	M05
C	BLDCHK…_C	Back	M06
D	BLDCHK…_D	Front	M01

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK…_E	Riight	M07
B (f)	BLDCHK…_F	Left	M04
C (g)	BLDCHK…_G	Left	M03
D (h)	BLDCHK…_H	Riight	M08

6.1.6 Octocopter "Flat-X" with DJI-Wookong – Variant 2

Figure 28 - Mapping Table: Octocopter - Flat X - DJI Wookong - Variant 2



Herkules Quad-ESC TOP

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front	M01
B	BLDCHK…_B	Left	M04
C	BLDCHK…_C	Back	M05
D	BLDCHK…_D	Right	M08

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK…_E	Back	M06
B (f)	BLDCHK…_F	Left	M03
C (g)	BLDCHK…_G	Front	M02
D (h)	BLDCHK…_H	Right	M07

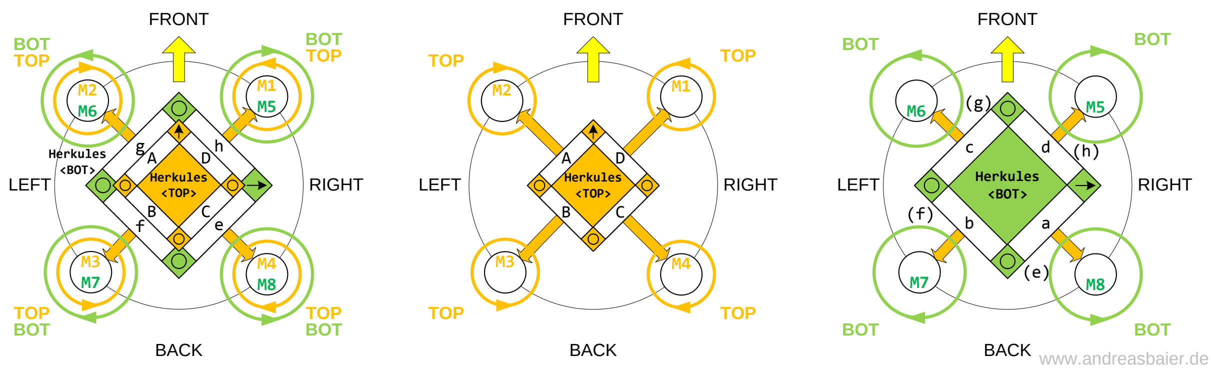
6.1.7 Octocopter “Coaxial-X” with DJI-Wookong

This Coaxial-Octo setup is a customized version using the custom mixer table.

The benefit of this configuration is that 2 separate quadcopters are working in 2 separate levels. In case one complete Herkules III Quad-ESC would fail, the copter would still be controllable as long as the total power is sufficient to lift the complete vehicle – at least a controllable descent should be possible.

Additionally, the Coaxial Octo setup flies with improved stability and the yaw behavior accurate.

Figure 29 - Mapping Table - Octocopter Coaxial-X with DJI Wookong



Herkules Quad-ESC TOP

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front L	M02
B	BLDCHK..._B	Back L	M03
C	BLDCHK..._C	Back R	M04
D	BLDCHK..._D	Front R	M01

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back R	M08
B (f)	BLDCHK..._F	Back L	M07
C (g)	BLDCHK..._G	Front L	M06
D (h)	BLDCHK..._H	Front R	M05

Important Note:

This special setting requires the user to modify the “customized mixer table”. Please refer to the flight controls user manual for more information!

Figure 30 - Mixer Table Coaxial-X8 with DJI-Wookong

Customize

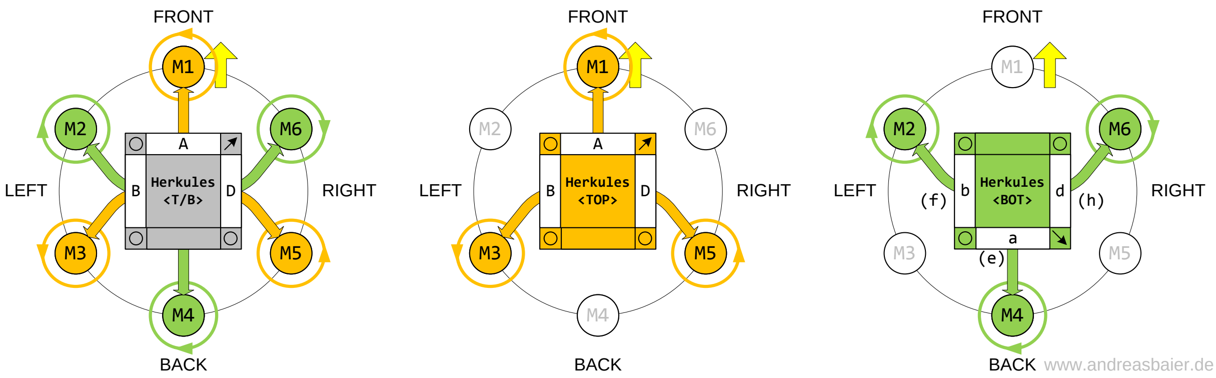
Motor	Throttle	Yaw	Pitch	Roll
M1	100 %	100 %	-100 %	-100 %
M2	100 %	-100 %	-100 %	100 %
M3	100 %	100 %	100 %	100 %
M4	100 %	-100 %	100 %	-100 %
M5	100 %	-100 %	-100 %	-100 %
M6	100 %	100 %	-100 %	100 %
F1/M7	100 %	-100 %	100 %	100 %
F2/M8	100 %	100 %	100 %	-100 %

Note: In order to improve efficiency of the Coaxial-Setup, it is possible to reduce the motor setpoint values of the TOP motors by 5-10%. This increases the overall efficiency and improves flight time.

The exact value depends on the used motors, propellers and average thrust of each motor. So the user must find the optimum value for each system separately

6.1.8 Hexacopter “Flat +” with DJI-Wookong – Variant 1

Figure 31 - Mapping Table: Hexacopter - Flat +" - DJI-Wookong - Variant 1



Herkules Quad-ESC TOP

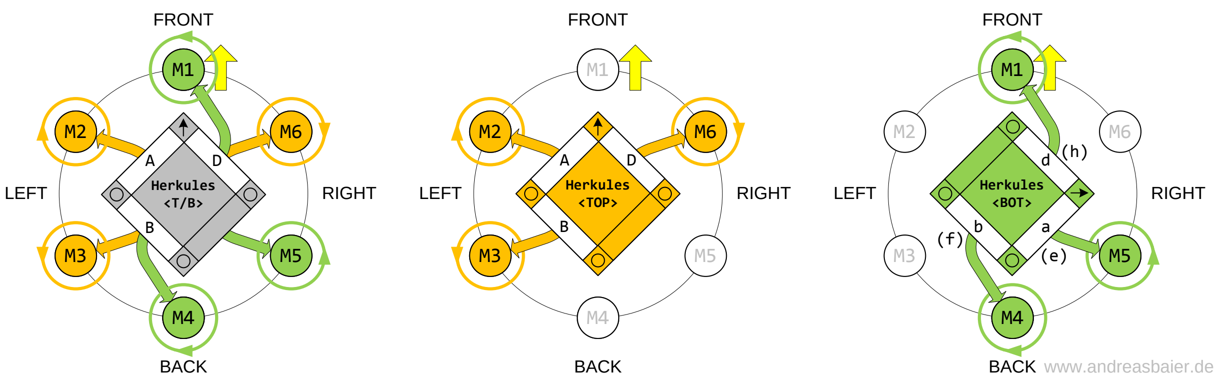
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front	M01
B	BLDCHK..._B	Back-Left	M05
C	BLDCHK..._C	-	-
D	BLDCHK..._D	Back-Right	M03

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back	M04
B (f)	BLDCHK..._F	Front-Left	M04
C (g)	BLDCHK..._G	-	-
D (h)	BLDCHK..._H	Front-Right	M02

6.1.9 Hexacopter “Flat +” with DJI Wookong – Variant 2

Figure 32 - Mapping Table: Hexacopter - Flat +" - DJI-Wookong - Variant 2



Herkules Quad-ESC TOP

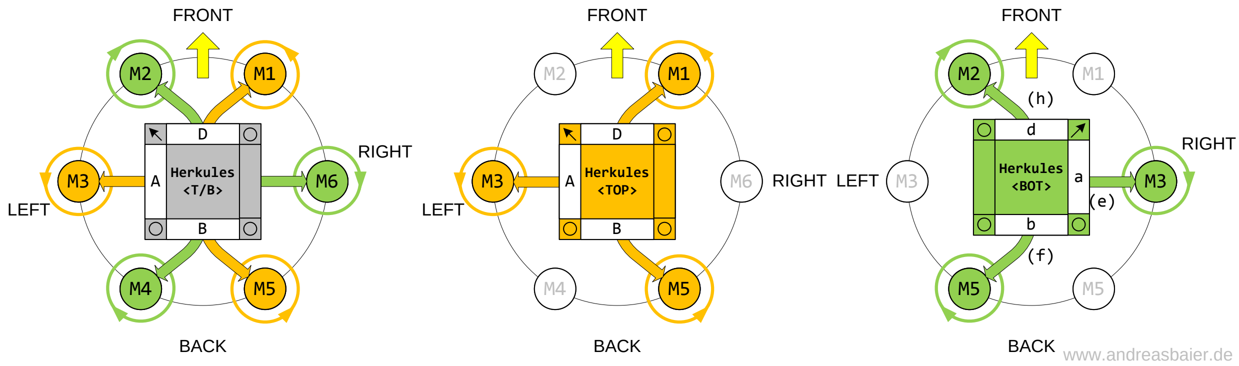
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front-Left	M06
B	BLDCHK..._B	Back-Left	M05
C	BLDCHK..._C	-	-
D	BLDCHK..._D	Front-Right	M02

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M03
B (f)	BLDCHK..._F	Back	M04
C (g)	BLDCHK..._G	-	-
D (h)	BLDCHK..._H	Front	M01

6.1.10 Hexacopter “Flat X” with DJI-Wookong – Variant 1

Figure 33 - Mapping Table: Hexacopter - Flat X" - DJI-Wookong - Variant 1



Herkules Quad-ESC TOP

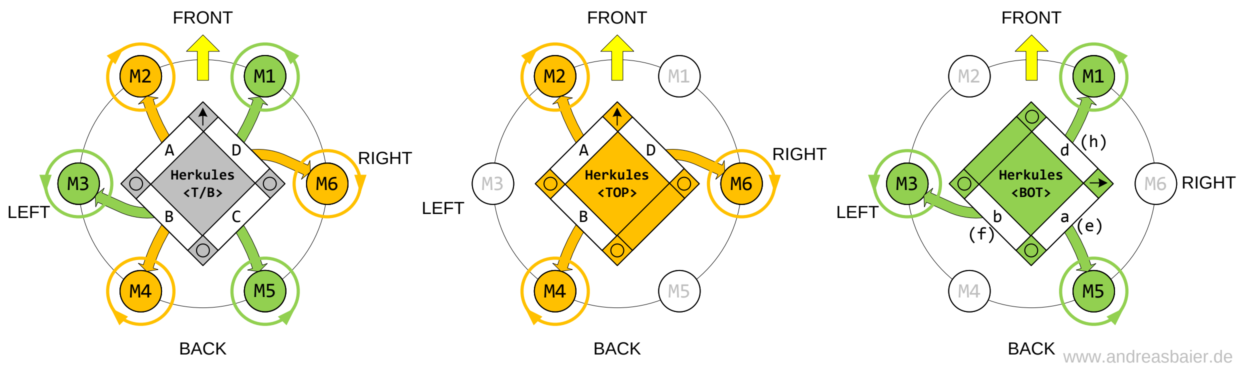
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Left	M03
B	BLDCHK…_B	Back-Right	M05
C	BLDCHK…_C	-	-
D	BLDCHK…_D	Front-Right	M01

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK…_E	Right	M03
B (f)	BLDCHK…_F	Back-Left	M05
C (g)	BLDCHK…_G	-	-
D (h)	BLDCHK…_H	Front-Left	M01

6.1.11 Hexacopter “Flat X” with DJI-Wookong – Variant 2

Figure 34 - Mapping Table: Hexacopter - Flat X" - DJI-Wookong - Variant 2



Herkules Quad-ESC TOP

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front-Left	M02
B	BLDCHK…_B	Back-Left	M04
C	BLDCHK…_C	-	-
D	BLDCHK…_D	Right	M06

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK…_E	Back-Right	M05
B (f)	BLDCHK…_F	Left	M03
C (g)	BLDCHK…_G	-	-
D (h)	BLDCHK…_H	Front-Right	M01

6.2 Setup with Mikrokopter FlightControl ME2.1 (I2C Control)

6.2.1 Pictures

Figure 35 - Mikrokopter with Molex connector

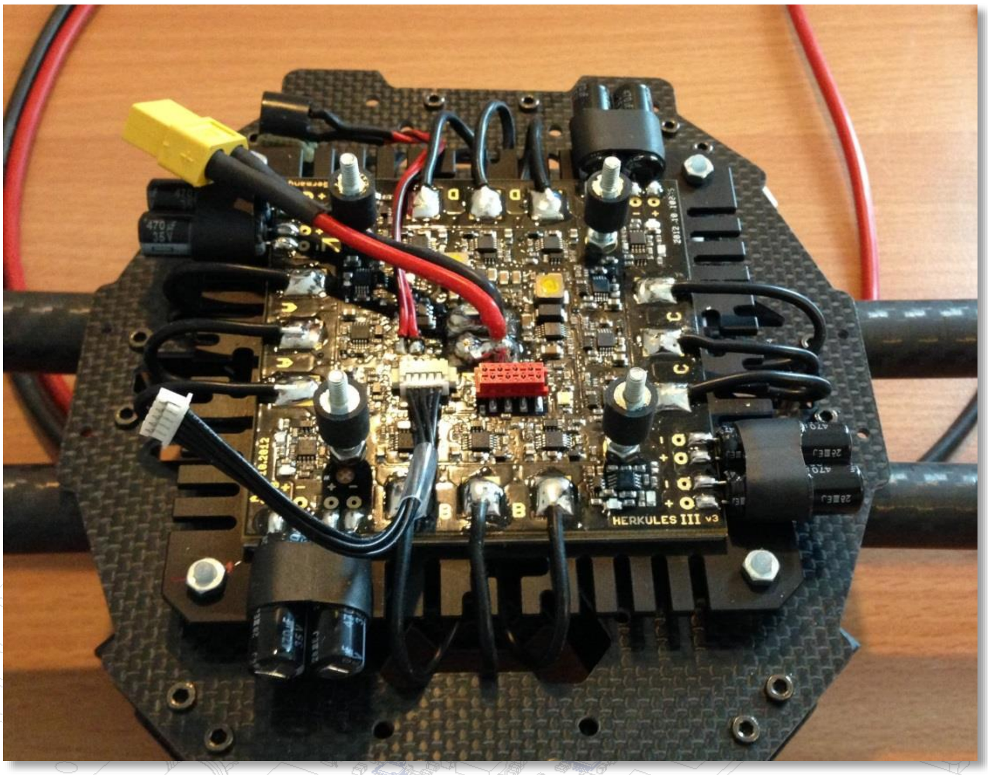


Figure 36 - Typical Setup with Mikrokopter Flight Control

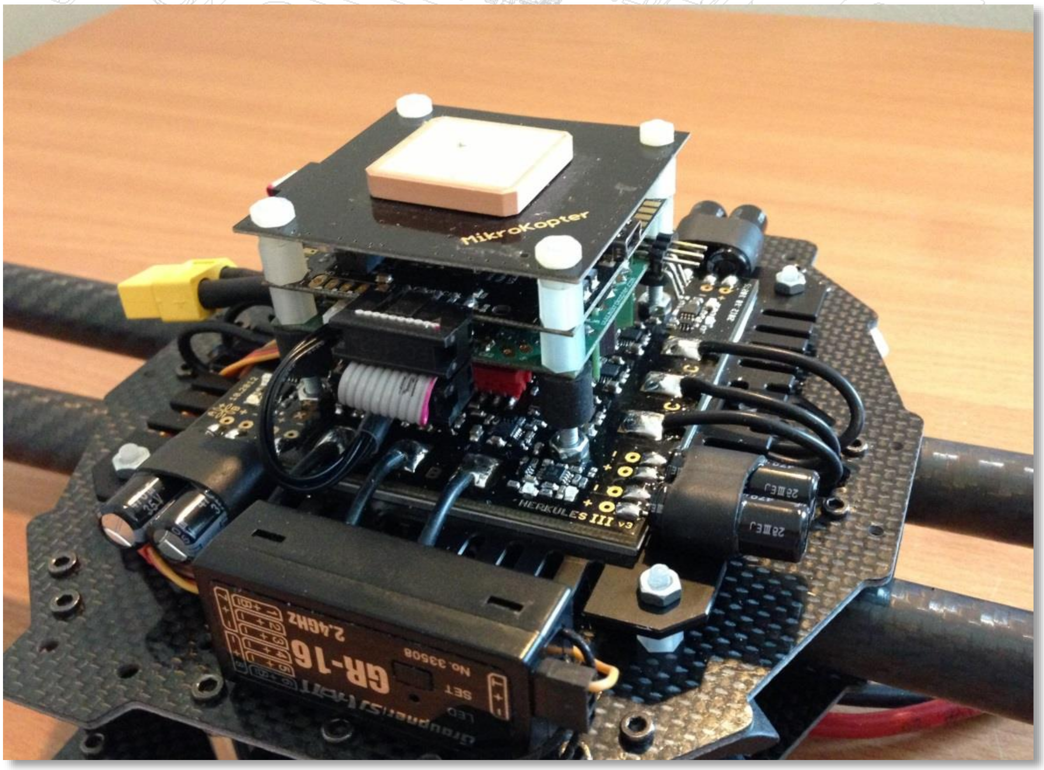
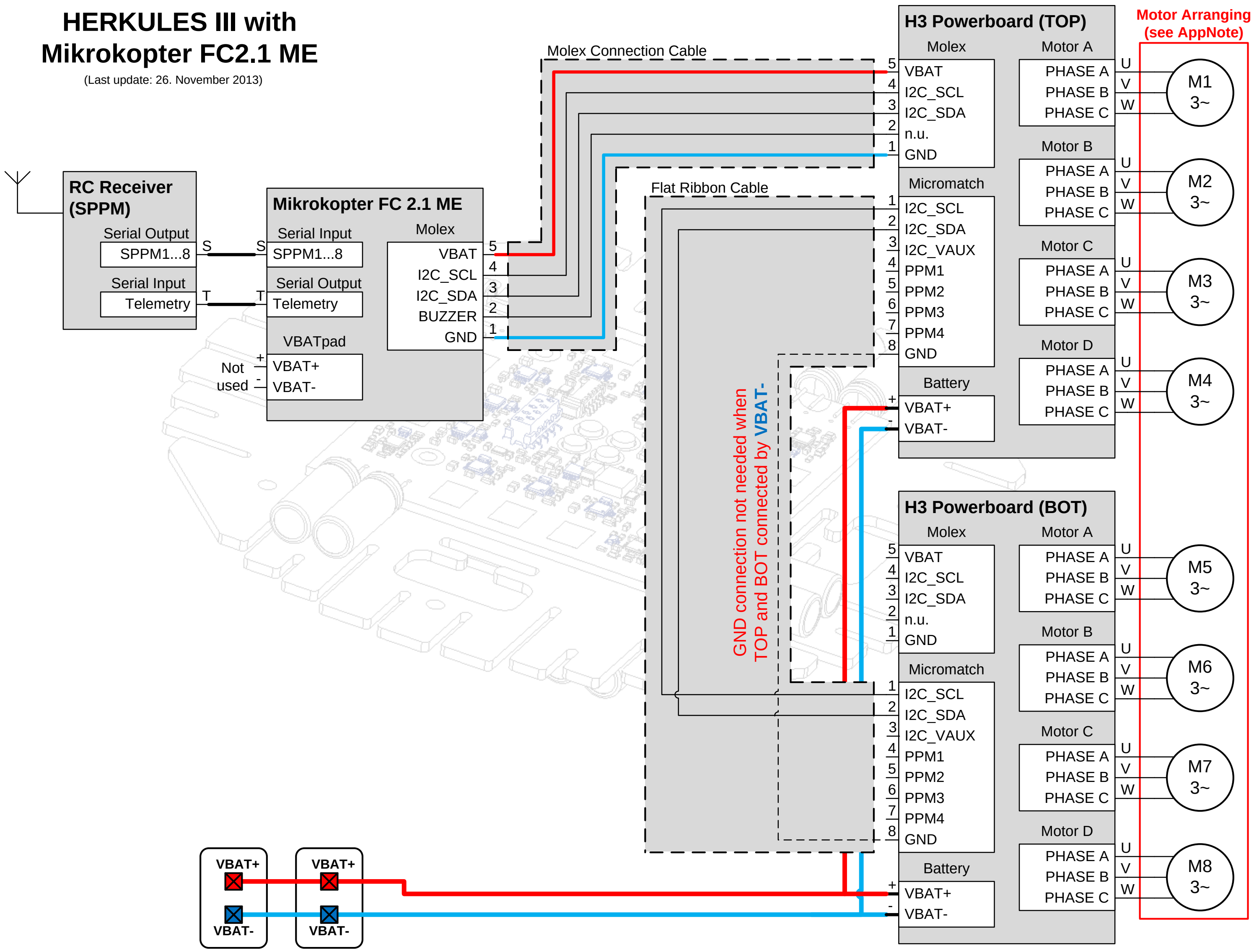
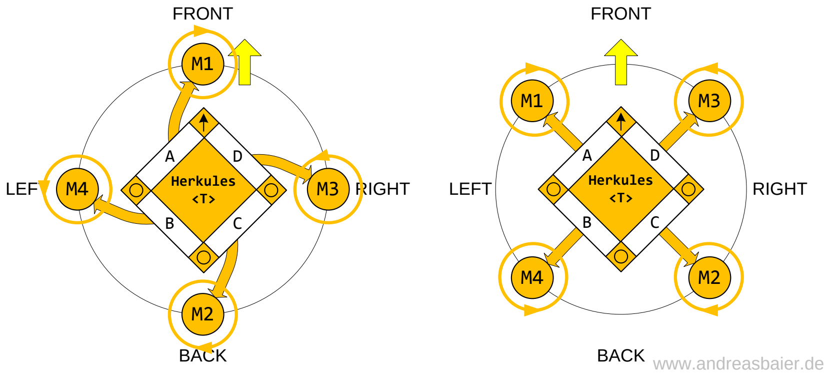


