

# Schnittstellenbeschreibung des AkkuMaster C5

Hinweis N1: Beim Lesen dieser Beschreibung wird vom Leser ein Grundwissen in englischer Sprache (ein De-facto-Standard in der „PC-Welt“) und in der Programmiersprache „C“ (De-facto-Industriestandard) verlangt.

Hinweis N2: Dies ist eine kurze Beschreibung, die zum Kennen lernen der Schnittstelle des Gerätes gedacht ist. Für mehr Information über Handhabung des Befehlsatzes beachten Sie bitte die Programmierbeispiele. Die Programmierbeispiele wurden mit der Hilfe der Entwicklungsumgebung „Pelles C für Windows“ geschrieben. Dieses Programm ist frei und kann z.B. von dieser Internet-Seite downgeladen werden: <http://www.christian-heffner.de>

Da sind die einige Konstanten und Deklarationen, die in der Beschreibung erwähnt werden können:

```
//-- constants --//

//C5 address
#define C5_RX_ADDRESS          0

//info cmds
#define CMD_GET_DEVICE_NAME    10
#define CMD_GET_DEVICE_VERSION 11

//log data cmds
#define CMD_GET_SERVICE_INFO    12
#define CMD_GET_SERVICE_DATA    13

//set mode cmds
#define CMD_ENTER_DF_MODE       14
#define CMD_ENTER_RC_MODE       15

//read dataflash cmds
#define CMD_DF_EXIT              16
#define CMD_DF_GET_INFO         17
#define CMD_DF_PREPARE_DATA      18
#define CMD_DF_GET_DATA         19
#define CMD_DF_GET_DATA_BULK     20

//remote control cmds
#define CMD_RC_EXIT              21
#define CMD_RC_START_CHARGE      22
#define CMD_RC_START_DISCHARGE   23
#define CMD_RC_STOP_SERVICE      24
#define CMD_RC_GET_SERVICE_DATA  25
#define CMD_RC_WRITE_LCD_XY      26
#define CMD_RC_SET_LEDS          27
#define CMD_RC_GET_KEY           28

//remote control errors
#define C5_ERR_NO                 0    //no error
#define C5_ERR_BUSY               1    //device busy
#define C5_ERR_NOT_READY          2    //device not ready
#define C5_ERR_PARAMETERS         3    //parameters value error
#define C5_ERR_NO_REPLY           4    //no Reply
#define C5_ERR_MAX_POWER          5    //max. power exceeded
#define C5_ERR_OVERHEATING        6    //device overheating

//accu type
#define TYPE_NICD                 1
#define TYPE_NIMH                 2
#define TYPE_PB                   3
#define TYPE_LI_ION               4
#define TYPE_LI_POLYMER           5

#define ACCU_TYPE_MAX             5

//programs
#define PGM_CHARGE                 1    // charge
#define PGM_DISCHARGE              2    // discharge
#define PGM_DC                     3    // discharge-charge
#define PGM_CDC                   4    // charge-discharge-charge
#define PGM_CYCLE                  5    // cycle
#define PGM_FORMING                6    // forming

#define ACCU_PROGRAM_MAX          6

//DataFlash CMD's answers
```

```

#define DF_EMPTY 0
#define DF_MANUAL_CHARGE_DATA 1
#define DF_SERVICE_DATA 2

#define DF_END 0

enum DATA_FLASH_TYPE
{
    C5_DF_EMPTY=0,
    C5_DF_MANUAL_CHARGE_DATA=1,
    C5_DF_SERVICE_DATA=2
};

enum DEVICE_ACTIVITY
{
    C5_NO_ACTIVITY=0,
    C5_SERVICE_PAUSE=1,
    C5_SERVICE_RUNNING=2,
    C5_MANUAL_CHARGE_PAUSE=3,
    C5_MANUAL_CHARGE_RUNNING=4
};

enum REMOTE_STATE
{
    C5_REMOTE_NO_SERVICE = 0,
    C5_REMOTE_STARTING_CHARGE = 1,
    C5_REMOTE_CHARGE_SERVICE = 2,
    C5_REMOTE_STARTING_DISCHARGE = 3,
    C5_REMOTE_DISCHARGE_SERVICE = 4,
    C5_REMOTE_OVERHEATING = 5
};

#define REMOTE_TIMEOUT 10 //sec

//LED's
#define PIEZO 0x01
#define LED_CHARGE 0x02
#define LED_DISCH 0x04
#define LED_READY 0x08
#define LED_ERROR 0x10

//Keyboard
#define KEY_NULL 0
#define KEY_UP 1
#define KEY_LEFT 2
#define KEY_DOWN 3
#define KEY_RIGHT 4
#define KEY_ENTER 5

#define KEY_UP_RPT KEY_UP+0x80
#define KEY_LEFT_RPT KEY_LEFT+0x80
#define KEY_DOWN_RPT KEY_DOWN+0x80
#define KEY_RIGHT_RPT KEY_RIGHT+0x80
#define KEY_ENTER_RPT KEY_ENTER+0x80

//LCD: ASCII numbers for user's chars
#define CHAR_UP 8
#define CHAR_LEFT 9
#define CHAR_DOWN 10
#define CHAR_RIGHT 11
#define CHAR_ENTER 12
#define CHAR_UPDOWN 13
#define CHAR_LR 14

//max & min values
#define ACCU_CURRENT_CHARGE_MAX 5000 //mA
#define ACCU_CURRENT_DISCHARGE_MAX 5000 //mA
#define ACCU_CURRENT_MIN 50 //mA
#define MAX_CHARGE_POWER 70 //Watt
#define MAX_DISCHARGE_POWER 30 //Watt
#define MAX_VOLTAGE_OUT 38000 //mV
#define MIN_VOLTAGE_OUT 100 //mV
#define MAX_VOLTAGE_DROP 4000 //mV

//-- end of constants --//

//prototypes

