

Referenz

Programmer

Produkt:	Programmer
Dokument:	Referenz Programmer
Version Dokument:	V3.10 Doc#: P0740
Version SW:	V3.10
Klassifizierung:	-
Dateiname:	ProgrammerReferenz.doc, 28.06.2017 13:54:00

Alle Marken-, Firmen- und Produktnamen in dieser Dokumentation werden nur für Erklärungen verwendet und sind in der Regel durch deren Inhaber geschützt.

Diese Dokumentation oder Teile daraus dürfen nicht ohne schriftliche Genehmigung von EDX Software Design in irgendeiner Form verarbeitet, kopiert, vervielfältigt oder verbreitet werden.

© EDX Software Design. Alle Rechte vorbehalten.

Änderungswesen

Version	Datum	Beschreibung der Änderung
3.10	16.06.17	EDX Software Design, E.Furrer + Erweiterungen für RX100 Derivate + Unterstützte Typen "R8C/Tiny" durch "R8C Family" neu mit Mx Series erweitert + Konfigurationsdatei angepasst Abschnitt <code>Target</code> und <code>Type</code> neu <code>R8C</code> + Erweiterung mit <code>Source Type</code> für DAT Datei Download von einem Server + Abschnitt <code>Process</code> mit <code>UpdateOnly</code> Beschreibung erweitert + Abschnitt <code>Process</code> mit <code>ResetAfterProgramming</code> erweitert
3.00	18.10.13	EDX Software Design, E.Furrer + Erweitert für Renesas RX Familien + Neue Kommandozeilen Parameter <code>+Position</code> , <code>+OnTop</code> , <code>+Silent</code> + Exit-Code erweitert mit detaillierter Auflösung + ID Code Check für RX beschrieben + Konfigurationsdatei erweitert mit Abschnitt <code>Target</code> und <code>Process</code> angepasst + Konfigurationsdatei <code>Process</code> erweitert mit <code>DetailedErrorMessage</code> + Konfigurationsdatei <code>Interface</code> erweitert mit <code>Type</code> + Dokumentation für die Version 3 überarbeitet
2.03	01.11.12	EDX Software Design, E.Furrer + Erweitert für M16C/65 + Erweitert mit dem Kommandozeilen Parameter <code>+AutoExit</code> und einer Beschreibung aller Parameter
2.02	11.01.12	EDX Software Design, E.Furrer + Neu können auch PNG Dateien aus dem DAT geladen werden + Abschnitt <code>Process</code> überarbeitet: Neu mit Intermezzo Dialog und getrennte Bestätigungstexte für <code>Program</code> und <code>Verify</code> + Neu mit Abschnitt <code>DeviceInstructionDefinition</code> für Instruktionsdefinitionen + Abschnitt <code>DeviceInstruction#</code> erweitert mit <code>Failure</code> + FTDI D2XX Treiber erweitert mit der Instruktion <code>CBBB[0xMW]</code>
2.01	28.09.11	EDX Software Design, E.Furrer + Kommandozeilen Parameter überarbeitet und erweitert + Hinweise zum erstellen von Logos ergänzt
2.00	17.05.11	EDX Software Design, E.Furrer + Dokumentation für die Version 2 vollständig überarbeitet und erweitert + Ziel-Hardware Konditionierung mit <code>DeviceInstructions</code> und Instruktionen + FTDI D2XX Treiber
1.03	10.11.08	EDX Software Design, E.Furrer + Erweitert mit Programming Mode Switch und Baudraten Liste + Erweitert mit <code>UpdateOnly</code> , <code>PreparationDlg</code> und <code>ConfirmationDlg</code> + Dokumentation allgemein überarbeitet und erweitert
1.02	12.02.08	EDX Software Design, E.Furrer + Erweiterung der Kommunikationsparameter mit <code>Interval</code> und <code>Waitstate</code> .
1.01	07.01.08	EDX Software Design, E.Furrer + Konfigurationsdatei, COM erweitert mit Filter
1.00	16.10.07	EDX Software Design, E.Furrer + Release
...		

Inhaltsverzeichnis

1.	Einleitung.....	4
1.1	Merkmale	4
1.2	Unterstützte Typen.....	4
1.3	Anforderungen	4
2.	Konzept	5
3.	Installation	6
3.1	Dateien.....	6
3.2	Installieren.....	6
3.3	Registrieren.....	6
3.4	Schnittstellen.....	6
3.4.1	COM - RS232	6
3.4.2	USB - Virtual COM Port.....	6
3.4.3	USB - FTDI D2XX.....	6
4.	Programmer	7
4.1	Anwendermodus	7
4.2	Entwicklermodus.....	8
4.2.1	Ziel-Hardware Konfiguration.....	8
4.2.2	Ziel-Hardware Konditionierung.....	8
4.3	Integration in Entwicklungsumgebungen	9
4.4	Kommandozeilen Parameter	9
4.5	Exit-Codes.....	10
4.6	ID Code Check.....	10
4.6.1	Renesas RX Family	10
4.6.2	Renesas R8C, M16C, M32C.....	10
5.	PDFC - Programmer Data File Compiler	11
6.	Konfigurationsdatei.....	11
6.1	Abschnitt [Files].....	11
6.2	Abschnitt [Target].....	12
6.3	Abschnitt [Process]	13
6.4	Abschnitt [Interface]	15
6.5	Abschnitt [Buttons]	16
6.6	Abschnitt [DeviceInstructionDefinition].....	16
6.7	Abschnitt [DeviceInstruction#].....	16
6.7.1	Instruktionen	17
7.	Anhang: ID Datei	18
8.	Anhang: Firmenlogo.....	18

1. Einleitung

Der *Programmer* ist eine Anwendung zum Programmieren von Renesas Microcontrollern. Über eine PC Schnittstelle (USB oder COM) wird der Microcontroller direkt in der Ziel-Hardware via SCI programmiert.