Figure 37 – Example Connection Scheme : HERKULES III with Mikrokopter FC2.1 ME



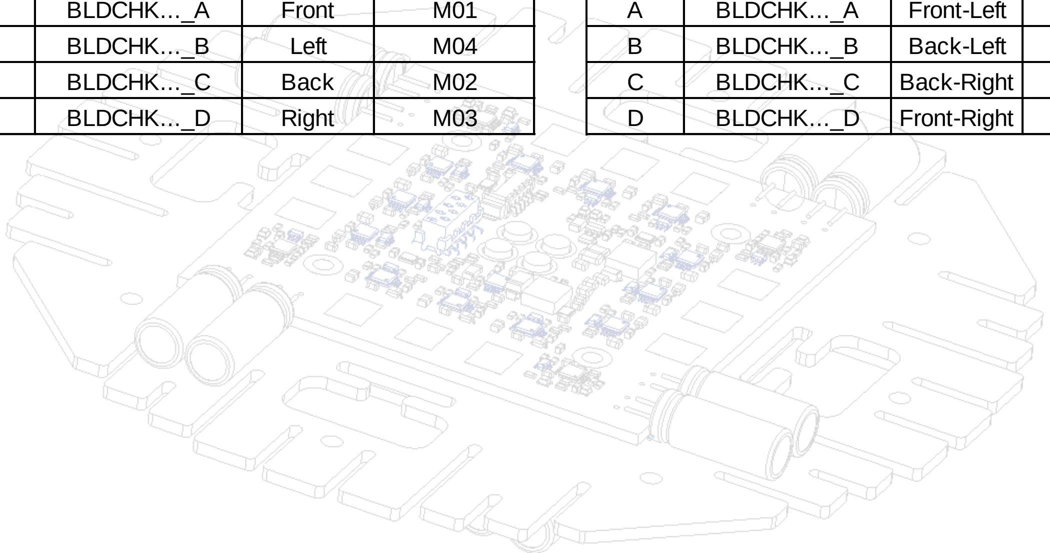
6.2.2 Quadcopter “+” and “X” with Mikrokopter

Figure 38 – Mapping Table: Quadcopter with Mikrokopter



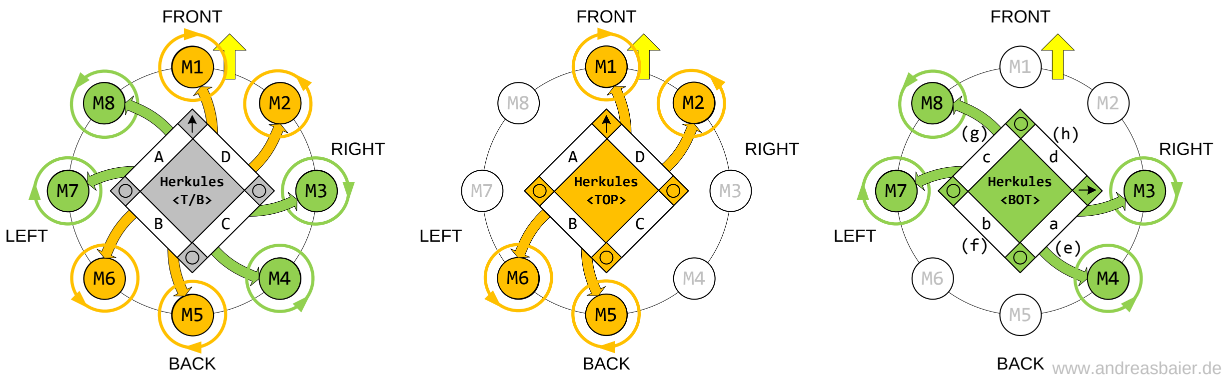
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front	M01
B	BLDCHK…_B	Left	M04
C	BLDCHK…_C	Back	M02
D	BLDCHK…_D	Right	M03

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front-Left	M01
B	BLDCHK…_B	Back-Left	M04
C	BLDCHK…_C	Back-Right	M02
D	BLDCHK…_D	Front-Right	M03



6.2.3 Octocopter “Flat +” with Mikrokopter – Variant 1

Figure 39 - Mapping Table: Octocopter Flat + - Mikrokopter - Variant 1



Herkules Quad-ESC TOP

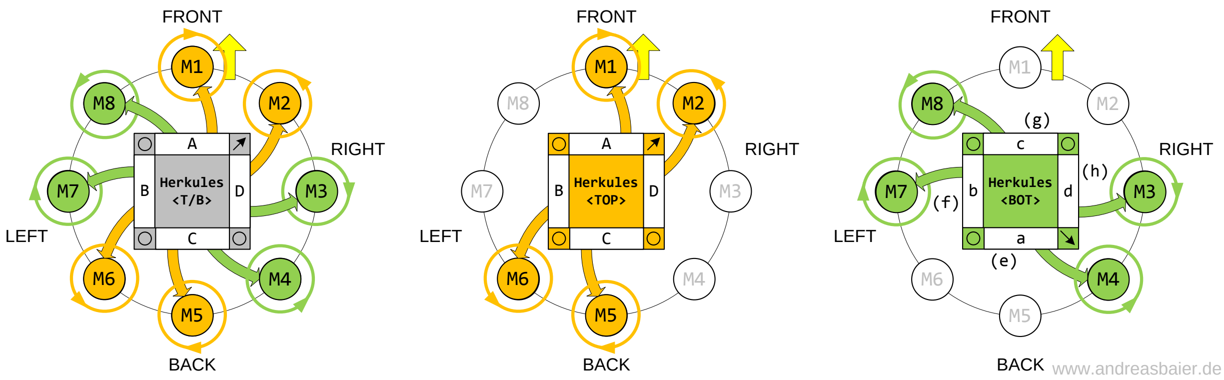
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front	M01
B	BLDCHK..._B	Back-Left	M06
C	BLDCHK..._C	Back	M05
D	BLDCHK..._D	Front-Right	M02

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M04
B (f)	BLDCHK..._F	Left	M07
C (g)	BLDCHK..._G	Front-Left	M08
D (h)	BLDCHK..._H	Right	M03

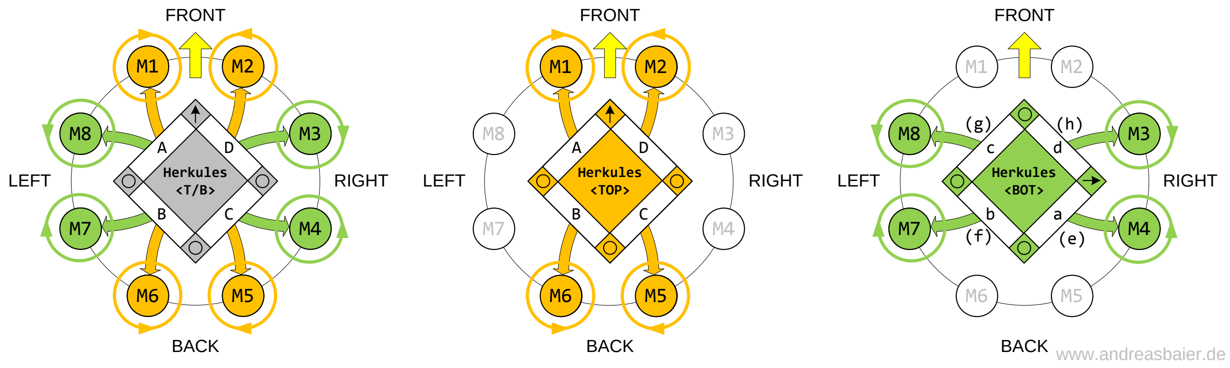
6.2.4 Octocopter “Flat +” with Mikrokopter – Variant 2

Figure 40 - Mapping Table: Octocopter Flat + - Mikrokopter - Variant 2



6.2.5 Octocopter “Flat X” with Mikrokopter – Variant 1

Figure 41 – Mapping Table: Octocopter Flat X - Mikrokopter -Variant 1



Herkules Quad-ESC TOP

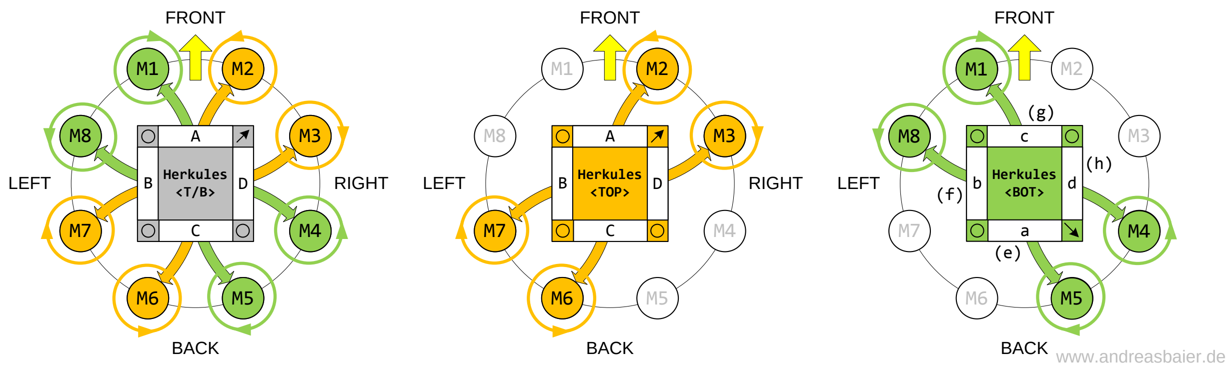
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front	M01
B	BLDCHK…_B	Back	M06
C	BLDCHK…_C	Back	M05
D	BLDCHK…_D	Front	M02

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK…_E	Right	M04
B (f)	BLDCHK…_F	Left	M07
C (g)	BLDCHK…_G	Left	M08
D (h)	BLDCHK…_H	Right	M03

6.2.6 Octocopter “Flat X” with Mikrokopter – Variant 2

Figure 42 – Mapping Table: Octocopter Flat X - Mikrokopter -Variant 2



Herkules Quad-ESC TOP

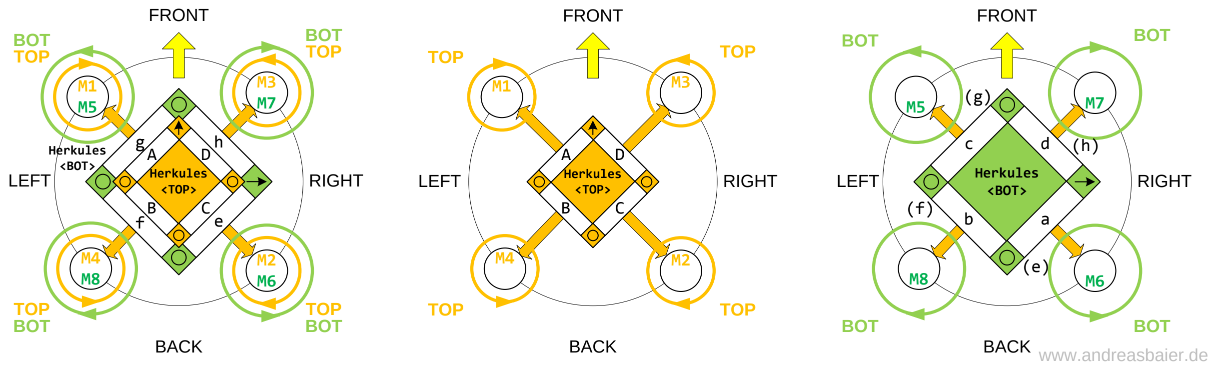
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK…_A	Front	M02
B	BLDCHK…_B	Left	M07
C	BLDCHK…_C	Back	M06
D	BLDCHK…_D	Right	M03

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK…_E	Back	M05
B (f)	BLDCHK…_F	Left	M08
C (g)	BLDCHK…_G	Front	M01
D (h)	BLDCHK…_H	Right	M04

6.2.7 Octocopter "Coaxial-X" with Mikrokopter

Figure 43 - Mapping Table: Octocopter Coaxial 8 - Mikrokopter - Customized



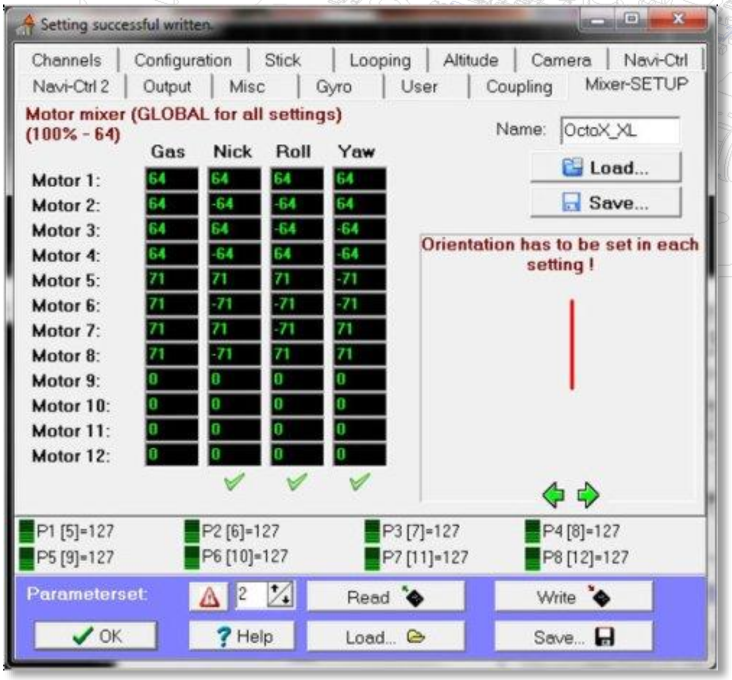
Herkules Quad-ESC TOP

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front-Left	M01
B	BLDCHK..._B	Back-Left	M04
C	BLDCHK..._C	Back-Right	M02
D	BLDCHK..._D	Front-Right	M03

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M06
B (f)	BLDCHK..._F	Back-Left	M08
C (g)	BLDCHK..._G	Front-Left	M05
D (h)	BLDCHK..._H	Front-Right	M07

Figure 44 - Mixer Table Coaxial-X8 with Mikrokopter



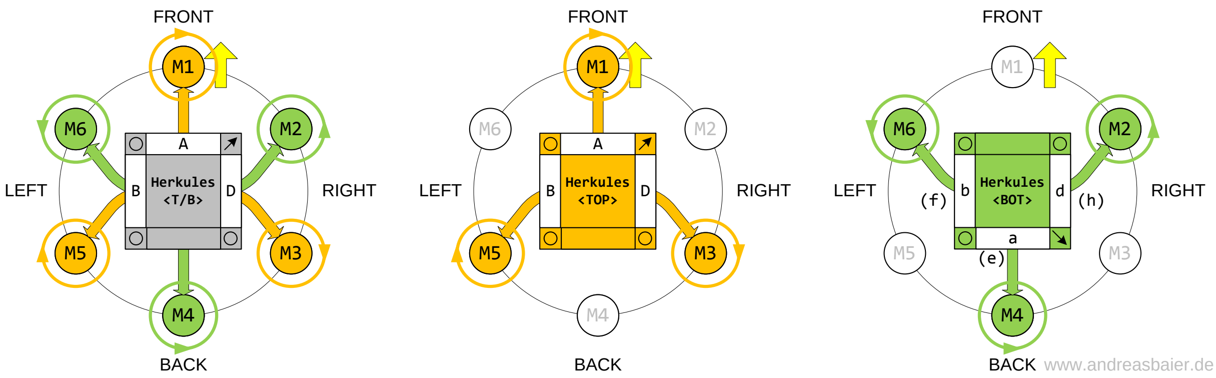
Note: In order to improve efficiency of the Coaxial-Setup, it is possible to reduce the motor setpoint values of the TOP Motors by 5-10%. This increases the overall efficiency and improves flight time.

The exact value depends on the used motors, propellers and average thrust of each motor. So the user must find the optimum value for each system separately

Please refer to the Flight Control Manual for more information according mixer table setup.

