Protocol Specification

Philipp Driess

July 29, 2014

Abstract

This document defines the message specification of the Victron BMV 700 and BlueSolar Grid Inverter devices. The message specification of the BlueSolar Inverter can be wrong, because it was reverse engineered with help of a logic analayzer. Feel free to extend it and if you find errors get in touch with me or write an patch.

Contents

1 Protocol Spezification - Victron BMV 700

```
LF =                "0x0d","0x0a" ;
Tab =               "0x09" ;
HexDigit =          [ a-fA-F0-9 ] ;
Byte =              HexDigit, HexDigit ;
Hex =               ( "0x", Byte ) ;
U_INT8 =            Hex ;
U_INT16 =           Hex, Hex ;
U_INT32 =           Hex, Hex, Hex,Hex ;
INT32 =             Hex, Hex, Hex,Hex ;
ASCII_Char =        Hex ;

PID =       LF, "PID", Tab, U_INT16 ;   (* Product ID *)
V =         LF, "V", Tab, U_INT16 ;     (* Main (battery) voltage *)
VS =        LF, "VS", Tab, U_INT16 ;    (* Auxiliary (starter) voltage *)
VM =        LF, "VS", Tab, U_INT16 ;    (* Mid-point voltage battery bank *)
DM =        LF, "VS", Tab, U_INT16 ;    (* Mid-point deviation battery bank *)
I =         LF, "I", Tab, INT32 ;       (* Current *)
T =         LF, "I", Tab, INT32 ;       (* Battery temperature *)
P =         LF, "P", Tab, INT32 ;       (* Instantaneous power *)
CE =        LF, "CE", Tab, INT32 ;      (* Consumed Amp Hours *)
SOC =       LF, "SOC", Tab, U_INT16 ;   (* Consumed Amp H State-of-charge *)
TTG =       LF, "TTG", Tab, U_INT16 ;   (* Time-to-go *)

Alarm =     LF, "Alarm", Tab, ASCII_Char[ 3 ] ; (* Alarm condition active *)
Relay =     LF, "Relay", Tab, ASCII_Char[ 3 ] ; (* Relay stat *)
AR =        LF, "AR", Tab, U_INT8 ;             (* Alarm reason *)
BMV =       LF, "BMV", Tab, U_INT16 ;           (* Model description *)
FW =        LF, "FW", Tab, U_INT16 ;            (* Firmware versio *)
H1 =        LF, "H1", Tab, INT32 ;              (* Alarm reason Depth of the
                                                   deepest discharge *)

H2 =        LF, "H2", Tab, INT32 ;      (* Depth of the last discharge *)
H3 =        LF, "H3", Tab, INT32 ;      (* Depth of the average discharge *)
H4 =        LF, "H4", Tab, U_INT16 ;    (* Number of charge cycles *)
H5 =        LF, "H5", Tab, U_INT16 ;    (* Number of full discharges *)
H6 =        LF, "H6", Tab, U_INT16 ;    (* Cumulative Amp Hours drawn *)
H7 =        LF, "H7", Tab, U_INT16 ;    (* Minimum main (battery) voltage *)
H8 =        LF, "H8", Tab, U_INT16 ;    (* Maximum main (battery)
                                           voltage *)
H9 =        LF, "H9", Tab, U_INT16 ;    (* Number of seconds since last full
                                           charge *)
H10 =       LF, "H10", Tab, U_INT16 ;   (* Number of automatic
                                           synchronizations *)
H11 =       LF, "H11", Tab, U_INT16 ;   (* Number of low main voltage alarms *)
H12 =       LF, "H12", Tab, U_INT16 ;   (* Number of high main voltage alarms *)
H13 =       LF, "H13", Tab, U_INT16 ;   (* Number of low auxiliary voltage
                                           alarms *)
```

```
H14 =      LF, "H14", Tab, U_INT16 ;  (* Number of high auxiliary voltage
                                         alarms *)
H15 =      LF, "H15", Tab, U_INT16 ;  (* Number of high auxiliary voltage
                                         alarms *)
H16 =      LF, "H16", Tab, U_INT16 ;  (* Maximum auxiliary (battery)
                                         voltage *)
H17 =      LF, "H17", Tab, U_INT16 ;  (* Amount of discharged energy *)
H18 =      LF, "H18", Tab, U_INT16 ;  (* Amount of charged energy * )

Checksum = LF, "Checksum", Tab, Hex ; (* CRC-8 * )

BMV_700_Block1 = PID, V, I, P, CE, SOC, TTG, Alarm, Relay, AR, BMV, FW,
                 Checksum ;

BMV_700_Block2 = H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, H11, H12, H13, H14,
                 H15, H16, H17, H18, Checksum ;

BMV_700_MSG =    BMV_700_Block1, BMV_700_Block2 ;
```