```

```

BOOL C5_Send_Command(unsigned char ADDR, unsigned char CMD,
                    unsigned char N, unsigned char *Data);
BOOL C5_Get_Reply(unsigned char *ADD, unsigned char *CMD,
                 unsigned char *N, unsigned char *Data);

//-- end of prototypes --//

```

Der Befehlssatz des Gerätes kann man auf 5 Gruppen aufteilen:

1. Kommandos für allgemeine Geräteinformation
2. Kommandos für Lesen von eingestellten Parameter und ermittelten Werten (Logger-Funktion)
3. Kommandos für Auswählen des Betriebsmodus
4. Kommandos für Lesen des Flashspeichers (eingebauter Datenlogger)
5. Kommandos für Fernsteuerung des Gerätes

Jede Gruppe ist nur in einem speziell dazu vorgesehenen Gerätezustand erreichbar. Die Gruppen 1, 2, und 3 sind erreichbar im Hauptbetriebsmodus. Dieser Modus wird automatisch beim Einschalten des Gerätes eingestellt. Die Gruppen 4 und 5 sind erreichbar entsprechend im Flashmodus und im Fernsteuerungsmodus. Für die Gruppe 3 ist wichtig, dass das Gerät sich im Hauptmenu (Accu service, Manual Charge oder Settings) befindet. In anderem Fall meldet sich es als „beschäftigt“.

Jeder Datenaustausch kann nur Steuergerät (Computer oder externer Mikroprozessor) initiieren.

Beim Ausführen des Kommandos kann sich das Gerät mit folgenden Fehlern zurückmelden:

C5_ERR_NO	: kein Fehler;
C5_ERR_BUSY	: das Gerät ist beschäftigt;
C5_ERR_NOT_READY	: das Gerät ist nicht bereit;
C5_ERR_PARAMETERS	: die geschickten Parameter sind falsch/nicht akzeptabel;
C5_ERR_NO_REPLY	: keine Meldung von Peripherie-Gerät;
C5_ERR_MAX_POWER	: es wurde maximal mögliche Leistung erreicht;
C5_ERR_OVERHEATING	: das Gerät ist überhitzt

## Kommandos für allgemeine Geräteinformation

### 1. Kommando: CMD\_GET\_DEVICE\_NAME;

Beschreibung: Mit diesem Kommando wird der Name des Gerätes ausgelesen.

Verwendung:

Kommando senden:

```

ADD = C5_RX_ADDRESS;
CMD = CMD_GET_DEVICE_NAME;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

ADD:	Adresse des Gerätes;
CMD:	Kommando;
Nbyte:	Anzahl der Bytes zum Versenden;
IO_Frame:	Sendepuffer;

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:

ADD:	Adresse des Empfängers;
CMD:	ausgeführtes Kommando;
Nbyte:	Anzahl der empfangenen Bytes;
IO_Frame:	Empfangspuffer;

Beschreibung der empfangenen Daten:

Der Empfangspuffer enthält den Namen des Gerätes (in ASCII).  
Nbyte = Länge der Zeichenkette.

### 2. Kommando: CMD\_GET\_DEVICE\_VERSION;

Beschreibung: Mit diesem Kommando wird die Firmware-Version des Gerätes ausgelesen.

Verwendung:

Kommando senden:

```

ADD = C5_RX_ADDRESS;
CMD = CMD_GET_DEVICE_VERSION;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