6.2.8 Hexacopter “Flat +” with Mikrokopter – Variant 1

Figure 45 - Mapping Table: Hexacopter "Flat +" - Mikrokopter - Variant 1



Herkules Quad-ESC TOP

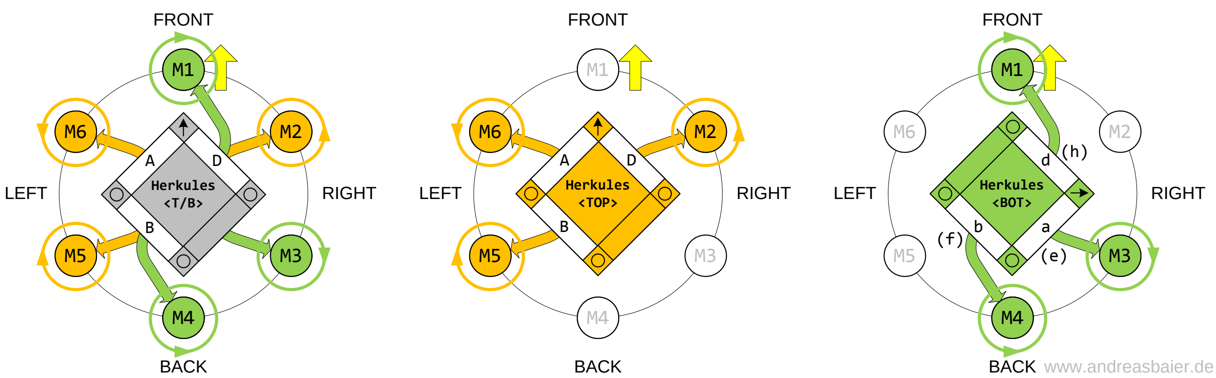
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front	M01
B	BLDCHK..._B	Back-Left	M05
C	BLDCHK..._C	-	-
D	BLDCHK..._D	Back-Right	M03

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back	M04
B (f)	BLDCHK..._F	Front-Left	M06
C (g)	BLDCHK..._G	-	-
D (h)	BLDCHK..._H	Front-Right	M02

6.2.9 Hexacopter “Flat +” with Mikrokopter – Variant 2

Figure 46 - Mapping Table: Hexacopter "Flat +" - Mikrokopter - Variant 2



Herkules Quad-ESC TOP

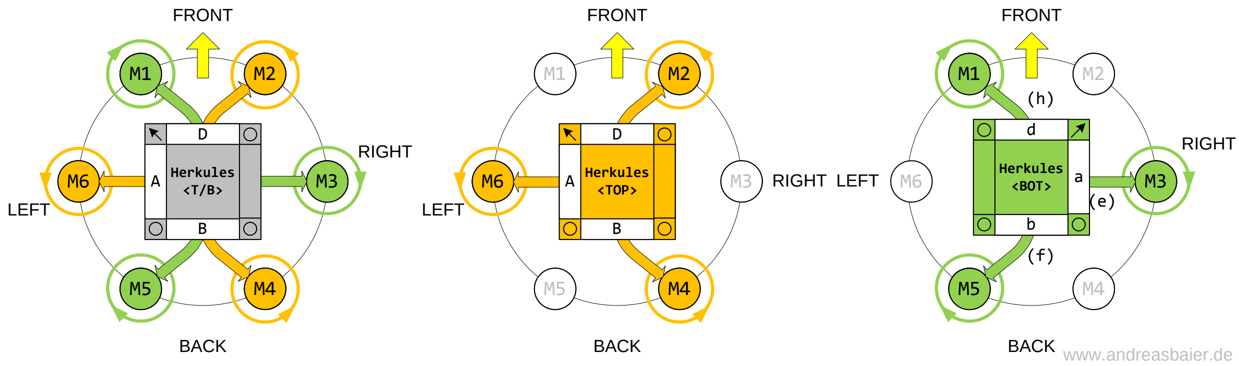
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front-Left	M06
B	BLDCHK..._B	Back-Left	M05
C	BLDCHK..._C	-	-
D	BLDCHK..._D	Front-Right	M02

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M03
B (f)	BLDCHK..._F	Back	M04
C (g)	BLDCHK..._G	-	-
D (h)	BLDCHK..._H	Front	M01

6.2.10 Hexacopter “Flat X” with Mikrokopter – Variant 1

Figure 47 - Mapping Table: Hexacopter "Flat X" - Mikrokopter - Variant 1



Herkules Quad-ESC TOP

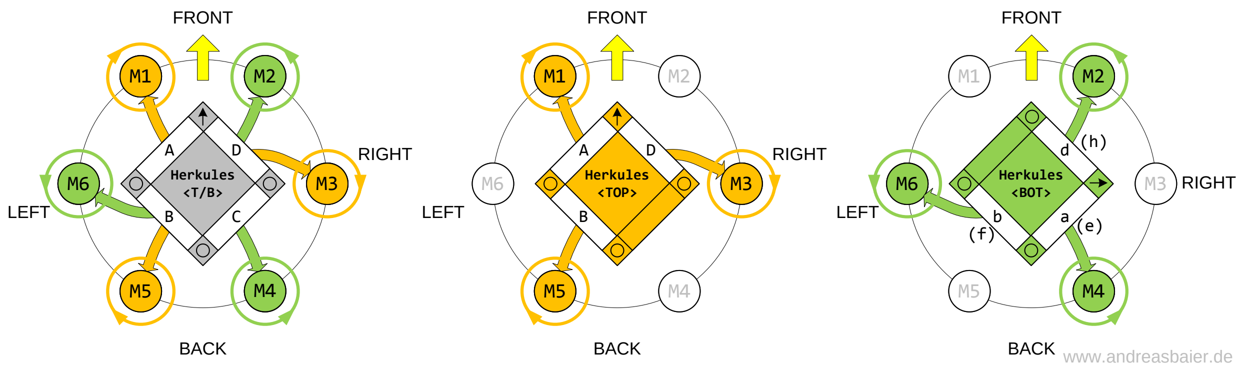
ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Left	M06
B	BLDCHK..._B	Back-Right	M04
C	BLDCHK..._C	-	-
D	BLDCHK..._D	Front-Right	M02

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Right	M03
B (f)	BLDCHK..._F	Back-Left	M05
C (g)	BLDCHK..._G	-	-
D (h)	BLDCHK..._H	Front-Left	M01

6.2.11 Hexacopter “Flat X” with Mikrokopter – Variant 2

Figure 48 - Mapping Table: Hexacopter "Flat X" - Mikrokopter - Variant 2



Herkules Quad-ESC TOP

ESC TOP	Bootsign	Motor	I2C-Address
A	BLDCHK..._A	Front-Left	M01
B	BLDCHK..._B	Back-Left	M05
C	BLDCHK..._C	-	-
D	BLDCHK..._D	Right	M03

Herkules Quad-ESC BOT (flipped)

ESC BOT	Bootsign	Motor	I2C-Address
A (e)	BLDCHK..._E	Back-Right	M04
B (f)	BLDCHK..._F	Left	M06
C (g)	BLDCHK..._G	-	-
D (h)	BLDCHK..._H	Front-Right	M02

7 RC Telemetry Systems

The Herkules III Telemetry Interface Board reads the diagnosis data of all connected Herkules III ESCs, calculates metric values and generates a specific Telemetry Protocol. The Telemetry Interface works as a stand-alone telemetry sensor which can transmit independently of the flight control all relevant telemetry data to the ground station. This enables telemetry from copters controlled by all “PPM-Flight Controls” like DJI Wookong, Multi-Wii or Ardupilot, assuming there is a radio downlink connected to the Herkules Telemetry Interface Board.

Important Note:

This chapter is only relevant for PPM controls like DJI Wookong / Naza used in combination with the external Herkules III Telemetry Interface!

It is NOT valid for Mikrokopter Flight Controls due to the fact that Mikrokopter reads out directly the Herkules III ESCs and generates the Telemetry messages on its own.

To bring out the telemetry data from the HERKULES **III** ESCs a special firmware, the “PPMT” or “I2CT” variant has to be flashed to the controllers.

Additionally, either the small telemetry interface-PCB or the HERKULES III Top cover plate is needed. These contain a small microcontroller. This controller reads data out periodically from the HERKULES III powerboards, evaluates the data and translates this to a telemetry protocol. The telemetry interface can be directly connected to a telemetry-capable receiver and emulates a 3rd party sensor.

When using using Mikrokopter as the flight control, no additional hardware is necessary. The telemetry data is directly read by the flight control an evaluated. The transmission to the ground station can be done as usual with a connected receiver at the flight control.

HERKULES III measures the following telemetry data:

- **ESC single temperature :** The temperature of each single ESC (max 16)
- **ESC single current :** The average current of each single ESC
- **Total Current :** summed-up current of all ESCs (= battery current)
- **Peak Current :** Highest battery current during flight
- **Peak Temperature :** Highest temperature on all HERKULES III ESCs
- **ESC with highest Temp :** Number of single ESC with highest Temp
- **Average Current :** Total battery average current
- **Cumulated Charge :** Battery consumption in mAh
- **Battery Voltage :** Total battery voltage
- **Motor Runtime :** Total runtime since motor start
- **Altitude :** Actual height above starting point

7.1 Graupner HoTT Telemetry

If activated in the transmitter menu, HERKULES III sends the display for EAM (Electrical Air Module).

The general setup and software update procedure is described in Chapter 5.7 "Update Procedure TELEMETRY **" on page 37.**

7.1.1 Setup Sequence

In order to install the target Telemetry System on the Datalogger, the correct files have to be copied from the "Firmwares"-folder to the root folder of the MicroSD-Card.

Important Note:

Please check if your Computer's operation system displays by standard filenames starting with a "." (dot) like .setup.hti. (Remember the dot in front of the name!

By default this might be hidden on Windows PC or MAC. Please referrer to your operating systems manual to "unhide" file names starting with dot.

1) Delete from the root direction of your MicroSD-Card the files

```
.setup.hti
HTI25_xxx.bin
 hticonf.txt
```

2) Copy from the subfolder "**Firmwares**" from any of the target Telemetry Systems subfolders (e.g. HOTT_vxx) the 3 files:

to the Root-folder the Micro-SD-card. (**NOTE**: copy the **Files** from the folder only, **NOT** the Folder **itself**!)

The root must contain then 3 files:

File Name	Description
.setup.hti	Dont change or delete this file!
HTI25_JETI_xxx.bin	New Firmware File to be updated
hticonf.txt	User-Configuration File. See description below

4) Edit User-Configuration File "hticonf.txt"

Open the hticonf.txt on the SD-card and change the alarm values and correction factors as described below.

Content of "hticonf.txt"

```
  800,   355,   335,   418,   216,   12,    0,    31,    0
  |      |      |      |      |      |      |      |      |JETI Duplex Mode
  |      |      |      |      |      |      |      |Selection of HoTT Sensors
  |      |      |      |      |      |      |0% IBAT correction factor
  |      |      |      |      |      |1,2A IBAT constant offset value
  |      |      |      |      |21,6V Battery Alarm threshold
  |      |      |      |4,18V = 100% Single Cell Voltage threshold
  |      |      |3,35V = 0% Single Cell Voltage threshold
  |      |3,55V Single Cell Alaram threshold
  |8000mAh Capacity Alarm threshold
```

Byte	Value		Example	Unit		Description	System
1	800	=	8000	mAh	Capacity	mAh Alarm Limit x 10mAh	All
2	355	=	3,55	V	LCV	Single Cell Voltage Alarm Limit x 10mV	All
3	335	=	3,35	V	CV 0%	0% Single Cell Voltage Limit x 10mV	HoTT Text
4	418	=	4,18	V	CV 100%	100% Single Cell Voltage Limit x 10mV	HoTT Text
5	216	=	21,6	V	VBAT	VBAT Alarm Limit / 10V	HiTec only
6	12	=	1,2	A	OFFSET	IBAT constant offset value for not measured load	All
7	0	=	0	%	GAIN	IBAT correction value in % (+/-) Values possible	All
8	31	=	1	ESC	SENSOR	Enable Graupner Hott Displays	HoTT only
		=	1	GAM	SENSOR	GPS	
		=	1	EAM	SENSOR	VARIO	
		=	1	VARIO	SENSOR	EAM	
		=	1	GPS	SENSOR	GAM	
9	0	=	0	0 = default	EXP	Jeti Timeout Config 0 = three Binary Messages (default) (15 Sensors active) => no Expander 1 = 2 Binary Messages (12 Sensors aktive) => with Expander	JETI EX only

Description:

Byte 1: 800 = Capacity Alarm threshold x 10 in mAh.

If capacity is exceeded, alarm is send from Telemetry System to Ground Station

=> 1000 means 10.000mAh

=> 220 means 2.200mAh

=> 500 means 5.000mAh

Byte 2: 355 = Lipo lowest cell voltage alarm threshold / 100 in V

If lowest cell voltage is lower than this, alarm is send from Telemetry System to GND

=> 355 means 3.55V

=> 345 means 3.45V

Byte 3: 335 = Lipo 0% Value for Text display fuel gauge

Value is only relevant for single cell sensor in Graupner Text Mode

=> 335 means 3.35V

Byte 4: 418 = Lipo 100% Value for Text display fuel gauge

Value is only relevant for single cell sensor in Graupner Text Mode

=> 418 means 4.18V

Byte 5: = not relevant for Graupner Telemetry

Byte 6: 12 = Offset correction value of total current

This value is added to the total battery current and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 12 means 1.2A

Byte 7: 0 = Gain correction value of total current

Total battery current is multiplied by this value [Ibat = (1+ Gain) x Ibat_power] and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 4 means 4%

Byte 8: 31 = Selection of Graupner HoTT Binary Display data (only relevant for Graupner HoTT!)

Select with this number how many Graupner Sensors you want to display.

Note: Default Value is 31, so all sensors will be send.

Value 3)	Enabled Sensor					Description
0	-	-	-	-	-	no sensor enabled
1	-	-	-	-	ESC	ESC only enabled
2	-	-	-	GAM	-	GAM only enabled
3	-	-	-	GAM	ESC	GAM and ESC enabled
4	-	-	EAM	-	-	EAM only enabled
5	-	-	EAM	-	ESC	EAM and ESC enabled
6	-	-	EAM	GAM	-	EAM and GAM enabled
7	-	-	EAM	GAM	ESC	EAM , GAM and ESC enabled
8	-	VARIO	-	-	-	VARIO only enabled
9	-	VARIO	-	-	ESC	VARIO and ESC enabled
10	-	VARIO	-	GAM	-	VARIO and GAM enabled
11	-	VARIO	-	GAM	ESC	VARIO , GAM and ESC enabled
12	-	VARIO	EAM	-	-	VARIO and EAM enabled
13	-	VARIO	EAM	-	ESC	VARIO , EAM and ESC enabled
14	-	VARIO	EAM	GAM	-	VARIO , EAM and GAM enabled
15	-	VARIO	EAM	GAM	ESC	VARIO , EAM , GAM and ESC enabled
16	GPS	-	-	-	-	GPS only enabled
17	GPS	-	-	-	ESC	GPS and ESC enabled
18	GPS	-	-	GAM	-	GPS and GAM enabled
19	GPS	-	-	GAM	ESC	GPS , GAM and ESC enabled
20	GPS	-	EAM	-	-	GPS and EAM enabled
21	GPS	-	EAM	-	ESC	GPS , EAM and ESC enabled
22	GPS	-	EAM	GAM	-	GPS , EAM and GAM enabled
23	GPS	-	EAM	GAM	ESC	GPS , EAM , GAM and ESC enabled
24	GPS	VARIO	-	-	-	GPS and VARIO enabled
25	GPS	VARIO	-	-	ESC	GPS , VARIO and ESC enabled
26	GPS	VARIO	-	GAM	-	GPS , VARIO and GAM enabled
27	GPS	VARIO	-	GAM	ESC	GPS , VARIO , GAM and ESC enabled
28	GPS	VARIO	EAM	-	-	GPS , VARIO and EAM enabled
29	GPS	VARIO	EAM	-	ESC	GPS , VARIO , EAM and ESC enabled
30	GPS	VARIO	EAM	GAM	-	GPS , VARIO , EAM and GAM enabled
31	GPS	VARIO	EAM	GAM	ESC	All Sensors Enabled (default)

Byte 9: 1 = not relevant for Graupner Telemetry

4) Put the SD-card back into the HERKULES III Datalogger

Power-on the HERKULES III and the firmware upgrade starts automatically.

During the update process the RED LED on the Datalogger blinks for about 15sec.

In case the already installed firmware is the same on the SD-Card, no update is done and the Datalogger starts working immediately.

7.1.2 Graupner HoTT Binary Displays

Figure 49 - Electrical Air Module (EAM)

Main Menu GAM

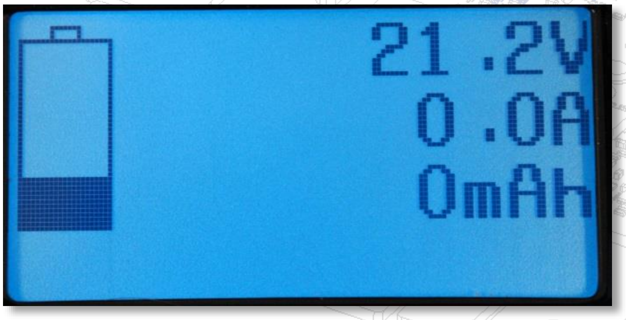
Battery Voltage [V]

Runtime [m:sec]

Capacity [mAh]

Altitude [m]

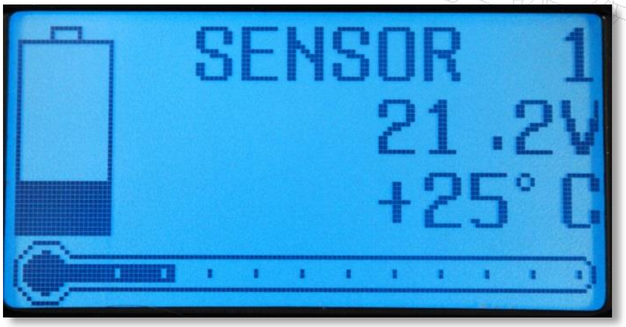
Current [A]



Battery Voltage [V]

Current [A]

Capacity [mAh]

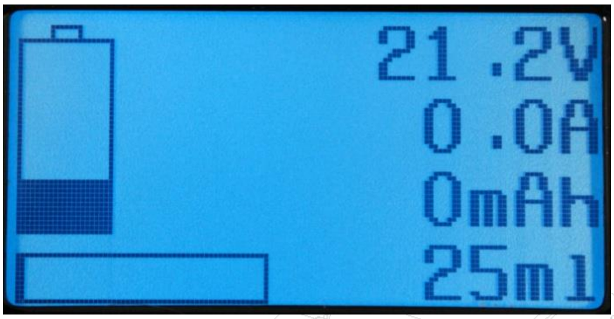


Battery Voltage [V]

Highest Temperature [°C]

Figure 50 - General Air Module (GAM)

Main Menu GAM



Battery Voltage [V]

Current [A]

Capacity [mAh]

Highest Temperatur [°C]

Battery Voltage [V]

Current [A]

Highest Temperature [°C]

Altitude [m]

Steigrate [m/sec][

7.1.3 Graupner HoTT Text Mode Display

Even though the Text Mode Display is quite tricky to access, the detailed information which can be displayed is very useful.

