Qiu Kommandos DRAFT Stand 2021-04-26

API-FW-Version: 1.0.0.25

U. Hülsberg

Byte-Order: Big-Endian (soweit nicht anders dokumentiert)
- Ausnahme: PULSE_MEAS_16_TIMER_IND, PULSE_MEAS_DATA_16_IND und pw-File: Little-Endian

Basis: 1-Byte-Kommandokodierung + Ausschluss reservierter Zeichen --> dann sind nur minimale SW-Änderungen notwendig--> testen, ob wir damit auskommen,

Definition Kommando (Kommando-Interpreter-Ebene)

Byte

0 CMD Kommando (Definition entsprechend QiuPlusCommandEvent.h)

1 Parameter/Data Parameter-/Daten-Bytes:
- Anzahl und Reihenfolge sind Kommando-spezifisch,
- Anzal kann auch Null sein,
- "X: Bezeichner" --> 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 QiuPlusCommandEvent.h)	Kodierung: ohne					
1	CMD	Kommando (Definition entsprechend QiuPlusCommandEvent.h)	Kodierung: ohne					
2	Parameter/Data Byte 1 High-Nibble-Hex als ASCII-Zeichen	Parameter-/Daten-Bytes: Anzahl und Reihenfolge sind Kommando-spezifisch, Anzal kann auch Null sein,	Kodierung: Intel-Hex-Format					
3	Parameter/Data Byte 1 Low-Nibble-Hex als ASCII-Zeichen							
4	Parameter/Data Byte 2 High-Nibble-Hex als ASCII-Zeichen							
5	Parameter/Data Byte 2 Low-Nibble-Hex als ASCII-Zeichen		Kodierung: Intel-Hex-Format					
	Parameter/Data Byte X High-Nibble-Hex als ASCII-Zeichen		K II					
N	Parameter/Data Byte X Low-Nibble-Hex als ASCII-Zeichen		Kodierung: Intel-Hex-Format					
N+1	ETX	Frame Ende (Definition entsprechend QiuPlusCommandEvent.h)	Kodieruna: ohne					