ADD:	Adresse des Gerätes;
------	----------------------

```

CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:      Sendepuffer;

```

Antwort erhalten:

```
C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );
```

hier:

```

ADD:           Adresse des Empfängers;
CMD:           ausgeführtes Kommando;
Nbyte:         Anzahl der empfangenen Bytes;
IO_Frame:      Empfangspuffer;

```

Beschreibung der empfangenen Daten:

```

Nbyte = 2;
IO_Frame[0] = low byte der Version(minor);
IO_Frame[1] = high byte der Version(major);
Beispiel: IO_Frame[0] = 12; IO_Frame[1] = 3; => Die Version ist 3.12

```

## Kommandos für Logger-Funktion

### 1. Kommando: CMD\_GET\_SERVICE\_INFO;

Beschreibung: Mit diesem Kommando werden der Betriebszustand und die Parametereinstellungen des Gerätes ausgelesen.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_GET_SERVICE_INFO;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Gerätes;
CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:      Sendepuffer;

```

Antwort erhalten:

```
C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );
```

hier:

```

ADD:           Adresse des Empfängers;
CMD:           ausgeführtes Kommando;
Nbyte:         Anzahl der empfangenen Bytes;
IO_Frame:      Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Betriebszustand-Code.

Der Betriebszustand kann folgende Werte haben:

```

C5_NO_ACTIVITY,
C5_SERVICE_PAUSE,
C5_SERVICE_RUNNING,
C5_MANUAL_CHARGE_PAUSE,
C5_MANUAL_CHARGE_RUNNING;

```

Je nach Betriebszustand werden folgende Daten vom Ladegerät gesendet:

1. IO\_Frame[0] = C5\_NO\_ACTIVITY;  
Nbyte = 1;

Es wird keine zusätzliche Information geliefert.

2. IO\_Frame[0] = C5\_SERVICE\_RUNNING oder  
IO\_Frame[0] = C5\_SERVICE\_PAUSE;  
Nbyte=14;

```

IO_Frame[1] = service program;           (unsigned char)
IO_Frame[2] = accu type;                  (unsigned char)
IO_Frame[3] = accu voltage(nominal) (low byte);  mV, (unsigned short int - 2 byte)
IO_Frame[4] = accu voltage(nominal) (high byte);
IO_Frame[5] = accu capacity(nominal) (low byte);  mAh, (unsigned long - 4 byte))
IO_Frame[6] = accu capacity(nominal) (next byte);
IO_Frame[7] = accu capacity(nominal) (next byte);
IO_Frame[8] = accu capacity(nominal) (high byte);
IO_Frame[9] = accu charge current (low byte);    mA, (unsigned short int - 2 byte)
IO_Frame[10] = accu charge current (high byte);
IO_Frame[11] = accu discharge current (low byte); mA, (unsigned short int - 2 byte)

```

```

IO_Frame[12] = accu discharge current (high byte);
IO_Frame[13] = accu temperature max.;          °C, (signed char)

```

```

3. IO_Frame[0] = C5_MANUAL_CHARGE_RUNNING oder
   IO_Frame[0] = C5_MANUAL_CHARGE_PAUSE;
   Nbyte=10;

```

```

IO_Frame[1] = U accu max. (low byte);          mV, (unsigned short int - 2 byte)
IO_Frame[2] = U accu max. (high byte);
IO_Frame[3] = I accu max. (low byte);          mA, (unsigned short int - 2 byte)
IO_Frame[4] = I accu max. (high byte);
IO_Frame[5] = charge time max. (low byte);    sec, (unsigned long - 4 byte)
IO_Frame[6] = charge time max. (next byte);
IO_Frame[7] = charge time max. (next byte);
IO_Frame[8] = charge time max. (high byte);
IO_Frame[9] = accu temperature max.;          °C, (signed char)

```

hier:

```

C5_NO_ACTIVITY           : es wird kein Lade-/Entladevorgang durchgeführt;
C5_SERVICE_PAUSE        : ein Lade-/Entladevorgang hat eine Pause;
C5_SERVICE_RUNNING      : es läuft ein Lade-/Entladevorgang;
C5_MANUAL_CHARGE_PAUSE  : ein manueller Ladeladevorgang hat eine Pause;
C5_MANUAL_CHARGE_RUNNING : es läuft ein manueller Ladeladevorgang;

