Table of Contents

Flashtool

Copyright © 2003 Guenter Ebermann

This is the documentation of the flashtool.

1 Introduction

The Bootstrap Loader Tool is a 32bit application that runs under Linux and under Microsoft Windows with cygwin.

The Bootstrap Loader Tool can download a program (in Intel Hex Format) from the PC via the bootstrap loader of the microcontroller in any location of the internal or external RAM.

BSL = Bootstraploader

uC = Microcontroller

Das Software-Paket besteht aus:

1. PC Programm flashtool
2. primary BSL, secondary BSL
3. uC Flashtool Programm

With the Bootstrap Loader Tool it is possible to download:

1. the initial 32 bytes that are expected by the C
2. program code in the Internal RAM
3. program code in the External RAM at any location

The program code in all three download modes must be in Intel Hex-File Format.

Zusaetzlich hat es noch features wie aus eprom auslesen aus bestimmten Adressen mit AUsgabe im Intel Hex File Format und ein eingebautes Terminal Programm.

2 flashtool Invocation

Hier einige Beispielszenarien zur Anwendung des Programmes:

1. Beispiel:

   Sie wollen das flashtool ins RAM spielen um spter ein Programm auf das EPROM flashen zu knnen.

   $ flashtool –write

   Nun wollen sie den EPROM lschen und kontrollieren:

   $ flashtool –delete $ flashtool –read –stop 55

   Und nun ein Programm z.B. bsl3_blinke_auf_LED_D13.H86 flashen und kontrollieren:

   $ flashtool –flash bsl3_blinke_auf_LED_D13.H86 $ flashtool –read –stop 77

   Jetzt mssen Sie DIP Schalter 1 abschalten und reseten. Das Programm wird nun aus den Flashspeicher ausgefhrt.

2. Beispiel:

   Sie wollen ein mit Keil compiliertes Programm dass im Intel Hex Format mit dem Dateinamen lauflicht.H86 vorliegt ins RAM speichern und ausfhren.

   $ flashtool –write lauflicht.H86

   Das Programm wird runter geladen und gestartet.

   $ flashtool –write –verbose lauflicht.H86

   Das gleiche mit mehr Augaben auf stdout.

3. Beispiel:

   Sie wollen einen neuen First (bsl1.H86) und Second Bootstrap loader (bsl2.H86) verwenden. Und dann ihr lauflicht Programm zum testen.

   $ flashtool –init

   Der uC wird initialisiert (0 Byte schicken).

   $ flashtool –bootstrap bsl1.H86

   Ersten bsl runterladen.

   $ flashtool –bootstrap bsl1.H86

   Zweiten BSL runterladen.

   $ flashtool –bootstrap3 lauflicht.H86

   Wollen Sie bei allen vorgefhrten Befehlen mehr Output geben sie berall die Option –verbose dazu.

4. Beispiel:

   Sie wollen mit einem bereits runtergeladenen uC Flashprogramm kommunizieren.

   $ flashtool –terminal

5. Beispiel:

   Sie wollen das Binaer Format fr ersten und dritten BSL sehen.

   $ flashtool –convert bsl1.H86 $ flashtool –convert3 bsl3.H86

3 Options

Flashtool invocation: ./flashtool [OPTIONS] INFILE\n

1. general options:

   Diese beiden Optionen gelten generell fr die Kombination mit mehreren anderen Optionen

   –output -o [FILENAME] output in file

   Wird bei optionen wie convert u. convert3 verwendet. Wird diese Option nicht angegeben wird Standardmssig INFILE.bin verwendet bei convert und INFILE.bin3 bei der Option convert3

   –verbose print more output

   Es werden Debugging Messages ausgegeben.

2. utilities:

   –terminal serial terminal programm - talk over serial line

   Das Flashtool Programm kann als Terminal Programm verwendet werden. Scheinbar gleichzeitiges schreiben auf serielle Schnittstelle und lesen von serielle Schnittstelle und Ausgabe auf stdout ist mglich.

   z.B: flashtoll –terminal –baud 57600 –device /dev/ttyS1

   oder flashtool –terminal // standardmssig wird 37600 baud und /dev/ttyS0 verwendet

3. standard switches:

   –version print version number and exit

   –help print this help and exit

4. serial setup:

   Mit diesen Optionen kann die Serielle Schnittstelle eingestellt werden. Werden Sie nicht angegeben wird standardmeassig 37600 Baud und /dev/ttyS0 verwendet

   –noinit no init of serial line - use stty

   Die Seriell Schnittstelle wird nicht initialisiert. Man kann also die Einstellungen manuell z.B. mit stty vornehmen.

   –baud -b [BAUDRATE] use baudrate eg. 57600

   –device -d [DEVICE] use device eg. /dev/ttyS0

5. converting functions:

   –convert convert from intel hex to bin

   Ein Intel hex file kann in ein Binaer File konvertiert werden.

   Bsp: flashtool –convert –output bsl1.bin bsl1.IHEX

   oder einfacher: flashtool –convert bsl1.IHEX // Ausgabe wird in File bsl1.bin automatisch geschrieben

   –convert3 convert from intel hex to bin for 3rd bsl

   Der dritte BSL (also das microcontroller flash-Programm) muss bevor es an den C167 geschickt wird in ein spezielles Format umgewndelt werden. Dies ist umbedingt

notwendig um z.B die Checksumme zu ueberpruefen und in die richtigen RAM Bereiche zu schreiben die im Intel hex file angegeben sind. Anwendung wie bei Option convert.

