Bedienungsanleitung „SETTER“ 1.2.0 R 0

RAMSER ELEKTROTECHNIK

www.ramser-elektro.com

Friedrich Ramser

Ramser Elektrotechnik

Copyright © by „Ramser Elektrotechnik“ 2022
Sämtliche Rechte vorbehalten. Dieses Handbuch darf nur mit schriftlicher Zustimmung von „Ramser Elektrotechnik“ vervielfältigt bzw. veröffentlicht werden. Auch nicht auszugsweise! Für Fehler technischer oder drucktechnischer Art und ihre Folgen übernehmen wir keine Haftung.
Alle Warenzeichen und Schutzrechte werden anerkannt.Änderungen im Sinne der Produktverbesserung vorbehalten. Die in den Beispielen verwendeten Firmen, Organisationen, Produkte, Domänennamen, E-Mail-Adressen, Logos, Personen, Orte und Ereignisse sind frei erfunden, soweit nichts anderes angegeben ist. Jede Ähnlichkeit mit bestehenden Firmen, Organisationen, Produkten, Domänennamen, E-Mail-Adressen, Logos, Personen, Orten oder Ereignissen ist rein zufällig.

 Copyright © by Ramser Elektrotechnik 2022

Vorwort:

Immer wieder kommt es vor, dass zwischen verschiedenen Systemen (wie PC, µC, …) und einem Mikrocontroller ein bidirektionaler Datenaustausch stattfinden muss.

Dieser sollte einfach, stabil und verlässlich ablaufen.

Das „SETTER Protokoll“ wurde genau für diese Voraussetzungen entwickelt.

Um es auch für den Endbenutzer zu ermöglichen, einzelne Werte von Baugruppen usw. zu ändern, wurde das Programm „Setter“ erstellt, welches als Freeware von unserer Homepage bezogen werden kann.

Change Log:

R 0: Erste Version

.

Inhalt

Vorwort: 1
Change Log: 1
1. Verbinden von Hardware oder Baugruppen mit einem Host 3
1.1. Generelles 3
1.2. USB / TTL Adapter mit DIP8 Adapter 3
2. Bedienung der „Setter“ Software 4
2.1. Allgemeine technische Daten 4
2.2. Hard- und Software Voraussetzungen 4
2.3. Starten von „Setter“ 5
2.4. Beschreibung des Menüs und der Button 6
2.5. Ablauf zum Ändern von Einstellungen 6
2.6. Ablauf zum Lesen von Einstellungen 6
3. Aufbau des Protokolls 7
3.1. Allgemeine Daten 7
3.2. Aufbau eines „Pakets“ 7
3.3. Rückgabewerte vom µC 8
3.4. Beispiele 8
3.5. Auflistung aller lesende Funktionen 9
3.6. Auflistung aller schreibende Funktionen 10
4. Verfügbare Funktionen nach Hardware 11
4.1. Powerbank Anti Off 11

1. Verbinden von Hardware oder Baugruppen mit einem Host

1.1. Generelles

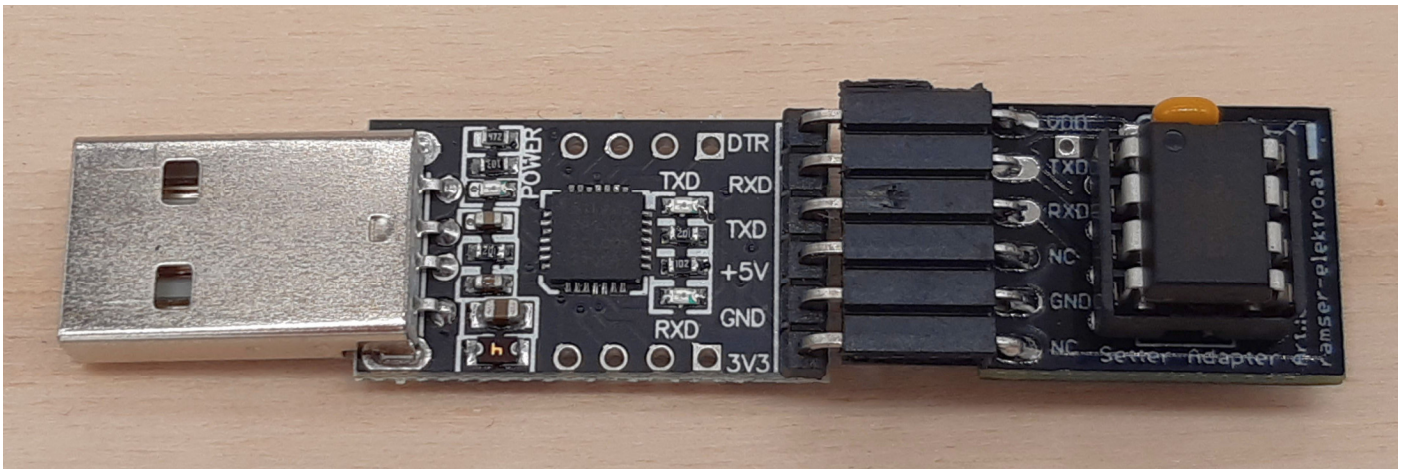
Der µC wird z.Bsp. mittels USB / TTL Adapter und einer Adapterplatine an ein freies USB-Port des Rechners gesteckt.

