PC Command	START	COMMAND	ADRESS		PARAM_LEN	PARAM	XMODEM CRC16	
	Byte	Byte	High-Byte	Low-Byte	Byte	PARAM_LEN x Byte	High-Byte	Low-Byte
	0x2F	0x30 - 0x3A	0x00 - 0xFF	0x00 - 0xFF	1-255 // 0=256	0x00 - 0xFF	0x00 - 0xFF	0x00 - 0xFF

INTERFACE Answer	START	COMMAND	ADRESS		PARAM_LEN	PARAM	ACK	XMODEM CRC16	
	Byte	Byte	High-Byte	Low-Byte	Byte	PARAM_LEN x Byte	Byte	High-Byte	Low-Byte
	0x2E	0x30 - 0x3A	0x00 - 0xFF	0x00 - 0xFF	1-255 // 0=256	0x00 - 0xFF	0x00 - 0x0F	0x00 - 0xFF	0x00 - 0xFF

Field name	**Min Value**	**Max Value**	**Description**
START	0x2E = 46 = '.'	0x2F = 47 = '/'	Escape character: (PC) must send **0x2F** / Interface must send **0x2E** in response
COMMAND	0x30 = 48 = '0'	0x3C = 58 = '<'	All chars are printable to better control with portmonitor
ADRESS	0x0000 = 0	0xFFFF = 65535	(Max for F330 is 0x1DFF). Only **Valid if Device Read or Write** (Big Endian) For all other commands the adress bytes will be ignored (better set to 0)
PARAM_LEN(n)	0x01 = 1	0x00 = 256	Length-Field for the following PARAM Block. To handle the whole Byte range from 0..256 a trick is used The minimum Value is 1 so there has to be allways 1 Byte in PARAM Values from 1..255 count what they say, but 0 means 256.
PARAM	0x00 = 0	0xFF = 255	A Data-block of PARAM_LEN count of Bytes. Only Valid if Device Read or Write. For other commands PARAM bytes will be ignored (so for other command set PARAM_LEN=1 and the single PARAM byte = 0)
ACK	0x00 = 'OK'	0x0F	Interface Response Field with OK or Error Code. Only send by Interface. Error Codes range is from 0x01 to 0x0F
XMODEM CRC16	0x0000 = 0	0xFFFF = 65535	As used in crc16.h of AVR-Gcc: Polynomial: x^16 + x^12 + x^5 + 1 (0x1021) Initial value: 0x0 This is the CRC used by the Xmodem-CRC protocol. All previous bytes are calculated from START to PARAM

Rem: The last 2 byte in sequence = CRC. Hex-Values are show when they are allways equal.

Command Table	HexVal	DecVal	Ascii	Meaning
cmd_InterfaceTestAlive	**30**	**48**	**0**	May be send by Master to check: Interface still present and responding ?
PC sends:	2F 30 00 00 01 00 CF D4			Data: 0
Interface responds	2E 30 00 00 01 00 00 44 C2			Data: 0
				Rem: *BLHeli-Setup sends this command 1 time/sec to check the interface connection*
cmd_ProtocolGetVersion	**31**	**49**	**1**	Retrieve Interface Protocoll version
PC sends:	2F 31 00 00 01 00 65 85			Data: 0
Interface responds	2E 31 00 00 01 bb 00 CRC			Data: bb = 1 Byte with interface protocol version number
				Rem: *The version number of this command table and handling*
cmd_InterfaceGetVersionStr	**32**	**50**	**2**	Retrieve Interface version as text.
PC sends:	2F 32 00 00 01 00 8B 57			Data: 0
Interface responds	2E 32 00 00 nn abc... 00 CRC			Data: nn = number of chars; abc... = chars with interface version text
				Rem: *Only the name of the interfaces (w/o the Rev. num)*
cmd_InterfaceGetVersion	**33**	**51**	**3**	Retrieve Interface version as byte value.
PC sends:	2F 33 00 00 01 00 21 06			Data: 0
Interface responds	2E 33 00 00 02 bb bb 00 CRC			Data: bb = 2 Byte with Interface version number I.Byte= 12.3 II.Byte= .4.5
				Rem: *Rev. Number of the interface*
cmd_InterfaceExit	**34**	**52**	**4**	Exit Interface PC Mode Resets the BESC's and restarts Boxes Display Mode
PC sends:	2F 34 00 00 01 00 46 D2			Data: 0 .
Interface responds	2E 34 00 00 01 00 00 42 63			Data: 0
				Rem: *Only valid for Dual mode Interfaces (Box with LCD), otherwise simply return OK*
cmd_DeviceReset	**35**	**53**	**5**	C2 Command: Reset connected Target (BESC)
PC sends:	2F 35 00 00 01 0n CRC			Data: 00-07 select the BESC channel '**V2** / **V1** always 00
Interface responds	2E 35 00 00 01 0n 00 CRC			Data: 00-07 ***V2** / **V1** always 00
				Rem: *Used as a single command will restart the BESC*
cmd_DeviceGetID	**36**	**54**	**6**	C2 Command: Retrieve Target MCU ID as byte value.
PC sends:	2F 36 00 00 01 00 02 51			Data: 0
Interface responds	2E 36 00 00 01 bb 00 CRC			Data: bb = 1 Byte with MCU ID = 0x0A for C8051F330 / ID = 0x08 for F310
				Rem: *BLHeli-Setup evaluates, if the chip is supported / not the interface*