Benutzeroberfläche, Attribute und Abläufe sind konfigurierbar und ermöglichen die Anforderungen der verschiedenen Einsatzgebiete optimal abzudecken. Während dem Entwickler umfassende Einstellungen und Funktionen zur Verfügung stehen, wird der normale Anwender vor unnötigen Optionen bewahrt.

Durch die Ziel-Hardware Konditionierung kann diese automatisch in den Programmiermodus gesetzt und nach der Programmierung in den Programmmodus zurückgesetzt werden.

Mit der Möglichkeit Ihr Firmenlogo in die Benutzeroberfläche des *Programmers* zu integrieren, wird die Unternehmensidentität unterstützt. Ihr einheitliches Erscheinungsbild wird nach innen in Produktion und Service sowie nach aussen in Vertrieb, Vertretungen oder direkt zum Kunden getragen.

1.1 Merkmale

- Keine Installation
- Asynchrone Datenübertragung ohne proprietäre Hardware Adapter
- Konfigurierbare Instruktionen zur Ziel-Hardware Konditionierung
- Automatischer ID Code Check
- Anwendermodus
 - + Customization mit Firmenlogo
 - + Freie Nutzung ohne Registrierung
 - + Konfigurierbare Benutzeroberfläche für einfache Bedienung
 - + Konfiguration mit Bild zur visuellen Unterstützung
 - + DAT Datei mit allen Informationen: Konfiguration-, Bild- und Quellcode-Daten
 - + DAT Datei Download von einem Server (FTP, FTPS, SFTP, ...)
 - + Datenschutz durch proprietäres DAT Dateiformat
 - + ID Code Check als "Update Only"
 - + Start aus der Kommandozeile mit selbständigem Schliessen und Exit-Codes
- Entwicklermodus
 - + Umfassende Einstellungen und Funktionen
 - + Vielfältige Konfigurationsmöglichkeiten in der INI Datei

1.2 Unterstützte Typen

- Renesas RX Family: RX100, RX200, RX600 Series und kompatible
- Renesas R8C Family: Tiny, Mx Series und kompatible Cores
- Renesas M16C/26, 62, 65, 80 Group und kompatible Cores
- Renesas M32C/80 Group und kompatible Cores
- weitere auf Anfrage

1.3 Anforderungen

Betriebssysteme: Die Anwendung ist für Microsoft Windows Desktop und Server Betriebssysteme erstellt worden und wird laufend an die neuen Betriebssysteme angepasst.
(..., Win7, 8, 10, ... Server 2008, 2012, 2016...)

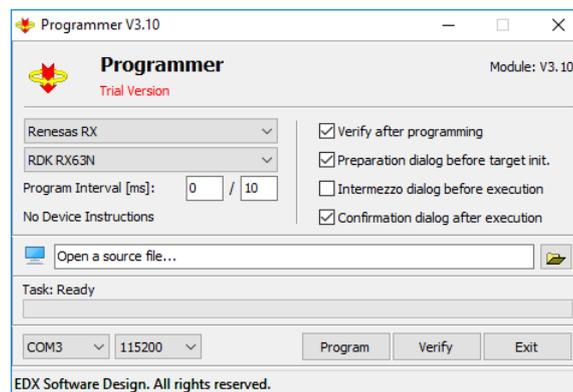
Schnittstellen: COM - RS232 mit Pegelanpassung zwischen PC und Ziel-Hardware
USB - Virtueller COM Port mit Baugruppen auf der Ziel-Hardware
USB - FTDI D2XX für verschiedene FTDIs (FT232R, FT2232D, ...)

2. Konzept

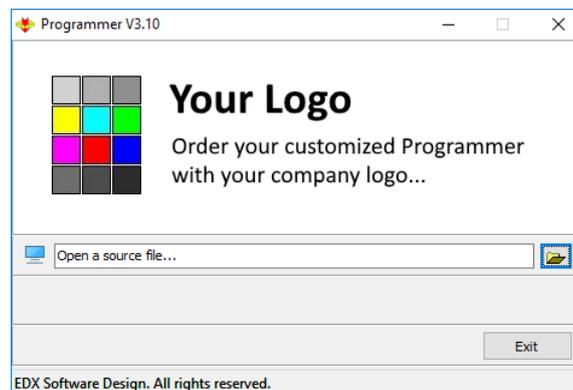
Die Anforderungen an den *Programmer* werden aus zwei unterschiedlichen Anwendungsgebieten formuliert. Während in der Entwicklung viele Einstellungsmöglichkeiten und Funktionen gefordert sind, ist dem normalen Anwender eine unkomplizierte Bedienung anzubieten.

Aufgrund dieser Anforderungen kann der *Programmer* im Entwicklermodus oder im Anwendermodus gestartet werden:

Im **Entwicklermodus** werden MOT Dateien mit dem Binärcode für die Ziel-Hardware geladen. Die Einstellungen erfolgen in der Benutzeroberfläche.



Im **Anwendermodus** werden DAT Dateien geladen. Diese enthalten den Quellcode und die nötigen Konfigurationen, Daten und Dateien für die Benutzeroberfläche, Funktionsabläufe und Schnittstellen mit Ziel-Hardware Konditionierungen.



