

Definition Kommando (Kommando-Interpreter-Ebene)		
Byte		
0	CMD	Kommando (Definition entsprechend GuiPlusCommandEvent.h)
1	Parameter/Data	Parameter (Daten-Bytes: - Anzahl und Reihenfolge sind Kommando-spezifisch, - Anzahl kann auch Null sein, - "X" Deleztchmer -> Kodierung ist im Quellcode definiert (TTR-Modul oder Fremd-Modul)
...		

Definition Frame und Kodierung (UART-Übertragungs-Ebene)		
Byte		Kodierung
0	STX	Frame Start (Definition entsprechend GuiPlusCommandEvent.h)
1	CMD	Kodierung: ohne
2	Parameter/Data Byte 1	Kodierung: ohne
3	Parameter/Data Byte 1	Kodierung: Intel-Hex-Format
4	Parameter/Data Byte 2	Kodierung: Intel-Hex-Format
5	Parameter/Data Byte 2	
...	...	
...	...	
...	Parameter/Data Byte X	Kodierung: Intel-Hex-Format
...	Parameter/Data Byte X	
N	Parameter/Data Byte X	Kodierung: Intel-Hex-Format
N+1	ETX	Kodierung: ohne

Cmd	Kommando	Kodierung	Beschreibung	zugehörige API-Funktion	Legacy	Offen / Diskussion	
Offene Kommandos (werden unbedingt ausgeführt)		Parameter (aller Kommandos): unsigned, wenn nicht anders angegeben					
System							
0	GET_API_VERSION_REQ	0x26	Firmware-Version abfragen	keine explizite API-Funktion - ist im Kommando-Handler implementiert	Y'	Version des API-FW-Teils (= Teleonic-Quellecode)	
0	GET_API_VERSION_CONF	0x27	Version: Major Minor Patch Build		Y''		
1	Major_H						
2	Major_L						
3	Minor_H						
4	Minor_L						
5	Patch_H						
6	Patch_L						
7	Build_H						
8	Build_L						
0	GET_APPL_VERSION_REQ	0x30	Firmware-Version abfragen	keine explizite API-Funktion - ist im Kommando-Handler implementiert		Version des Applikation-FW-Teils (= BioSign-Quellecode)	
0	GET_APPL_VERSION_CONF	0x31	Version: Major Minor Patch Build				
1	Major_H						
2	Major_L						
3	Minor_H						
4	Minor_L						
5	Patch_H						
6	Patch_L						
7	Build_H						
8	Build_L						
Schnittstellen-Kontrolle (USB/BT)							
0	GIU_REGISTER_REQ	0x32	verbindet den Giu exklusiv mit dem anfordernden Kommunikations-Kanal 0: Retrigger/Timeout Mode wird nicht aktiviert → kein automatisches RELEASE nach Timeout (für Entwicklung/Debugging) 1: Retrigger/Timeout Mode wird aktiviert	keine explizite API-Funktion - ist im Kommando-Handler implementiert			
0	GIU_REGISTER_CONF	0x33	State ConnectState	0: QP_API_SUCCESS X: QP_API_ERROR (ApIReturnValues.h) 0: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal			
0	GIU_RELEASE_REQ	0x34	der verbundene Kommunikations-Kanal löst die exklusive Verbindung mit dem Giu	keine explizite API-Funktion - Kommando-Handler implementiert			
0	GIU_RELEASE_CONF	0x35	State ConnectState	0: QP_API_SUCCESS X: QP_API_ERROR (ApIReturnValues.h) 0: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal		weitere Release-Events: - automatisches Release durch Timeout: erfordert periodisches Triggern des verbundenen Kanals (durch noch zu def. Kommando oder vorh. Kommandos) -> Timeout-Zeit definieren - automatisches Release durch FW-Events: BT disconnected (TODO: ist. Zustand notwendig), USB disconnected - denkbar wäre auch Nutzer-initiiertes Release per Tastendruck (ungetriggert) - mehrstufige Verfahren oder nur alternativ implementieren? (zuerst mindestens ein weiteres Release-Event ist notwendig) -> CCL_Besprechungsprotokoll_2016-11-05_Tekko_Schnittstellen-Kontrolle_FW-Update.pdf	