Graupner Text Mode: Overview Parameters

```
VBAT: 114.5V  |  8123W
CURR: 178.4A  |212Apk
CAPA: 14502mAh|129.2A
TEMP: 134°C|M1|  ALT:
TIME:154'54min|254.3m
1RUN 2RUN 3RUN 4RUN
5RUN 6RUN 7--- 8STP
9RUN10ERR11OT!12TO!
```

Line 1: VBAT, Power

Line 2: Current, Peak-Current

Line 3: mAh, Average Current (mAh / Runtime)

Line 4: Hottest Temp, Hottest ESC

Line 5: Runtime, Altitude

Line 6: Motor Status 1…4

Line 7: Motor Status 5…8

Line 8: Motor Status 9…12 only if Motors 9…12 are connected

Graupner Tex t Mode: Single ESC Value

Line 1..8: Average ESC Current, ESC Temperature, ESC Status

```
M1avg:  17.6A  29°C  OK
M2avg:  15.6A  26°C  RUN
M3avg:  18.6A  30°C  RUN
M4avg:  18.6A  28°C  ERR
M5avg:  18.6A  28°C  ---
M6avg:  14.3A  31°C  RUN
M7avg:  --.-A  --°C  ---
M8avg:  --.-A  --°C  ---
```

Graupner Text Mode: Single-Cell Voltages (Battery 1..4):

B11:	4.05V	178mR	78%
B12:	3.98V	78mR	99%
B13:	3.75V	45mR	99%
B14:	3.65V	123mR	99%
B15:	4.05V	178mR	99%
B16:	4.05V	178mR	99%
B17:	4.05V	178mR	99%
B18:	4.05V	178mR	99%

B21:	4.05V	178mR	78%
B22:	3.98V	78mR	99%
B23:	3.75V	45mR	99%
B24:	3.65V	123mR	99%
B25:	4.05V	178mR	99%
B26:	4.05V	178mR	99%
B27:	4.05V	178mR	99%
B28:	4.05V	178mR	99%

B31:	4.05V	178mR	78%
B32:	3.98V	78mR	99%
B33:	3.75V	45mR	99%
B34:	3.65V	123mR	99%
B35:	4.05V	178mR	99%
B36:	4.05V	178mR	99%
B37:	4.05V	178mR	99%
B38:	4.05V	178mR	99%

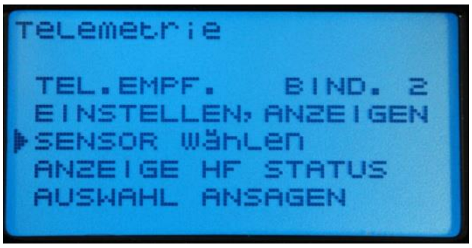
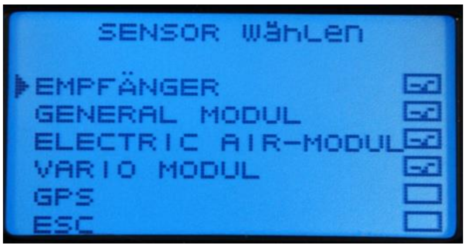
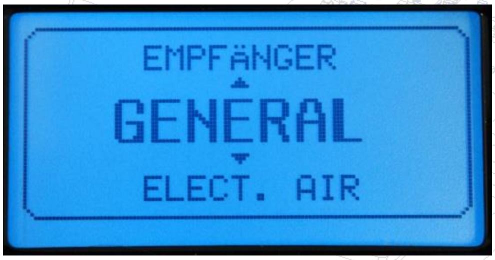
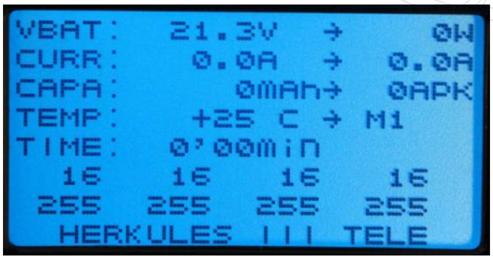
B41:	4.05V	178mR	78%
B42:	3.98V	78mR	99%
B43:	3.75V	45mR	99%
B44:	3.65V	123mR	99%
B45:	4.05V	178mR	99%
B46:	4.05V	178mR	99%
B47:	4.05V	178mR	99%
B48:	4.05V	178mR	99%

Graupner Text Mode: Single-Cell of BAT1 and BAT2 in one Screen (voltage, %)

B11:	405	96	B21:	376	75
B12:	378	86	B22:	376	75
B13:	382	89	B23:	376	75
B14:	365	96	B24:	376	75
B15:	405	96	B25:	376	75
B16:	405	96	B26:	376	75
B17:	405	96	B27:	376	75
B18:	405	96	B28:	376	75

Accessing the Text Mode Display

Due to the cumbersome HoTT-Software implementation, accessing of the Text-Display is a bit tricky.

<table>
<tr><td>
</td><td>Step 1
From the main menu, press for at least 3 seconds the “ESC” - key</td></tr>
<tr><td>
</td><td>Step 2:
Select your target sensors
(Since the implementation of AUTOSCAN this feature is not usable any more)
And escape with “ESC” this menu again</td></tr>
<tr><td>
</td><td>Step 3:
Then move back to “DATAVIEW” menu</td></tr>
<tr><td>
</td><td>Step 4:
Press on the Left Cursor the “UP-KEY” and select “GENERAL”</td></tr>
<tr><td>
</td><td>Step 5:
Press the Left Cursor a few times to reach “RIGHT-KEY” until you see the HERKULES III Textmode Display.</td></tr>
</table>

7.2 JETI Duplex Telemetry

The HERKULES III Telemetry Interface supports the old Jeti Text mode and the Jeti Binary Mode. Most of the telemetry information is available in both modes. Please refer to the JETI user manual on setup the sensors in your system.

The general setup and software update procedure is described in Chapter 5.7 "Update Procedure TELEMETRY **" on page 37.**

The available Data is described below.

7.2.1 Setup Sequence

In order to install the target Telemetry System on the Datalogger, the correct files have to be copied from the "Firmwares"-folder to the root folder of the MicroSD-Card.

Important Note:

Please check if your Computer's operation system displays by standard filenames starting with a "." (dot) like .setup.hti. (Remember the dot in front of the name!

By default this might be hidden on Windows PC or MAC. Please referrer to your operating systems manual to "unhide" file names starting with dot.

1) Delete from the root directory of your MicroSD-Card these files:

```
.setup.hti
HTI25_xxx.bin
 hticonf.txt
```

2) Copy these 3 files from the subfolder "**Firmwares**" from any of the target system's subfolders (e.g. JETI_vxx) the 3 files **to the Root-folder of the Micro-SD-card.**

(**NOTE**: copy the **Files** from the folder only, **NOT** the Folder **itself**!)

The root must contain these 3 files:

File Name	Description
.setup.hti	Dont change or delete this file!
HTI25_JETI_xxx.bin	New Firmware File to be updated
hticonf.txt	User-Configuration File. See description below

4) Edit User-Configuration File “hticonf.txt”

Open the hticonf.txt on the SD-card and change the alarm values and correction factors as described below.

Content of "hticonf.txt"

```
  800,  355,  335,  418,  216,  12,    0,    31,    0
   |     |     |     |     |     |     |      |     |JETI Duplex Mode
   |     |     |     |     |     |     |      |Selection of HoTT Sensors
   |     |     |     |     |     |     |0% IBAT correction factor
   |     |     |     |     |     |1,2A IBAT constant offset value
   |     |     |     |     |21,6V Battery Alarm threshold
   |     |     |     |4,18V = 100% Single Cell Voltage threshold
   |     |     |3,35V = 0% Single Cell Voltage threshold
   |     |3,55V Single Cell Alaram threshold
   |8000mAh Capacity Alarm threshold
```

Byte	Value		Example	Unit		Description	System
1	800	=	8000	mAh	Capacity	mAh Alarm Limit x 10mAh	All
2	355	=	3,55	V	LCV	Single Cell Voltage Alarm Limit x 10mV	All
3	335	=	3,35	V	CV 0%	0% Single Cell Voltage Limit x 10mV	HoTT Text
4	418	=	4,18	V	CV 100%	100% Single Cell Voltage Limit x 10mV	HoTT Text
5	216	=	21,6	V	VBAT	VBAT Alarm Limit / 10V	HiTec only
6	12	=	1,2	A	OFFSET	IBAT constant offset value for not measured load	All
7	0	=	0	%	GAIN	IBAT correction value in % (+/-) Values possible	All
8	31	=	1	ESC	SENSOR	Enable Graupner Hott Displays	HoTT only
		=	1	GAM	SENSOR	GPS	
		=	1	EAM	SENSOR	VARIO	
		=	1	VARIO	SENSOR	EAM	
		=	1	GPS	SENSOR	GAM	
9	0	=	0	0 = default	EXP	Jeti Timeout Config 0 = three Binary Messages (default) (15 Sensors active) => no Expander 1 = 2 Binary Messages (12 Sensors aktive) => with Expander	JETI EX only

Description:

Byte 1: 800 = Capacity Alarm threshold x 10 in mAh.

If capacity is exceeded, alarm is send from Telemetry System to Ground Station

=> 1000 means 10.000mAh

=> 220 means 2.200mAh

=> 500 means 5.000mAh

Byte 2: 355 = Lipo lowest cell voltage alarm threshold / 100 in V

If lowest cell voltage is lower than this, alarm is send from Telemetry System to GND

=> 355 means 3.55V

=> 345 means 3.45V

Byte 3: 335 = not relevant for JETI Telemetry

Byte 4: 418 = not relevant for JETI Telemetry

Byte 5: = not relevant for JETI Telemetry

Byte 6: 12 = Offset correction value of total current

This value is added to the total battery current and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 12 means 1.2A

Byte 7: 0 = Gain correction value of total current

Total battery current is multiplied by this value [Ibat = (1+ Gain) x Ibat_power] and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 4 means 4%

Byte 8: 31 = not relevant for JETI Telemetry

Byte 9: 1 = No. of send parameter in Jeti Duplex (only relevant for Jeti Duplex)

1 = 12 parameters active (default)

0 = 15 parameters active (may cause timeout with expander).

Select 0 only when using NO JETI Expander. Otherwise the number of values would cause timeouts in Data transmission.

4) Put the SD-card back to the HERKULES III Datalogger

Power-on the HERKULES III and the firmware upgrade starts automatically.

During update process the RED LED on the Datalogger blinks for about 15sec.

In case the already installed firmware is the same on the SD-Card, no update is done and the Datalogger starts working immediately.

7.2.2 JETI Text Mode (JETI-Box)

Welcome-Screen 0)

			T	E	L	E	M	E	T	R	Y				
		H	E	R	K	U	L	E	S		I	I	I		

Overview Screen 1)

9	9	.	9	V		9	9	.	9	A		9	9	9	W
9	9	9	9	9	m	A	h		-	9	9	C	:	M	1

Line 1: Volt, Current, Watt

Line 2: mAh, Temp, Hottest ESC

Overview-Screen 2)

9	9	.	9	V		9	9	.	9	A		9	9	9	W
9	9	9	9	9	m	A	h		9	9	9	:	9	9	s

Line 1: Volt, Current, Watt,

Line: mAh, Motor Runtime

Overview Screen 3)

9	9	.	9	V		9	9	9	A		+	2	4	6	m
9	9	9	9	9	m	A	h		-	9	9	C	:	M	1

Line 1: Volt, Current, Altitude

Line 2: mAh, Temp, Hottest ESC

Overview Screen 4)

9	9	.	9	V		9	9	9	A		+	2	4	6	m
9	9	9	9	9	m	A	h		9	9	9	:	9	9	s

Line 1: Volt, Current, Altitude

Line 2: mAh, Runtime

Screen 5) Motor Status

1	R	U	N		S	T	P		O	T	!		O	C	!
5	S	T	P		S	T	P		S	T	P		S	T	P

Line 1: Status Motor 1…4

Line 2: Status Motor 5…8

RUN = Motor is running

STP = Motor stopped

OC! = Motor stopped due to overcurrent

OT! = Motor stopped due to overtemp

Screen 5…8) Single ESC Values

Displays the single ESC values like ESC current and temperature

M	1	:		1	5	.	5	A		1	0	5	°	C	
M	2	:		1	7	.	5	A		1	0	5	°	C	

M	3	:		9	9	.	5	A		1	0	5	°	C	
M	4	:		9	9	.	5	A		1	0	5	°	C	

M	5	:		9	9	.	5	A		1	0	5	°	C	
M	6	:		9	9	.	5	A		1	0	5	°	C	

M	7	:		9	9	.	5	A		1	0	5	°	C	
M	8	:		9	9	.	5	A		1	0	5	°	C	

7.2.3 JETI Binary Mode (JETI-EX)

The HERKULES III Telemetry System emulates different JETI-EX sensors, which can be arranged by the user individually in the JETI RC-Transmitter or the JETI Box Profi.

Available Telemetry Values:

- Battery Voltage VBAT in V
- Total Battery Current IBAT in A
- Hottest Temperature in °C
- Hottest ESC No.
- Used Charge in mAh
- Altitude in meter
- Motor Status (Run, Stop, Fail, TO, OC, OT)

7.3 Futaba S.BUS

The HERKULES III Telemetry System emulates different Futaba / Robbe SBUS sensors. HERKULES III telemetry emulates 8 Futaba Sensors, 2x F1678, 5x F1713, 1x F1712 and re-uses 13 of 31 time slots.

7.3.1 Supported Receivers

Currently only 2 types of receivers are supported: R7003 and R7008 are the only receivers from Futaba, which support external telemetry systems.

7.3.2 Emulated Futaba Sensors

The following Futaba sensors are emulated by HERKULES III

Table 7 - Emulated Futaba Sensors

Emulated Futaba Sensor			Applied Herkules Sensor Values			
ID	Name	Value	ID	Short	Range / Dim	Description
1	F1678 Robbe Current Sensor	Current	1	Ibat	[A]	Actual Battery Current
		Voltage	2	Vbat	[V]	Actual Battery Voltage
		Capacity	3	Capacity	[mAh]	Actual Battery Capacity
2	F1678 Robbe Current Sensor	Current	4	IbatAvg	[A]	Average Battery Current
		Voltage	5	LCV	[1…8]	Lowest Lipo Cell Voltage
		Capacity	6	MotorTime	[min:sec]	Actual Motor Runtime e.g. 1.643mAh = 16min 43sec
3	F1713 Robbe Temperature	Temperature	7	LCN	[1…8]	Lowest Lipo Cell No.
4	F1713 Robbe Temperature	Temperature	8	LBN	[1…4]	Lowest Battery No.
5	F1712 Robbe Vario Sensor	Vario	9	Vario	[m/s]	Ascend / Descend rate in m/s
		Altitude	10	Altitude	[m]	Actual height above Ground in m
6	F1713 Robbe Temperature	Temperature	11	TempMax	[°C]	Maximum Temperature on all Herkules ESCs
7	F1713 Robbe Temperature	Temperature	12	nTempMax	[1…16]	Number of Hottest ESC
8	F1713 Robbe Temperature	Temperature	13	nMotFail	[1…16]	Number of ESC with any failure (OT, OC, STALL, etc.)

7.3.3 Software Preparation

All Futaba transmitters must have the latest firmware. The T14SG must have version 3.0 or higher.

Please refer to the user manual of your transmitter on how to update to the latest firmware.

7.3.4 Hardware Preparation HERKULES Telemetry v1

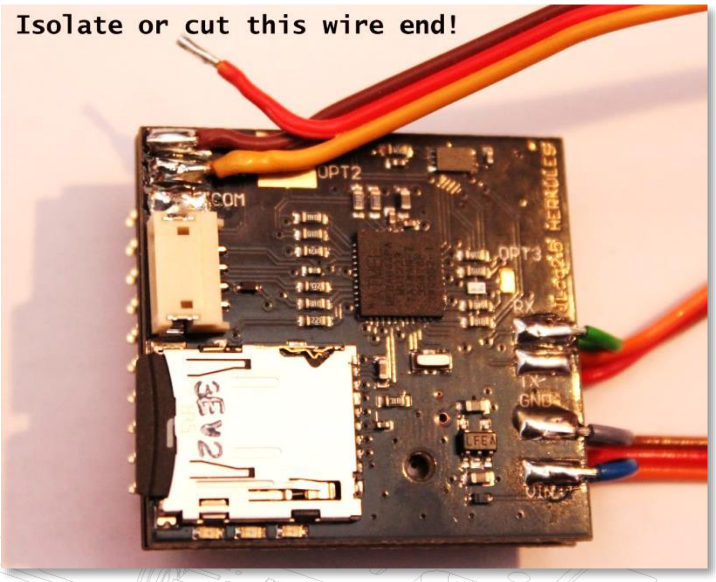
As Futaba uses an inverted polarity of data signals and a non-standard connector setup, the telemetry-cable of the HERKULES has to be resoldered.

Important Note:

Please use a small solder iron tip to remove and re-solder the cables carefully. After soldering, check carefully that there is no short circuit between cables and ensure that there are no solder bubbles on the electronics.

“Figure 51” on page 76 shows the modified Telemetry Cable of the HERKULES III Datalogger for Futaba Sensors.

Figure 51 - Modification of PPM Cable with Futaba

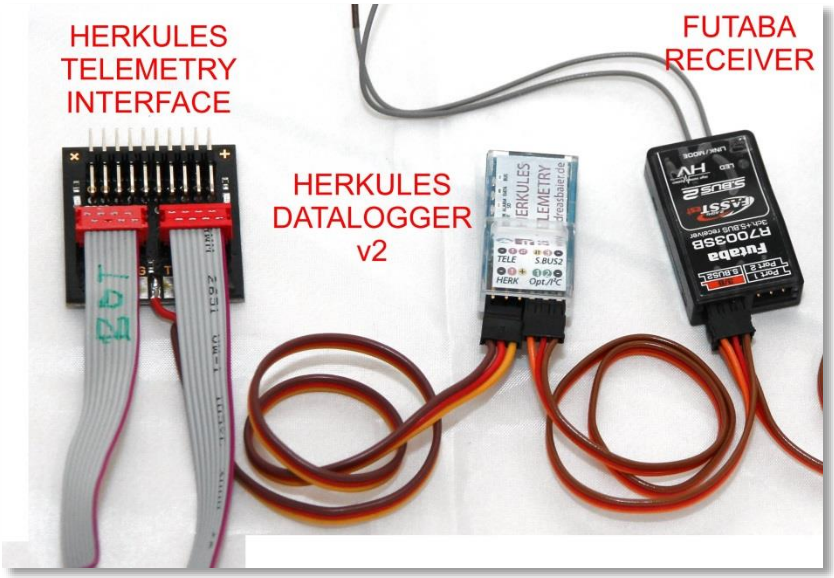


a) Remove the red-cable from the middle solder pad and cut the end or isolate the open end with a small shrink tube
b) Remove the yellow wire (close to the white connector) from the solder pad "COM"
c) Now solder the open wire cable to the central solder pad called "SB II"
d) The finished installation should look like shown in picture above.

7.3.5 Hardware Preparation HERKULES Telemetry v2

With Herkules Telemetry Datalogger v2 no re-soldering is necessary any more. Connect the HERKULES TELEMETRY INTERFACE to the DATALOGGER input “HERK” and connect the SBUS-Output to the FUTABA Recievers SBUS input. That’s all.

Figure 52 - Connection of Telemetry Interface v2 with Futaba



7.3.6 Setup Sequence

Futaba Sensors have to be registered once on a telemetry capable transmitter (e.g. Futaba T14SG) or Futaba Telemetry Box.

Please watch also the online-video, describing the register-process:

https://vimeo.com/97568410

Important Note:

Please bind first a telemetry capable receiver to your transmitter. Otherwise the transmitter will not accept the Herkules III Sensor registration. For details on the general registration process of telemetry sensors, please refer to the Futaba user guide of your Futaba transmitter.

After registering, the alarm thresholds have to be set either in the transmitter (or telemetry box) or on "hticonf.txt" on the HERKULES III Datalogger SD-Card.

Note: This registration process has to be repeated completely if the register process has failed or the sensors have been deleted.

a) Copy all telemetry files to the SD-Card
b) Activate the register flag in the file ".setup.hti" on the Micro-SD card
c) put the Micro-SD card to the Datalogger and power on
d) Register 8 sensors in Transmitter

Figure 53 shows the connection of the Datalogger SBUS output to the Transmitters I/F input for Futaba Sensors.

Figure 53 - Register HERKULES TELEBOARD as FUTABA Sensor

In order to install the target Telemetry System on the Datalogger, the correct files have to be copied from the “Firmwares”-folder to the root folder of the MicroSD-Card.

Important Note:

Please check if your Computer’s operation system displays by standard filenames starting with a “.” (dot) like .setup.hti. (Remember the dot in front of the name!

By default this might be hidden on Windows PC or MAC. Please referrer to your operating systems manual to “unhide” file names starting with dot.