3. Installation

3.1 Dateien

Datei	Kurzbeschreibung
ReadMe_Vnnnnn.txt	Textdatei mit Informationen zu den gelieferten Dateien
Programmer.exe	Anwendung zum Programmieren von Renesas Microcontroller
PDFC.exe	<i>Programmer Data File Compiler</i> Anwendung zum Erstellen von DAT Dateien für den Anwendermodus
Registration.exe	Anwendung zum Registrieren des <i>Programmers</i> und des <i>Programmer Data File Compiler</i>

3.2 Installieren

- EXE- und Beispiel-Dateien in ein Verzeichnis auf dem Computer kopieren
- Für den Entwicklermodus eine Verknüpfung mit dem Parameter *+Developer* erstellen
- Weitere Verknüpfungen nach Wunsch im Windows Start Menü und/oder auf dem Desktop erstellen

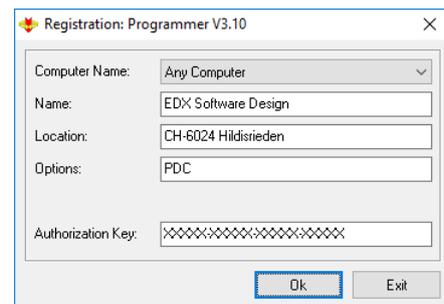
Die Installation von Treibern ist von den Schnittstellen abhängig und nicht Bestandteil dieser Dokumentation.

3.3 Registrieren

Der *Programmer* im Anwendermodus benötigt keine Registrierung und kann von Unternehmen mit einer gültigen Standardlizenz frei verbreitet und genutzt werden.

Die Verwendung des *Programmers* im Entwicklermodus und des *PDFC - Programmer Data File Compiler* setzen die Registrierung der Lizenz voraus.

Für die Registrierung werden Administratorenrechte benötigt, so dass die Lizenz für alle Anwender auf dem Computer gespeichert werden kann.



Anwendung: Registration.exe

3.4 Schnittstellen

3.4.1 COM - RS232

Die Verbindung von der seriellen Schnittstelle des PCs auf die Ziel-Hardware muss über eine Pegelanpassung erfolgen. Für Details ist die Beschreibung zum verwendeten Derivat zu konsultieren.

3.4.2 USB - Virtual COM Port

Auf dem Markt sind verschiedene USB Baugruppen für die Integration auf der Ziel-Hardware erhältlich. Für Details sind die jeweiligen Hersteller oder Vertriebsorganisationen anzufragen. Die virtuellen COM Ports werden mit Treibern von den jeweiligen Hersteller auf dem System installiert. Treiber und dessen Installation sind nicht Bestandteil des *Programmers*.

3.4.3 USB - FTDI D2XX

Future Technology Devices International Ltd. (FTDI www.ftdichip.com) ist ein Spezialist für die einfache Anbindung von USB Geräten. Mit dem FTDI D2XX Treiber wird über USB direkt mit dem FTDI Chip kommuniziert. Das und die Konfigurierbarkeit der USB Parameter im FTDI Chip eröffnen neue Perspektiven in Funktionalität und Stabilität.

Es sollte darauf geachtet werden, dass im FTDI Chip und Treiber die VCP (Virtual COM Port) Option deaktiviert ist, so dass das Betriebssystem nicht immer automatisch einen VCP einrichtet.

Treiber und dessen Installation sind nicht Bestandteil des *Programmers* und müssen direkt beschafft werden, sofern diese nicht automatisch installiert werden.

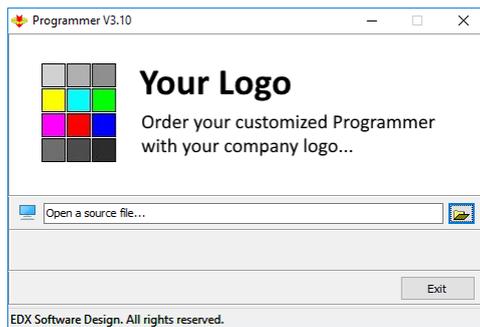
(FTD2XX.DLL Dateiversionen älter als V3.02.00 werden vom *Programmer* ignoriert.)

4. Programmier

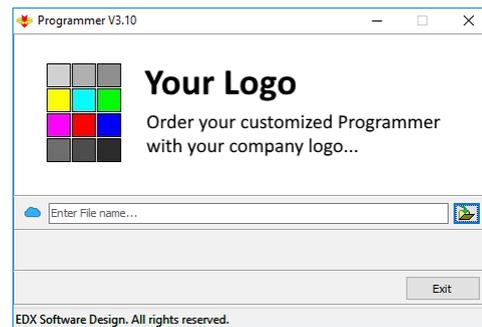
4.1 Anwendermodus

Die Anwender Benutzeroberfläche sorgt nicht nur für das perfekte Erscheinungsbild, sondern führt den Benutzer sicher durch die Bedienung. Der Benutzer wird vor unnötigen Einstellungen und Funktionen bewahrt.

Der Benutzer kann nur eine DAT Datei laden, in der alle Konfigurationen und Daten sowie der Quellcode für die Ziel-Hardware enthalten sind.

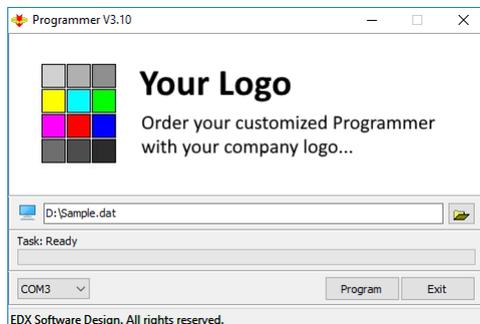


☰ Öffnen der DAT Datei aus einem Dateiverzeichnis mit dem Standard *Datei-Öffnen-Dialog*.