2 Protocol Spezification - Victron MPPT Chargers

```
LF =                "0x0d","0x0a" ;
Tab =               "0x09" ;
HexDigit =          [ a-fA-F0-9 ] ;
Byte =              HexDigit, HexDigit ;
Hex =               ( "0x", Byte ) ;
U_INT8 =            Hex ;
U_INT16 =           Hex, Hex ;
U_INT32 =           Hex, Hex, Hex,Hex ;
ASCII_Char =        Hex ;

V =   LF, "V", Tab, U_INT16 ;                       (* Main  (battery) voltage *)
VPV = LF, "V", Tab, U_INT16(* can be wrong *) ;    (* Panel voltage *)
PPV = LF, "V", Tab, U_INT16(* can be wrong *) ;    (* Panel power *)
I =   LF, "I", Tab, U_INT32 ;                       (*  Current *)
H19 = LF, "H19", Tab, U_INT16(* can be wrong *) ;  (* Yield total *)
H20 = LF, "H20", Tab, U_INT16(* can be wrong *) ;  (* Yield today *)
H21 = LF, "H21", Tab, U_INT16(* can be wrong *) ;  (* Maximum power today *)
H22 = LF, "H22", Tab, U_INT16(* can be wrong *) ;  (* Yield yesterday *)
H23 = LF, "H23", Tab, U_INT16(* can be wrong *) ;  (* Maximum power yesterday *)
ERR = LF, "ERR", Tab, U_INT16(* can be wrong *) ;  (* Error code *)
FW =  LF, "FW", Tab, U_INT16 ;                      (* Firmware version *)
PID = LF, "PID", Tab, U_INT16 ;                     (* Product ID *)
SER = LF, "SER#", Tab, U_INT16 (* can be wrong *); (* Serial number *)

Checksum =          LF, "Checksum", Tab, U_INT8 ;   (* CRC-8 *)

MPPT_Block =        V, VPV, PPV, I, H19, H20, H21, H22, H23, ERR, FW, PID, SER#,
                    Checksum ;

VICTRON_MPPT_MSG = MPPT_Block ;
```