```

```

service program          : der Typ des Lade-/Entladeprogramm;
accu type                : der Typ des Akkus;
accu voltage (nominal)   : Nennwert der Akkuspannung;
accu capacity (nominal)  : Nennwert der Akkukapazität;
accu charge current      : eingestellter Ladestrom;
accu discharge current   : eingestellter Entladestrom;
accu temperature max.    : eingestellte maximal mögliche Akkutemperatur;
U accu max.              : eingestellte maximal mögliche Ladespannung;
I accu max.              : eingestellter maximal möglicher Ladestrom;
charge time max.         : eingestellte maximal mögliche Ladezeit;

```

## 2. Kommando: CMD\_GET\_SERVICE\_DATA;

Beschreibung: Mit diesem Kommando werden die Daten des Betriebszustandes ausgelesen.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_GET_SERVICE_DATA;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Gerätes;
CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:      Sendepuffer;

```

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Empfängers;
CMD:           ausgeführtes Kommando;
Nbyte:         Anzahl der empfangenen Bytes;
IO_Frame:      Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Betriebszustand-Code.

Der Betriebszustand kann folgende Werte haben:

```

C5_NO_ACTIVITY,
C5_SERVICE_PAUSE,
C5_SERVICE_RUNNING,
C5_MANUAL_CHARGE_PAUSE,
C5_MANUAL_CHARGE_RUNNING;

```

Je nach Betriebszustand werden folgende Daten vom Ladegerät gesendet:

```

1. IO_Frame[0] = C5_NO_ACTIVITY oder
   IO_Frame[0] = C5_SERVICE_PAUSE oder
   IO_Frame[0] = C5_MANUAL_CHARGE_PAUSE;
   Nbyte = 1;

```

Es wird keine zusätzliche Information geliefert.

```

2. IO_Frame[0] = C5_SERVICE_RUNNING oder
   IO_Frame[0] = C5_MANUAL_CHARGE_RUNNING;
   Nbyte=12;

   IO_Frame[1] = U accu load (low byte);           mV, (unsigned short int - 2 byte)
   IO_Frame[2] = U accu load (high byte);
   IO_Frame[3] = U accu unload (low byte);         mV, (unsigned short int - 2 byte)
   IO_Frame[4] = U accu unload (high byte);
   IO_Frame[5] = I accu charge/discharge (low byte); mA, (signed short int - 2 byte)
   IO_Frame[6] = I accu charge/discharge (high byte);
   IO_Frame[7] = charged/discharged capacity (low byte); mAh, (signed long - 4 byte)
   IO_Frame[8] = charged/discharged capacity (next byte);
   IO_Frame[9] = charged/discharged capacity (next byte);
   IO_Frame[10] = charged/discharged capacity (high byte);
   IO_Frame[11] = accu temperature;                °C, (signed char)

```

hier:

```

U accu load           : die gemessene Eingangsspannung bei Laden/Entladen
                      : (Lade-/Entladestrom fließt);
U accu unload         : die gemessene Eingangsspannung bei Laden/Entladen
                      : (Lade-/Entladestrom abgeschaltet);
charged/discharged capacity : gemessene Lade-/Entladekapazität;
accu temperature      : gemessene Akkutemperatur;

```

## Kommandos für Auswählen des Betriebsmodus

### 1. Kommando: CMD\_ENTER\_DF\_MODE;

Beschreibung: Mit diesem Kommando wird das Gerät in den Flashmodus versetzt. Es wird dabei eine Timeout-Funktion gestartet. Wenn innerhalb der eingestellten Zeit(REMOTE\_TIMEOUT) vom Steuergerät kein Kommando ausgeführt wird, kehrt das Gerät in den Hauptmodus zurück.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_ENTER_DF_MODE;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:      Adresse des Gerätes;
CMD:      Kommando;
Nbyte:    Anzahl der Bytes zum Versenden;
IO_Frame: Sendepuffer;

```

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:

```

ADD:      Adresse des Empfängers;
CMD:      ausgeführtes Kommando;
Nbyte:    Anzahl der empfangenen Bytes;
IO_Frame: Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

### 2. Kommando: CMD\_ENTER\_RC\_MODE;

Beschreibung: Mit diesem Kommando wird das Gerät in den Fernsteuerungsmodus versetzt. Es wird dabei eine Timeout-Funktion gestartet. Wenn innerhalb der eingestellten Zeit(REMOTE\_TIMEOUT) vom Steuergerät kein Kommando ausgeführt wird, kehrt das Gerät in den Hauptmodus zurück.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_ENTER_RC_MODE;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:      Adresse des Gerätes;
CMD:      Kommando;
Nbyte:    Anzahl der Bytes zum Versenden;
IO_Frame: Sendepuffer;

