

ds30 Loader
Console manual

Table of contents

Introduction.....	3
ds30 Loader	3
Prerequisites and Requirements.....	3
Trademarks.....	3
Usage	4
Requirements	4
Installation.....	4
Starting	4
Windows.....	4
Linux	4
MAC OS X.....	4
Usage syntax.....	4
Options summary.....	4
Return value	5
Examples.....	5
Windows.....	5
Linux	6

Introduction

ds30 Loader

ds30 Loader is a boot loader supporting PIC12, PIC16, PIC18, PIC24, and dsPIC families of MCUs from Microchip. It supports all devices in each family out of the box (those in production). The firmware is written in assembler. The PC clients run on Windows, Linux, and Mac OS X.

Prerequisites and Requirements

.NET framework 2.0 or Mono

Trademarks

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Usage

Requirements

ds30 Loader console requires a .NET framework to run. For Windows there are two different frameworks available; .NET framework from Microsoft and Mono sponsored by Novell. For Linux and MAC OS X, only Mono is available. Mono is constantly being developed and bug fixed. It is recommended to use the latest version. Download links are available in appendix A.

Installation

The ds30 Loader console does not require to be installed; it can be run directly from the bin director. However, for Windows users an installation is included for those people who would like the application to be installed and have short cuts created on the start menu.

Starting

Windows

- Use run on the start menu, browse to ds30LoaderConsole.exe
- Run from command prompt: ds30LoaderConsole.exe

Linux

Run command: mono ds30LoaderConsole.exe

MAC OS X

Run command: mono ds30LoaderConsole.exe

Usage syntax

The syntax is:

```
ds30loaderconsole options
```

Options summary

When unknown option or illegal option value is detected, the download is aborted. For more information about each command refer to the GUI manual.

Short option name	Long option name	Description
-h	--help	Display help
-l	--debugmode	Activate debugmode
-o	--non-interactive	Start download without user interaction
-z	--list-ports	List available ports
n/a	--parse-only	Parse hex file then exit
n/a	--find	Find boot loader then exit

-f=filename	--file=filename	Hex-file
-d=device	--device=devicename	Devicename
-k=portname	--port=portname	Portname
-r=baudrate	--baudrate=baudrate	Baudrate
-p	--write-program	Write program
-e	--write-EEPROM	Write EEPROM
n/a	--pic-can-id=id	CAN id of device to download to
n/a	--ds30-can-id=id	CAN id of the ds30 Loader application
n/a	--ext	CAN extended frames
n/a	--dlc=dlc	CAN transmit data length code
-g	--no-goto	Don't write goto to bootloader
-s	--allow-bl-overwrite	Allow overwrite of bootloader
-c	--write-configs	Download configs
-v	--customplacement	Custom bootloader placement, pages from end
-w	--customsize	Custom bootloader size, page(s)
-x	--auto-baudrate	Send auto baud rate sync character
-y	--echo-verification	Verify tx echoes
n/a	--add-crc	Writes a checksum of the flash
n/a	--disable-empty-pages	Empty pages will no be written
n/a	--ht=hello timeout	Time to keep sendin hello before aborting
-a=time	--polltime=polltime	Polltime [ms], default is 100
-t=time	--timeout=timeout	Time before timeout [ms], default is 5000
-q=cmd	--reset-command=cmd	Reset device by specified command, this option may need to be enclosed in " under Linux, se Linux example in the next section
-u=baudrate	--reset-baudrate=br	Baudrate for reset command
-m	--reset-dtr	Reset device by dtr
-n	--reset-RTS	Reset device by rts
-b=time	--resettime=time	Resettime [ms]
-i	--activate-dtr	Activate device by dtr
-j	--activate-RTS	Activate device by rts
n/a	--pw=password	Password to send to ds30 Secure Loader

Return value

0 is returned after a successful download, else -1.

Examples

Windows

```
ds30loaderconsole --file=d:\serlcd.hex --device=pic18f2550 --port=com1  
--baudrate=115200 --write-program
```

Linux

```
mono ds30loaderconsole.exe -p -f=serlcd.hex "-q=a;b" -d=pic18f2550  
-k=/dev/ttyUSB0 -r=11520
```