Step 1) Delete from the root directory of your MicroSD-Card the files all as follows

File Name	Description
.setup.hti	Dont change or delete this file except described
HTI25_FUTABA_xxx.bin	New Firmware File to be updated
hticonf.txt	User-Configuration File. See description below

with the attached USB-Card reader or SD-Adapter for the Micro-SD card.

Open the file ".setup.hti" with a text editor and activate the "Register Flag" in the ".setup.hti".

Change the next-to-last value "132" to "148" or copy the file ".setup.hti" from the subfolder "register" in the downloaded software across the file ".setup.hti" on the SD-card (if asked, overwrite it).

SETUP.HTI for Datalogger v1

Content of ".setup.hti" default
38400,0,0,10,0,12,82,17, ... , 16,5,16,16,**132**,192

Content of ".setup.hti" Register Mode Active
38400,0,0,10,0,12,82,17, ... ,16,5,16,16,**148**,192

SETUP.HTI for Datalogger v2

Content of ".setup.hti" Register Mode Active
38400,0,128,10,0,12,82,17, ... ,16,16,16,**132**,143,0,0,3

Content of ".setup.hti" Register Mode Active
38400,0,128,10,0,12,82,17, ... ,16,16,16,**148**,143,0,0,3

Step 2) Plug the Micro-SD card into the HERKULES III Telemetry Interface

The HERKULES III Telemetry Interface now reads the file ".setup.hti" on next start and clears the register flag 148 back to 132.

Step 3) Connect the HERKULES III Telemetry Interface to the HERKULES powerboard via the red Micromatch cable and power-on the system on a current limited power supply.

Wait about 20sec until the firmware is updated.

After a successful update, a new file "version.txt" is created on the SD card. This file contains the version name the firmware.

Important Note

Don’t remove the SD-Card or power supply from the Telemetry Interface until the registration process at the Futaba transmitter is completed. A power-off exits the register mode and you have to start the register process again starting at Step 1)

Note on Futaba Telemetry Terminals

Only latest firmware’s o T-Box supports HERKULES III Telemetry and the sensor type F1678.

There is a bug in the Futaba transmitter software. If the transmitter is switched-off and on again during flight, it sets the actual altitude to zero and will report a wrong altitude. Please complain at Futaba and ask for a bugfix of this problem.

Step 4) Connect the Telemetry Interface Cable of the Datalogger to the I/F connector on the back of your transmitter and start the registration process of sensors according the description in the user manual of your transmitter. (See Figure 53 on page 77)

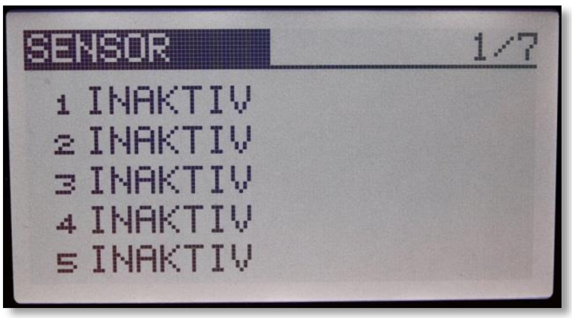
Step 5) Connect the Telemetry Interface Cable of the Datalogger to the Receivers “S.BUS” input and restart transmitter, receiver and HERKULES III Telemetry Interface.

Register Example for Futaba T14SG:

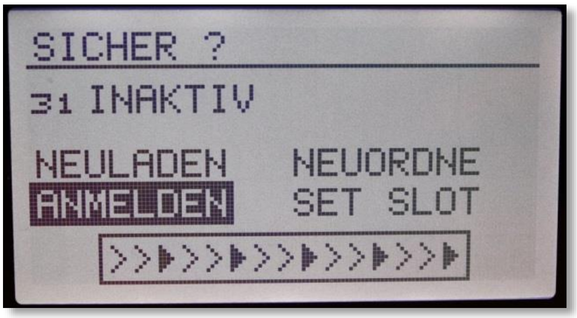
1. Double click on “LNK” button on cursor panel
2. Scroll down to “SENSOR” and scroll down to the lowest sensor (31)
3. Select “RELOAD / NEULADEN” to clear all sensors
4. Select “REGISTER / ANMELDEN” to start Register process 8 times to register all sensors until the message “COMMUNICATION ERROR” appears.
5. Restart Transmitter and Receiver and Connect the Telemetry Interface to the Receiver
6. Arrange your 4 most interesting telemetry data on the Telemetry front page of the transmitter

1

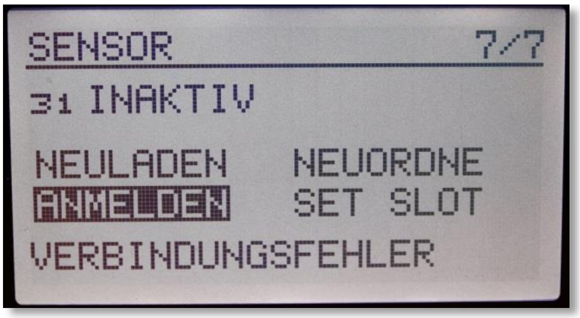
2



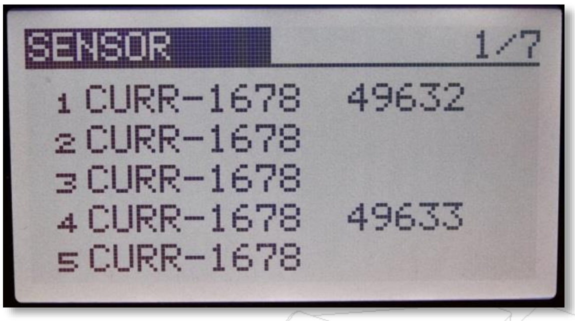
3



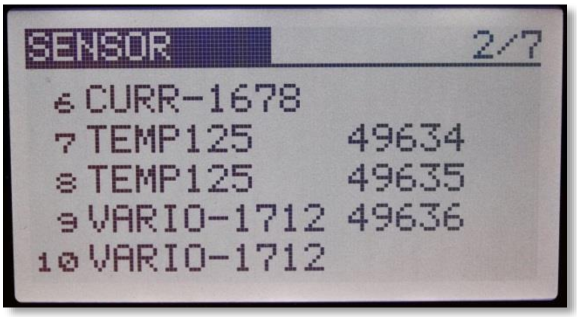
4



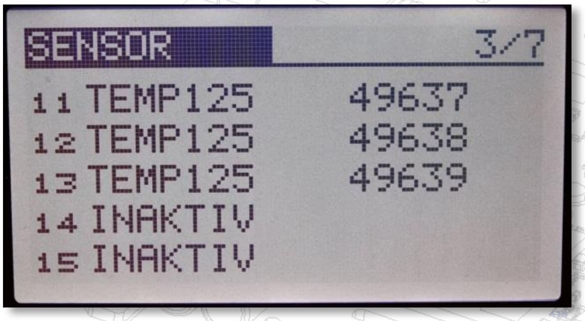
5



6



7



8

Now Set your most interesting values in the main telemetry display according your user manual. (8)

7.3.7 Setup of Alarms

Futaba telemetry alarms are set in the transmitter. Please refer to the Futaba user manual on how to set these alarm thresholds.

HERKULES III Telemetry allows the user to set alarms in the File "hticonf.txt" on the Micro-SD-Card of the HERKULES III Telemetry. In this case, the Transmitter alarms have to be set to special values.

Table 8 - Futaba Possible Alarms set by Herkules III Telemetry in the file “hticonf.txt”

Byte	Value		Example	Unit		Description	System
1	800	=	8000	mAh	Capacity	mAh Alarm Limit x 10mAh	All
2	355	=	3,55	V	LCV	Single Cell Voltage Alarm Limit x 10mV	All
3	335	=	3,35	V	CV 0%	0% Single Cell Voltage Limit x 10mV	HoTT Text
4	418	=	4,18	V	CV 100%	100% Single Cell Voltage Limit x 10mV	HoTT Text
5	216	=	21,6	V	VBAT	VBAT Alarm Limit / 10V	HiTec only
6	12	=	1,2	A	OFFSET	IBAT constant offset value for not measured loads / 10A	All
7	0	=	0	%	GAIN	IBAT correction value in % (+/-) Values possible	All
8	31	=	1	ESC	SENSOR	Enable Graupner Hott Displays	HoTT only
		=	1	GAM	SENSOR	GPS	
		=	1	EAM	SENSOR	VARIO	
		=	1	VARIO	SENSOR	EAM	
		=	1	GPS	SENSOR	GAM	
9	0	=	0	0 = default	EXP	Jeti Timeout Config 0 = three Binary Messages (default) (15 Sensors active) => no Expander 1 = 2 Binary Messages (12 Sensors aktive) => with Expander	JETI EX only

Table 9 - Futaba hticonf.txt

Content of "hticonf.txt"

```
800, 355, 335, 418, 216, 12, 0, 31, 0
                                    JETI Duplex Mode
                                Selection of HoTT Sensors
                            0% IBAT correction factor
                        1,2A IBAT constant offset value
                    21,6V Battery Alarm threshold
                4,18V = 100% Single Cell Voltage threshold
            3,35V = 0% Single Cell Voltage threshold
        3,55V Single Cell Alaram threshold
    8000mAh Capacity Alarm threshold
```

Edit User-Configuration File "hticonf.txt"

Open the hticonf.txt on the SD-card and change the alarm values and correction factors as described below.

Description:

Byte 1: 800 = Capacity Alarm threshold x 10 in mAh.

If capacity is exceeded, alarm is send from Telemetry System to Ground Station

=> 1000 means 10.000mAh

=> 220 means 2.200mAh

=> 500 means 5.000mAh

Byte 2: 355 = Lipo lowest cell voltage alarm threshold / 100 in V

If lowest cell voltage is lower than this, alarm is send from Telemetry System to GND

=> 355 means 3.55V

=> 345 means 3.45V

Byte 3: 335 = not relevant for Futaba Telemetry

Byte 4: 418 = not relevant for Futaba Telemetry

Byte 5: = not relevant for Futaba Telemetry

Byte 6: 12 = Offset correction value of total current

This value is added to the total battery current and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 12 means 1.2A

Byte 7: 0 = Gain correction value of total current

Total battery current is multiplied by this value [Ibat = (1+ Gain) x Ibat_power] and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 4 means 4%

Byte 8: 31 = not relevant for Futaba Telemetry

Byte 9: 1 = not relevant for Futaba Telemetry

7.3.8 Infos on Futaba Telemetry Alarms

Lowest Cell Voltage (LCV):

LCV alarm can be generated either by HERKULES III Telemetry or Futaba Transmitter.

a) Alarm by HERKULES III Telemetry

Set the LCV Alarm value in the file "hticonf.txt" on the Micro-SD-Card to an alarm value, e.g. 355 will send LCV alarm when the lowest cell voltage falls below 3.55V. In case of cell under voltage, HERKULES III Telemetry adds 10.0V to the displayed voltage. In this case, please set the alarm in the transmitter > 5.0V.

b) Alarm by Futaba Transmitter

If you want to make the alarm from the transmitter only, disable the LCV alarm value by writing a value e.g. 100 (1.00V) to the "hticonf.txt". This will disable the alarms by HERKULES III. Enable the LCV alarm in the transmitter e.g. < 3.6V. Now set in the transmitter a capacity threshold e.g. < 3.55V.

Capacity (mAh):

A capacity alarm can be generated either by HERKULES III Telemetry or the Futaba Transmitter.

a) Alarm by HERKULES III Telemetry

Set the mAh Alarm value in the file "hticonf.txt" on the Micro-SD-Card to an alarm value, e.g. 800 will send mAh alarm when capacity exceeds 8000mAh. In case of alarm, HERKULES III Telemetry negates the capacity value (e.g. to -8000mAh) and sends an alarm for 6 seconds. The alarm is repeated every 20sec until motors are stopped.

b) Alarm by Futaba Transmitter

If you want to make the alarm threshold from the transmitter only, disable mAh alarm value in by writing a value e.g. 10000 (100000mAh) to the "hticonfig.txt" which will never be reached. This will disable the alarms by HERKULES III. Now enable the mAh alarm in the transmitter by writing e.g. >8000mAh threshold.

Important Note:

Futaba can display max 32768mAh of capacity. In case this value is exceeded, the value starts at -32768 again. This is a bug of the Futaba Software. Please complain at Futaba and ask for firmware update.

7.3.9 Summary

We have to apologize for this uncomfortable way to register Futaba sensors. But this is related to the "very strange" telemetry concept of Futaba and it is beyond our control. Other systems are more straight-forward and are more user friendly.

7.4 Spektrum Telemetry

7.4.1 Emulated Spektrum Sensors

The following Spektrum sensors are emulated by HERKULES III

Table 10 - Emulated Spektrum Sensors

Emulated Spektrum Sensor			Applied Herkules Sensor Values			
ID	Name	Value	ID	Short	Range / Dim	Description
1	Telemetry	Altitude	1	Altitude	[m]	Actual height above Ground in m
		Current	2	Ibat	[A]	Actual Battery Current
2	Powerbox	Voltage 1	3	Vbat	[V]	Actual Battery Voltage
		Voltage 2	4	LCV	[V]	Lowest Lipo Cell Voltage
		Capacity 1	5	Capa_Rest	[mAh]	Remaining Battery Capacity
		Capacity 2	6	Capa_Used	[mAh]	Consumed Battery Capacity
4	G-Force	x-axis	7	Altitude	[m]	Actual height above Ground in m
		x-axis-max	8	Vario	[m/s]	Ascend / Descend rate in m/s
		y-axis	9	TempMax	[°C]	Maximum Temperature on all Herkules ESCs
		y-axis-max	10	nTempMax	[1...16]	Number of Hottest ESC
		z-axis	11	nMotFail	[1...16]	Number of ESC with any failure (OT, OC, STALL, etc.)
		z-axis-max	12	LCN	[1...8]	Lowest Lipo Cell No.
		z-axis-min	13	LBN	[1...4]	Lowest Battery No.

The Alarams have to be setup in the file ".setup.hti" stored on the SD card as described in the next chapter. Depending on the type of Transmitter there are various displays and display summaries in the Spektrum Transmitters available. Please find below some examples.

A) Telemetry

Telemetry

Altitude:	**Alti**	m
Current:	**Ibat**	A

B) Powerbox

PowerBox

Bat 1:	**Vbat**	V
Bat 2:	**LCV**	V
Capacity 1:	**Capa_Rest**	mAh
Capacit 2:	**Capa_Used**	mAh

C) GForce

GForce

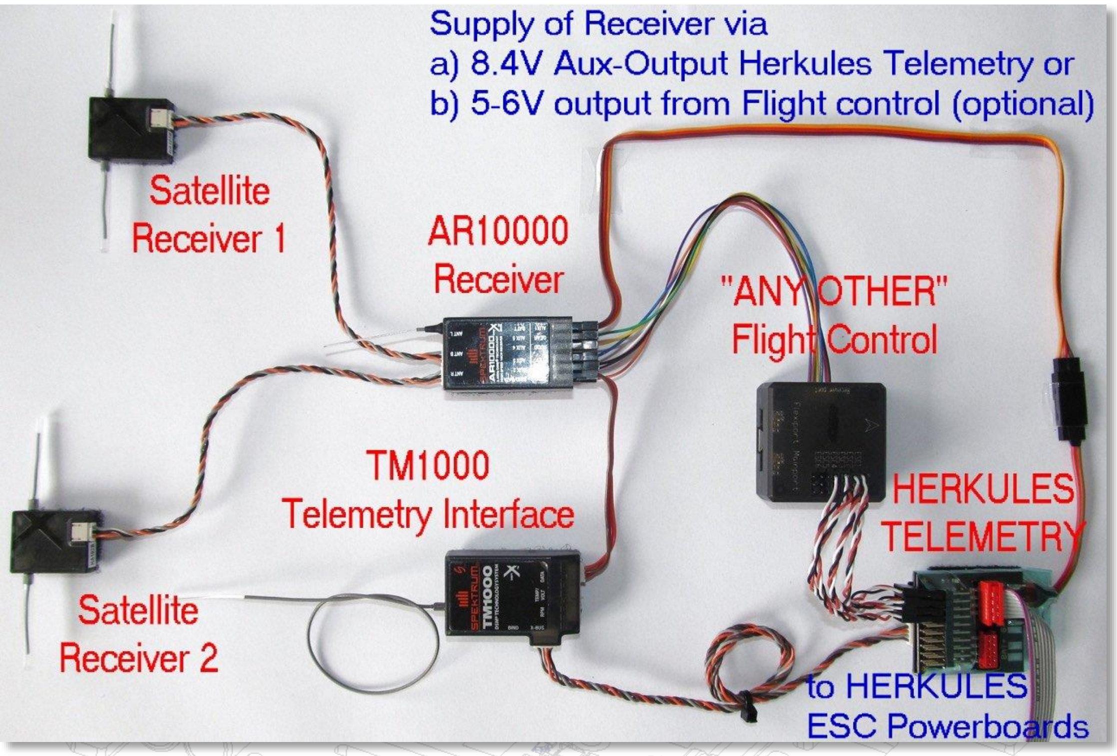
	Axis	Max	Min
X:	**Alti**	**Vario**	
Y:	**TempMax**	**nTempMax**	
Z:	**nMotFail**	**LCN**	**LBN**

7.4.2 Wiring Diagram

Figure 54 shows an example of wiring Spektrum Telemetry components with Herkules Telemetry System. The supply of the AR10000 receiver can be done either by a 5-6V output from the flight control system or via the 8.4V output from the Herkules Telemetry interface.

For more details please check chapter "**4.5 - The HERKULES III Telemetry Interface**" on page 20.

Figure 54 – Wiring Example of Spektrum Telemetry Components



7.4.3 Setup Sequence

In order to install the target Telemetry System on the Datalogger, the correct files have to be copied from the "Firmwares"-folder to the root folder of the MicroSD-Card.

Important Note:

Please check if your Computer's operation system displays by standard filenames starting with a "." (dot) like .setup.hti. (Remember the dot in front of the name!

By default this might be hidden on Windows PC or MAC. Please referrer to your operating systems manual to "unhide" file names starting with dot.

1) Delete from the root directory of your MicroSD-Card these files:

```
.setup.hti
HTI25_xxx.bin
 hticonf.txt
```

2) Copy these 3 files from the subfolder "**Firmwares**" from any of the target system's subfolders (e.g. JETI_vxx) the 3 files **to the Root-folder of the Micro-SD-card.**

(**NOTE**: copy the **Files** from the folder only, **NOT** the Folder **itself**!)

The root must contain these 3 files:

File Name	Description
.setup.hti	Dont change or delete this file!
HTI25_JETI_xxx.bin	New Firmware File to be updated
hticonf.txt	User-Configuration File. See description below

4) Edit User-Configuration File “hticonf.txt”

Open the hticonf.txt on the SD-card and change the alarm values and correction factors as described below.