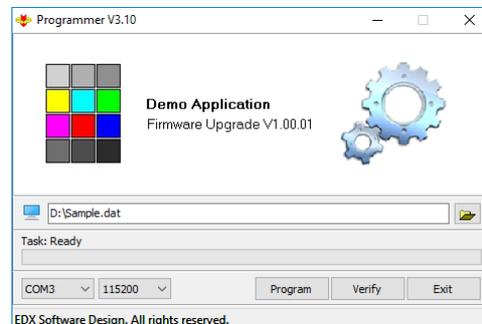


☁ Download der DAT Datei von einem Server. Die Funktionalität wird nach den Anforderungen des Kunden mit Benutzeridentifikation usw. implementiert (FTP, FTPS, SFTP, ...)

Nach dem Laden der DAT Datei wird die konfigurierte Benutzeroberfläche angezeigt.



Die einfachste Konfiguration mit dem Schnittstellen Kombinationsfeld und der Funktionstaste [Program]



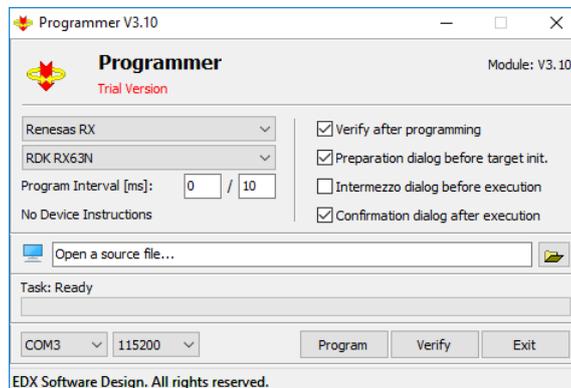
Die umfangreichste Konfiguration mit Bild, Kombinationsfeldern für die Schnittstelle und Bitrate sowie den Funktionstasten [Program] und [Verify]

Der *Programmer* kann auch aus der Kommandozeile gestartet werden. Die DAT Datei wird als Parameter übergeben. Optional können die Schnittstelle und weitere Parameter bestimmt werden. (Siehe 4.4 Kommandozeilen Parameter)

Beispiele: C:\...\Programmer.exe +DAT:Sample1.dat
 C:\...\Programmer.exe +DAT:"Sample 1.dat"
 C:\...\Programmer.exe +DAT:Sample1.dat +Port:COM2
 C:\...\Programmer.exe +DAT:Sample1.dat +Port:COM2 +AutoExit

4.2 Entwicklermodus

In der Entwickler Benutzeroberfläche stehen alle Einstellungen und Funktionen zur Verfügung. Die Ziel-Hardware wird direkt mit MOT Dateien programmiert.



Mit einer Lizenz registriert, kann der *Programmer* im Entwicklermodus gestartet werden. Der nötige Parameter kann beispielsweise in einer Verknüpfung auf den *Programmer* definiert werden.

Beispiel: `C:\..\Programmer.exe +Developer`

Im Entwicklermodus wird die Konfiguration aus dem `Programmer.ini` gelesen. Die Konfigurationen entsprechen der Konfiguration im Anwendermodus, so dass diese einfach übernommen werden können.

4.2.1 Ziel-Hardware Konfiguration

Microcontroller der *Renesas RX Family* benötigen weitergehende Einstellungen. Diese können im `Programmer.ini` in selbstdefinierbaren Abschnitten bestimmt werden. Die Abschnittbezeichnung setzt sich aus der Typenbezeichnung und einem eindeutigen Namen getrennt mit einem Doppelpunkt zusammen. Die Attribute entsprechen denen der Anwendermodus Konfiguration und sind beschrieben unter "6.2 Abschnitt [Target]".

Beispiele:

```
[Renesas RX: RDK RX63N (LittleEndian)]   Blau: Ziel-Hardware Typ
[Renesas RX: RDK RX63N (BigEndian)]       Grün: Eindeutiger Name
```

4.2.2 Ziel-Hardware Konditionierung

Gleich wie im Anwendermodus ist es möglich Schnittstellen Instruktionen zur Ziel-Hardware Konditionierung zu definieren. Siehe dazu "6.6 Abschnitt [DeviceInstructionDefinition]" und "6.7 Abschnitt [DeviceInstruction#]"

4.3 Integration in Entwicklungsumgebungen

Der *Programmer* kann über Kommandozeilen Parametern in automatisieren Abläufen integriert werden, wie dies im Anwendermodus oben beschrieben ist. Mit registrierter Standard Lizenz ist es auch möglich den *Programmer* mit der Konfigurationsdatei direkt zu starten.

Wenn die Konfigurations- und Quelldatei im gleichen Verzeichnis mit gleichem Namen (z.B.: Sample.cfg und Sample.mot) vorhanden sind, muss nur die Konfigurationsdatei mit +CFG: übergeben werden:

```
Programmer.exe +CFG:Sample\Sample.cfg
```

Eine andere Quelldatei wird mit +MOT: bestimmt:

```
Programmer.exe +CFG:Sample\Sample.cfg +MOT:Anywhere\Any.mot
```

Wird die Schnittstelle in der Konfigurationsdatei mit dem Filter nicht auf einen Port reduziert, muss die Schnittstelle mit +Port: zusätzlich übergeben werden:

```
Programmer.exe +CFG:Sample\Sample.cfg +Port:COM1
Programmer.exe +CFG:Sample\Sample.cfg +MOT:Sample\Any.mot +Port:COM2
```