Kommando Definition

fs Kommando Kodie	erung Beschreibung	zugehörige API-Funktion	Legacy	Kommando und API Tabelle y Offen / Diskussion
offene Kommandos (werden unbedingt ausgeführt)	Parameter (aller Kommandos): unsigned, wenn nicht anders angegeben			
System				
	Firmware-Version abfragen	keine explizite API-Funktion - ist im Kommando-Handler implementiert	'p'	
GET_API_VERSION_CONF 0x2 Major_H Major_L	Version: Major.Minor.Patch.Build		'P'	
Minor_H Minor_L				
Patch_H Patch_L Build_H				
Build_L				
GET_APPL_VERSION_REQ 0x3	Firmware-Version abfragen	keine explizite API-Funktion - ist im Kommando-Handler implementiert		Version des Applikation-FW-Teils (= BioSign-Quellcode)
GET_APPL_VERSION_CONF 0x3 Major_H	Version: Major.Minor.Patch.Build			
Major_L Minor_H Minor_L				
Patch_H Patch_L				
Build_H Build_L				
Cobrittatallan Kantualla (ISD/DT)				
	verbindet den Qiu exklusiv mit dem anfordernden Kommunikations-Kanal	keine explizite API-Funktion - ist im Kommando-Handler implementiert		
RetriggerMode	O: Retrigger/Timeout Mode wird nicht aktiviert> kein automatisches RELEASE nach Timeout (für Entwicklung/Debugging) 1: Retrigger/Timeout Mode wird aktiviert			
QIU_REGISTER_CONF 0x3 State	0: QP_API_SUCCESS X: QP_API_ERROR (ApiReturnValues.h)			
ConnectState	O: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal			
	der verbundene Kommunikations-Kanal löst die exklusive Verbindung mit dem Qiu	keine explizite API-Funktion - Kommando-Handler implementiert		
QIU_RELEASE_CONF 0x3	35			weitere Release-Events: - automatisches Release duch Timeout: erfordert periodisches triggern des verbundenen Kanals (durch noch zu def. Kommando oder vorh. Kommando),> Timeout-Zeit definieren - automatisches Release duch HW-Events: BT disconnected (TODO: ext. Zuarbeit notwendig), USB disconnected - denkbar wäre auch Nutzer-initiiertes Release per Tastendruck (ungünstig), - mehrere/alle Verfahren oder nur alternativ implementieren? (aber mindesten ein weiteres Release-Event ist notwendig!)> COL_Besprechungsprotokoll_2018-11-05_Telko_Schnittstellen-Kontrolle_FW-Update.pdf
State	0: QP_API_SUCCESS X: QP_API_ERROR (ApiReturnValues.h)			mentere/alie vertanten oder nur alternativ implementieren? (aber mindesten ein welteres Release-Event ist notwendig!)> COL_Besprechungsprotokoll_2018-11-05_Telko_Schnittstellen-Kontrolle_FW-Update.pdf
ConnectState	0: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal			
QIU_COMMAND_REJECT_CONF 0x3 State ConnectState	X: QP_API_ERROR (ApiReturnValues.h)	keine explizite API-Funktion - Kommando-Handler implementiert		
ConnectState CmdState	O: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal 2: verbunden mit BT-Kommunikations-Kanal O: QP_API_SUCCESS			
Command	X: QP_API_ERROR_CMD_UNKNOW *** Kodierung modifiziert *** (ApiReturnValues.h) Kommando, welches per REQ an den Qui gesendet wurde			
geschützte Kommandos				
(werden nur nach Register ausgeführt)	Funktion zum Retriggern von QIU REGISTER REQ	uint8 t Oll IAPI retriggerRegisterTimeout (uint16 t timeout)		
QIU_REGISTER_RETRIGGER_CONF 0x3	> muss periodisch aufgerufen werden, sonst nach Timeout automatisches Release	uint8_t QIUAPI_retriggerRegisterTimeout (uint16_t timeout) uint16_t QIUAPI_getRemainingTimeoutTime(void)		
State ConnectState	0: QP_API_SUCCESS 0: Kommunikations-Kanal nicht verbunden 1: verbunden mit USB-Kommunikations-Kanal			
RemainingTimeRaw	2: verbunden mit BT-Kommunikations-Kanal 0xFF: RemainingTime [s] > 2,5 0x00 0xFE: RemainingTime [s] = RemainingTimeRaw * 8 / 1024			
Power / Akku (Akku-Ladung)				wenn möglich, sollten Kommandos und zugehörige API-Funktionen bezüglich Call- und Return-Parameter identisch sein
POWER_CONTROL_REQ 0x5 State	Spannung ein-/aus-schalten, Auswahl der zu schaltenden Spannung per VoltageSelect 0: aus 1: ein	uint8_t QIUAPI_analogPowerControl(uint8_t state); uint8_t QIUAPI_ledPowerControl(uint8_t state);	'A'	