cmd_DeviceInitFlash	**37**	**55**	**7**	C2 Command: Enable Flash access to Target MCU
PC sends:	2F 37 00 00 01 0n CRC			Data: 00-07 select the BESC channel *__V2__ / **V1** always 00
Interface responds V1..V3	2E 37 00 00 01 bb 00 CRC			Data: **V1 =** always 00 /*__V2__=00-07 /**V3** = Derivative ID
Interface responds V4	2E 37 00 00 03 aa bb cc 00 CRC			Data: aa=DeviceID bb=DerivativeID cc=LineState
				Rem: LineState: bit 0 = C2CK, bit 1 = C2D (0=Low/1= high) should be both high -> 11b

cmd_DeviceEraseAll	**38**	**56**	**8**	C2 Command: Erase whole memory of Target MCU
PC sends:	2F 38 00 00 01 00 CD F9			Data: 0
Interface responds	2E 38 00 00 01 00 00 49 80			Data: 0
				Rem:

cmd_DevicePageErase	**39**	**57**	**9**	C2 Command: Erase one page in memory of Target MCU
PC sends:	2F 39 00 00 01 bb CRC			Data: bb = 1 Byte with the page number
Interface responds	2E 39 00 00 01 bb 00 CRC			Data: bb = 1 Byte with the page number
				Rem:

cmd_DeviceRead	**3A**	**58**	**:**	C2 Command: Read memory of Target MCU
PC sends:	2F 3A hi lo 01 nn CRC			Data: hi lo = start address; nn = number of bytes to read
Interface responds	2E 3A hi lo nn bbb... 00 CRC			Data: hi lo = start address; nn = number of data bytes; bbb... = data bytes
				Rem: *nn = 0 means: read 256 bytes*

cmd_DeviceWrite	**3B**	**59**	**;**	C2 Command: Write to memory of Target MCU
PC sends:	2F 3B hi lo nn bbb... CRC			Data: hi lo = start address; nn = number of data bytes; bbb... = data bytes
Interface responds	2E 3B hi lo 01 00 00 CRC			Data: hi lo = start address
				Rem: nn = 0 means: read 256 bytes
				Rem: Writes are internally verified (Interface reads back after write and compares)

cmd_DeviceC2CK_LOW	**3C**	**60**	**<**	C2 Command: Set C2 clock line C2CK to low
PC sends:	2F 3C 00 00 01 0n CRC			Data: 00-07 select the BESC channel '**V2** / **V1** always 00
Interface responds	2E 3C 00 00 01 0n 00 CRC			Data: 00-07 *__V2__ / **V1** always 00
				Rem: *Not yet implemented in BLHeli-Setup; may help recover wrong flashed BESC*

Errror codes

If a command sequence is send by the master and the interface fails to proceed, it will answer with an Error code.

Interface Error Response	2E cc hi lo 01 00 er CRC	Data: 00 cc = command which failed; hi+lo = address value which failed; er = Error Code

Error codes defined for ACK

ACK_OK	**0x00**	Operation succeeded. No Error.
ACK_I_UNKNOWN_ERROR	**0x01**	Failure in the interface for unknown reason
ACK_I_INVALID_CMD	**0x02**	Interface recognized an unknown command
ACK_I_INVALID_CRC	**0x03**	Interface calculated a different CRC / data transmission form Master failed
ACK_I_VERIFY_ERROR	**0x04**	Interface did a successful write operation over C2, but the read back data did not match
ACK_C2_INVALID_COMMAND	**0x05**	C2 communication failed and the Status was 0x00 instead of 0x0D
ACK_C2_COMMAND_FAILED	**0x06**	C2 communication failed and the Status was 0x02 or 0x03 instead of 0x0D
ACK_C2_UNKNOWN_ERROR	**0x07**	C2 communication failed and the Status was of unknow value instead of 0x0D
ACK_I_INVALID_CHANNEL	**0x08**	Interface recognized: unavailable Port is adressed in Multi BESC Mode *** V2 only**
ACK_C2_GENERAL_ERROR	**0xFF**	C2 communication failed for unknown reason

History:	V1.0	Intial release
	V2.0	Added Support für Multiple BESC Handling
		Interface Name starting with "m..." indicates: this is a multiple BESC Interface
		The following Commands got a new parameter 0-7 which selects the BESC Channel 1..8
		Once selected, the Channel will remain activ till another one is selcted.
		cmd_DeviceC2CK_LOW
		cmd_DeviceReset
	.	cmd_DeviceInitFlash
		To enable Interfaces with less than 8 channels ACK_I_INVALID_CHANNEL is added
		Interface will respond if a Channel higher than supported is addressed.
	V3.0	cmd_DeviceInitFlash returns the Silabs device Derivative ID
	V4.0	cmd_DeviceInitFlash combines cmd_DeviceReset + cmd_DeviceGetID + cmd_DeviceInitFlash
		and returns DeviceID, DerivativeID and LineState for C2D and C2CK wires
	V5.0	cmd_InterfaceGetVersion now returns 2 bytes.
		(first byte = 2 digit main+ 1.digit sub / second byte 3. and 4. digit sub)
		Length of cmd_InterfaceGetVersionStr is no longer fixed to 12 but variable length