Die Kommunikation erfolgt danach im TTL Level vom µC zum Host.

Vom Host zum µC muss TX und RX gedreht (gekreuzt) werden!

Dies kann z.Bsp. mittels Adapterplatine geschehen.

1.2. USB / TTL Adapter mit DIP8 Adapter



- Artikelnummer: 002024
- Link: USB auf UART TTL Adapter mit DTR Ausgang CP210 - Mit DIP 8 adapter

Für folgende Bausätze/Hardware verwendbar:

- Powerbank Anti Off (FW Version: >= R5, PCB Version: RX (egal))

2. Bedienung der „Setter“ Software

2.1. Allgemeine technische Daten

- „Setter“ wurde in der .Net Plattform von Microsoft für Windows erstellt.

- Sämtliche Komponenten sind selbst erstellt worden, um einer Abhängigkeit von Drittanbietern zu entgehen und Zukunftsorientiert zu agieren.

2.2. Hard- und Software Voraussetzungen

Zum Ausführen von „Setter“ ist folgende Minimalkonfiguration Ihrer Hard und Software Voraussetzung:

Prozessor: Mindestens 1Ghz

Speicherausstattung: Für den Betrieb von „Setter“ werden mindestens 256MB freier Arbeitsspeicher benötigt.

Installiertes .Net Framework Version 4.5

Unterstütze Betriebssysteme:

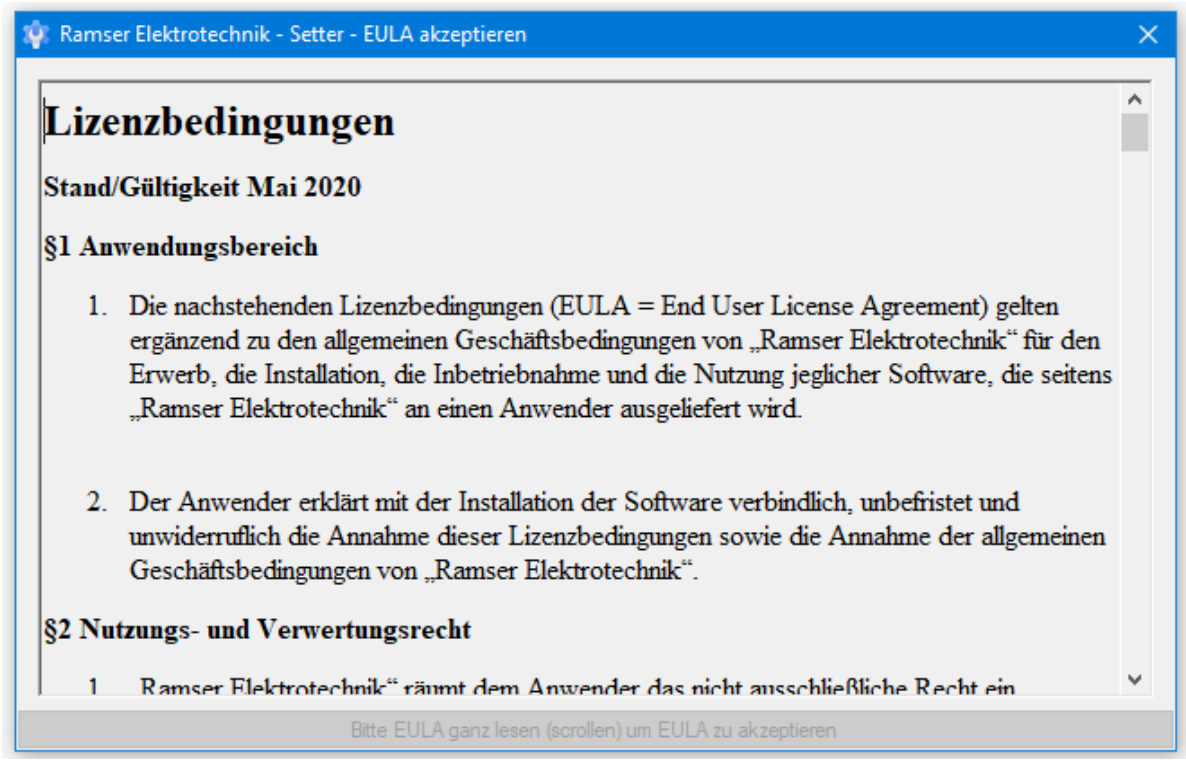
- Microsoft Windows 11
- Microsoft Windows 10
- Microsoft Windows 8.1
- Microsoft Windows 7 SP1
- Microsoft Windows Server 2022 (Essentials, Standard, Datacenter)*
- Microsoft Windows Server 2019 (Essentials, Standard, Datacenter)*
- Microsoft Windows Server 2016 (Essentials, Standard, Datacenter)*
- Microsoft Windows Server 2012 R2 (Foundation, Essentials, Standard, Datacenter)*
- Microsoft Windows Server 2012 (Foundation, Essentials, Standard, Datacenter)*
- Microsoft Windows Small Business Server 2011 SP1*
- Microsoft Windows Server 2008 R2 SP1 (Foundation, Standard, Enterprise, Datacenter)*

Der Benutzer benötigt keine Administratorrechte.

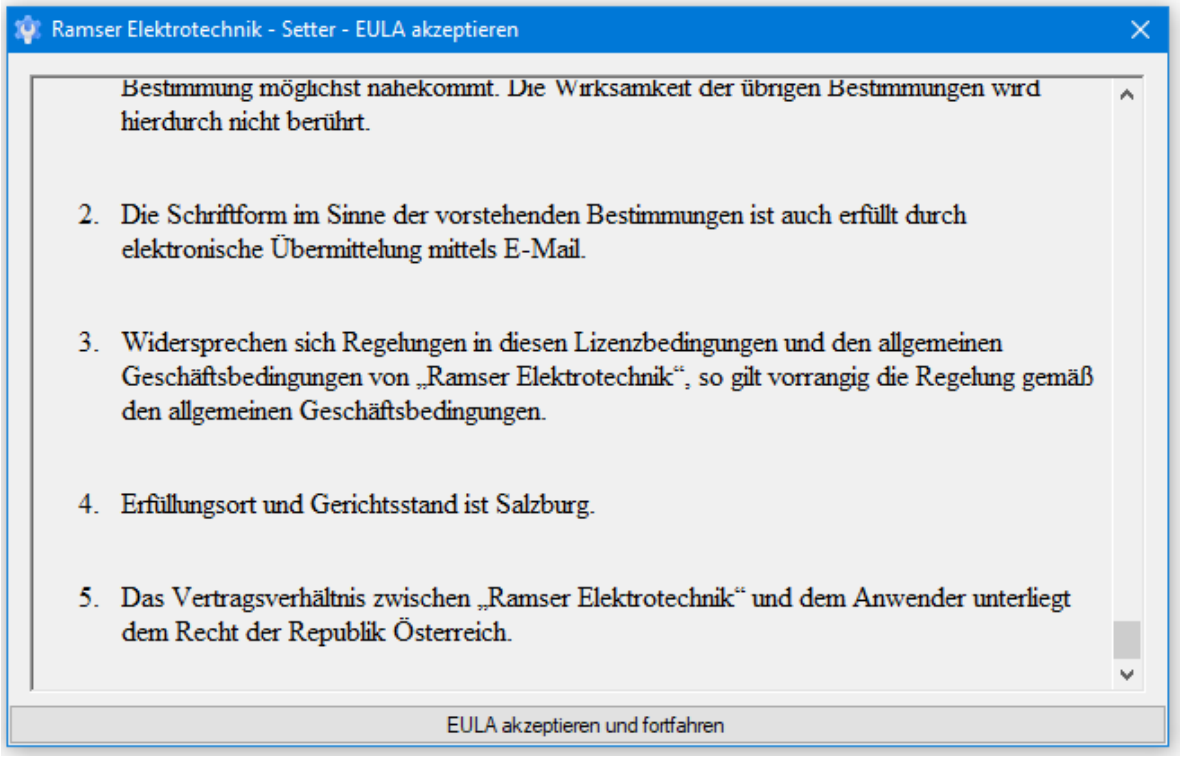
Es muss ein passender USB / TTL Adapter mit dem PC verbunden und als COM Port erkannt werden!

