Stand:28.2.2000

Beschreibung der seriellen Schnittstelle des AMC4

1.) Beim Kommunikationsaufbau zwischen HW und PC wird als erstes die Version abgefragt, d.h. der PC sendet ein CMD_RD_VERS (= 0x16). Der AMC4 sendet daraufhin folgende 11 Byte:

CMD_RD_VERS; (0x16)
VERSION_NUMBER; (0x1)
VERSION_INDEX; (0x07)
VERSION_DAY; (0x05)
VERSION_MONTH; (0x06)
VERSION_YEAR >> 8; (1999/256)
VERSION_YEAR & 255; (1999 & 255)
COMPILE_OPTION1; (0 für 500mA Version; 1 für 2A Version)
COMPILE_OPTION2; (0 für 6 Tasten Version; 1 für 4 Tasten Version)
COMPILE_OPTION3; (0 für AMC2 bzw. AMC3; 1 für AMC4)
COMPILE_OPTION4; (0 für Deutsch; 1 für Englisch)

Falls die Versions und Index Nummer kleiner als die oben angegebene ist, meldet der PC einen „Versions-Fehler. Eventuell kanne auch noch das Datum kontrolliert werden. Beim AMC4 sind folgende Compile-Optionen gesetzt: Compile_Option 1, 2 und 3.

Für alle anderen Commandos muss die Kanalnummer mitgesendet werden. Das gesendete Kommando besteht immer aus 8 Bit. Die obersten 2 Bit sind für die Kanal-Nummer reserviert. D.h. die gewünschte Kanal-Nummer wird mit 64 multipliziert und das Commando hinzu addiert. Dabei entspricht Kanal 1 der 0, Kanal 2 der 1, Kanal 3 der 2 und Kanal 4 der 3.
*Beispiel: Das Commando CMD_RD_SET hat für den Kanal 1 den Wert 0x11 (0*64 + 0x11), für den Kanal 2 den Wert 0x51 (1*64 + 0x11), für den Kanal 3 den Wert 0x91 (2*64 + 0x11) und für den Kanal 4 den Wert 0xD1 (3*64 + 0x11)*
Die folgenden Commando-Beschreibungen beziehen sich zur Vereinfachung nur noch auf den Kanal 1.
Kann der AMC4 das gesendete Commando nicht verstehen, sendet er immer ein 0x80 zurück.

2.) Ist die Versionsabfrage in Ordnung, wird CMD_RD_SET (=0x11). Als Antwort erhält man folgende 14 Bytes zurück:

CMD_RD_SET;	(0x11)	
status_code & STATUS_MASK;	(status_code & 0x87)	/* status */ [1])
error_definition[error_number[tmp_ch]-1].display_number; [2])		/* error number
program[tmp_ch];	(1...9)	/* program number */
akku_type[tmp_ch];	(NiCd=0; NiMH=1; PB=2)	/* akku type */
cell_count[tmp_ch];	(1…12)	/* cell count */
nominal_capacity[tmp_ch] /256;	(100 ... 20000)	/* nominal capacity */
nominal_capacity[tmp_ch] & 0xff;		
discharge_current[tmp_ch] / 256;	(50...2000)	/* discharge current */
discharge_current[tmp_ch] & 0xff;		
setup_charge_current[tmp_ch] / 256;	(50...2000)	/* charge current */
setup_charge_current[tmp_ch] & 0xff;		
repeat_time[tmp_ch] / 256;	(30...7200)	/* wait time */
repeat_time[tmp_ch] & 0xff;	(bzw. 1440...43200 bei Program 6)	/* wait time */

[1])
die relevanten Bits bei status_code haben folgende Bedeutung:

Bit0	charge_akku	(entsprechender Kanal lädt Akku)
Bit1	discharge_akku	(entsprechender Kanal entlädt Akku)
Bit3	trickle_charge_akku	(entsprechender Kanal befindet sich im Modus „Erhaltungsladung“)
Bit7	progress	(entsprechender Kanal ist aktiv, d.h. er lädt bzw. entlädt)