Content of "hticonf.txt"

```
800,  355,  335,  418,  216,  12,   0,   31,    0
                                                 JETI Duplex Mode
                                            Selection of HoTT Sensors
                                       0% IBAT correction factor
                                  1,2A IBAT constant offset value
                             21,6V Battery Alarm threshold
                       4,18V = 100% Single Cell Voltage threshold
                 3,35V = 0% Single Cell Voltage threshold
           3,55V Single Cell Alaram threshold
     8000mAh Capacity Alarm threshold
```

Byte	Value		Example	Unit		Description	System
1	800	=	8000	mAh	Capacity	mAh Alarm Limit x 10mAh	All
2	355	=	3,55	V	LCV	Single Cell Voltage Alarm Limit x 10mV	All
3	335	=	3,35	V	CV 0%	0% Single Cell Voltage Limit x 10mV	HoTT Text
4	418	=	4,18	V	CV 100%	100% Single Cell Voltage Limit x 10mV	HoTT Text
5	216	=	21,6	V	VBAT	VBAT Alarm Limit / 10V	HiTec only
6	12	=	1,2	A	OFFSET	IBAT constant offset value for not measured load	All
7	0	=	0	%	GAIN	IBAT correction value in % (+/-) Values possible	All
8	31	=	1	ESC	SENSOR	Enable Graupner Hott Displays	HoTT only
		=	1	GAM	SENSOR	GPS	
		=	1	EAM	SENSOR	VARIO	
		=	1	VARIO	SENSOR	EAM	
		=	1	GPS	SENSOR	GAM	
9	0	=	0	0 = default	EXP	Jeti Timeout Config 0 = three Binary Messages (default) (15 Sensors active) => no Expander 1 = 2 Binary Messages (12 Sensors aktive) => with Expander	JETI EX only

Description:

Byte 1: 800 = Capacity Alarm threshold x 10 in mAh.

If capacity is exceeded, alarm is send from Telemetry System to Ground Station

=> 1000 means 10.000mAh

=> 220 means 2.200mAh

=> 500 means 5.000mAh

Byte 2: 355 = Lipo lowest cell voltage alarm threshold / 100 in V

If lowest cell voltage is lower than this, alarm is send from Telemetry System to GND

=> 355 means 3.55V

=> 345 means 3.45V

Byte 3: 335 = not relevant for JETI Telemetry

Byte 4: 418 = not relevant for JETI Telemetry

Byte 5: = not relevant for JETI Telemetry

Byte 6: 12 = Offset correction value of total current

This value is added to the total battery current and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 12 means 1.2A

Byte 7: 0 = Gain correction value of total current

Total battery current is multiplied by this value [Ibat = (1+ Gain) x Ibat_power] and can be used to estimate the not measured loads like LEDs, Flight Control, Gimbal in case of supplied via main battery

=> 4 means 4%

Byte 8: 31 = not relevant for JETI Telemetry

Byte 9: 1 = No. of send parameter in Jeti Duplex (only relevant for Jeti Duplex)

1 = 12 parameters active (default)

0 = 15 parameters active (may cause timeout with expander).

Select 0 only when using NO JETI Expander. Otherwise the number of values would cause timeouts in Data transmission.

4) Put the SD-card back to the HERKULES III Datalogger

Power-on the HERKULES III and the firmware upgrade starts automatically.

During update process the RED LED on the Datalogger blinks for about 15sec.

In case the already installed firmware is the same on the SD-Card, no update is done and the Datalogger starts working immediately.

7.5 LOGVIEW Serial Protocol

In case the user wants to use a separate Telemetry Interface e.g. a wireless transceiver like WI.232 an Open-Format Protocol can be accessed on the Telemetry Interface.

Check chapter “9 Control Protocol and Communication Interface” for detailed protocol description.

In case you have special requirements, please contact the Herkules development team via www.andreasebaier.de.

8 Analyzing Telemetry Data with LogView

When using the HERKULES III Telemetry Interface, the internal Datalogger collects lots of data which can be analyzed with the powerful Tool “LogView”.

Note: This is only valid for PPM Controls and NOT for Mikrokopter Flight Controls. Mikrokopter uses its own Datalogging features in conjuction with the Navigation board.

8.1 Installation and Setup

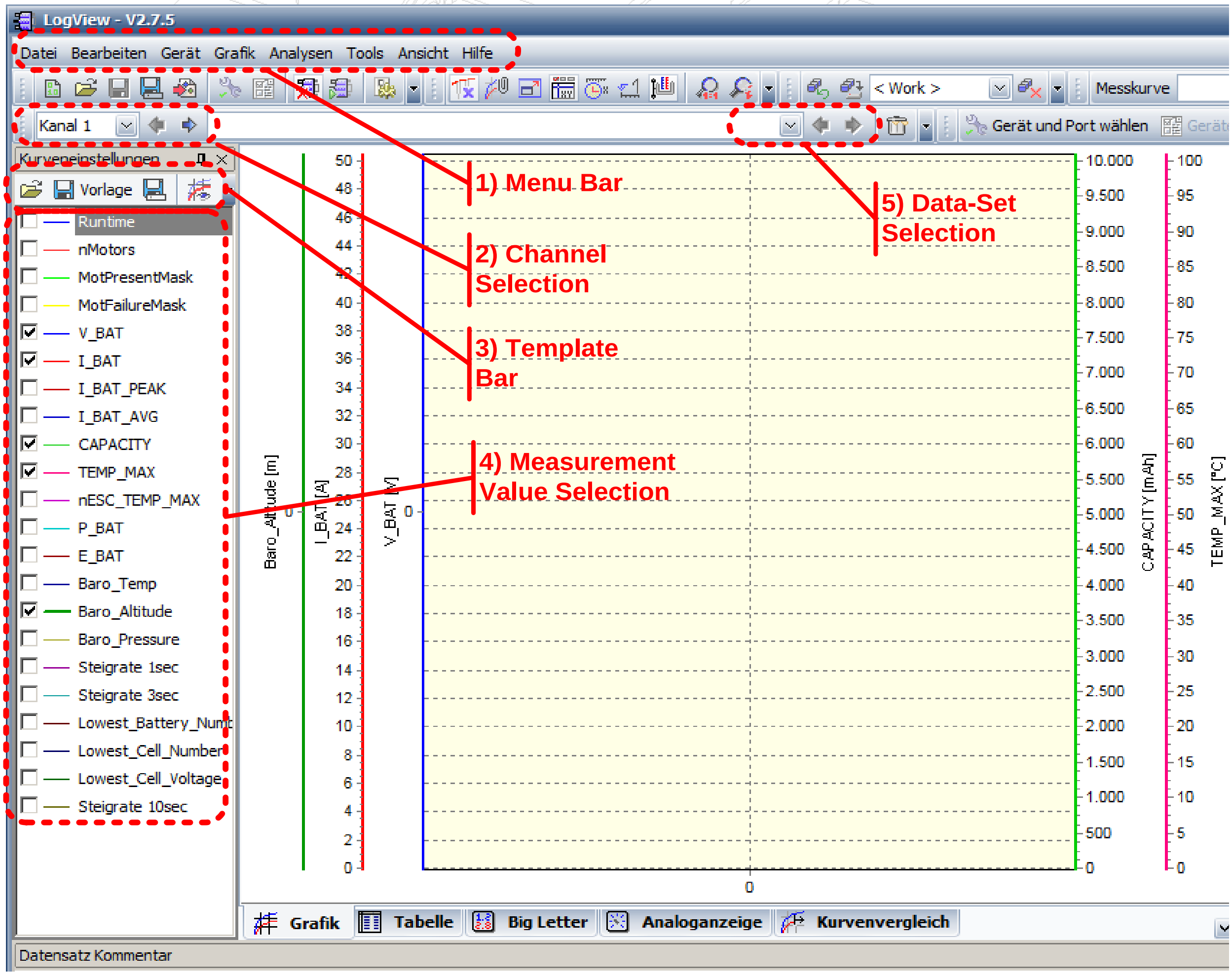
LogView is a very powerful and flexible software for visualizing data. You can download the software “for free” but the program is “**Donation Ware**” and its developers would be grateful if you donate payment for their efforts.

This documentation describes only the basic steps to import Herkules III Datalogging files but the software has much more functionality. Please refer to the very detailed documentation and help files of LogView.

Download and Installation

First go to www.logview.info and download the Tool Logview v.2.x. The software is available in different language files and please donate to the developers as described above.

After Installation, you get the main window as described below.

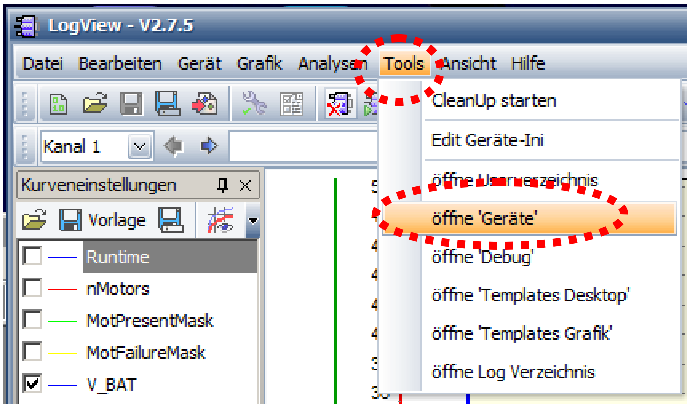
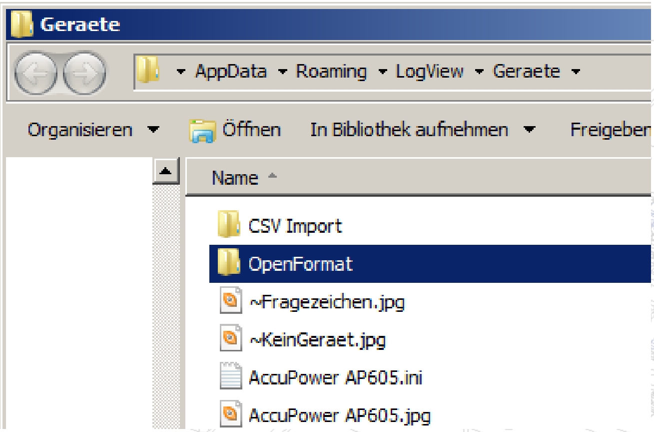
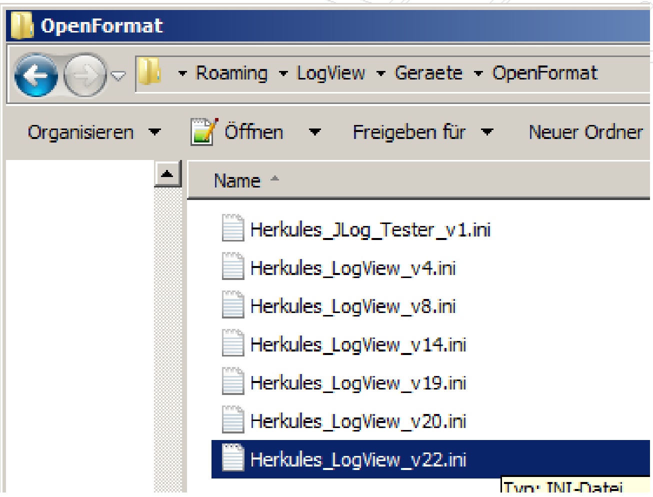


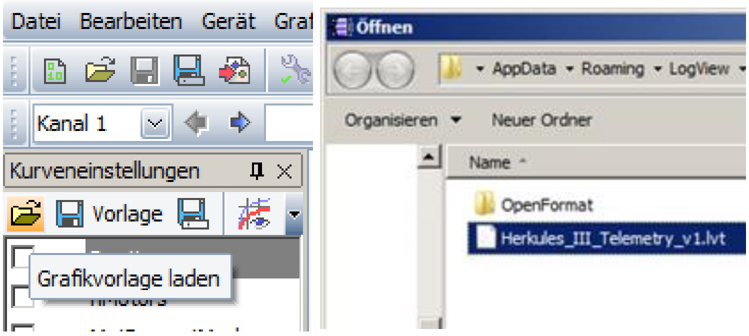
1) Menu Bar: Top Left for setup, load and save
2) Channel Selection: You can choose here 1 of 6 channels delivered by the HERKULES Telemetry
3) Template Bar: Load and save templates, view types, scaling of axes and personal adaptations

4) Measurement Value selection: Select the displayed measurement values

Install the HERKULES Template Files

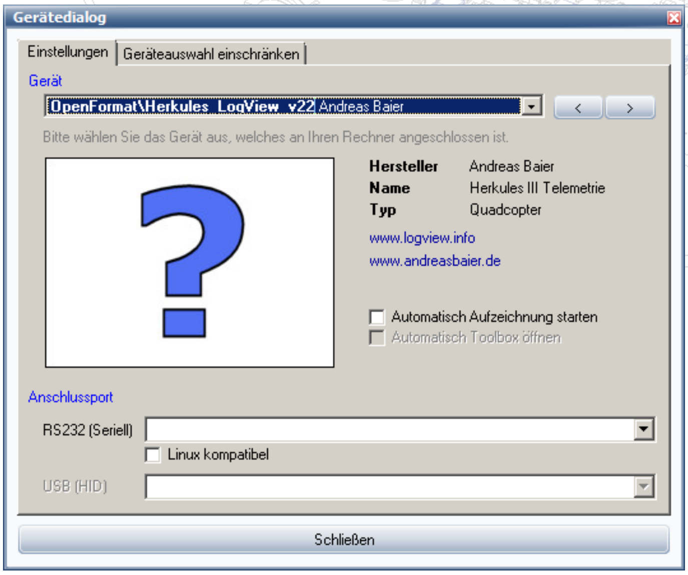
Now install the HERKULES Template Files into the OpenFormat folder of LogView:

	Click on **"Tools" > "open Devices"** (Click on **"Tools" > "öffne Geräte")**
	Double Click and open the Folder **"OpenFormat"**
	and Copy the file **"HERKULES_LogView_vxx.ini"** from the HERKULES Setup Folder into here.



Now click on the Template Bar on "Load Template" and copy from the HERKULES Setup folder the file "**HERKULES_II_Telemetry_vx.lvt**" into here.

Click on **"Device" > "Select Port"**
Click on **"Gerät" > "Gerät und Port wählen"**



Select the **"OpenFormat"** Device "**HERKULES_LogView_vxx**"

The comport must be set to a valid value and then click **"Close"**

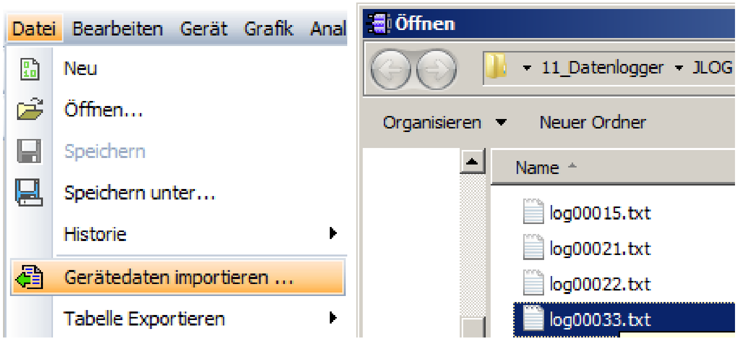
Then click "Schließen"

Now the basic setup of the OpenFormat is told to LogView and you can start importing your logged measurement files from the MicroSD Card.

8.2 Importing Files

To view the HERKULES III Datalogging files the files have to be imported.

NOTE: Don't mix this with **"open".** The files must be **"Imported"**



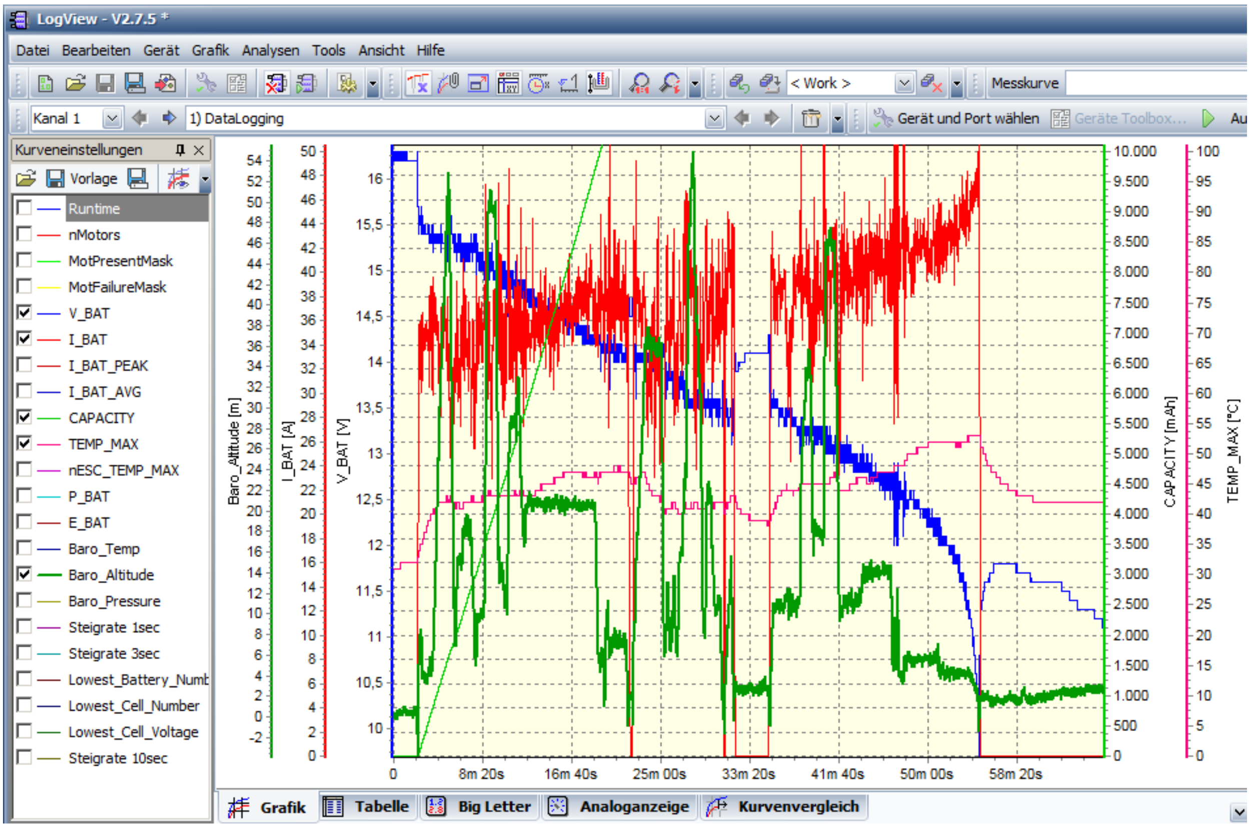
Click on "File Import"

Click on "Datei Geräted. Importieren"

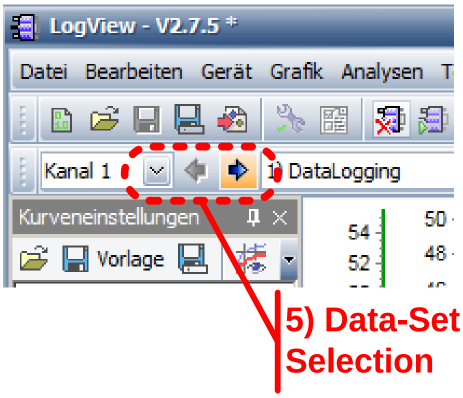
And select from your SD-Card subfolder the Logfile, e.g.

"log00033.txt".