4.4 Kommandozeilen Parameter

Parameter	Beschreibung
+Developer	Entwicklermodus Startet den <i>Programmer</i> im Entwicklermodus.
+DAT: +CFG: +MOT:	Dateien Dem <i>Programmer</i> können verschiedene Dateien und Dateikombinationen übergeben und mit den entsprechenden Parameterpräfixen definiert werden.
+Port:	Schnittstellenselektion Aus dem Schnittstellen Kombinationsfeld wird die definierte Schnittstelle ausgewählt. Wenn die Schnittstelle nicht zur Verfügung steht, bleibt die erste Schnittstelle ausgewählt.
+Position:X,Y	Fensterposition Der <i>Programmer</i> wird an der definierten Position geöffnet. (Zahlen und Komma ohne Abstände). Die aktuelle Fensterposition wird im Entwicklermodus beim Verschieben des Fensters in der Statusleiste angezeigt.
+OnTop	Fensteroption Der <i>Programmer</i> wird über allen Fenstern geöffnet und bleibt über normalen Fenstern im Vordergrund.
+AutoExit	Selbständiges Schliessen Beim Start aus der Kommandozeile wird der <i>Programmer</i> nach dem Programmieren oder bei einem Fehler geschlossen. Der Erfolg wird durch den Exit-Code mitgeteilt. (Siehe 4.5 Exit-Codes)
+Silent	Fehlermeldungen unterdrücken Dem Anwender werden keine Fehlermeldungen gezeigt. Der Erfolg kann durch den Exit-Code ermittelt werden. (Siehe 4.5 Exit-Codes)

4.5 Exit-Codes

Hex	Dez	Beschreibung
0x00	0	Funktion erfolgreich abgeschlossen
0xFF	255	Fehler beim Start der Anwendung
0xFA	250	Fehler beim Laden der Konfiguration
0xF9	249	Fehler beim Initialisieren des Download-Clients
0xF8	248	Fehler beim Anmelden am Server (Benutzer/Passwort falsch)
0xF7	247	Verzeichnis auf dem Server nicht gefunden
0xF6	246	Datei auf dem Server nicht verfügbar
0xF5	245	Fehler beim Download
0xF4	244	Fehler nach dem Download
0xF1	241	Fehler beim Initialisieren
0xEA	234	Fehler beim Öffnen der Schnittstelle
0xE5	229	Fehler beim Schliessen der Schnittstelle
0xCA	202	Fehler beim Verbindungsaufbau
0xC5	197	Fehler beim Wechsel der Kommunikationsgeschwindigkeit
0xAA	170	Fehler während dem Einrichten der MCU
0xA5	165	Fehler während dem Anpassen an die MCU
0x80	128	ID Code Check oder Programmiermodusfehler
0x6A	106	Blank Check Fehler
0x65	101	Fehler während dem Löschen
0x4A	74	Fehler während dem Programmieren
0x45	69	Fehler während dem Verifizieren

4.6 ID Code Check

4.6.1 Renesas RX Family

Der ID Code Check wird mit dem ID Code aus der MOT Datei durchgeführt. Falls keine ID zur Verfügung steht oder die ID nicht passt, wird mit einem ID Code Check Überlauf versucht die ZHW zu löschen. Der Ablauf wird mit einem ID Fehler abgebrochen.

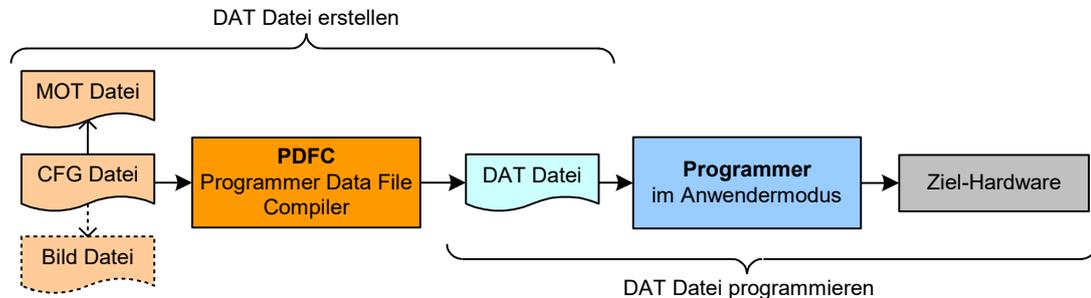
4.6.2 Renesas R8C, M16C, M32C

Zur Vereinfachung des Ablaufs führt der *Programmer* den ID Code Check mit verschiedenen ID Codes automatisch aus:

1. ID Code Konstanten 0x00 und 0xFF
2. ID Code aus der MOT Datei
3. ID Code aus der ID Datei (nur im Entwicklermodus). Siehe auch "7. Anhang: ID Datei"

5. PDFC - Programmer Data File Compiler

Mit dem *PDFC - Programmer Data File Compiler* werden DAT Dateien anhand der CFG Konfigurationsdatei, der MOT Binärcodedatei und optional mit einer Bilddatei generiert. Das DAT Dateiformat ist proprietär, so dass die Daten geschützt sind.



In der CFG Konfigurationsdatei können Einstellungen zum *Programmer* im Anwendermodus definiert werden: Benutzeroberfläche, Funktionsabläufe, Schnittstellen mit Ziel-Hardware Konditionierungen usw. Im Detail sind die Einstellungen beschrieben unter "6. Konfigurationsdatei".

Der PDFC wird mit der Konfigurationsdatei als Parameter gestartet. Die erstellte Datei wird im Verzeichnis und mit dem Namen der Konfigurationsdatei mit der Dateierweiterung DAT gespeichert.