1	State						
2	ConnectState						
0	GIU_COMMAND_REJECT_CONF	0x37	State ConnectState CmdState Command	keine explizite API-Funktion - Kommando-Handler implementiert X: QP_API_ERROR (ApIReturnValues.h) 0: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal 0: QP_API_SUCCESS X: QP_API_ERROR_CMD_UNKNOWN *** Kodierung modifiziert *** (ApIReturnValues.h) Kommando, welches per REQ an den Giu gesendet wurde			
geschützte Kommandos (werden nur nach Register ausgeführt)							
0	GIU_REGISTER_RETRIGGER_REQ	0x38	Funktion zum Retriggen von GIU_REGISTER_REQ -> muss periodisch aufgerufen werden, sonst nach Timeout automatisches Release	uint8_t GIUAPI_retriggerRegisterTimeout (uint16_t timeout) uint16_t GIUAPI_getRemainingTimeoutTime(void)			
0	GIU_REGISTER_RETRIGGER_CONF	0x39	State ConnectState RemainingTimeRaw	0: QP_API_SUCCESS 0: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal 0xFF: RemainingTime [s] > 2.5 0x00: (bFE: RemainingTime [s] + RemainingTimeRaw * 8 / 1024			
Power / Akku (Akku-Ladung)		wenn möglich, sollten Kommandos und zugehörige API-Funktionen bezüglich Call- und Return-Parameter identisch sein					
0	POWER_CONTROL_REQ	0x5A	Spannung ein-/aus-schalten, Auswahl der zu schaltenden Spannung per VoltageSelect State VoltageSelect	uint8_t GIUAPI_analogPowerControl(uint8_t state); uint8_t GIUAPI_ledPowerControl(uint8_t state);	X'		
1	State		0: aus 1: ein				
2	VoltageSelect		X: QP_API_ANALOG_SUPPLY_VOLTAGE X: QP_API_LED_SUPPLY_VOLTAGE				
0	POWER_CONTROL_CONF	0x5B	VoltageSelect	X: QP_API_UNDEFINED_VOLTAGE X: QP_API_ANALOG_SUPPLY_VOLTAGE X: QP_API_LED_SUPPLY_VOLTAGE	Y'		
0	GET_VOLTAGE_REQ	0x5C	Spannung(en) abfragen, Auswahl der zu messenden Spannung per VoltageSelect VoltageSelect	uint16_t GIUAPI_getAccuVoltage(void); uint16_t GIUAPI_getUsbVoltage(void); uint16_t GIUAPI_getDigiVoltage(void);	Y'		
1	VoltageSelect		X: QP_API_ACCU_VOLTAGE X: QP_API_USB_VOLTAGE X: QP_API_DIG_SUPPLY_VOLTAGE				
0	GET_VOLTAGE_CONF	0x5D	VoltageSelect	X: QP_API_UNDEFINED_VOLTAGE X: QP_API_ACCU_VOLTAGE X: QP_API_USB_VOLTAGE X: QP_API_DIG_SUPPLY_VOLTAGE	Y'		
2	VoltageRaw_H		Da 2x16-Bits liefert nur der ADC-Schnitt Wert "VoltageRaw" Da Umrechnung in den Spannungspegel muss in der Software erfolgen Da abhängig von LP-Revision und FW-Version skaliert				
3	VoltageRaw_L		ACCU_VOLTAGE, USB_VOLTAGE, DIG_SUPPLY_VOLTAGE LP-Revision >= 1.2.0.0.0, FW-Version >= 0.0.0.23: Voltage[V] = VoltageRaw * 5.0 / 4096				
0	GET_CHARGE_STATE_REQ	0x5A	Ladestatus abfragen	uint8_t GIUAPI_getChargeState(void)			
0	GET_CHARGE_STATE_CONF	0x5B	ChargeState	Version 1 (implementiert): 0x0: PREPARED_FOR_CHRG charging (Anzeige im HWFW-Testprogramm: PRECH) 0x1: no battery charge im HWFW-Testprogramm: NOBAT 0x2: maintenance - full (Anzeige im HWFW-Testprogramm: MFLU) 0x3: fault (Anzeige im HWFW-Testprogramm: FAULT) Version 2 (implementiert): 0x0: no battery (Anzeige im HWFW-Testprogramm: NOBAT) 0x1: not charging (Anzeige im HWFW-Testprogramm: NOTCH) 0x2: PREPARED charging (Anzeige im HWFW-Testprogramm: PRECH) 0x3: TOP-CHG charging (Anzeige im HWFW-Testprogramm: TPOCH) 0x4: maintenance (Anzeige im HWFW-Testprogramm: MAINT) 0x5: fault (Anzeige im HWFW-Testprogramm: FAULT)			
