

# M16C-Flasher – Einstieg

---

## Allgemein

---

Der M16C-Flasher dient zum Programmieren von Renesas-Mikrocontrollern. Der Programmiervorgang wird im Standard Serial Mode 2 (asynchron), d.h. einfach per serielltem Kabel zwischen PC und Mikrocontroller durchgeführt. Zusätzliche Hardware ist nicht erforderlich. Es muss auf die unterschiedlichen Pegel beim PC (+12V & -12V) und dem Mikrocontroller (5V & 0V) geachtet werden. (z.B. MAX232 verwenden)

Als NON-PROFIT-Version ist der M16C-Flasher nur für den privaten, nicht-kommerziellen Einsatz frei, ansonsten muss eine entsprechende Lizenz erworben werden.

## Hilfetexte

---

Die meisten Funktionen sind selbsterklärend. Bei einigen Buttons oder Optionen erscheinen kleine Hilfetexte (ToolTips), wenn der Mauszeiger einige Zeit darüber verweilt.

Nachfolgend wird nur auf den ersten Einstieg und auf einige grundsätzliche Funktionen bzw. Probleme eingegangen.

## Erster Start

---

Als erstes sollte der Comport und die Baudrate eingestellt werden. Bitte beachten Sie, dass die maximale Baudrate von dem verwendeten Mikrocontroller und der Quarzfrequenz abhängt. Für einen ersten Test empfehle ich 9600 Baud. (damit sollte es immer funktionieren).

Als Protokoll 'Revision 2' und 'Standard Connect' wählen. Dies entspricht dem aktuellen Datenblättern.

Verwendet die Hardware die Steuerleitungen für Reset und CNVSS, so ist diese Option zu aktivieren und auf richtigen Pegel zu achten (Mehr hierzu unter 'Steuerleitungen').

Natürlich muss auch der entsprechende Mikrocontroller ausgewählt werden.

## Verbinden

---

Bevor man eine Funktion ausführen kann (Programmieren, Löschen, Auslesen, ...) muss eine Software-Verbindung mit dem Bootloader des Mikrocontroller hergestellt werden. Sofern man die Steuerleitungen nicht verwendet, muss man die Leitung CNVSS entsprechend jumpern und einen Reset beim Mikrocontroller ausführen.

Durch Betätigen von 'Connect' wird versucht, eine Verbindung mit dem Mikrocontroller herzustellen. Bei Erfolg wird die Version des Bootloaders angezeigt (VERx.xx).

Jetzt kann die gewünschte Funktion (z.B. Programmieren) ausgeführt werden.

## Die ID des Mikrocontrollers

---

Als Schutz gegen Veränderung und Auslesen des Mikrocontrollerspeichers wird vom Mikrocontroller eine 7stellige sogenannte ID verwendet. Damit man Zugriff auf die Funktionen des Bootloaders bekommt, muss man die ID des im Mikrocontroller befindlichen Programmes kennen. Sollte der Mikrocontroller unbeschrieben sein, erhält man immer Zugriff. Mehr Informationen zur ID finden Sie im Datenblatt des Mikrocontrollers, sowie in der Dokumentation zu ihrem Compiler.

Die NON-PROFIT Version des M16C-Flasher schreibt immer die ID 0x00 0x00 0x00 0x00 0x00 0x00 0x00 in den Mikrocontroller, unabhängig, was im MOT-File steht.

# M16C-Flasher – Einstieg

---

Ausserdem besteht die Möglichkeit, jede programmierte ID in eine Datei (ID.LOG) zu schreiben. So haben Sie immer einen Überblick über die verwendeten IDs.

Standardmässig verwendet der M16C-Flasher immer die letzte programmierte ID als aktuelle ID. Dies ist auch in den meisten Fällen sinnvoll.

Arbeitet man aber an verschiedenen Geräten mit unterschiedlichen Programmen und unterschiedlicher ID, so ist diese Einstellung nicht günstig. Hier bietet der M16C-Flasher die Option '**Extract ID from File**'. Es wird also immer die ID verwendet, die im zu programmierenden File enthalten ist. Dies sollte ja auch die ID im Mikrocontroller sein.

## 'ID mismatch' - Fehler

---

Sollte die im M16C-Flasher eingestellte ID nicht mit der im Mikrocontroller eingestellten ID übereinstimmen, gibt es einen '**ID mismatch**'-Fehler. Stellen Sie die korrekte ID ein, schalten Sie die Versorgungsspannung des Mikrocontrollers kurz ab und versuchen Sie es erneut. Ohne Abschalten der Versorgungsspannung ist ein weiterer Versuch **nie** erfolgreich. Dies ist ein Schutzmechanismus, damit nicht einfach beliebige IDs in kurzer Folge ausprobiert werden können.

## Option 'Load File once'

---

Wenn man oft die gleiche Datei hintereinander programmiert, kann man sich den 'Datei auswählen'-Dialog ersparen, indem man die '**Load File once**'-Option aktiviert. Es wird nur einmal die Datei abgefragt, danach kann man einfach auf den 'Prog'-Button drücken, um die Datei erneut zu programmieren.

## Steuerleitungen

---

Ein erheblicher Komfortgewinn und eine Zeitersparnis erhält man, wenn man bei seiner Hardware die Steuerleitungen RTS & DTR verwendet. Mit Hilfe dieser Leitungen kann der Mikrocontroller resetet und in den Bootloader-Modus gebracht werden. Ein umständliches Jumpen und Reseten ist nicht mehr erforderlich. Man muss, wie bei den Datenleitungen auch, die unterschiedlichen Pegel beachten. Ein Schaltungsbeispiel mit Z-Diode und Widerständen findet man unter 'Settings' im M16C-Flasher-Programm.

Damit die Steuerleitungen von der verwendeten Logik unabhängig sind, kann man die Signale getrennt invertieren und so an seine Hardware anpassen.

Wird der externe Bus des Mikrocontrollers benutzt, ist auf die richtigen Pegel von **Ready & Hold** zu achten!

## AutoFlash-Funktion

---

Um die Benutzerinteraktion so gering wie möglich zu halten, kann man die AutoFlash-Funktion aktivieren.

Eine einmal programmierte MOT-Datei wird im Hintergrund auf Veränderung überwacht und dann automatisch übertragen.

Erzeugt der Compiler eine neue MOT-Datei, so wird diese sofort vom M16C-Flasher übertragen – einfacher geht es nicht.