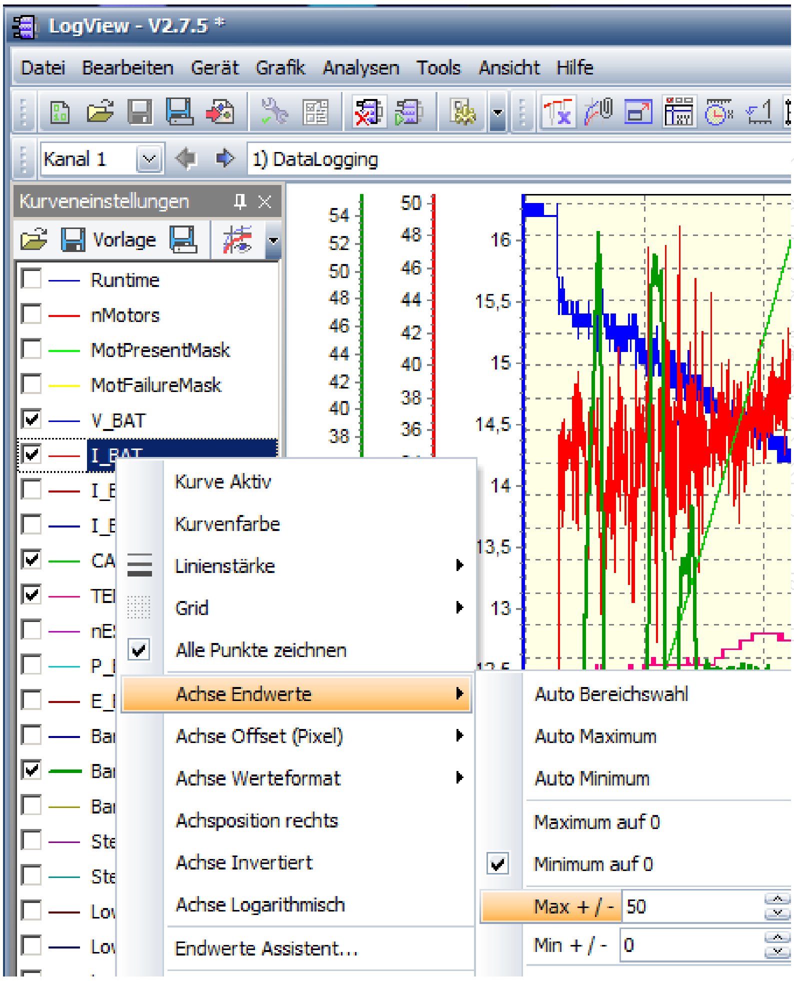
The Importing starts and the data is displayed.



8.3 Analyzing Data



Click the small blue arrow on the channel selection bar to switch between one of the six Herkules III Data channels.



Click with the right mouse key on a measurement value and change settings as you wish.

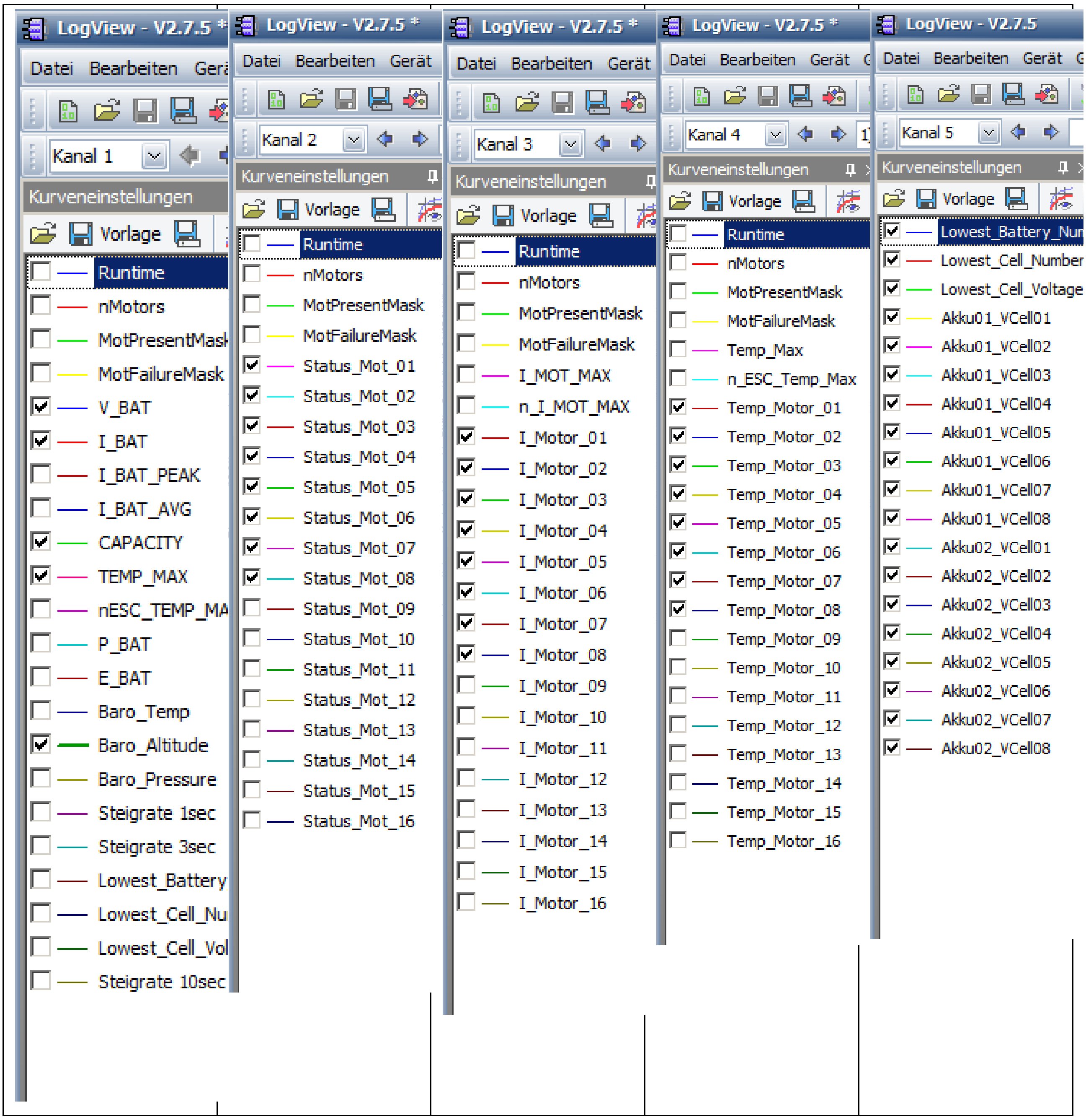
You can change e.g.
- Line weight
- Line color
- Axis scaling
- Axis position (left right)
- and many more….

For detailed explanation, please refer to the manual and help files of LogView.

8.4 Available Channels

Currently there are **6 Channels** with different data sets available.

- Channel 1: **Summary Data.** Most important measurements in one overview
- Channel 2: **Motor Status.**
- Channel 3: **Single Motor Currents**
- Channel 4: **Single ESC Temperatures**
- Channel 5: **Single Cell Voltages** of Battery 1 and 2 (or Single ESC VBAT in CH6-version)
- Channel 6: **Single Cell Voltages** of Battery 3 and 4 (or Single ESC RPM in CH6-version)



CHANNEL 1 (Summary Data)

Item #	Symbol	Description
1	Start	Start-Sign
2	Channel	Channel
3	State	State: Output: 0 = Binary, 1 = ASCII
4	Timestamp	Timestamp (Systemtime)
5	Runtime	Runtime (Motor Runtime)
6	nMotors	Number of deteceted Motors
7	MotPresentMask	Detected Motors Mask 0 = no Motor detected at init 1 = detected
8	MotFailureMask	Motor Failure Mask 0 = no Failure 1 = Failure
9	V_BAT	Battery Voltage
10	I_BAT	Battery Current (sum of all Motors)
11	I_BAT_PEAK	Peak Battery Current
12	I_BAT_AVG	Average Battery Current (CAPA/RUNTIME)
13	CAPA	Capacity
14	Temp_max	Maximum ESC temperature
15	nESC_Temp_Max	ESC-No. with max temperature
16	P_BAT	Battery Power
17	E_BAT	Energy
18	Baro_Temp	Temperature of the pressure Sensor
19	Baro_Altitude	Altitude over ground (zero at power-on)
20	Baro_Pressure	Air Pressure
21	Steigrate / 1sec	Climbrate per 1sec
22	Steigrate / 3sec	Climbrate per 3sec
23	LBN	Lowest Battery Number
24	LCN	Lowest Cell Number
25	LCV	Lowest Cell Voltage
26	Steigrate / 10sec	Climbrate per 10sec
27	Baro_Altitude_Max	Altitude over ground (zero at power-on)
28	Checksum	no checksum sent!
29	Carriege Return	
30	Line Feed	

CHANNEL 2 (Motor Status)

Item #	Symbol	Description
1	Start	Start-Sign
2	Channel	Channel
3	State	State: Output: 0 = Binary, 1 = ASCII
4	Timestamp	Timestamp (Systemtime)
5	Runtime	Runtime (Motor Runtime)
6	nMotors	Number of deteceted Motors
7	MotPresentMask	Detected Motors Mask 0 = not detected at init 1 = detected
8	MotFailureMask	Motor Failure Mask 0 = no Failure 1 = Failure
9	Status_Mot_01	0 = Motor is not Used (not detected)
10	Status_Mot_02	1 = STOP
11	Status_Mot_03	2 = RUN
12	Status_Mot_04	3 = STALL
13	Status_Mot_05	4 = TIMEOUT
14	Status_Mot_06	5 = OVERTEMP
15	Status_Mot_07	6 = OVERCURRENT
16	Status_Mot_08	7 = frei
17	Status_Mot_09	
18	Status_Mot_10	
19	Status_Mot_11	
20	Status_Mot_12	
21	Status_Mot_13	
22	Status_Mot_14	
23	Status_Mot_15	
24	Status_Mot_16	
25	Checksum	no checksum sent!
26	Carriege Return	
27	Line Feed	
28		
29		
30		

CHANNEL3 (Motor Currents)

Item #	Symbol	Description
1	Start	Start-Sign
2	Channel	Channel
3	State	State: Output: 0 = Binary, 1 = ASCII
4	Timestamp	Timestamp (Systemtime)
5	Runtime	Runtime (Motor Runtime)
6	nMotors	Number of deteceted Motors
7	MotPresentMask	Detected Motors Mask 0 = not detected at init 1 = detected
8	MotFailureMask	Motor Failure Mask 0 = no Failure 1 = Failure
9	I_MOT_MAX	Maximum single Motor Current
10	n_I_MOT_MAX	No. of motor with highest current
11	I_MOT_01	Actual Current Motor 01
12	I_MOT_02	Actual Current Motor 02
13	I_MOT_03	Actual Current Motor 03
14	I_MOT_04	Actual Current Motor 04
15	I_MOT_05	Actual Current Motor 05
16	I_MOT_06	Actual Current Motor 06
17	I_MOT_07	Actual Current Motor 07
18	I_MOT_08	Actual Current Motor 08
19	I_MOT_09	Actual Current Motor 09
20	I_MOT_10	Actual Current Motor 10
21	I_MOT_11	Actual Current Motor 11
22	I_MOT_12	Actual Current Motor 12
23	I_MOT_13	Actual Current Motor 13
24	I_MOT_14	Actual Current Motor 14
25	I_MOT_15	Actual Current Motor 15
26	I_MOT_16	Actual Current Motor 16
27	Checksum	no checksum sent!
28	Carriege Return	
29	Line Feed	

CHANNEL 4 (ESC Temperatures)

Item #	Symbol	Description
1	Start	Start-Sign
2	Channel	Channel
3	State	State: Output: 0 = Binary, 1 = ASCII
4	Timestamp	Timestamp (Systemtime)
5	Runtime	Runtime (Motor Runtime)
6	nMotors	Number of deteceted Motors
7	MotPresentMask	Detected Motors Mask 0 = not detected at init 1 = detected
8	MotFailureMask	Motor Failure Mask 0 = no Failure 1 = Failure
9	Temp_max	Maximum ESC temperature
10	n_ESC_Temp_Max	ESC-No. with max temperature
11	TEMP_MOT_01	Actual Temp ESC 01
12	TEMP_MOT_02	Actual Temp ESC 02
13	TEMP_MOT_03	Actual Temp ESC 03
14	TEMP_MOT_04	Actual Temp ESC 04
15	TEMP_MOT_05	Actual Temp ESC 05
16	TEMP_MOT_06	Actual Temp ESC 06
17	TEMP_MOT_07	Actual Temp ESC 07
18	TEMP_MOT_08	Actual Temp ESC 08
19	TEMP_MOT_09	Actual Temp ESC 09
20	TEMP_MOT_10	Actual Temp ESC 10
21	TEMP_MOT_11	Actual Temp ESC 11
22	TEMP_MOT_12	Actual Temp ESC 12
23	TEMP_MOT_13	Actual Temp ESC 13
24	TEMP_MOT_14	Actual Temp ESC 14
25	TEMP_MOT_15	Actual Temp ESC 15
26	TEMP_MOT_16	Actual Temp ESC 16
27	Checksum	no checksum sent!
28	Carriege Return	
29	Line Feed	

CHANNEL 5 (Single Cell Voltages)

Item #	Symbol	Description
1	Start	Start-Sign
2	Channel	Channel
3	State	State: Output: 0 = Binary, 1 = ASCII
4	Timestamp	Timestamp (Systemtime)
5	LBN	Lowest Battery Number
6	LCN	Lowest Cell Number
7	LCV	Lowest Cell Voltage
8	AKKU01_VCELL01	Single Cell Voltage Battery 01
9	AKKU01_VCELL02	Single Cell Voltage Battery 01
10	AKKU01_VCELL03	Single Cell Voltage Battery 01
11	AKKU01_VCELL04	Single Cell Voltage Battery 01
12	AKKU01_VCELL05	Single Cell Voltage Battery 01
13	AKKU01_VCELL06	Single Cell Voltage Battery 01
14	AKKU01_VCELL07	Single Cell Voltage Battery 01
15	AKKU01_VCELL08	Single Cell Voltage Battery 01
16	AKKU02_VCELL01	Single Cell Voltage Battery 02
17	AKKU02_VCELL02	Single Cell Voltage Battery 02
18	AKKU02_VCELL03	Single Cell Voltage Battery 02
19	AKKU02_VCELL04	Single Cell Voltage Battery 02
20	AKKU02_VCELL05	Single Cell Voltage Battery 02
21	AKKU02_VCELL06	Single Cell Voltage Battery 02
22	AKKU02_VCELL07	Single Cell Voltage Battery 02
23	AKKU02_VCELL08	Single Cell Voltage Battery 02
24	Checksum	no checksum sent!
25	Carriege Return	
26	Line Feed	

CHANNEL 6 (Single Cell Voltages)

Item #	Symbol	Description
1	Start	Start-Sign
2	Channel	Channel
3	State	State: Output: 0 = Binary, 1 = ASCII
4	Timestamp	Timestamp (Systemtime)
5	LBN	Lowest Battery Number
6	LCN	Lowest Cell Number
7	LCV	Lowest Cell Voltage
8	AKKU03_VCELL01	Single Cell Voltage Battery 03
9	AKKU03_VCELL02	Single Cell Voltage Battery 03
10	AKKU03_VCELL03	Single Cell Voltage Battery 03
11	AKKU03_VCELL04	Single Cell Voltage Battery 03
12	AKKU03_VCELL05	Single Cell Voltage Battery 03
13	AKKU03_VCELL06	Single Cell Voltage Battery 03
14	AKKU03_VCELL07	Single Cell Voltage Battery 03
15	AKKU03_VCELL08	Single Cell Voltage Battery 03
16	AKKU04_VCELL01	Single Cell Voltage Battery 04
17	AKKU04_VCELL02	Single Cell Voltage Battery 04
18	AKKU04_VCELL03	Single Cell Voltage Battery 04
19	AKKU04_VCELL04	Single Cell Voltage Battery 04
20	AKKU04_VCELL05	Single Cell Voltage Battery 04
21	AKKU04_VCELL06	Single Cell Voltage Battery 04
22	AKKU04_VCELL07	Single Cell Voltage Battery 04
23	AKKU04_VCELL08	Single Cell Voltage Battery 04
24	Checksum	no checksum sent!
25	Carriege Return	
26	Line Feed	

8.5 Zooming, Tips and Tricks

There are lots of shortcuts and useful features for analyzing your data. Here are the most important ones:

- **Zooming IN:** click and hold left mouse key and draw a window from top left to bot right across your zoom area
- **Zooming OUT:** click and hold left mouse key and draw a window from bot right to top left and zoom back to 100%
- **Zooming only the X-Axes:** Click and hold CTRL-key while pressing the left mouse key and zoom in from left to right and zoom out from right to left
- **Zooming only the Y-Axes:** Click SHIFT-key while pressing the left mouse key and zoom in from left to right and zoom out from right to left
- **Panning along the X-Axes:** While in Zoom Mode, click CTRL KEY and press middle mouse key (or mouse wheel) to Pan left and Right along the X-Axes without scaling the Y-Axes.
- **Set the scale of single ESC** y-axes manually like motor currents to the same value to be able to compare the values exactly

Exporting to Excel

All measured date could be exported e.g. to Excel or any other CSV formats.

Click on **“File” > “Export”**and select the file you want and which format you want to export to.

9 Control Protocol and Communication Interface

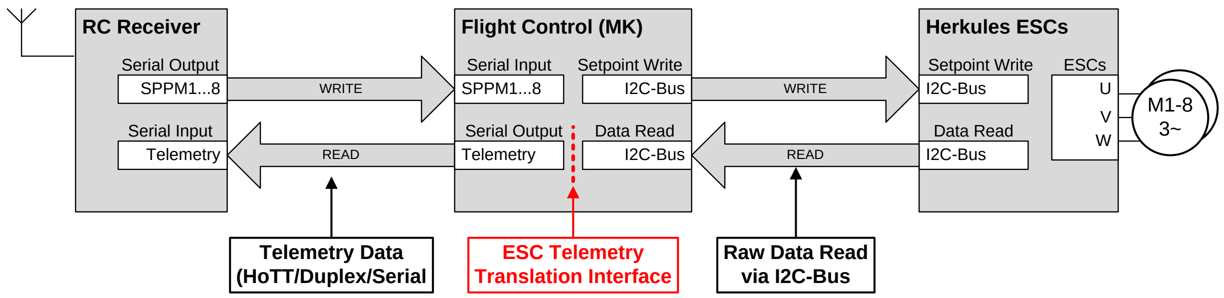
9.1 Overview

HERKULES III can be controlled via standard PPM signal or via serial communication. An I2C-Bus enables the user to read out a lot of telemetry data from the ESCs. In general there are two different control modes possible with the HERKULES ESCs:

A) I2C Control Mode - Setpoint write via I2C (Mikrokopter Standard)

This mode does all communication, setpoint write and data read via I2C bus. The Flight Control takes care of timing the write and read commands, interprets the raw-data from the HERKULES III ESCs and generates together with its own data a serial telemetry protocol which can be fed into a telemetry capable RC receiver.

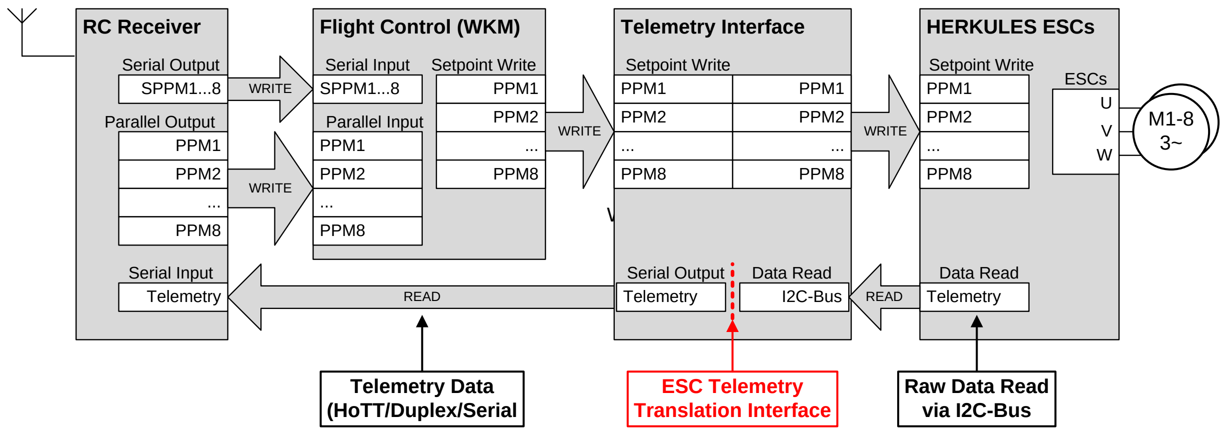
Figure 55- Control Mode A) I2C Control and Read



B) PPM Control Mode - Setpoint write via PPM (RC-Standard)

This mode controls the Setpoint of each ESC with a dedicated PPM signal and reads the Data via I2C-Bus. An additional interface controller, the "**HERKULES III Telemetry Interface**" reads the data from each ESC via I2C-Bus and translates it to a serial telemetry protocol which can be fed into to a telemetry capable RC-receiver. This mode "bypasses" the flight control and has the benefit of independence and flexibility. The Telemetry Interface Board provides additional interface connectors for standard PPM-signals for easy connection to the flight control signals via standard Servo patch cables.