VoltageSelect	X: QP_API_ANALOG_SUPPLY_VOLTAGE X: QP_API_LED_SUPPLY_VOLTAGE			
POWER_CONTROL_CONF 0x5 VoltageSelect	X: QP_API_UNDEFINED_VOLTAGE X: QP_API_ANALOG_SUPPLY_VOLTAGE X: QP_API_LED_SUPPLY_VOLTAGE		'a'	
	X: QP_API_LED_SUPPLY_VOLTAGE			
GET_VOLTAGE_REQ 0x5	Spannung(en) abfragen, Auswahl der zu messenden Spannung per VoltageSelect	uint16_t QIUAPI_getAccuVoltage(void); uint16_t QIUAPI_getUsbVoltage(void); uint16_t QIUAPI_get3V3Voltage(void);	'u'	
VoltageSelect	X: QP_API_ACCU_VOLTAGE X: QP_API_USB_VOLTAGE X: QP_API_DIG_SUPPLY_VOLTAGE	uint16_t QIUAPI_get3V3Voltage(void);		
GET_VOLTAGE_CONF 0x5 VoltageSelect	X: QP_API_UNDEFINED_VOLTAGE	1	'U'	
	X: QP_API_ACCU_VOLTAGE X: QP_API_USB_VOLTAGE X: QP_API_DIG_SUPPLY_VOLTAGE			
VoltageRaw_H VoltageRaw_L	Die API-Funktion liefert nur den ADC-RAW-Wert "VoltageRaw", Die Umrechnung in den Spannungswert muss in der Applikation erfolgen. Die Abhängigkeit von LP-Revision und FW-Version beachten! 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			
	31, 3 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,			
GET_CHARGE_STATE_REQ 0x3	Ladestatus abfragen	uint8_t QIUAPI_getChargeState(void)		
GET_CHARGE_STATE_CONF 0x3 ChargeState				
	Version 1 (implementiert): 0x0: PRE/FAST/TOP-OFF charging (Anzeige im HW/FW-Testprogramm: PFTCH) 0x1: no battery (Anzeige im HW/FW-Testprogramm: NOBAT) 0x2: maintenance - full (Anzeige im HW/FW-Testprogramm: MFULL) 0x3: fault (Anzeige im HW/FW-Testprogramm: FAULT) Version 2 (implementiert): 0x4: no bettery. (Anzeige im HW/FW-Testprogramm: NOBAT)			
	0xA: no battery (Arzeige im HW/FW-restprogramm: NOTCH) 0xB: not charging (Anzeige im HW/FW-Testprogramm: NOTCH) 0xC: PRF/FAST charging (Anzeige im HW/FW-Testprogramm: PRFCH)			
	0xD: TOP-OFF charging (Anzeige im HW/FW-Testprogramm: TPOCH) 0xE: maintenance (Anzeige im HW/FW-Testprogramm: MAINT) 0xF: fault (Anzeige im HW/FW-Testprogramm: FAULT)			
IRLED_EXT_CONTROL_REQ 0x0	externe IRLED ein-/aus-schalten	uint8_t QIUAPI_irledExtControl(uint8_t state);	'E'	
State	0: aus 1: ein			
IRLED_EXT_CONTROL_CONF 0x0	07		'e'	
	interne IRLED ein-/aus-schalten	uint8_t QIUAPI_irledIntControl(uint8_t state);	יןי	
State	0: aus 1: ein			
IRLED_INT_CONTROL_CONF 0x0			'i'	
	DA LED dimmen	uint8_t QIUAPI_dimLed(uint8_t col, uint8_t pwmi);	'G' 'B' 'R'	R' drei separate Kommandos zusammengefasst
LedId DimValue	'G', 'B', 'R' (unsigned): 0 65, 0: aus, 65: max			
DIM_LED_CONF 0x0	DB COMMITTEE OF THE COM		'g' 'b' 'r'	
	LEDs im Balken steuern	uint8_t QIUAPI_ledbarControl(uint8_t ledVector);	'D'	
LedVector	LedVectorBit[7]: left LED LedVectorBit[0]: right LED		_	
	LedVectorBit[x] 0: LED off 1: LED on			
LEDBAR_CONTROL_CONF 0x0			'd'	
	LEDs als Lauflicht steuern	uint8_t QIUAPI_ledbarControlRow(uint8_t direction)	neu	
	direction LEFT: 0x00 direction RIGHT: 0x01			
EARCLIP STATE REQ	Status externer Ohrclip abfragen	Liint8 t Oll IAPI, paralinState(void)		Status kann auch per GET_EVENTSTATUS_VECTOR_REQ abgefragt werden _> EARCUR_STATE_REQ/CONE redundant_> entforman2
	Status externer Ohrclip abfragen	uint8_t QIUAPI_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
State 0x1	0: nicht angeschlossen 1: angeschlossen			
FLASH_READ_DATA_REQ 0x1	Daten (32 Byte) aus Flash lesen	uint8_t QIUAPI_flashReadData(uint16_t sectorNumber, uint8_t partNumber, uint8_t *data);	'h'	
SectorNumber_H	SectorNumber: - 0 MAX (theoretisch nutzbarer Wertebereich entsprechend API-Definition, MAX wird bestimmt von Größe FLASH-IC) - MIN MAX (real nutzbarer Wertebereich oberhalb Filesystem-FLASH-Bereich> abhängig von API-Version)	unito_t data);		
SectorNumber_L PartNumber	PartNumber: 0 127			
FLASH_READ_DATA_CONF 0x1 Data_1	13		'H'	
Data_2				