```

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

```

hier:
  ADD:           Adresse des Empfängers;
  CMD:           ausgeführtes Kommando;
  Nbyte:         Anzahl der empfangenen Bytes;
  IO_Frame:      Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

## Kommandos für Auslesen des Flashspeichers

### 1. Kommando: CMD\_DF\_GET\_INFO;

Beschreibung: Mit diesem Kommando wird die Information über Inhalt des Flashspeichers (Datenlogger) ausgelesen. Der Datenzeiger des Flashs wird auf ersten Datensatz eingestellt (falls vorhanden).

Verwendung:

```

Kommando senden:
  ADD = C5_RX_ADDRESS;
  CMD = CMD_DF_GET_INFO;
  Nbyte = 0;
  C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

```

hier:
  ADD:           Adresse des Gerätes;
  CMD:           Kommando;
  Nbyte:         Anzahl der Bytes zum Versenden;
  IO_Frame:      Sendepuffer;

```

Antwort erhalten:

```

  C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

```

hier:
  ADD:           Adresse des Empfängers;
  CMD:           ausgeführtes Kommando;
  Nbyte:         Anzahl der empfangenen Bytes;
  IO_Frame:      Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Flashtyp-Code.  
Der Flashtyp kann folgende Werte haben:  
DF\_EMPTY,  
DF\_MANUAL\_CHARGE\_DATA,  
DF\_SERVICE\_DATA;

Je nach Flashtyp werden folgende Daten vom Ladegerät gesendet:

1. IO\_Frame[0] = DF\_EMPTY;  
Nbyte = 1;

2. IO\_Frame[0] = DF\_MANUAL\_CHARGE\_DATA;  
Nbyte = 11;

```

IO_Frame[1] = DataFlash save interval;      sec,   (unsigned char)
IO_Frame[2] = U accu max. (low byte);        mV,    (unsigned short int - 2 byte)
IO_Frame[3] = U accu max. (high byte);
IO_Frame[4] = I accu max. (low byte);        mA,    (unsigned short int - 2 byte)
IO_Frame[5] = I accu max. (high byte);
IO_Frame[6] = charge time max. (low byte);   sec,    (unsigned long - 4 byte)
IO_Frame[7] = charge time max. (next byte);
IO_Frame[8] = charge time max. (next byte);
IO_Frame[9] = charge time max. (high byte);
IO_Frame[10] = accu temperature max.;        °C,    (signed char)

```

3. IO\_Frame[0] = DF\_SERVICE\_DATA;  
Nbyte = 15;

```

IO_Frame[1] = DataFlash save interval;      sec,   (unsigned char)
IO_Frame[2] = service program;              (unsigned char)
IO_Frame[3] = accu type;                    (unsigned char)
IO_Frame[4] = accu voltage(nominal)(low byte); mV,    (unsigned short int - 2 byte)
IO_Frame[5] = accu voltage(nominal)(high byte);
IO_Frame[6] = accu capacity(nominal)(low byte); mAh,  (unsigned long - 4 byte)
IO_Frame[7] = accu capacity(nominal)(next byte);
IO_Frame[8] = accu capacity(nominal)(next byte);
IO_Frame[9] = accu capacity(nominal)(high byte);
IO_Frame[10] = accu charge current(low byte); mA,    (unsigned short int - 2 byte)

```

```

IO_Frame[11] = accu charge current(high byte);
IO_Frame[12] = accu discharge current(low byte); mA, (unsigned short int - 2 byte)
IO_Frame[13] = accu discharge current(high byte);
IO_Frame[14] = accu temperature max.; °C, (signed char)

```

hier:

DataFlash save interval: Zeitabstand zwischen Messungen;

für den Rest: siehe oben;

## 2. Kommando: CMD\_DF\_PREPARE\_DATA;

Beschreibung: Mit diesem Kommando werden die Daten im Flash für das Auslesen vorbereitet. Dabei wird der Datenzeiger des Flashs auf ersten Datensatz eingestellt. Dieses Kommando ist für einen abgebrochenen Lesevorgang des Flashspeichers vorgesehen.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_DF_PREPARE_DATA;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Gerätes;
CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:     Sendepuffer;

```

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Empfängers;
CMD:           ausgeführtes Kommando;
Nbyte:         Anzahl der empfangenen Bytes;
IO_Frame:     Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

## 3. Kommando: CMD\_DF\_GET\_DATA;

Beschreibung: Mit diesem Kommando wird der laufenden Datensatz des Flashs ausgelesen. Danach wird der Datenzeiger des Flashs auf nächsten Datensatz eingestellt.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_DF_GET_DATA;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Gerätes;
CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:     Sendepuffer;

```

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Empfängers;
CMD:           ausgeführtes Kommando;
Nbyte:         Anzahl der empfangenen Bytes;
IO_Frame:     Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Wenn ein Datensatz vorhanden ist, werden 9 Bytes als Antwort zurückgesendet, wenn nicht: 1Byte.