2.3. Starten von „Setter“

Beim Setter handelt es sich um ein „1 EXE Programm“.
Außer der gedownloadeten EXE File, wird keine weitere Datei benötigt.
Die EXE wird mit einem Doppelklick gestartet.



Danach müssen die Benutzerbestimmungen (EULA) gelesen und zugestimmt werden.
Dies geschieht, indem die Benutzerbestimmungen (EULA) gelesen und ganz nach unten gescrollt wird.



Danach kann mit dem Button „EULA akzeptieren und fortfahren“ der EULA zugestimmt werden. Die Einstellung wird dauerhaft gespeichert.

Mit der Bestätigung der Benutzerbestimmungen (EULA), wird das Hauptmenü angezeigt.

2.4. Beschreibung des Menüs und der Button

Menüeintrag:

- Datei – Beenden:
  Beendet das Programm. Eventuelle Einstellungen bleiben gespeichert.

- Hardware:
  Wählen Sie die passende Hardware aus. Z.Bsp. „Powerbank Anti Off“

- COM-Port:
  Wählen Sie das richtige COM Port aus. Mit dem Button „Liste aktualisieren“, werden die verfügbaren COM-Ports in die Liste eingetragen.

- ? – Infos zu den Icons:
  Gibt die generellen Copyright Infos für die verwendeten Icons wieder

- ? – COM Status anzeigen:
  Zeigt die ganze Kommunikation zwischen Hardware und PC an.
  Einzelne Einträge können markiert und danach mit dem Betätigen der rechten Maustaste in die Zwischenablage kopiert werden bzw. kann auch die ganze Liste geleert werden.

  ? – Fehler anzeigen
  Zeigt die eventuell aufgetretenen Fehlermeldungen an.
  Einzelne Einträge können markiert und danach mit dem Betätigen der rechten Maustaste in die Zwischenablage kopiert werden bzw. kann auch die ganze Liste geleert werden.

Buttons:
Generell: Buttons können von der ausgewählten Hardware abhängig sein!

- „Lese Einstellungen“:
  Liest die Einstellungen bzw. aktuellen Werte der verbunden Hardware ein und zeigt diese in den einzelnen Eingabefelder an. Sollte ein Fehler auftreten, so wird dieser angezeigt.

- „Schreibe Einstellungen“:
  Schreibt die in den jeweiligen Eingabefeldern eingetragenen Werte zu der verbundenen Hardware. Sollte ein Fehler auftreten, so wird dieser angezeigt.

- „Zurücksetzen“:
  Setzt die Eingabefelder auf die Werkseinstellungen zurück.

2.5. Ablauf zum Ändern von Einstellungen

1. Verbinden Sie die jeweilige Hardware mit dem PC
2. Vergewissern Sie sich, dass der jeweilige Adapter als COM Port erkannt wurde
3. Starten sie den „Setter“ und wählen Sie das richtige COM-Port und Hardware aus
4. Nehmen Sie die gewünschten Einstellungen in den Eingabefelder vor
5. Betätigen Sie den Button „Schreibe Einstellungen“
6. Die Einstellungen wurden übernommen.