–hextobin convert raw hex to bin

Anwendung wie convert. Es kann ein Ascii HEX File in Binaer gewandelt werden.

6. bootstrap:

   –init only send the 0 byte

   Nur Null Byte senden und Antwort checken. Um eine Ausgabe zu erhalten mit verbose anwenden.

   z.B: flashtool –init –verbose

   –bootstrap transfer the bootstrap loaders (1st, 2nd, 3rd)

   Damit wird das angegebene File im Intel Hex oder im (vorher umgewandelten mit Option convert oder convert3) Binaer Format vorliegend Zeichen fr Zeichen auf den C167 bertragen.

   flashtool –init // zuerst Kennung versenden

   flashtool –bootstrap bsl1.IHEX // dann bsl1 runterladen

   –bootstrap3 transfer 3rd bootstrap loader

   Um den dritten BSL zu tansferieren sollte man diese Funktion verweden da die Checksum nach jeder versandten Zeile berprft wird.

   –write automates the downloading of data

   Hier wird sofort ein Programm (Intel Hex oder im pseudo Binaer Format (convertiert mit convert3), Erkennung erfolgt automatisch) in den RAM geflasht.

   z.B: flashtool –write bsl3_lauflicht_schnell.H86 // Ein Lauflicht wird runtergeladen und automatisch gestartet.

7. communictation with bsl3:

   –flash write ihex on eprom

   Es wird ein Intel Hex File Zeile fr Zeile auf den EPROM transferiert.

   –delete delete eprom

   Es wird das gesamte eprom geloescht (mit FF beschrieben)

   –read read from flash memory

   Es wird vom Eprom ausgelesen und im Hexfileformat am Bildschirm ausgegeben. Dazu am Besten mit stop eine Stopadresse angeben

   –start -s [ADDRESS] only with –read

   –stop -t [ADDRESS] only with –read

   –info print info of connected board an processor

4 Technical Doku

Ablauf bei flashtool –write:

1. Micro Controller initialisieren:

   Dazu wird ASCII 0 an den Microcontroller geschickt. Der Microcontroller misst die Zeitdauer dieses ASCII 0 Zeichens um die Baudrate der Verbindung zu messen und sich darauf selbststndig einzustellen. Der Microcontroller antwotet auf dieses ASCII 0 mit 0xC5, dies entspricht 197 im dezimal System. Das Computerprogramm wartet auf diese Antwort.

2. Senden des primary Bootstrap Loaders:

   Der Bootstrap loader wird seriell bertragen. Dieses eigentstndische kleine Assembler Programm macht nichts anderes als den secondary BSL zu lesen und ins RAM zu schreiben.

3. Senden des secondary BSL:

   Der zweite BSL kann Checksum berechnen und retounieren und an die angegebe Adresse ins RAM schreiben. Dazu muss ein Intel hex filein ein pseudo binaer format konvertiert werden dass checksum und Adresse enthlt.

4. Sendes des uC Programms:

   Dass Haupt-C Programm auf den Microcontroller transferieren. Kommunikation zwischen Computer und Microcontroller: Nun endlich kann das Computer Programm mit dem Microcontroller programm ber den definierten Befehlssatz kommunizieren. Dieser Befehlsatz hat immer die form einer Intel HEX Zeile.

5 Bootstrap Loader

Der Mikrocontroller geht in den BootStrap Loader Mode, wenn das Pin P0L.4 auf LOW gesetzt ist und ein Hardware Reset erfolgte.

Mikrocontroller initialisieren: Dazu wird ASCII 0 an den Mikrocontroller geschickt. Der Mikrocontroller misst die Zeitdauer dieses ASCII 0 Zeichens um die Baudrate der Verbindung zu messen und sich darauf selbststndig einzustellen. Der Mikrocontroller antwortet auf dieses ASCII 0 mit 0xC5, dies entspricht 197 im Dezimalsystem. Das Computerprogramm wartet auf diese Antwort.

BSL 1 und 2: Der Mikrocontroller erwartet nun 32 Bytes, die im Internen RAM von der Adresse 0xFA40 H bis 0xFA5F H gespeichert werden. Zum Ausfhren des Codes springt der BSL zur Adresse 0xFA40 H.

Der BSL 1 und 2 wurden dem Produkt der Fa. Siemens (AP1644) entnommen. Im BSL 1 mussten noch folgende Variablen gesetzt werden:

NUMBER_BYTES EQU 0AAH ;Gre des BSL 2

STARTADDRESS EQU 0F600H ;IRAM Startadresse

Der BSL 1 speichert schlielich den BSL 2 ab der STARTADDRESS (0x0F600 H).

Der BSL 2 ist nunmehr fr das runterladen des Flashprogramms zustndig. Dazu wird das Flashprogramm im Externen RAM abgelegt. Alle Dateien werden als Intel Hex File (*.H86) auf den Mikrocontroller geladen.

Index