Figure 56 - Control Mode B) PPM Control and I2C Read



ESC Telemetry Translation Interface

Dependent on Control mode, the Translation to a specific Telemetry Protocoll (HoTT/Duplex/Serial) is done either by the Flight Control (Mikrokopter) or the HERKULES III Telemetry Interface Board. For details see the protocol description.

9.2 I2C Address Range

Each ESC on the HERKULES III board is individual and stands-alone. So each ESC has to be addressed separately and the diagnosis information is also separately available for each ESC.

Each ESC has to be programmed with a unique address. There are maximum of 16 motors addressable.

The 8-bit address-byte is built of a 7-bit I2C-address and one read/write (RW) bit.

Write addresses start at 0x52 (RW-Bit=0) and the read addresses start at 0x53 (RW-Bit=1).

Table 11 - I2C WRITE Address Range

			8-Bit I2C-Adress-Byte								Addr
Address-Range			7-Bit Address								[hex] (incl.
ESC	[dec]	[hex]	A6	A5	A4	A3	A2	A1	A0	RW	RW-BIT)
M1	41	0x29	0	1	0	1	0	0	1	0	0x52
M2	42	0x2A	0	1	0	1	0	1	0	0	0x54
M3	43	0x2B	0	1	0	1	0	1	1	0	0x56
M4	44	0x2C	0	1	0	1	1	0	0	0	0x58
M5	45	0x2D	0	1	0	1	1	0	1	0	0x5A
M6	46	0x2E	0	1	0	1	1	1	0	0	0x5C
M7	47	0x2F	0	1	0	1	1	1	1	0	0x5E
M8	48	0x30	0	1	1	0	0	0	0	0	0x60
M9	49	0x31	0	1	1	0	0	0	1	0	0x62
M10	50	0x32	0	1	1	0	0	1	0	0	0x64
M11	51	0x33	0	1	1	0	0	1	1	0	0x66
M12	52	0x34	0	1	1	0	1	0	0	0	0x68
M13	53	0x35	0	1	1	0	1	0	1	0	0x6A
M14	54	0x36	0	1	1	0	1	1	0	0	0x6C
M15	55	0x37	0	1	1	0	1	1	1	0	0x6E
M16	56	0x38	0	1	1	1	0	0	0	0	0x70

Table 12 - I2C READ Address Range

			8-Bit I2C-Adress-Byte								Addr
Address-Range			7-Bit Address								[hex] (incl.
ESC	[dec]	[hex]	A6	A5	A4	A3	A2	A1	A0	RW	RW-BIT)
M1	41	0x29	0	1	0	1	0	0	1	1	0x53
M2	42	0x2A	0	1	0	1	0	1	0	1	0x55
M3	43	0x2B	0	1	0	1	0	1	1	1	0x57
M4	44	0x2C	0	1	0	1	1	0	0	1	0x59
M5	45	0x2D	0	1	0	1	1	0	1	1	0x5B
M6	46	0x2E	0	1	0	1	1	1	0	1	0x5D
M7	47	0x2F	0	1	0	1	1	1	1	1	0x5F
M8	48	0x30	0	1	1	0	0	0	0	1	0x61
M9	49	0x31	0	1	1	0	0	0	1	1	0x63
M10	50	0x32	0	1	1	0	0	1	0	1	0x65
M11	51	0x33	0	1	1	0	0	1	1	1	0x67
M12	52	0x34	0	1	1	0	1	0	0	1	0x69
M13	53	0x35	0	1	1	0	1	0	1	1	0x6B
M14	54	0x36	0	1	1	0	1	1	0	1	0x6D
M15	55	0x37	0	1	1	0	1	1	1	1	0x6F
M16	56	0x38	0	1	1	1	0	0	0	1	0x71

9.3 I2C Communication Sequence and Timing

Each I2C-Communication causes a certain interrupt latency in the motor controllers. And excessively high communication rate could disturb the motor commutation calculation and therefore the communication time should be limited a reasonable update rate. Be aware that an excessively high I2C communication rate could influence the motor timing and commutation stability!

The telemetry readout frequency should be set to a **maximum of 10Hz** in order to minimize the influence of communication load on the processor.

A good practice is to split the communication into frames like this:

Figure 57 - I2C communication Frame

WRITE																				READ					
S	M1			S	M2			S	M3			S	M4			S	Mx (5..12)			S	Mx				P
	AD	WR			AD	WR			AD	WR			AD	WR			AD	WR			AD	RD			
	82	HB	LB		84	HB	LB		86	HB	LB		88	HB	LB		xx	HB	LB		83	B1	B2	B3	
	A	A	A		A	A	A		A	A	A		A	A	A		A	A	A		A				

One frame should contain all setpoints for the motors (M1…16) but only one read parameter for one motor the diagnosis The address of the read parameter is increased after each communication frame and lowers therefore the update rate of the I2C communication interface.

9.4 I2C-Mode : Setpoint Write and Data Read via I2C

The motor setpoints and the read-back data is transferred via I2C-BUS. The protocol is compatible with the standard Mikrokopter Flight Control protocol. For more details please see www.mikrokopter.de .

Main Features of the I2C protocol

- setpoint write resolution 8 bit (up to 12 bit reserved, but not implemented yet)
- telemetry read of motor current, motor status, ESC temperature and battery voltage

9.4.1 Setpoint Write via I2C in I2C-Mode

In order to write the motor setpoint to the ESCs the following sequence has to be sent:

Figure 58 - I2C Send Sequence

BYTE 0 = Address									BYTE 1 = Setpoint High									BYTE 2 = Setpoint Low							
7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
A6	A5	A4	A3	A2	A1	A0	0	=>	S7	S6	S5	S4	S3	S2	S1	S0	=>	-	-	-	-	S3	S2	S1	S0
I2C-Addr							W		Setpoint									Setpoint LB (optional)							

I2C-Address M1-16			Motor Setpoint		Motor Setpoint Low
M1:0x52 (read) M16: 0x70 (read)	WRITE = 0	=>	Range: 0…250d Resolution: 8 Bit Dutycycle: 0…94% Offset: 12 digit	=>	Optional Lowbyte Not used in Herkules III Only detection of new I2C Protocoll (MK)

Specification Limits of I2C Protocoll

(all parameters are valid for 25°C ambient temperature, otherwise noted)

No.	Description	Symbol	min	typ	max	Unit	Comment
1	I2C bus speed	SPEED_I2C			100	kHz	s
2	Setpoint Resolution	RES_SP_I2C		235		digit	depending on F_PWM
3	Setpoint update rate	T_PER_I2C	2.5		200	msec	
4	Setpoint timeout detection threshold	T_TO_I2C		500		msec	after T_TO_I2C has elapsed, the motor stops
5	Setpoint update frequency	F_PER_I2C	2		450	Hz	
6	Motor OFF Detection Time	VAL_OFF_I2C		9		digit	If setpoint < VAL_HI_OFF, Motor stops
7	Motor START Detection Threshold	VAL_START_I2C		10		digit	If setpoint > VAL_HI_OFF, Motor starts
8	Motor 100% Threshold	VAL_100_I2C		245		digit	If setpoint > VAL_HI_100, Motor Dutycyle 100%

Table 13 - I2C Control Interface Specification

The **1st byte (BYTE0)** is the address-byte with the R/W BIT set to zero (write).

The **2nd byte (BYTE1)** is the motor setpoint high byte. It can have any value between 0 and 255d.

The **3rd byte (BYTE2)** is the motor setpoint low byte. This byte is reserved for future usage. Currently this byte has no function due to setpoint resolution is max. 8 bit. There is no need to send the Byte2!

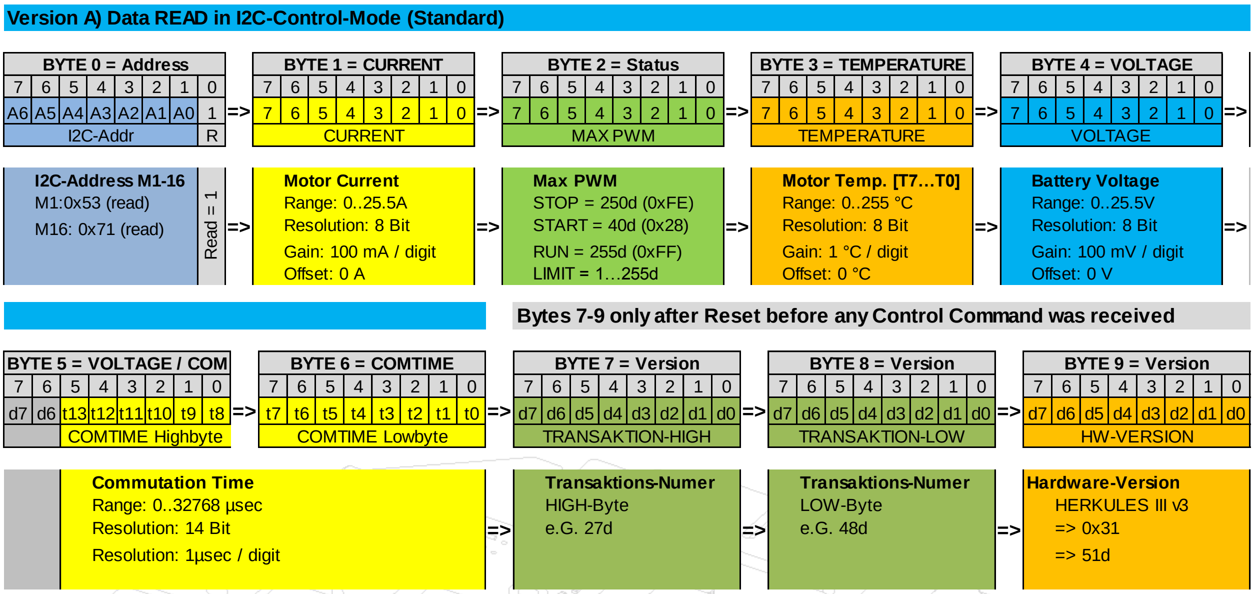
Communication behavior

- The Setpoint is calculated by Setpoint = Byte1 x 256 + Byte2
- The Motor stops at a setpoint below VAL_OFF_I2C
- The Motor starts running at value higher than VAL_START_I2C.
- The Max setpoint is VAL_100_I2C. If a value higher than this is sent, the ESC goes from PWM mode to full power without current modulation.

9.4.2 Diagnosis Read via I2C in I2C-Mode

To Read Telemetry Data from the ESCs the following sequence has to be sent:

Figure 59 - I2C Read Sequence



BYTE0 is the read address with RW-Bit = 1. It’s the same 7-bit address as in write-mode but with RW-bit set different.

BYTE1 reads the actual average current from the addressed ESC. HERKULES III measures internally the current with a resolution of 10bit but due to the Mikrokopter protocol limitation the read-back range is limited to 25.5A.

BYTE2 reads the MAX-PWM value from the ESC. In Idle/Stop mode, the Value is 250d (0xFE) and signals the new I2C-Protocoll for BL2.0 with full telemetry read-back.

During starting of the motor, the byte changes to 40d (0x28) and signals a running starting sequence.

After starting, when the motor is running the value is 255d.

BYTE3 reads the ESC temperature value. HERKULES III measures internally the temperature between -40 and +150°C but due to the Mikrokopter protocol limitation the read-back range is limited to positive values between 0°C and 255°C.

BYTE4 is the battery voltage measured at the ESC. HERKULES III measures internally from 0V to +55V but due to Mikrokopter protocol limitation the read-back value is limited to +25.5V.

BYTE 5 and 6 is the Motor Commutation Time. The Motors field rotation frequency can be calculated by:
FIELD_FREQ[RPM] = 10.000.000 / COMTIME [µs].
The Motor Shaft Frequency can be calculated by:
MOTOR_RPM [1/min] = FIELD_FREQ / MOTOR_POLES.

Byte 7 and 8 are the software Version Number Highbyte and Lowbyte

Byte 9 is the Herkules Hardware-ID

9.5 PPM-Mode : Setpoint Write via PPM and Data Read via I2C

In PPM Mode, the setpoint is written to each ESC with a separate PWM wire and the data information is read via I2C bus. The benefit of this mode is the control wire redundancy and independency from each other. If one controller fails or pulls the I2C-bus low or high only the read back communication is disturbed but the setpoint controlling is working in parallel for each controller.

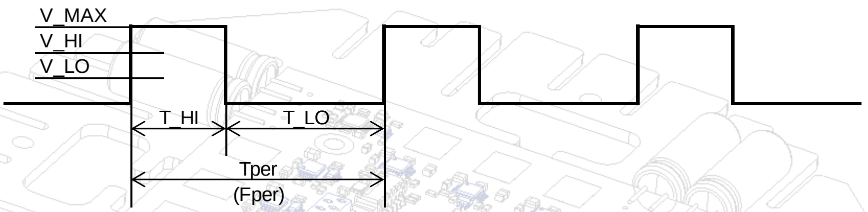
9.5.1 Setpoint Write via PPM Signal

The PPM signal is a widely used RC-standard. The motor setpoint is proportional to the high-time of the PPM control signal. The repetition rate of the signal can be up to 400Hz. This enables a very fast setpoint update rate and supports the flight control for very accurate and stable attitude regulation.

Write:

- Setpoint Write with standard PPM Signal (RC-standard) with 8 bit resolution
- Fast update rate of up to 400 Hz

Figure 60 - PPM Control Signal



(all parameters are valid for 25°C ambient temperature, otherwise noted)

No.	Description	Symbol	min	typ	max	Unit	Comment
1	Setpoint Resolution in PPM Control Mode	RES_SP_PPM			256	digit	depending on F_PWM
2	Setpoint Update Period	T_PER_PPM	2.5		200	msec	
3	Setpoint update frequency	F_PER_PPM	2		450	Hz	
4	Setpoint Timout detection	T_TO_PPM		500		msec	after T_TO_PPM has elapsed, the motor stops
5	Motor OFF Detection Time	T_HI_OFF_PPM		1080		µsec	If T_HI < T_HI_OFF, Motor Stops
6	Motor START Detection Threshold	T_HI_START_PPM		1090		µsec	If T_HI > T_HI_START, Motor Starts
7	Motor 100% Threshold	T_HI_100_PPM		1900		µsec	If T_HI > T_HI_100, Motor Dutycyle 100%

Table 14 - PPM Control Interface Specification

9.5.2 Telemetry Read via I2C in PPM-Mode

The readback format from each ESC is similar to the one above.

Figure 61 - Data Read in PPM Control Mode via I2C

Data READ in PPM-Control Mode (Version 2) => Standard (since 01/2014)

BYTE 0 = Address							
7	6	5	4	3	2	1	0
A6	A5	A4	A3	A2	A1	A0	1
I2C-Addr							R

=>

BYTE 1 = Status / CURR							
7	6	5	4	3	2	1	0
-	F	R	TO	OT	OC	C9	C8
STATUS						CURR	

=>

BYTE 2 = CURRENT							
7	6	5	4	3	2	1	0
C7	C6	C5	C4	C3	C2	C1	C0
CURRENT							

=>

BYTE 3 = TEMPERATURE							
7	6	5	4	3	2	1	0
T7	T6	T5	T4	T3	T2	T1	T0
TEMPERATURE							

I2C-Address M1-16
M1:0x53 (read)
M16: 0x71 (read)

Read = 1

=>

Motor Status
F: Fail, R: Run
TO: Timeout
OT: Overtemp
OC: Overcurrent

Motor Current [C9…C0]
Range: 0..250 A
Resolution: 10 Bit
Gain: 244.4 mA/digit
Offset: 0 A

=>

Motor Temp. [T7…T0]
Range: -40..215 °C
Resolution: 8 Bit
Gain: 1 °C / digit
Offset: 40 °C

BYTE 4 = Voltage							
7	6	5	4	3	2	1	0
V9	V8	V7	V6	V5	V4	V3	V2
VOLTAGE							

=>

BYTE 5 = VOLTAGE / COM							
7	6	5	4	3	2	1	0
V1	V0	t13	t12	t11	t10	t9	t8
VOLT		COMTIME Highbyte					

=>

BYTE 6 = COMTIME							
7	6	5	4	3	2	1	0
t7	t6	t5	t4	t3	t2	t1	t0
COMTIME Lowbyte							

=>

Battery Voltage [V9…V0]
Range: 0..55 V
Resolution: 10 Bit
Gain:53.76 mV / digit
Offset: 0 V

Commutation Time
Range: 0..32768 µsec
Resolution: 14 Bit
Resolution: 2µsec / digit

=>

Bytes 7-9 only after Reset before any Control Command was received

BYTE 7 = SW-Version							
7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0
TRANSAKTION-HIGH							

=>

BYTE 8 = SW-Version							
7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0
TRANSAKTION-LOW							

=>

BYTE 9 = HW-Version							
7	6	5	4	3	2	1	0
d3	d2	d1	d0	d3	d2	d1	d0
Main				Revision			

Transaktions-Numer
HIGH-Byte
e.G. 27d

=>

Transaktions-Numer
LOW-Byte
e.G. 48d

=>

Hardware-Version
HERKULES III v3
=> 0x31
=> 51d

BYTE0 is the read address with RW-Bit = 1. Same 7-bit address as write-mode but with RW-bit set different.

BYTE1 contains the Motor Status flags + the two MSBs of the motor current. The available motor flags are

- R: Run : set to high if motor is starting, set to low if motor has stopped
- F: General Failure : set to high if any failure is present
- TO: Timeout : set to high if no setpoint update is detected and timeout period has elapsed
- OT: Over Temperature : set to high if ESC threshold is higher than T_OT_LIMIT
- OC: Over Current : set to high if Motor current is higher than I_OC_DIS

BYTE2 reads the actual average current from the addressed ESC. HERKULES III measures internally the current with a resolution of 10bit.

BYTE3 reads the ESC temperature value. HERKULES measures internally the temperature between -40 and +215°C.

BYTE4 and 5 is the battery voltage measured at the ESC. HERKULES III measures internally from 0V to +55V.

Byte 7 and 8 are the software Version Number Highbyte and Lowbyte

Byte 9 is the Herkules Hardware-ID

Contact Information

HERKULES III is developed, manufactured and distributed by the following partners.

Manufacturing and Development

HerSi Electronic Development GmbH & Co. KG
Roentgenstr. 1
93055 Regensburg
Germany
web: www.hersi.de

Development Partners

DeltaBytes Elektronik
Andreas Baier
Himmelreich 12
96361 Stenbach am Wald
Germany
web: www.andreasbaier.de

Distributors and Technical Support

Globe Flight GmbH
Borsigstrasse 9
93092 Barbing
Germany
web: www.globe-flight.de

Kopterworx Aerial Solutions
Earthline Matjaz Kranjcevic s.p.
Grajska pot 14a
8373 Leskovec pri Krskem
Slovenia
web: www.kopterworx.com