Beispiel: C:\..\PDFC.exe Sample.cfg

```

C:\WINDOWS\system32\cmd.exe
X:\Programmer\PDFC_Sample_RX63N\Sample.cfg
Programmer Data File Compiler 3.10.06
Copyright © EDX Software Design. All rights reserved.

Read configuration file: Sample_RX63N\Sample.cfg
Read source file: Sample_RX63N\RDKRX63N_REN_MTU.mot
Read image file: Sample_RX63N\PictureSample2.png
Create data file: Sample_RX63N\Sample.dat
Successfully done

Press any key to continue . . .
  
```

6. Konfigurationsdatei

6.1 Abschnitt [Files]

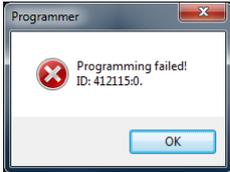
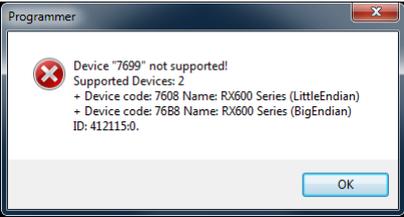
Schlüssel	Beschreibung
Source	Dateiname der MOT Datei Die MOT Datei enthält den Binärcode, der in die Ziel-Hardware zu programmieren ist. Beispiele: Source = Test1.mot Source = C:\ProjektA\Test1.mot
Image	Dateiname der Bilddatei (Optional) In der Anwender Benutzeroberfläche des <i>Programmers</i> wird das hier definierte Bild nach dem Laden der DAT Datei angezeigt. Unterstützt werden ausschliesslich PNG und BMP Dateien mit den gleichen Attributen wie unter "8. Anhang: Firmenlogo" beschrieben. Beispiele: Image = TestSystem1.bmp Image = C:\ProjektA\TestSystem1.bmp

6.2 Abschnitt [Target]

Schlüssel	Beschreibung
Type	<p>Ziel-Hardware MCU Typen</p> <ul style="list-style-type: none"> • RX Renesas RX Family: RX200, RX600 Series und compatible • R8C Renesas R8C Family: Tiny, Mx Series und compatible Cores • R8C/Tiny Retourkompatibilität entspricht R8C • M16C26 Renesas M16C/26, 26A, 28 und compatible Cores • M16C62 Renesas M16C/62A, 62N, 62P und compatible Cores • M16C65 Renesas M16C/65 Group und compatible Cores • M16C80 Renesas M16C/80 Group und compatible Cores • M32C80 Renesas M32C/80 Group und compatible Cores <p>Beispiel: Type = RX</p>
InitialBitrate	<p>Bitrate für den Verbindungsaufbau</p> <p>Beispiel: InitialBitRate = 9600</p>
DeviceCode (nur RX)	<p>ASCII Code zur Produktbezeichnung des Chips Bei einem falschen Code wird im Entwicklermodus ein Fehlerdialog mit den vom Chip unterstützten Codes ausgegeben. Im Anwendermodus werden die Chip Informationen nur angezeigt, wenn DetailedErrorMessages im Abschnitt Process gesetzt ist.</p> <p>Beispiel: DeviceCode = 7608</p>
ClockMode (nur RX)	<p>Taktmodus (<i>Clock mode</i>*) Falls dieser von einer Device nicht unterstützt wird, mit -1 deaktivieren.</p> <p>Beispiel: ClockMode = 0</p>
Frequency (nur RX)	<p>MCU Taktrate in Herz (<i>Frequency input to the RX</i>*)</p> <p>Beispiel: Frequency = 12000</p>
ClockTypeCount (nur RX)	<p>Anzahl der Takttypen (<i>Number of clock types</i>*) <i>For example, 2 indicates two clock types; that is, a system clock and a peripheral clock.*</i></p> <p>Beispiel: ClockTypeCount = 2</p>
MultiplicationRatio1 (nur RX)	<p>ICLK Verhältnis <i>Multiplication/division ratio of the input frequency to obtain the system clock (ICLK) A positive value indicates a multiplication ratio (for example, 4 = multiplied by 4) A negative value indicates a division ratio (for example, -2 = divided by 2)*</i></p> <p>Beispiel: MultiplicationRatio1 = 4</p>
MultiplicationRatio2 (nur RX)	<p>PCLK Verhältnis <i>Multiplication/division ratio of the input frequency to obtain the peripheral clock (PCLK). This value is represented in the same format as multiplication ratio 1.*</i></p> <p>Beispiel: MultiplicationRatio2 = 4</p>