3 Protocol Spezification - Victron BlueSolar Grid Inverter

```
HexDigit =                          [ a-fA-F0-9 ] ;
Byte =                              HexDigit, HexDigit ;
Hex =                               ( "0x", Byte ) ;

Size =                              Hex ;
U_INT32 =                           Hex, Hex, Hex, Hex ;
U_INT16 =                           Hex, Hex ;
U_INT8 =                            Hex ;
ASCII_Code =                        Hex ;

ID1 =                               "0x01","0x04" ;
ID2 =                               "0x01","0x03" ;
KW_Total =                          U_INT32 ;
Run_Time_Day =                      U_INT16 ;
Run_Time_Total =                    U_INT32 ;
Temp =                              U_INT16 ;
A_CPU_BUS_U =                       U_INT16 ;
Checksum =                          U_INT16 ; (* CRC-16 (Modbus) *)
DC_Voltage =                        U_INT16 ;
AC_Voltage =                        U_INT16 ;
AC_Current =                        U_INT16 ;
DC_Voltage =                        U_INT16 ;
AC_Freq =                           U_INT16 ;
PV_Watt =                           U_INT16 ;
KW_Day  =                           U_INT16 ;
B_CPU_AC_U =                        U_INT16 ;
DCI1 =                              U_INT8 ;
SoftwareVersion =                   Hex, Hex ;
Serialnumber =                      Hex, Hex, Hex, Hex, Hex, Hex ;
ModbusAddress =                     Hex ;
VPV_Start_V =                       U_INT16 ;
T_Starts =                          U_INT16 ;
Code =                              U_INT16 ;
VAC_Low_Voltage_V =                 U_INT16 ;
VAC_Ultra_Low_Voltage_V =           U_INT16 ;
Time_Low_Voltage =                  U_INT16 ;
Time_Ultra_Low_Voltage =            U_INT16 ;
VAC_High_Voltage_V =                U_INT16 ;
VAC_Ultra_High_Voltage_V =          U_INT16 ;
Time_High_Voltage =                 U_INT16 ;
Time_Ultra_High_Votlage =           U_INT16 ;
Low_Frequency =                     U_INT16 ;
Ultra_Low_Frequency =               U_INT16 ;
Time_Low_Frequency =                U_INT16 ;
Time_Ultra_Low_Frequency =          U_INT16 ;
High_Frequency =                    U_INT16 ;
```

```
Ultra_High_Frequency =              U_INT16 ;
Time_High_Frequency =               U_INT16 ;
Time_Ultra_High_Frequency =         U_INT16 ;
DCI_Limit =                         U_INT16 ;
Full_Power_Limit =                  U_INT16 ;
Active_Power_Reduction_rate_pf_Hz = U_INT16 ;
KPV_Gain =                          U_INT16 ;
KAC_Gain =                          U_INT16 ;
KA_Bus_Gain =                       U_INT16 ;
KB_CPU_AC_Gain =                    U_INT16 ;
DCI_Shift =                         U_INT16 ;
Inverter_Mode =                     ASCII_Code [12] ;
Fac_High_SPI_Hz =                   U_INT16 ;
Fac_Low_SPI_Hz =                    U_INT16 ;