[2])

error_definition[error_number[tmp_ch]-1].display_number:

```
1       {"Akku defekt! bzwZel.-Zahl fals ch", ERROR_TYP_FATAL,  1},
2       {"Akku verpolt!   Ueberpruefen!   ", ERROR_TYP_WARNING,  2},
3       {"Zellenzahl     ueberpruefen!   ", ERROR_TYP_WARNING,  3},
4       {"Lade-Schluss-Spgnicht erreicht  ", ERROR_TYP_ERROR,    4},
5       {"Ladezeit       ueberschritten  ", ERROR_TYP_ERROR,    5},
6       {"Anschluss Ladek-abel/Kurzschluss", ERROR_TYP_WARNING,  6},
7       {"Akku voll oder  Akku hochohmig  ", ERROR_TYP_ERROR,    7},
8       {"Kein Akku      angeschlossen!  ", ERROR_TYP_ERROR,    8},
9       {"Sicherung defektPruefen/erneuern", ERROR_TYP_FATAL,   98},
10      {"EEPROM defekt,  anService wenden", ERROR_TYP_FATAL,   99},
11      {"Programm beendetKein Akku ang.  ", ERROR_TYP_FATAL,    8},
12      {"Kein Akku      angeschlossen!  ", ERROR_TYP_ERROR,    8},
13      {"Warnung 70:     Zeitueberschr.  ", ERROR_TYP_WARNING,  70},
14      {"Warnung 71:     Delta U         ", ERROR_TYP_WARNING,  71},
15      {"Warnung 72:     Spannung zu hoch", ERROR_TYP_WARNING,  72},
16      {"Warnung 73:     Ladestrom       ", ERROR_TYP_WARNING,  73},
17      {"Kühlkörper-Temp.zu hoch !       ", ERROR_TYP_FATAL,    17};
```

Danach wird CMD_RD_SET2 gesendet. Daraufhin erhält man folgende 5 Bytes zurück:

```
CMD_RD_SET2;                    (0x21)
akku_number[tmp_ch];            (0...7)          /* number of akku data set */
max_cycle_count[tmp_ch];        (1...9)          /* number of maximal cycles */
charge_current[tmp_ch] / 256;   (50...2000)      /* charge current*/
charge_current[tmp_ch] & 0xff;
```

3.) Wenn ein Kanal schon aktiv ist, kann man die aktuellen Daten mit dem CMD_RD_MEAS und dem CMD_RD_MEAS2 auslesen. Dabei erhält man folgende 16 Bytes zurück:

```
CMD_RD_MEAS;                                    (0x12)
actual_discharge_capacity[tmp_ch] / 256;        (0...20000)     /* actual discharge capacity */
actual_discharge_capacity[tmp_ch] & 0xff;                       /* actual discharge capacity */
actual_charge_capacity[tmp_ch] /256;            (0...20000)     /* actual charge capacity */
actual_charge_capacity[tmp_ch] & 0xff;                          /* actual charge capacity */
(akku_voltage_rms[tmp_ch] / WIN_MA_FACTOR) /256;                /* actual akku voltage */
(akku_voltage_rms[tmp_ch] / WIN_MA_FACTOR) & 0xff;              /* actual akku voltage */
discharge_time_h[tmp_ch];                                       /* discharge time hours */
discharge_time_m[tmp_ch];                                       /* discharge time minutes */
discharge_time_s[tmp_ch];                                       /* discharge time seconds */
charge_time_h[tmp_ch];                                          /* charge time hours */
charge_time_m[tmp_ch];                                          /* charge time minutes */
charge_time_s[tmp_ch];                                          /* charge time seconds */
cycle_count[tmp_ch];                                            /* charge count */
repeat_time[tmp_ch] / 256;                                      /* actual wait time */
repeat_time[tmp_ch] & 0xff;                                     /* actual wait time */
```

*) WIN_MA_FACTOR = 5

Mit dem CMD_RD_MEAS2 erhält man folgende 3 Byte zurück:

```
CMD_RD_MEAS2;                                   (0x22)
old_discharge_capacity[tmp_ch] / 256;                           /* actual discharge capacity */
old_discharge_capacity[tmp_ch] & 0xff;                          /* actual discharge capacity */
```

4.) Mit dem CMD_WR_PARA kann man mit 10 Byte folgende Parameter in den AMC4 schreiben (man sollte sich aber vorher vergewissern, daß der gewünschte Kanal nicht aktiv ist):

```
program[tmp_ch] = rx_buffer[1];                                 (1...9)
repeat_time[tmp_ch] = rx_buffer[2] * MINUTES_PER_DAY;           (0...5, 1...30 bei Program  6)
akku_type[tmp_ch] = rx_buffer[3];                               (0 ... 2)
cell_count[tmp_ch] = rx_buffer[4];                              (1...12)
```

nominal_capacity[tmp_ch] = rx_buffer[6] + rx_buffer[5] * 256; (100...20000)
discharge_current[tmp_ch] = rx_buffer[8] + rx_buffer[7] * 256; (50...2000)
charge_current[tmp_ch] = rx_buffer[10] + rx_buffer[9] * 256; (50...2000)
setup_charge_current[tmp_ch] = charge_current[tmp_ch]; (50...2000)