2.6. Ablauf zum Lesen von Einstellungen

1. Verbinden Sie die jeweilige Hardware mit dem PC
2. Vergewissern Sie sich, dass der jeweilige Adapter als COM Port erkannt wurde
3. Starten sie den „Setter“ und wählen Sie das richtige COM-Port und Hardware aus
4. Betätigen Sie den Button „Lese Einstellungen“
5. Die Einstellungen werden angezeigt.

3. Aufbau des Protokolls

3.1. Allgemeine Daten

Basiert auf eine serielle Kommunikation, deren Einstellungen immer gleich sind:

- Baudrate: 9600 Baud
- Datenbits: 8
- Stoppbits: 1
- Parität: Keine
- Flussteuerung: Keine

3.2. Aufbau eines „Pakets“

- Es werden immer 5 Bytes übertragen.
- Werden weniger, mehr oder fehlerhafte Bytes übertragen, so wird dies durch Checksummen erkannt.
- Es gibt lesende und schreibende Funktionen.
- Wird ein Paket zum µC übertragen, so sendet dieser auch jedes Mal ein Paket (gleicher Aufbau) zurück

Senderichtung →				
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Command	High Byte	Low Byte	Checksum	Endmarker

Bedeutung / Definition der Bytes	
Command	Gibt an, welcher Befehl / welche Funktion ausgeführt werden soll Funktionscodes sind fix definiert. Jede Hardware verwendet andere bzw. eventuell auch gleiche Funktionscodes.
High Byte & Low Byte	Enthält die Nutzdaten aufgeteilt in Lo und Hi Teil eines Uint16 oder Byte. Bei lesenden Funktionen ist das Byte1 (High Byte) immer 0x01 und das Byte2 (Low Byte) immer 0x02. Bei verschiedenen Hardwaresystemen kann der Wertebereich eingeschränkt sein!!!
Checksum	Prüfsumme. Aufbau: Byte0 XOR Byte1 XOR Byte2
Endmarker	Markiert das Paketende. Ist immer 0xFF

Beispiele für Lo / Hi Byte Aufbau eines Uint16 Wertes			
Wert der Nutzdaten		Byte 1	Byte 2
Dezimal	Hex	High Byte	Low Byte
0	0x00	0x00	0x00
1	0x01	0x00	0x01
160	0xA0	0x00	0xA0
255	0xFF	0x00	0xFF
256	0x100	0x01	0x00
300	0x12C	0x01	0x2C
4000	0xFA0	0x0F	0xA0
12000	0x2EE0	0x2E	0xE0
16383	0x3FFF	0x3F	0xFF

3.3. Rückgabewerte vom µC

- Lesende Operation:
  Wird eine lesende Funktion vom PC (o.ä.) an dem µC übertragen, so sendet dieser wieder ein 5 Byte Paket zurück, welches den aktuellen Wert des µC enthält.
  Also: Gleicher Funktionscode, Aktuelle Nutzdaten vom µC, Prüfsumme, Endmarker

Pakete bei lesendem Zugriff					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	Command	High Byte	Low Byte	Checksum	Endmarker
Vom µC empfangen	Command	Aktuelle Daten des µC		Checksum	Endmarker

- Schreibende Operation:
  Es wird dasselbe Paket zurückgesendet

Pakete bei schreibenden Zugriff					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	Command	High Byte	Low Byte	Checksum	Endmarker
Vom µC empfangen	Command	High Byte	Low Byte	Checksum	Endmarker

- Nullpaket:
  Wird bei einer Operation das Paket „0x00, 0x00, 0x00, 0x00, 0x00,“ zurückgegeben, so wurde das Paket nicht richtig vom µC empfangen (z.Bsp.: Prüfsumme falsch).