1. Nbyte = 1;  
IO\_Frame[0] = DF\_EMPTY;
2. Nbyte = 9;  
IO\_Frame[0] = U accu unload (low byte); mV, (unsigned short int - 2 byte)  
IO\_Frame[1] = U accu unload (high byte);



```

IO_Frame[2] = I accu charge/discharge (low byte);      mA, (signed short int - 2 byte)
IO_Frame[3] = I accu charge/discharge (high byte);
IO_Frame[4] = charged/discharged capacity (low byte);  mAh, (signed long - 4 byte)
IO_Frame[5] = charged/discharged capacity (next byte);
IO_Frame[6] = charged/discharged capacity (next byte);
IO_Frame[7] = charged/discharged capacity (high byte);
IO_Frame[8] = accu temperature;                        °C, (signed char)

```

#### 4. Kommando: **CMD\_DF\_GET\_DATA\_BULK;**

Beschreibung: wie das Kommando **CMD\_DF\_GET\_DATA**, aber wenn das Kommando abgeschlossen ist, startet es sich automatisch wieder. Es wird so oft passieren, bis das Ende der Log-Daten festgestellt wird. In diesem Fall werden die ganzen Daten vom Ladegerät mit maximal möglicher Geschwindigkeit zum PC gesendet. Die „Datenflut“ kann abgebrochen werden (falls es nötig ist), wenn ein anderes Kommando aus der „CMD\_DF“-Gruppe ans Gerät vom PC gesendet wird.

#### 5. Kommando: **CMD\_DF\_EXIT;**

Beschreibung: Mit diesem Kommando wird das Gerät in den Hauptmodus zurückgesetzt.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_DF_EXIT;
Nbyte = 0;
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Gerätes;
CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:     Sendepuffer;

```

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Empfängers;
CMD:           ausgeführtes Kommando;
Nbyte:         Anzahl der empfangenen Bytes;
IO_Frame:     Empfangspuffer;

```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

## Kommandos für Fernsteuerung

#### 1. Kommando: **CMD\_RC\_START\_CHARGE;**

Beschreibung: Mit diesem Kommando wird ein Ladevorgang gestartet.

Verwendung:

```

Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_RC_START_CHARGE;

```

```

IO_Frame[0] = U accu max (low byte);    mV, (unsigned short int - 2 byte)
IO_Frame[1] = U accu max (high byte);
IO_Frame[2] = I accu max (low byte);    mA, (unsigned short int - 2 byte)
IO_Frame[3] = I accu max (high byte);

```

Nbyte = 4;

```

C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );

```

hier:

```

ADD:           Adresse des Gerätes;
CMD:           Kommando;
Nbyte:         Anzahl der Bytes zum Versenden;
IO_Frame:     Sendepuffer;

```

U accu max: maximal mögliche Ausgangsspannung;

I accu max: maximal möglicher Ausgangsstrom;

Die maximal mögliche Leistung sollte dabei nicht überschritten werden( $U \cdot I \leq \text{MAX\_CHARGE\_POWER}$ ).

Antwort erhalten:

```

C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );

```

hier:  
ADD:                   Adresse des Empfängers;  
CMD:                   ausgeführtes Kommando;  
Nbyte:                  Anzahl der empfangenen Bytes;  
IO\_Frame:              Empfangspuffer;

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

## **2. Kommando: CMD\_RC\_START\_DISCHARGE;**

Beschreibung: Mit diesem Kommando wird ein Entladevorgang gestartet.

Verwendung:

Kommando senden:  
ADD = C5\_RX\_ADDRESS;  
CMD = CMD\_RC\_START\_DISCHARGE;

IO\_Frame[0] = I accu discharge max (low byte);   mA, (unsigned short int - 2 byte)  
IO\_Frame[1] = I accu discharge max (high byte);

Nbyte = 2;

C5\_Send\_Command(ADD, CMD, Nbyte, IO\_Frame );

hier:  
ADD:                   Adresse des Gerätes;  
CMD:                   Kommando;  
Nbyte:                  Anzahl der Bytes zum Versenden;  
IO\_Frame:              Sendepuffer;

I accu discharge max: maximal möglicher Entladestrom;  
Die maximal mögliche Leistung sollte dabei nicht überschritten  
werden( $U \cdot I \leq \text{MAX\_DISCHARGE\_POWER}$ ).