Page 2

```
0: QP_API_SUCCESS
X: QP_API_ERROR_FLASH_SECTOR_RANGE_MIN (ApiReturnValues.h)
X: QP_API_ERROR_FLASH_SECTOR_RANGE_MAX (ApiReturnValues.h)
  0 FLASH_WRITE_DATA_REQ
                                                                0x14 Daten (32 Byte) in Flash schreiben
                                                                                                                                                                                                                                1 SectorNumber_H
                                                                            - 0 ... MAX (theoretisch nutzbarer Wertebereich entsprechend API-Definition, MAX wird bestimmt von Größe FLASH-IC)
- MIN ... MAX (real nutzbarer Wertebereich oberhalb Filesystem-FLASH-Bereich --> abhängig von API-Version)
2 SectorNumber_L
3 PartNumber
                                                                             PartNumber: 0 ... 127
  0 FLASH_WRITE_DATA_CONF
                                                                           0: QP_API_SUCCESS
X: QP_API_ERROR_FLASH_SECTOR_RANGE_MIN (ApiReturnValues.h)
X: QP_API_ERROR_FLASH_SECTOR_RANGE_MAX (ApiReturnValues.h)
                                                          0x16 Flash-Sektor löschen
  0 FLASH_ERASE_SECTOR_REQ
                                                                                                                                                                                                                                                  uint8_t QIUAPI_flashEraseSector(uint16_t sectorNumber);
    1 SectorNumber_H
                                                                             - 0 ... MAX (theoretisch nutzbarer Wertebereich entsprechend API-Definition, MAX wird bestimmt von Größe FLASH-IC)
                                                                             - MIN ... MAX (real nutzbarer Wertebereich oberhalb Filesystem-FLASH-Bereich ---> abhängig von API-Version)
 2 SectorNumber_L
  0 FLASH_ERASE_SECTOR_CONF
                                                                           0: QP_API_SUCCESS
X: QP_API_ERROR_FLASH_SECTOR_RANGE_MIN (ApiReturnValues.h)
X: QP_API_ERROR_FLASH_SECTOR_RANGE_MAX (ApiReturnValues.h)
       Messung / Parameter
                                                                                                                                                                                                                                                                                                                                      'M' 'X' zwei separate Kommandos zusammengefasst
                                                      0x1A Messung starten/stoppen
    0 PULSE_MEAS_CONTROL_REQ
                                                                                                                                                                                                                                            MeasMode
                                                                           MeasMode[0]: 0: Messung stoppen (dann alle weiteren Bits don't care), 1: Messung starten MeasMode[1]: 0: Online Mode deaktiviert, 1: Online Mode aktiviert, MeasMode[2]: 0: Recording deaktiviert, 1: Recording aktiviert, MeasMode[3]: 0: Samples = ADC-Puls-Signal, 1: Samples = Emulation Sägezahn/Rampe,
  0 PULSE_MEAS_CONTROL_CONF
                                                                                                                                                                                                                                                                                                                                                       wird CONF benötigt? --> sonst ohne (wir vorher)
 0 PULSE_MEAS_DATA_16_IND
                                                                  1D Online-Messwerte (Pulsmessung): 16 Messwerte
  1 Data_1_L
                                                                              Data[15..0]: rsss dddd dddd dddd
                                                                              s: 3-Bit SequenzNumber: 1, 2, ... 6, 7, 6, ... 2, 1, 2, ... *** modifiziert *** d: 12-Bit ADC-Value
2 Data_1_H
3 Data_2_L
 4 Data_2_H
 5 Data_3_L
 6 Data_3_H
31 Data_16_L
32 Data_16_H
  0 GET_RESP_RATE_REQ
                                                                                                                                                                                                                                                             uint8_t QIUAPI_getRespRate(void);
                                                                                                                                                                                                                                                                                                                                         't' Kommandonamen bitte anpassen, falls med. unpassend gewählt
                                                               0x1E Atemfrequenz (Atemzüge pro Minute) abfragen
  0 GET_RESP_RATE_CONF
      RespRate
                                                                             0: Atemfrequenz 5
1: Atemfrequenz 6
                                                                             2: Atemfrequenz 7
3: Atemfrequenz 8
                                                                             4: Atemfrequenz 9
  0 SET_RESP_RATE_REQ
                                                                 0x20 Atemfrequenz (Atemzüge pro Minute) einstellen
                                                                                                                                                                                                                                                       uint8_t QIUAPI_setRespRate(uint8_t respRate);
                                                                             1: Atemfrequenz 6
2: Atemfrequenz 7
                                                                              3: Atemfrequenz 8
                                                                             4: Atemfrequenz 9
 0 SET_RESP_RATE_CONF 0x21
                                                                                                                                                                                                                                                                uint8_t QIUAPI_getGain(void);
                                                          0x22 Verstärkung abfragen
   0 GET_GAIN_CONF
                                                                              0: Verstärkung 479 (53,6 dB)
1: Verstärkung 1277 (62,1 dB)
2: Verstärkung 2553 (68,1 dB)
                                                                               3: Verstärkung 4096 (72,2 dB)
   0 SET_GAIN_REQ
                                                                                                                                                                                                                                                             uint8_t QIUAPI_setGain(uint8_t gain);
                                                                             0: Verstärkung 479 (53,6 dB)