3.4. Beispiele

Lesen Hardware ID. Rückgabewert: 0x01 (=1dec)					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	0x01	0x01	0x02	0x02	0xFF
Vom µC empfangen	0x01	0x00	0x01	0x00	0xFF

Lese Firmware Revision. Rückgabewert: 0x05 (=5dec)					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	0x02	0x01	0x02	0x01	0xFF
Vom µC empfangen	0x02	0x00	0x05	0x07	0xFF

Lese Einstellung „Einschaltverzögerung Ausgang“. Rückgabewert: 0x50 (= 160dec)					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	0x63	0x01	0x02	0x60	0xFF
Vom µC empfangen	0x63	0x00	0x50	0x33	0xFF

Schreibe Einstellung „Einschaltverzögerung Ausgang“. Wert: 0xA0 (= 160dec)					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	0x84	0x00	0xA0	0x24	0xFF
Vom µC empfangen	0x84	0x00	0xA0	0x24	0xFF

Schreibe Einstellung „Einschaltverzögerung Ausgang“. Wert: 0x0FA0 (= 4000dec)					
Operation	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Zum µC gesendet	0x85	0x0F	0xA0	0x2A	0xFF
Vom µC empfangen	0x85	0x0F	0xA0	0x2A	0xFF

3.5. Auflistung aller lesende Funktionen

Lesende Funktionen (Byte 1 immer 0x01, Byte 2 immer 0x02) - PC liest Daten vom µC					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Funktion / Bedeutung
Command	High Byte	Low Byte	Checksum	Endmarker	
0x01	0x01	0x02	Prüfsumme	0xFF	Lese Hardware ID
0x02	0x01	0x02	Prüfsumme	0xFF	Lese Firmware Revision
0x10	0x01	0x02	Prüfsumme	0xFF	Lese Statuswort 1
0x15	0x01	0x02	Prüfsumme	0xFF	Lese System Modus
0x20	0x01	0x02	Prüfsumme	0xFF	Lese aktuelle VDD Spannung
0x21	0x01	0x02	Prüfsumme	0xFF	Lese Wert von ADC1
0x22	0x01	0x02	Prüfsumme	0xFF	Lese Wert von ADC2
0x23	0x01	0x02	Prüfsumme	0xFF	Lese Differenz von ADC1 - ADC2
0x60	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung vom „Schaltpunkt Taster 1“
0x61	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung vom „PowerUpTimer“
0x62	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung der „Entprellzeit“
0x63	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung der „Einschaltverzögerung Ausgang“
0x64	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung der „Puls Zeit Ausgang“
0x65	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung der „Minimum Auszeit des Ausgangs“
0x66	0x01	0x02	Prüfsumme	0xFF	Lese Einstellung des „Software BOR“

3.6. Auflistung aller schreibende Funktionen

Schreibende Funktionen - PC schreibt Daten zum µC					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Funktion / Bedeutung
Command	High Byte	Low Byte	Checksum	Endmarker	
0x70	0x00	LowByte	Prüfsumme	0xFF	Schreibe System Modus
0x75	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Statuswort 1
0x81	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung vom „Schaltpunkt Taster 1“
0x82	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung vom „PowerUpTimer“
0x83	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung der „Entprellzeit“
0x84	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung der „Einschaltverzögerung Ausgang“
0x85	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung der „Puls Zeit Ausgang“
0x86	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung der „Minimum Auszeit des Ausgangs“
0x87	HighByte	LowByte	Prüfsumme	0xFF	Schreibe Einstellung des „Software BOR“

4. Verfügbare Funktionen nach Hardware

4.1. Powerbank Anti Off

Hardwareeigenschaften:
- µC: PIC12F1572
- Maximaler Wert der Nutzdaten: 0x3FFF (=16383dec)
- Hardware ID: 0x01 (=1dec)
- Unterstützte FW: > 0x05 (=5dec) (R.5 - Erste Auslieferung KW34/2022 - 19.08.2022)

Unterstütze lesende Funktionen:
- 0x01 - Lese Hardware ID
- 0x02 - Lese Firmware Revision
- 0x63 - Lese Einstellung der „Einschaltverzögerung Ausgang“
- 0x64 - Lese Einstellung der „Puls Zeit Ausgang“

Unterstütze schreibende Funktionen:
(Wertebereich jeweils 0.1-600.0 Sekunden)
- 0x84 - Schreibe Einstellung der „Einschaltverzögerung Ausgang“
- 0x85 - Schreibe Einstellung der „Puls Zeit Ausgang“