Antwort erhalten:

C5\_Get\_Reply( &ADD, &CMD, &Nbyte, IO\_Frame );

hier:  
ADD:                   Adresse des Empfängers;  
CMD:                   ausgeführtes Kommando;  
Nbyte:                  Anzahl der empfangenen Bytes;  
IO\_Frame:              Empfangspuffer;

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

## **3. Kommando: CMD\_RC\_STOP\_SERVICE;**

Beschreibung: Mit diesem Kommando wird ein laufender Lade-/Entladevorgang beendet.

Verwendung:

Kommando senden:  
ADD = C5\_RX\_ADDRESS;  
CMD = CMD\_RC\_STOP\_SERVICE;  
Nbyte = 0;  
C5\_Send\_Command(ADD, CMD, Nbyte, IO\_Frame );

hier:  
ADD:                   Adresse des Gerätes;  
CMD:                   Kommando;  
Nbyte:                  Anzahl der Bytes zum Versenden;  
IO\_Frame:              Sendepuffer;

Antwort erhalten:

C5\_Get\_Reply( &ADD, &CMD, &Nbyte, IO\_Frame );

hier:  
ADD:                   Adresse des Empfängers;  
CMD:                   ausgeführtes Kommando;  
Nbyte:                  Anzahl der empfangenen Bytes;  
IO\_Frame:              Empfangspuffer;

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

#### 4. Kommando: **CMD\_RC\_GET\_SERVICE\_DATA;**

Beschreibung: Mit diesem Kommando werden die Messdaten ausgelesen.

Verwendung:

```
Kommando senden:  
ADD = C5_RX_ADDRESS;  
CMD = CMD_RC_GET_SERVICE_DATA;  
Nbyte = 0;  
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );
```

hier:

```
ADD:           Adresse des Gerätes;  
CMD:           Kommando;  
Nbyte:         Anzahl der Bytes zum Versenden;  
IO_Frame:      Sendepuffer;
```

Antwort erhalten:

```
C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );
```

hier:

```
ADD:           Adresse des Empfängers;  
CMD:           ausgeführtes Kommando;  
Nbyte:         Anzahl der empfangenen Bytes;  
IO_Frame:      Empfangspuffer;
```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Betriebszustand-Code.

Der Betriebszustand kann folgende Werte haben:

```
C5_REMOTE_NO_SERVICE,  
C5_REMOTE_STARTING_CHARGE,  
C5_REMOTE_CHARGE_SERVICE,  
C5_REMOTE_STARTING_DISCHARGE,  
C5_REMOTE_DISCHARGE_SERVICE,  
C5_REMOTE_OVERHEATING;
```

Je nach Betriebszustand werden folgende Daten vom Ladegerät gesendet:

1. IO\_Frame[0] = C5\_REMOTE\_NO\_SERVICE oder  
IO\_Frame[0] = C5\_REMOTE\_STARTING\_CHARGE oder  
IO\_Frame[0] = C5\_REMOTE\_STARTING\_DISCHARGE oder  
IO\_Frame[0] = C5\_REMOTE\_OVERHEATING;

Nbyte = 4;

```
IO_Frame[1] = U accu unload (low byte);           mV, (unsigned short int - 2 byte)  
IO_Frame[2] = U accu unload (high byte);  
IO_Frame[3] = accu temperature;                     °C, (signed char)
```

2. IO\_Frame[0] = C5\_C5\_REMOTE\_CHARGE\_SERVICE oder  
IO\_Frame[0] = C5\_REMOTE\_DISCHARGE\_SERVICE;

Nbyte = 8;

```
IO_Frame[1] = U accu load (low byte);              mV, (unsigned short int - 2 byte)  
IO_Frame[2] = U accu load (high byte);  
IO_Frame[3] = U accu unload (low byte);            mV, (unsigned short int - 2 byte)  
IO_Frame[4] = U accu unload (high byte);  
IO_Frame[5] = I accu charge/discharge (low byte);  mA, (signed short int - 2 byte)  
IO_Frame[6] = I accu charge/discharge (high byte);  
IO_Frame[7] = accu temperature;                     °C, (signed char)
```

hier:

```
C5_REMOTE_NO_SERVICE           : es wird kein Lade-/Entladevorgang durchgeführt;  
C5_REMOTE_STARTING_CHARGE      : es wird ein Ladevorgang gestartet;  
C5_REMOTE_CHARGE_SERVICE       : es läuft ein Ladevorgang;  
C5_REMOTE_STARTING_DISCHARGE   : es wird ein Entladevorgang gestartet;  
C5_REMOTE_DISCHARGE_SERVICE    : es läuft ein Entladevorgang;  
C5_REMOTE_OVERHEATING          : das Gerät ist überhitzt;
```

für den Rest: siehe oben;

#### 5. Kommando: **CMD\_RC\_WRITE\_LCD\_XY;**

Beschreibung: Mit diesem Kommando wird zum Display des Gerätes eine Zeichenkette gesendet. Das Kommando ist für eine Steuerung mit einem externen Mikroprozessor vorgesehen.