1: Verstärkung 1277 (62,1 dB)
2: Verstärkung 2553 (68,1 dB)
3: Verstärkung 4096 (72,2 dB)
                                                           0x28 Datum/Zeit abfragen
                                                                                                                                                                                                                                                             Calendar QIUAPI_getDatetime(void)
                                                                                                                                                                                                                                                                                                                                                    Zeit als 64 Bit UTC-Wert? Rückgabe im "Calendar"-Format siehe driverlib rtc_c.h
                                                                                                                                                                                                                                                                                                                                      'Z' -RTC-Lib speichert Zeit als 16bit yr, 8bit mo, 8bit dy, 8bit hr, 8bit mi, 8bit sc
                                                              0x2A Datum/Zeit setzen
                                                                                                                                                                                                                                                                                                                                     'z' Übergabe im "Calendar"-Format siehe driverlib rtc_c.h
                                                                                                                                                                                                                                                       uint8_t QIUAPI_setDatetime(Calendar newTime)
 0 SET_DATETIME_CONF 0x2B
                                                               0x3C FW-Update vorbereiten (Shutdown) auf Qiu-Applikations-Ebene - Messungen beenden, Daten speichern, ggf. noch LED-Signalisierung, ...
                                                                                                                                                                                                                                                                 **** kein Bestandteil der API ****
                                                                              - Statusmaschine deaktivieren,
  0 PREPARE_FW_UPDATE_APPL_CONF 0x3D
                                                                                                                                                                                                                                                                                                                                                      - API-Funktion wird von Qiu-Applikation als (letzte) Funktion vor dem FW-Update aufgerufen
- oder alternativ: Shutdown Qiu-Applikation und Qiu-System/API wird bereits (remote) vom Smartphone-FW-Update-Modul initiiert/gesteuert,
- danach Übernahme der Steuerung durch das Smartphone-FW-Update-Modul (Interface/Protokoll: BSL)
    1 PREPARE_FW_UPDATE_SYSTEM_REQ 0x3E
                                                                                                                                                                                                                                                             QIUAPI_PrepareFwUpdateSystem()
                                                                          Das PREPARE_FW_UPDATE_SYSTEM_CONF wird vom aktivierten BSL-Modul generiert/gesendet.

Nach Empfang des PREPARE_FW_UPDATE_SYSTEM_CONF Übernahme der Steuerung durch das Smartphone-FW-Update-
   1 PREPARE_FW_UPDATE_SYSTEM_CONF
                                                                             Modul (Interface/Protokoll: BSL)
  0 PLAY_SOUND_REQ
                                                                                                                                                                                                                                                     uint8_t QIUAPI_playSound(uint8_t soundNumber);
                                                               0x2C Sound abspieler
                                                                                                       (Vorschlag: Tasten-Klick OK)
(Vorschlag: Tasten-Klick Error)
(Vorschlag: Power-On)
(Vorschlag: Power-Off)
    1 SoundNumber
                                                                              2: Test-Sound
3: Test-Sound
                                                                                                         (Vorschlag: Reminder - für Empathy-Funktion)

    - das CONF wird sofort gesendet, sobald das Abspielen des Sound gestartet wurde noch bevor das Abspielen des Sounds beendet ist,
    - das Abspielen der Sounds hat Flash-Zugriffe zur Folge, diese erfolgen im HW-Timer/ISR-Kontext!
    --> alle anderen API-/Kommando-Calls, die ebenfalls Flash-Zugriffe generieren, müssen auf Applikations-Ebene (Qiu-intern) bzw. Smartphone-/PC-Applikation synchronisiert werden,

  0 PLAY_SOUND_CONF
                                                                                                                                                                                                                                                          uint8_t QIUAPI_getPushButtonState(void)
   0 PUSHBUTTON_STATE_CONF
                                                                             0: Taster nicht gedrückt
1: Taster gedrückt
   0 GET_ACCEL_RAW_REQ
                                                                0x56 Beschleunigung-Sensor abfragen (RAW-Werte)
                                                                                                                                                                                                                                                    uint8_t QIUAPI_getAccelRaw(uint8_t * accelRawBuff)
   0 GET_ACCEL_RAW_CONF
                                                                               1: QP_API_ERROR (ApiReturnValues.h) API: oder weiterer spezielle Error-Werte definieren in ApiReturnValues.h
        AccelRaw_X_H
                                                                             Beschleunigung-Wert X-Achse (RAW)
       AccelRaw_X_L
         AccelRaw_Y_H
                                                                              Beschleunigung-Wert Y-Achse (RAW)
        AccelRaw_Y_L
        AccelRaw_Z_H
                                                                              Beschleunigung-Wert Z-Achse (RAW)
       AccelRaw_Z_L
                                                                                mrechnung AccelRaw --> Beschleunigung in g
ccelRaw[15..0]: dddd dddd ddxx
D: 14 Accel-Daten-Bits (signed / 2er-Komplement-Darstellung)
x: ungültige Bits
ccel[g] = 14-Accel-Daten-Bit-Wert * 0,244 / 1000
       Status / Events / Notification
```