Time_Fac_High_SPI_Hz =              U_INT16 ;
Time_Fac_Low_SPI_Hz =               U_INT16 ;
Connection_Fac_High_Hz =            U_INT16 ;
Connection_Fac_Low_Hz =             U_INT16 ;

Request01 = 0x01, 0x04, 0x00, 0x14, 0x00, 0x11, 0x70, 0x02 ;
Request02 = 0x01, 0x04, 0x00, 0x00, 0x00, 0x14, 0xF0, 0x05 ;
Request03 = 0x01, 0x04, 0x03, 0xE8, 0x00, 0x14, 0x70, 0x75 ;
Request04 = 0x01, 0x03, 0x00, 0x00, 0x00, 0x0C, 0x45, 0xCF ;
Request05 = 0x01, 0x03, 0x00, 0x64, 0x00, 0x09, 0x45, 0xCF ;
Request06 = 0x01, 0x03, 0x00, 0xC8, 0x00, 0x12, 0x44, 0x39 ;
Request07 = 0x01, 0x03, 0x00, 0xDC, 0x00, 0x02, 0x05, 0xF1 ;
Request08 = 0x01, 0x03, 0x01, 0x2C, 0x00, 0x06, 0x05, 0xFD ;
Request09 = 0x01, 0x03, 0x01, 0x90, 0x00, 0x14, 0x44, 0x14 ;
Request10 = 0x01, 0x03, 0x03, 0xE8, 0x00, 0x14, 0xC5, 0xB5 ;
Request11 = 0x01, 0x03, 0x04, 0x10, 0x00, 0x14, 0x45, 0x30 ;
Request12 = 0x01, 0x03, 0x04, 0x24, 0x00, 0x05, 0xC4, 0xF2 ;
Request13 = 0x01, 0x03, 0x04, 0x4C, 0x00, 0x14, 0x85, 0x22 ;
Request14 = 0x01, 0x03, 0x01, 0xF5, 0x00, 0x06, 0xD4, 0x06 ;

Reply01   = ID1, Size, KW_Total, Run_Time_Day, Run_Time_Total, Hex[ 12 ], Temp,
            A_CPU_BUS_U, Hex[ 8 ], Checksum ;

Reply02   = ID1, Size, DC_Voltage, Hex[ 10 ], AC_Voltage, Hex[ 4 ],
            AC_Current, Hex[ 4 ], AC_Freq, Hex[ 8 ], PV_Watt, Hex[ 2 ],
            KW_Day, Checksum ;

Reply03   = ID1, Size, Hex[ 12 ], B_CPU_AC_U, Hex[ 22 ], DCI1, Hex[ 2 ],
            Checksum ;

Reply04   = ID2, Size, Hex[ 4 ], SoftwareVersion, Hex[ 6 ], Serialnumber,
            Hex[ 5 ], ModbusAddress, Checksum ;

Reply05   = ID2, Size, VPV_Start_V, Hex[ 6 ], T_Starts, Hex[ 8 ], Checksum ;

Reply06   = ID2, Size, Code, VAC_Low_Voltage_V, VAC_Ultra_Low_Voltage_V,
```

```
            Time_Low_Voltage, Time_Ultra_Low_Voltage, VAC_High_Voltage_V,
            VAC_Ultra_High_Voltage_V, Time_High_Voltage,
            Time_Ultra_High_Votlage, Low_Frequency, Ultra_Low_Frequency,
            Time_Low_Frequency, Time_Ultra_Low_Frequency, High_Frequency,
            Ultra_High_Frequency, Time_High_Frequency,
            Time_Ultra_High_Frequency, DCI_Limit, Checksum ;

Reply07   = ID2, Size, Hex[ 4 ], Checksum ;

Reply08   = ID2, Size, Hex[ 12 ], Checksum ;

Reply09   = ID2, Size, Hex[ 4 ], Full_Power_Limit, Hex[ 2 ],
            Active_Power_Reduction_rate_pf_Hz, Hex[ 30 ],
            Checksum;

Reply10   = ID2, Size, Hex[ 12 ], KPV_Gain, Hex[ 14 ], KAC_Gain, Hex[ 6 ],
            KA_Bus_Gain, Hex[ 2 ], Checksum ;

Reply11   = ID2, Size, Hex[ 12 ], KB_CPU_AC_Gain, Hex[ 26 ], Checksum ;

Reply12   = ID2, Size, Hex[ 8 ], DCI_Shift, Checksum ;

Reply13   = ID2, Size, Inverter_Mode, Hex[ 28 ], Checksum ;

Reply14   = ID2, Size, Fac_High_SPI_Hz, Fac_Low_SPI_Hz, Time_Fac_High_SPI_Hz,
            Time_Fac_Low_SPI_Hz, Connection_Fac_High_Hz, Connection_Fac_Low_Hz,
            Checksum ;
```