* Ausschnitte aus einem RX User's Manual: Hardware

6.3 Abschnitt [Process]

Schlüssel	Beschreibung
UpdateOnly	<p>0: Deaktiviert (Standard) 1: Um sicherzustellen, dass nur bestimmte Quelldateien programmiert werden können, muss der ID Code der MOT Datei mit der Ziel-Hardware übereinstimmen.</p> <p>a) MOT Datei ohne ID Code. Abbruch mit Fehlermeldung: <i>"No ID found in code - update failed!"</i> b) ZHW ohne ID Code. Abbruch mit Fehlermeldung: <i>"Can not update an empty target!"</i> c) ZHW und MOT mit unterschiedlichem ID Code. Abbruch mit Fehlermeldung: <i>"ID check failed for update!"</i> d) ZHW und MOT mit gleichem ID Code. ZHW wird vollständig gelöscht und mit der neuen MOT Datei programmiert.</p> <p>Beispiel: UpdateOnly = 0</p>
DetailedErrorMessages	<p>0: Funktionsfehlermeldungen (Standard) 1: Fehlermeldungen mit Fehlerursache und möglichen Inhalten die von der Ziel-Hardware gelesen worden sind.</p> <p>Beispiel: DetailedErrorMessages = 0 DetailedErrorMessages = 1</p> <div style="display: flex; justify-content: space-around;">   </div>
VerifyAfterProgramming	<p>0: Nur programmieren 1: Nach dem Programmieren wird automatisch das Verifizieren durchgeführt. (Standard)</p> <p>Beispiel: VerifyAfterProgramming = 1</p>
ResetAfterProgramming	<p>0: keine Funktion (Standard) 1: Wenn die DAT Datei per Download in den <i>Programmer</i> geladen worden ist, werden nach dem Programmieren die Einstellungen zurückgesetzt und die Daten im <i>Programmer</i> gelöscht.</p> <p>Beispiel: ResetAfterProgramming = 0</p>
PreparationDlg IntermezzoDlg ConfirmationDlg	<p>Im Ablauf können verschiedene Dialoge für die Interaktion mit dem Benutzer angezeigt werden:</p> <ol style="list-style-type: none"> 1. Ablauf auslösen über die Taste [Program] oder [Verify] oder Startparameter 2. <i>Vorbereitungsdialog anzeigen wenn aktiviert:</i> PreparationDlg = 1 3. Ziel-Hardware wird konditioniert falls definiert 4. <i>Pausendialog anzeigen wenn aktiviert:</i> IntermezzoDlg = 1 5. Ziel-Hardware wird programmiert und/oder geprüft 6. <i>Bestätigungsdialog anzeigen wenn aktiviert:</i> ConfirmationDlg = 1
PreparationText	<p>Text^o im Vorbereitungsdialog Der Dialog wird nur angezeigt, wenn PreparationDlg = 1 ist.</p> <p>Beispiel: PreparationText = Reset target.</p>

Schlüssel	Beschreibung
IntermezzoText	<p>Text[Ⓞ] im Pausendialog Der Dialog wird nur angezeigt, wenn IntermezzoDlg = 1 ist.</p> <p>Beispiel: IntermezzoText = Activate programming mode.</p>
ConfirmationText ProgramConfirmationText VerifyConfirmationText	<p>Text[Ⓞ] im Bestätigungsdialog Der Bestätigungsdialog wird nach erfolgreichem Ablauf angezeigt, wenn ConfirmationDlg = 1 ist. Funktionsunabhängig mit ConfirmationText oder mit ProgramConfirmationText und VerifyConfirmationText Funktionsabhängig.</p> <p>Beispiel: ConfirmationText = Programming done Restart oder ProgramConfirmationText = Programming is done successfully. VerifyConfirmationText = Verify is done successfully.</p>

① Zeilenumbruch im Dialog wird mit zwei Pipes "||" eingeleitet.

6.4 Abschnitt [Interface]

Schlüssel	Beschreibung
Type	<p>Mit der Definition des Schnittstellentyps kann zur Zeitoptimierung des Ablaufs, die Suche nach verfügbaren Schnittstellen bestimmt werden. (Standard = COM; FTDI)</p> <p>COM: Suche nach verfügbaren RS232 Schnittstellen FTDI: Suche nach verfügbaren FTDI D2XX Geräten</p> <p>Beispiele: Type = COM; FTDI Type = FTDI</p>
Filter	<p>Im Schnittstellen Kombinationsfeld werden nach dem Laden der Konfiguration nur die verfügbaren Schnittstellen angezeigt. Durch die hier definierte Zeichenkette kann die Auswahl der Schnittstellen einbegrenzt werden.</p> <p>Beispiele: Filter = PCI Serial Filter = FT2232 USB UART B</p>
BitrateVisible	<p>0: Kombinationsfeld Bitrate ausblenden. 1: Kombinationsfeld Bitrate anzeigen. (Standard)</p> <p>Beispiele: BitrateVisible = 1</p>
BitrateList	<p>Liste der zur Verfügung zu stellenden Bitraten Die möglichen Bitraten sind von der Ziel-Hardware, dem Microcontroller und der Taktrate abhängig. Für Details ist die Beschreibung zum verwendeten Microcontroller zu konsultieren.</p> <p>Beispiel: BitrateList = 9600; 19200; 38400</p>
Bitrate	<p>Standardeinstellung der Bitrate in der Bitratenliste.</p> <p>Beispiel: Bitrate = 38400</p>
Interval	<p>Zeitabstand in Millisekunden zwischen Programmierkommandos Renasas empfiehlt bei Kommunikationsfehlern beim Ausführen von Programmierkommandos die Zeit zwischen 5 und 50ms zu setzen. (R8C, M16C und M32C) (Standard = 0)</p> <p>Beispiel: Interval = 40</p>
Waitstate	<p>Warteeinheiten in Millisekunden beim Lesen von Daten zur Optimierung der Kommunikationsgeschwindigkeit respektive Anpassung an die Schnittstelle. (Standard = 10)</p> <p>Beispiel: Waitstate = 10</p>

6.5 Abschnitt [Buttons]

Schlüssel	Beschreibung
Types	<p>Liste der anzuzeigenden Funktionstasten.</p> <p>Program: Programmieren der Ziel-Hardware Verify: Prüfen des Binärcodes in der Ziel-Hardware (Nicht für RX Typen)</p> <p>Beispiele: Types = Program Types = Program; Verify Types = Verify; Program</p>

6.6 Abschnitt [DeviceInstructionDefinition]

Zur besseren Übersicht können Instruktionen und Instruktionslisten definiert werden, die dann im Abschnitt `DeviceInstruction#` zur Verfügung stehen.