0 GET_EVENTSTATUS_VECTOR_REQ 0x2E Event-/Status-Vektor lesen
0 GET_EVENTSTATUS_VECTOR_CONF 0x2F

```
bitweise kodiert:
StatusVector_1[0]: externer Pulssensor (Ohrclip), 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
       StatusVector_1
   2 EventVector_1
                                                                          bitweise kodiert:
                                                                         Alle Event-Vektor-Bits werden nach dem GET automatisch gelöscht
EventVector_1[0]: 0: keine Accel-Sensor-Events, 1: Accel-Sensor-Events vorhanden
         --> externe Dokumentation des Filesystems
   0 GET_DIR_FIND_FIRST_FILE_REQ 0x40 File-Liste anfordern, erstes Element (= erster Datei-Name)
   0 GET_DIR_FIND_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 Filenamenlänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen

O: File vorhanden, Filename Bytes sind gültig
---> nächster Call GET_DIR_FIND_NEXT_FILE_REQ

1: kein (weiteres) File vorhanden, Filename Bytes sind ungültig
---> kein nächster Call GET_DIR_FIND_NEXT_FILE_REQ notwendig
  17 FilenameState
                                                                            2: allgemeiner Fehler, Filename Bytes sind ungültig
   0 GET_DIR_FIND_NEXT_FILE_REQ
                                                      Ox50 File-Liste anfordern, weitere Elemente (= nächster Dateiname)
  0 GET_DIR_FIND_NEXT_FILE_CONF
   1 Filename ASCII byte 1
                                                                          max. Länge 16 Byte (inklusive Extension-Trenner '.')
   2 Filename ASCII byte 2
                                                                          falls Filenamenlänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen
  16 Filename ASCII byte 16

O: File vorhanden, Filename Bytes sind gültig
---> nächster Call GET_DIR_FIND_NEXT_FILE_REQ

1: kein (weiteres) File vorhanden, Filename Bytes sind ungültig
---> kein nächster Call GET_DIR_FIND_NEXT_FILE_REQ notwendig
   17 FilenameState
                                                                            2: allgemeiner Fehler, Filename Bytes sind ungültig
                                                                      Generiert die Datei "DirContent.txt" im Filesystem mit aktuellem Directory-Inhalt, überschreibt vorherige "DirContent.txt",
    0 CREATE_DIR_CONTENT_FILE_REQ
   0 CREATE_DIR_CONTENT_FILE_CONF 0x5F
                                                              0x42 File löschen
   0 DEL_FILE_REQ
  1 Filename ASCII byte 1
                                                                          max. Länge 16 Byte (inklusive Extension-Trenner '.')
  2 Filename ASCII byte 2
   16 Filename ASCII byte 16
                                                                          falls Filenamenlänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen
                                                                            signed, Return/Error-Code entsprechend FS-Error-Code: FS_ERROR_XXXXX (FS.h)
   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 Filenamenlänge < 16 Byte: am Ende nicht benötigte Bytes mit 0x00 auffüllen

READ_FILE_SELECT_REQ_ACTION_OPEN: Read-File auf Qiu-Seite zum lesen öffnen,

READ_FILE_SELECT_REQ_ACTION_CLOSE: das geöffnete Read-File auf Qiu-Seite schließen (Filename ist nicht relevant),