4 Protocol Spezification - Victron Multiplus and Quadro

```
HexDigit =                [ a-fA-F0-9 ] ;
Byte =                    HexDigit, HexDigit ;
Hex =                     ( "0x", Byte ) ;

U_INT32 =                 Hex, Hex, Hex, Hex ; (* Little-Endian *)
U_INT24 =                 Hex, Hex, Hex ;      (* Little-Endian *)
U_INT16 =                 Hex, Hex ;           (* Little-Endian *)
INT16 =                   Hex, Hex ;           (* Little-Endian *)
U_INT8 =                  Hex ;
ASCII_Code =              Hex ;

Length =                  U_INT8 ;
Checksum =                Hex ;    (* CRC-8 *)
CMD_0 =                   "0x56" ;
CMD_1 =                   "0x4c" ;
CMD_2 =                   "0x20" ;
CMD_3 =                   "0x41" ;

Frame =                   "0xff" ;
Frame_Type_0 =            "0x00" ; (* DC-Info Frame *)
Frame_Type_1 =            "0x01" ; (* AC Info L1 Frame *)
Frame_Type_2 =            "0x02" ; (* AC Info L2 Frame *)
Frame_Type_3 =            "0x03" ; (* AC Info L3 Frame *)
Frame_Type_4 =            "0x04" ; (* AC Info L4 Frame *)
Frame_Type_5 =            "0x05" ; (* MasterMultiLED Frame *)

LED_ON =                  Hex ;    (* 0-Bit: Mains, 1-Bit: Absorption,
                                      2-Bit: Bulk,3-Bit: Float,
                                      4-Bit: Inverter, 5-Bit: Overload,
                                      6-Bit: Low battery, 7-Bit: Temperature *)

LED_BLINK =               Hex ;    (* 0-Bit: Mains, 1-Bit: Absorption,
                                      2-Bit: Bulk,3-Bit: Float,
                                      4-Bit: Inverter, 5-Bit: Overload,
                                      6-Bit: Low battery, 7-Bit: Temperature *)

VersionNumber =           U_INT32
Phase_Info =              U_INT8 ;
DC_Voltage =              U_INT16 ;
DC_Current_Inverter =     U_INT24 ;
DC_Current_Charger =      U_INT24 ;
Inverter_Periode =        U_INT8 ;
BF_Factor =               U_INT8 ;
Inverter_Factor =         U_INT8 ;
Mains_Voltage =           U_INT16 ;
Mains_Current =           INT16 ;
```

```
Inverter_Voltage =         U_INT16 ;
Inverter_Current =         INT16 ;
Mains_Periode =            U_INT8 ;
AC_Input_Configuration  =  Hex ; (* 0-Bit: last active AC0,
                                    1-Bit: last active AC1,
                                    2-Bit: Input Current overwritten by Panel
                                    3-7-Bit: not defined *)
MIN_INP_Current_Limit =    U_INT16 ;
MAX_INP_Current_Limit =    U_INT16 ;
Actual_INP_Current_Limit = U_INT16 ;

MK2_FRAME = Length, Frame, CMD_0, VersionNumber, Checksum ; (* Mk2 frame *)
Request01 = 0x02, 0xff, 0x4c, 0xb3 ;        (* request to get LED Status *)
Request02 = 0x03, 0xff, 0x46, 0x00 , 0xb8 ; (* request to get InfoFrame0 *)
Request03 = 0x03, 0xff, 0x46, 0x01 , 0xb7 ; (* request to get InfoFrame1 *)
Request04 = 0x03, 0xff, 0x46, 0x02 , 0xb6 ; (* request to get InfoFrame2 *)
Request05 = 0x03, 0xff, 0x46, 0x03 , 0xb5 ; (* request to get InfoFrame3 *)
Request06 = 0x03, 0xff, 0x46, 0x04 , 0xb4 ; (* request to get InfoFrame4 *)
Request07 = 0x03, 0xff, 0x46, 0x05 , 0xb3 ; (* request to get InfoFrame5 *)

Reply01 =   Length, Frame, CMD_1, LED_ON, LED_BLINK, Checksum ;

Reply02 =   Length, CMD_2, Frame_Type_0, Hex[ 4 ], Phase_Info, DC_Voltage,
            DC_Current_Inverter, DC_Current_Charger, Inverter_Periode,
            Checksum ;

Reply03 =   Length, CMD_2, Frame_Type_1, BF_Factor, Inverter_Factor,
            Mains_Voltage, Mains_Current, Inverter_Voltage, Inverter_Current,
            Mains_Periode, Checksum ;

Reply04 =   Length, CMD_2, Frame_Type_2, BF_Factor, Inverter_Factor,
            Mains_Voltage, Mains_Current, Inverter_Voltage, Inverter_Current,
            Mains_Periode, Checksum ;

Reply05 =   Length, CMD_2, Frame_Type_3, BF_Factor, Inverter_Factor,
            Mains_Voltage, Mains_Current, Inverter_Voltage, Inverter_Current,
            Mains_Periode, Checksum ;

Reply06 =   Length, CMD_2, Frame_Type_4, BF_Factor, Inverter_Factor,
            Mains_Voltage, Mains_Current, Inverter_Voltage, Inverter_Current,
            Mains_Periode, Checksum ;

Reply07 =   Length, CMD_3, Frame_Type_5, Hex[ 4 ], AC_Input_Configuration,
            MIN_INP_Current_Limit, MAX_INP_Current_Limit,
            Actual_INP_Current_Limit, Checksum ;
```