Über- bzw. Unterschreitet einer der oben angegebenen Parameter den Gültigkeitsbereich, meldet der AMC4 folgendes zurück:
CMD_WR_PARA (0x14)
0x80
Sind die Werte in Ordnung meldet der AMC4:
CMD_WR_PARA (0x14)
0x00

Mit CMD_WR_PARA2 werden die restlichen Parameter in den AMC4 geschrieben:
akku_number[tmp_ch] = rx_buffer[1]; (0...7)
max_cycle_count[tmp_ch] = rx_buffer[2]; (1...9)
repeat_time[tmp_ch] = rx_buffer[4] + rx_buffer[3] * 256; (30...7200; 1440...43200 bei Prog. 6)

Über- bzw. Unterschreitet einer der oben angegebenen Parameter den Gültigkeitsbereich, meldet der AMC4 folgendes zurück:
CMD_WR_PARA (0x24)
0x80
Sind die Werte in Ordnung meldet der AMC4:
CMD_WR_PARA (0x24)
0x00
Danach schreibt der AMC4 die Parameter in sein EEPROM (d.h. selbst nach Ausschalten und wieder Einschalten des AMC4 sind die Parameter noch vorhanden.

5.) Um ein Programm des AMC4 zu starten, muss CMD_START (=0x15) gesendet werden.
Als Antwort erhält man folgende 2 Byte zurück:
CMD_START (0x15)
0x00 (bei erfolgreichem Start)
bzw.
0x80 (bei Fehler)

5a) Hat der AMC4 auf CMD_START mit 0x00 geantwortet, muß nach ca. 1 Sek. das Commando CMD_START_NOW (0x19) gesendet werden. Als Antwort erhält man folgende 2 Byte zurück:
CMD_START_NOW (0x19)
0x00 (bei erfolgreichem Start)
bzw.
0x80 (bei Fehler)

5b) Hat der AMC4 auf CMD_START_NOW mit 0x00 geantwortet muß nach einer weiteren Sekunde das Commando CMD_ASK_WAIT (0x17) gesendet werden. Mit diesem Commando wird abgefragt, ob der AMC4 das gewünschten Programm schon gestartet hat. Der AMC4 kann mit folgenden Bytes antworten:
CMD_ASK_WAIT (0x17)
0x00 (bei erfolgreichem Start)
bzw.
0x80 (Gesamtstrom zu hoch! Kanal in den Wait-Modus setzen?)

5c) Soll der Gesamtstrom nicht geändert werden und der AMC4-Kanal in den Wait-Modus gehen, dann muß das Commando CMD_WAIT (0x18) gesendet werden. Als Antwort erhalten Sie folgende 2 Byte zurück:
CMD_WAIT (0x19)
0x00 (bei erfolgreichem Start)
bzw.
0x80 (bei Fehler)
Sollen die Einstellungen aber neu vorgemommen werden, muß das Commando CMD_STOP (0x13) gesendet werden.

6.) Um ein Programm des AMC4 zu stoppen, muss CMD_STOP (=0x13) gesendet werden.
Als Antwort erhält man folgende 2 Byte zurück:

CMD_STOP	(0x13)
0x00	(bei erfolgreichem Start)
bzw.	
0x80	(bei Fehler)

7.) Man kann den AMC4 dazu veranlassen, die aktuellen Parameter aus dem EEPROM zu lesen, dazu muss CMD_EE_RD (=0x23) gesendet werden.
Als Antwort erhält man folgende 2 Byte zurück:

CMD_EE_RD	(0x23)
0x00	(bei erfolgreichem Start)
bzw.	
0x80	(bei Fehler)

8.) Man kann den AMC4 dazu veranlassen, die aktuellen Parameter in das EEPROM zu schreiben, dazu muss CMD_EE_WR (=0x25) gesendet werden.
Als Antwort erhälten man die folgenden 2 Byte zurück:

CMD_EE_RD	(0x25)
0x00	(bei erfolgreichem Start)
bzw.	
0x80	(bei Fehler)

Anmerkung:
Es kann vorkommen, daß bei Benutzung mehrerer Kanäle zur gleichen Zeit, der maximale Lade- bzw. Entladestrom von 2 A überschritten wird. Um dies zu verhindern, müssen von PC-Seite Vorkehrungen getroffen werden, daß nicht mehr als insgesamt 2A Lade- bzw. Entladestrom eingestellt werdeen können.