0	IRED_EXT_CONTROL_REQ	0x06	externe IRED ein-/aus-schalten State	uint8_t IQUAPI_iredExtControl(uint8_t state);	E'		
1	State		0: aus 1: ein				
0	IRED_EXT_CONTROL_CONF	0x07					
0	IRED_INT_CONTROL_REQ	0x08	interne IRED ein-/aus-schalten State	uint8_t IQUAPI_iredIntControl(uint8_t state);	T'		
1	State		0: aus 1: ein				
0	IRED_INT_CONTROL_CONF	0x09					
0	DIM_LED_REQ	0x0A	LED dimmen ledId DimValue	uint8_t IQUAPI_dimLed(uint8_t col, uint8_t pwm);	G' B' R'	drei separate Kommandos zusammengefasst	
1	ledId		G: 'G', B: 'B', R: 'R'				
2	DimValue		(unsigned) 0 ... 65, 0: aus, 65: max				
0	DIM_LED_CONF	0x0B			Y' Y'		
0	LEDBAR_CONTROL_REQ	0x0C	LEDs im Balken steuern ledVector	uint8_t IQUAPI_ledbarControl(uint8_t ledVector);	D'		
1	ledVector		ledVectorBit0: left LED ledVectorBit1: right LED ledVectorBit2: left LED 0: LED off 1: LED on				
0	LEDBAR_CONTROL_CONF	0x0D					
			LEDs als Lauflicht steuern direction LEFT: 0x00 direction RIGHT: 0x01	uint8_t IQUAPI_ledbarControlRow(uint8_t direction)	neu		
0	EARCLIP_STATE_REQ	0x10	Status externer Ohrclip abfragen	uint8_t IQUAPI_earclipState(void);	C'	Status kann auch per GET_EVENTSTATUS_VECTOR_REQ abgefragt werden -> EARCLIP_STATE_REQ_CONF redundant -> entfernen? Da ADC abgefragt werden muss besser eigene Funktion	
0	EARCLIP_STATE_CONF	0x11	State		C'		
1	State		0: nicht angeschlossen 1: angeschlossen				
Flash							
0	FLASH_READ_DATA_REQ	0x12	Daten (32 Bytes) aus Flash lesen SectorNumber_H SectorNumber_L PageNumber	uint8_t IQUAPI_flashReadData(uint16_t sectorNumber, uint8_t pageNumber, uint8_t tData);	Y'		
1	SectorNumber_H		0 ... MAX: theoretisch nutzbarer Wertebereich entsprechend API-Definition, MAX wird bestimmt von Größe FLASH-IC MIN ... MAX: nach hardware Randbereich darstellt Flashbereich-IC-Adress-Bereich -> abhängig von API-Version				
2	SectorNumber_L						
3	PageNumber		PageNumber: 0 ... 127				
0	FLASH_READ_DATA_CONF	0x13	Data_1 Data_2		Y'		
1	Data_1						
2	Data_2						

1	StatusVector_1		<div>bitweise kodiert: StatusVector_1[0]: externer Pulssensor (Dhrdp), 0: nicht angesteckt, 1: angesteckt StatusVector_1[1]: Akku-Ladung, 0: keine Ladung per USB, 1: Ladung per USB StatusVector_1[2]: USB-Connect, 0: nicht verbunden, 1: verbunden StatusVector_1[3]: BT-Connect, 0: nicht verbunden, 1: verbunden</div>
2	EventVector_1		<div>bitweise kodiert: Alle Event-Vector-Bits werden nach dem GET automatisch gelöscht EventVector_1[0]: 0: keine Accel-Sensor-Events, 1: Accel-Sensor-Events vorhanden</div>
Filesystem			
→ externe Dokumentation des Filesystems			
Filetransfer			
0	GET_DIR_FND_FIRST_FILE_REQ	0x40	File-Liste anfordern, erstes Element (= erster Date-Name)
0	GET_DIR_FND_FIRST_FILE_CONF	0x41	
1	Filename ASCII byte 1		max. Länge 16 Byte (inklusive Extension-Trenner ";")
2	Filename ASCII byte 2		
...			