Jedes File-Lesen muss nach dem letzten READ_FILE_BLOCK_REQ/CONF explizit mit CLOSE abgeschlossen werden!
  0 READ_FILE_SELECT_CONF
                                                                            signed, Return/Error-Code entsprechend FS-Error-Code: FS_ERROR_XXXXX (FS.h)
                                                                            Bestätigung/Echo 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
                                                               0x52 Block aus Read-File lesen
    0 READ_FILE_BLOCK_REQ
                                                                          READ_FILE_BLOCK_REQ_ACTION_NEXT: nächsten Block anfordern, READ_FILE_BLOCK_REQ_ACTION_REPEAT: vorherigen Block nochmals anfordern,
                                                                            >= 0: Anzahl der enthaltenen Datenbytes (0 = keine (weiteren) Datenbytes vorhanden);
                                                                          < 0: Error-Code
 3 Checksum_H
                                                                            Checksummen-Wert ist unbestimmt / ungültig (Feature ist noch nicht implementiert)
  4 Checksum_L
                                                              0x5C Block aus Read-File lesen, Funktion mit alternativer Kodierung
    0 READ_FILE_BLOCK_ALT_REQ
                                                                           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
                                                                            >= 0: Anzahl der enthaltenen Datenbytes (0 = keine (weiteren) Datenbytes vorhanden);
                                                                          < 0: Error-Code
 2 State_L
3 Checksum_H
 4 Checksum_L
    0 FS_STORAGE_CLEAN_REQ
                                                               0x60 | führt Garbage Collection im Filesystem aus (per FS_STORAGE_CleanOne() Calls)
                                                                          1 ... 254: Anzahl der FS_STORAGE_CleanOne() Calls, welche maximal ausgeführt werden sollen, 255: FS_STORAGE_CleanOne() Calls werden ausgeführt, bis Flash vollständig "cleaned" ist,
   1 FsStorageCleanOneThreshold
   0 FS_STORAGE_CLEAN_CONF
                                                               0x61 wird gesendet, wenn Garbage Collection beendet wurde (kann im Bereich von Sekunden bis Minuten liegen)
   1 FsStorageCleanOneCount_H
                                                                            Anzahl der FS_STORAGE_CleanOne() Calls, die ausgeführt wurden,
  2 FsStorageCleanOneCount_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
                                                                          0xFFFFFFFF: Fehler
 7 FreeSpaceKB_2
  8 FreeSpaceKB_L
        System Status Vector
```

Reserved Events Teletronic

0 GET_SYSTEM_STATUS_REQ 0 GET_SYSTEM_STATUS_CONF 0x49 Error-Vektor lesen 1 ErrorVector1 ErrorVector_1[0] = 1: HFXT externe Clock nicht gestartet
ErrorVector_1[1] = 1: LFXT externe Clock nicht gestartet 2 ErrorVector2 3 ErrorVector3 4 ErrorVector4 5 FileSystemStatus

PULSE_MEAS_16_TIMER_IND 0xD1 es sind (mind.) 16 Samples in Measurement-Queue verfügbar --> auslesen/verarbeiten Data[15..0]: rsss dddd dddd dddd r: reserviert s: 3-Bit SequenzNumber: 1, 2, ... 6, 7, 6, ... 2, 1, 2, ... *** modifiziert *** d: 12-Bit ADC-Value

Kommando und API Tabelle