6.7 Abschnitt [DeviceInstruction#]

Entsprechend der zu unterstützenden Schnittstellen können eine oder mehrere dieser Abschnitte definiert werden. Das Doppelkreuz # dient als Platzhalter für die aufsteigende Nummerierung beginnend bei Null.

Schlüssel	Beschreibung
Caption	<p>Bezeichnung der Definition In der Entwickler Benutzeroberfläche des <i>Programmers</i> wird diese Bezeichnung zur selektierten Schnittstelle angezeigt.</p> <p>Beispiel: Caption = FTDI Test Adapter</p>
Id	<p>Erkennungskriterium Bezeichnung oder Teilbezeichnung der Schnittstelle für die Zuweisung der Instruktionskonfiguration. (Hinweis: In den FTDI Chips kann die Beschreibung definiert werden)</p> <p>Beispiel: Id = FT2232 USB UART B</p>
Initial	<p>Einleitende Instruktionen * Mit Strichpunkt getrennte Instruktionen um beispielsweise die Ziel-Hardware in den Programmiermodus zu setzen.</p> <p>Beispiel: Initial = SetRTS; SetDTR; W=50; ClrDTR; W=100; SetDTR; W=200</p>
Failure	<p>Instruktionen beim einem Funktionsabbruch * Mit Strichpunkt getrennte Instruktionen um beispielsweise die Ziel-Hardware mit einem Reset in einen sicheren Zustand zu setzen.</p> <p>Beispiel: Failure = SetRTS; SetDTR; W=50; ClrDTR; W=100; SetDTR; W=200</p>
Final	<p>Abschliessende Instruktionen * Mit Strichpunkt getrennte Instruktionen um beispielsweise die Ziel-Hardware mit einem Reset in den Programmmodus zurückzusetzen.</p> <p>Beispiel: Final = ClrRTS; ClrDTR; W=50; SetDTR</p>

* Die Möglichkeiten sind von der Schnittstelle abhängig und im Folgenden beschrieben unter "6.7.1 Instruktionen"

6.7.1 Instruktionen

Instruktionen sind Befehle an die Schnittstelle mit speziellen Funktionen wie beispielsweise das Ansteuern einzelner Pins.

6.7.1.1 Allgemeine Instruktionen

Instruktion	Beschreibung
W=N	N: Wartezeit in Millisekunden

6.7.1.2 COM - RS232

Instruktion	Beschreibung
<p>Nicht implementiert</p> SetRTS ClrRTS SetDTR ClrDTR	<p>Die RS232 Hardware Kontrollsignale können als Ausgänge benutzt werden, sofern diese nicht anderweitig im Einsatz sind.</p> <p>Bemerkung: Unter Windows wird zum Umschalten der Bitrate der Port geschlossen und mit neuer Bitrate geöffnet. Beim Schliessen des Ports wird RTS und DTR vom Betriebssystem zurückgesetzt.</p>

6.7.1.3 USB - FDTI D2XX Treiber

Instruktion	Beschreibung
SetRTS ClrRTS SetDTR ClrDTR	<p>Die RS232 Hardware Kontrollsignale können als Ausgänge benutzt werden, sofern diese nicht anderweitig im Einsatz sind. Das Verhalten der Signale des eingesetzten Chips sollte aber auf jeden Fall vorab geprüft werden.</p> <p>Bemerkung: Tests haben gezeigt, dass die Signale gehalten werden, selbst wenn die Bitrate umgeschaltet wird.</p>
CBBB[0xMW]	<p>CBUS Bit Bang für FTDI FT232R Im Bit Bang Modus können die CBUS IOs als Ausgänge benutzt werden.</p> <p>Wert Bitmaskiert: CBUS0 = Bit 1, CBUS1 = Bit 2, CBUS2 = Bit 3, CBUS3 = Bit 4 Modus Bitmaskiert: CBUS0 = Bit 5, CBUS1 = Bit 6, CBUS2 = Bit 7, CBUS3 = Bit 8</p> <p>CBBB[0xMW] W Wert am Pin: Low=0, High=1 (Bitmaskiert) M Modus am Pin: Input=0, Output=1 (Bitmaskiert)</p>

7. Anhang: ID Datei

Der *Programmer* kann im Entwicklermodus für R8C, M16C oder M32C Typen die ID Datei für den ID Code Check laden. Die ID Datei hat den gleichen Namen wie die MOT Datei aber mit der Dateierweiterung ID. Beide Dateien müssen im gleichen Verzeichnis abgelegt sein.

Beispiele für einen M16C62:

ID Dateiinhalt 1: -ID1234

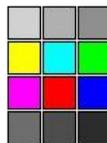
ID Dateiinhalt 2: -ID12340000000000

Für beide Beispiele ist der ID Code 12340000000000. 12_h ist für die Adresse FFFDF_h, 34_h ist für die Adresse FFFE3_h und 00_h ist für die übrigen Adressen FFEB_h, FFEF_h, FFFF3_h, FFFF7_h und FFFFB_h.

8. Anhang: Firmenlogo

Da die Farbe des Applikationsfensters von den System Themen abhängt, sollte der Hintergrund des Logos neutral gehalten werden.

Höhe: max. 160 Pixel
Breite: max. 470 Pixel
Farbtiefe: max. 24 Bit
Formate: PNG, BMP, JPG, JPEG, GIF



Your Logo

Order your customized Programmer
with your company logo...

Zusätzlich zum Logo kann ein Text bestimmt werden, der in der Statusleiste angezeigt wird, wenn der Mauszeiger auf dem Logo steht.