16	Filename ASCII byte 16		falls Filenamenslänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen
17	FilenameState		0: File vorhanden, Filename Bytes sind gültig → nächster Call GET_DIR_FND_NEXT_FILE_REQ 1: kein (weiteres) File vorhanden, Filename Bytes sind ungültig → kein nächster Call GET_DIR_FND_NEXT_FILE_REQ notwendig 2: allgemeiner Fehler, Filename Bytes sind ungültig
0	GET_DIR_FND_NEXT_FILE_REQ	0x50	File-Liste anfordern, weitere Elemente (= nächster Dateiname)
0	GET_DIR_FND_NEXT_FILE_CONF	0x51	
1	Filename ASCII byte 1		max. Länge 16 Byte (inklusive Extension-Trenner ";")
2	Filename ASCII byte 2		
...			
16	Filename ASCII byte 16		falls Filenamenslänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen
17	FilenameState		0: File vorhanden, Filename Bytes sind gültig → nächster Call GET_DIR_FND_NEXT_FILE_REQ 1: kein (weiteres) File vorhanden, Filename Bytes sind ungültig → kein nächster Call GET_DIR_FND_NEXT_FILE_REQ notwendig 2: allgemeiner Fehler, Filename Bytes sind ungültig
0	CREATE_DIR_CONTENT_FILE_REQ	0x6E	Generiert die Datei "DirContent.txt" im Filesystem mit aktuellem Directory-Infat, überschreibt vorhandige "DirContent.txt"
0	CREATE_DIR_CONTENT_FILE_CONF	0x6F	
0	DEL_FILE_REQ	0x42	File löschen
1	Filename ASCII byte 1		max. Länge 16 Byte (inklusive Extension-Trenner ";")
2	Filename ASCII byte 2		
...			
16	Filename ASCII byte 16		falls Filenamenslänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen
0	DEL_FILE_CONF	0x43	signed , Return/Error-Code entsprechend FS-Error-Code: FS_ERROR_XXXXX (FS-h)
1	State		
0	READ_FILE_SELECT_REQ	0x44	File zum Lesen auswählen/öffnen
1	Filename ASCII byte 1		max. Länge 16 Byte (inklusive Extension-Trenner ";")
2	Filename ASCII byte 2		
...			
16	Filename ASCII byte 16		falls Filenamenslänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen
17	Action		READ_FILE_SELECT_REQ_ACTION_OPEN: Read-File auf Qu-Seite zum lesen öffnen, READ_FILE_SELECT_REQ_ACTION_CLOSE: das geöffnete Read-File auf Qu-Seite schließen (Filename ist nicht relevant), wenn File-Lesen muss nach dem letzten (READ_FILE_BLOCK_READ_CONF) explizit mit CLOSE abgebrochen werden!
0	READ_FILE_SELECT_CONF	0x45	signed , Return/Error-Code entsprechend FS-Error-Code: FS_ERROR_XXXXX (FS-h)
1	State		
2	Action		Beibehaltung/Änder der Action des READ_FILE_SELECT_REQ
3	FileSize_H		bei Action = READ_FILE_SELECT_REQ_ACTION_OPEN: Größe des geöffneten Files
4	FileSize_3		bei Action = READ_FILE_SELECT_REQ_ACTION_CLOSE: 0
5	FileSize_2		
6	FileSize_L		
0	READ_FILE_BLOCK_REQ	0x52	Block aus Read-File lesen
1	Action		READ_FILE_BLOCK_REQ_ACTION_NEXT: nächsten Block anfordern, READ_FILE_BLOCK_REQ_ACTION_REPEAT: vorherigen Block nochmals anfordern,
0	READ_FILE_BLOCK_CONF	0x53	== 0: Anzahl der enthaltenen Datenbytes (0 = keine (weiteren) Datenbytes vorhanden); < 0: Error-Code
1	State_H		
2	State_L		
3	Checksum_H		
4	Checksum_L		Checksum-Wert ist unbestimmt ! ungültig (Feature ist noch nicht implementiert)!
5	Data_1		
6	Data_2		
...			