Verwendung:

```
Kommando senden:  
ADD = C5_RX_ADDRESS;  
CMD = CMD_RC_WRITE_LCD_XY;
```

```
IO_Frame[0] = x-coord. , 0..15
IO_Frame[1] = y-coord. , 0..1
IO_Frame[2..17] = Zeichenkette
```

```
Nbyte = 3 .. 18;
```

```
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );
```

```
hier:
  ADD:           Adresse des Gerätes;
  CMD:           Kommando;
  Nbyte:         Anzahl der Bytes zum Versenden;
  IO_Frame:      Sendepuffer;
```

```
Antwort erhalten:
  C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );
```

```
hier:
  ADD:           Adresse des Empfängers;
  CMD:           ausgeführtes Kommando;
  Nbyte:         Anzahl der empfangenen Bytes;
  IO_Frame:      Empfangspuffer;
```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

#### **6. Kommando: CMD\_RC\_SET\_LEDS;**

Beschreibung: Mit diesem Kommando werden die LEDs des Gerätes gesteuert. Das Kommando ist für eine Steuerung mit einem externen Mikroprozessor vorgesehen.

Verwendung:

```
Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_RC_SET_LEDS;
```

```
IO_Frame[0] = LEDs;
```

```
Nbyte = 1;
```

```
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );
```

```
hier:
  ADD:           Adresse des Gerätes;
  CMD:           Kommando;
  Nbyte:         Anzahl der Bytes zum Versenden;
  IO_Frame:      Sendepuffer;
```

```
Antwort erhalten:
  C5_Get_Reply( &ADD, &CMD, &Nbyte, IO_Frame );
```

```
hier:
  ADD:           Adresse des Empfängers;
  CMD:           ausgeführtes Kommando;
  Nbyte:         Anzahl der empfangenen Bytes;
  IO_Frame:      Empfangspuffer;
```

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;

#### **7. Kommando: CMD\_RC\_GET\_KEY;**

Beschreibung: Mit diesem Kommando wird der Joystick des Gerätes abgefragt. Das Kommando ist für eine Steuerung mit einem externen Mikroprozessor vorgesehen. Die Abfrage des Joysticks sollte nicht öfter als 20 Mal pro Sekunde stattfinden.

Verwendung:

```
Kommando senden:
ADD = C5_RX_ADDRESS;
CMD = CMD_RC_GET_KEY;
```

```
Nbyte = 0;
```

```
C5_Send_Command(ADD, CMD, Nbyte, IO_Frame );
```

```
hier:
  ADD:           Adresse des Gerätes;
  CMD:           Kommando;
  Nbyte:         Anzahl der Bytes zum Versenden;
```

IO\_Frame:            Sendepuffer;

Antwort erhalten:

C5\_Get\_Reply( &ADD, &CMD, &Nbyte, IO\_Frame );

hier:

ADD:                Adresse des Empfängers;  
CMD:                ausgeführtes Kommando;  
Nbyte:              Anzahl der empfangenen Bytes;  
IO\_Frame:           Empfangspuffer;

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Key-Code.

#### **8. Kommando: CMD\_RC\_EXIT;**

Beschreibung: Mit diesem Kommando wird das Gerät in den Hauptmodus zurückgesetzt.

Verwendung:

Kommando senden:

ADD = C5\_RX\_ADDRESS;

CMD = CMD\_RC\_EXIT;

Nbyte = 0;

C5\_Send\_Command(ADD, CMD, Nbyte, IO\_Frame );

hier:

ADD:                Adresse des Gerätes;  
CMD:                Kommando;  
Nbyte:              Anzahl der Bytes zum Versenden;  
IO\_Frame:           Sendepuffer;

Antwort erhalten:

C5\_Get\_Reply( &ADD, &CMD, &Nbyte, IO\_Frame );

hier:

ADD:                Adresse des Empfängers;  
CMD:                ausgeführtes Kommando;  
Nbyte:              Anzahl der empfangenen Bytes;  
IO\_Frame:           Empfangspuffer;

Beschreibung der empfangenen Daten:

Das Byte IO\_Frame[0] enthält den Fehler-Code. Wenn IO\_Frame[0] = C5\_ERR\_NO --> kein Fehler;  
Nbyte=1;