...	Data_XX		
0	READ_FILE_BLOCK_ALT_REQ	0x5C	Block aus Read-File lesen, Funktion mit alternativer Kodierung
1	Action		READ_FILE_BLOCK_REQ_ACTION_NEXT: nächsten Block anfordern, READ_FILE_BLOCK_REQ_ACTION_REPEAT: vorherigen Block nochmals anfordern,
0	READ_FILE_BLOCK_ALT_CONF	0x5D	== 0: Anzahl der enthaltenen Datenbytes (0 = keine (weiteren) Datenbytes vorhanden); < 0: Error-Code
1	State_H		
2	State_L		
3	Checksum_H		
4	Checksum_L		
5	Data_1		
6	Data_2		
...			
...	Data_XX		
0	FS_STORAGE_CLEAN_REQ	0x60	Rührt Garbage Collection im Filesystem aus (per FS_STORAGE_CleanOnce() Calls)
1	FsStorageCleanOnceThreshold		1 ... 254: Anzahl der FS_STORAGE_CleanOnce() Calls, welche maximal ausgeführt werden sollen, 255: FS_STORAGE_CleanOnce() Calls werden ausgeführt, bis Flash vollständig "cleaned" ist.
0	FS_STORAGE_CLEAN_CONF	0x61	wird gesendet, wenn Garbage Collection beendet wurde (kann im Bereich von Sekunden bis Minuten liegen)
1	FsStorageCleanOnceCount_H		Anzahl der FS_STORAGE_CleanOnce() Calls, die ausgeführt wurden,
2	FsStorageCleanOnceCount_L		
3	FsErrorCode_H		signed , Return/Error-Code entsprechend FS-Error-Code: FS_ERROR_XXXXX (FS-h)
2	FsErrorCode_L		
0	GET_VOLUME_MEM_INFO_REQ	0x46	Speicher-Status des Filesystems abfragen
0	GET_VOLUME_MEM_INFO_CONF	0x47	Speicher-Status des Filesystems:
1	VolumeSizeKB_H		Datenträger-Laufwerk-Größe [KB]
2	VolumeSizeKB_3		unsigned Integer 32
3	VolumeSizeKB_2		0xFFFFFFFF: Fehler
4	VolumeSizeKB_L		
5	FreeSpaceKB_H		freier Speicher auf Datenträger-Laufwerk [KB]
6	FreeSpaceKB_3		unsigned Integer 32
7	FreeSpaceKB_2		0xFFFFFFFF: Fehler
8	FreeSpaceKB_L		
System Status Vector			
0	GET_SYSTEM_STATUS_REQ	0x48	
0	GET_SYSTEM_STATUS_CONF	0x49	Error-Vektor lesen
1	ErrorVector1		bitweise kodiert: ErrorVector_1[0] = 1: HPXT externe Clock nicht gestartet ErrorVector_1[1] = 1: LPXT externe Clock nicht gestartet
2	ErrorVector2		
3	ErrorVector3		
4	ErrorVector4		
5	FileSystemStatus		
6			
7			
8			
9			
10			
Reserved Events Teletronic			
0	PULSE_MEAS_16_TIMER_RD	0xD7	es sind (min.) 16 Samples in Measurement-Queue verfügbar → auslesen/verarbeiten
1	Data_L		Data[15-0]: mess 6553 6553 6553
			r: measured
			s: 3-Bit SequenceNumber: 1, 2, ..., 6, 7, 6, ..., 2, 1, 2, ..., *** modifiziert ***
			d: 12-Bit ADC-Value
1	Data_H		