ds30 Loader
GUI manual

Table of contents

Document history

Rev C

Minor improvements
Updated with new options

Rev B

New feature: auto baud rate
New feature: echo verification
New command: re-parse hex-file

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.

Getting started

Requirements

ds30 Loader GUI requires a .NET framework to run. For Windows t here 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 GUI 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

- Double click ds30 Loader GUI.exe from Explorer
- Click shortcut on the start menu
- Use run on the start menu, browse to ds30 Loader GUI.exe
- Start from command from prompt

Linux

Run command: mono ds30\ Loader\ GUI.exe

MAC OS X

Run command: mono ds30\ Loader\ GUI.exe
If there are graphics related problems different environment variables can set to change how mono does graphics:
export MONO_MWF_USE_QUARTZ_BACKEND=1
export MONO_MWF_USE_CARBON_BACKEND=1
export MONO_MWF_MAC_FORCE_X11=1

First start

When started the first time it will look something like the screenshot to the right, depending on your operating system set-up.

If you are an experienced boot loader user you will probably want to switch to advanced mode on the menu View\Advanced mode. Doing so will enable more settings and features.
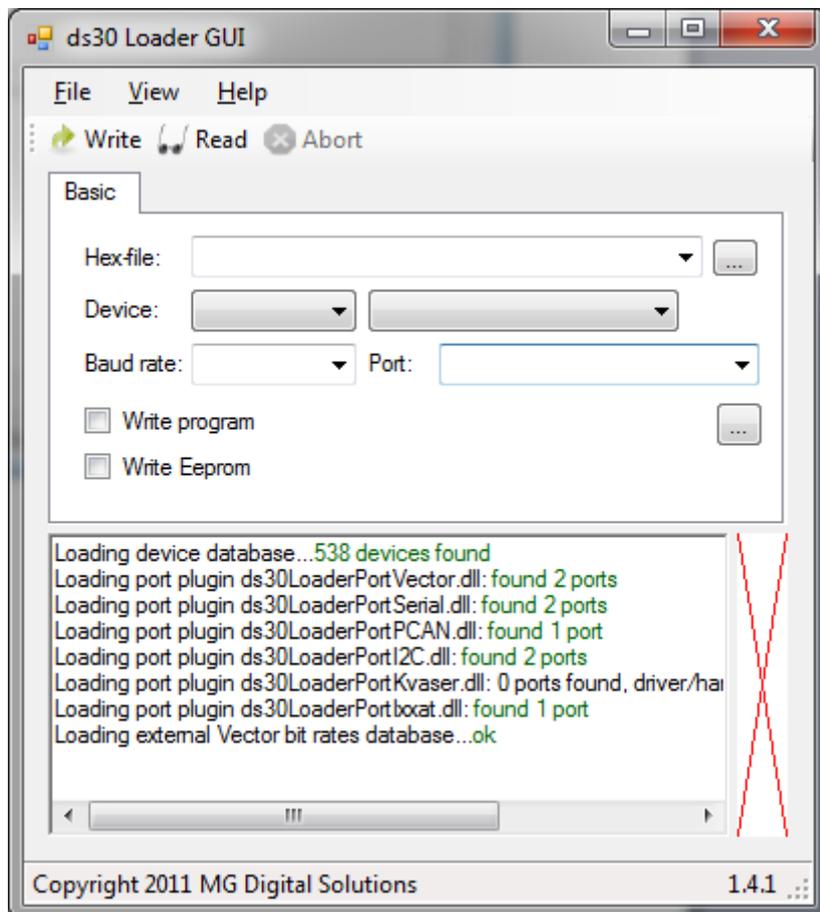
Window elements

The ds30 Loader GUI consists of 5 main parts; the menu, the toolbar, the tab pages, the output text box and the graphical hex-file representation.

Menus

File\Exit

Saves all settings and closes the application.

Options\Debug mode

When debug mode is enabled, additional information is outputted during parsing of a hex-file and during write. This option is normally only used when looking for causes of boot loader malfunction. When toggled, the hex-file is re-parsed.

Options\Reset all settings to default

When clicked, all settings are set to their default values for the current file. Text boxes and drop down boxes retain their values.

Commands\Abort

Aborts the current write or check for boot loader operation. This command is also available as a toolbar button.
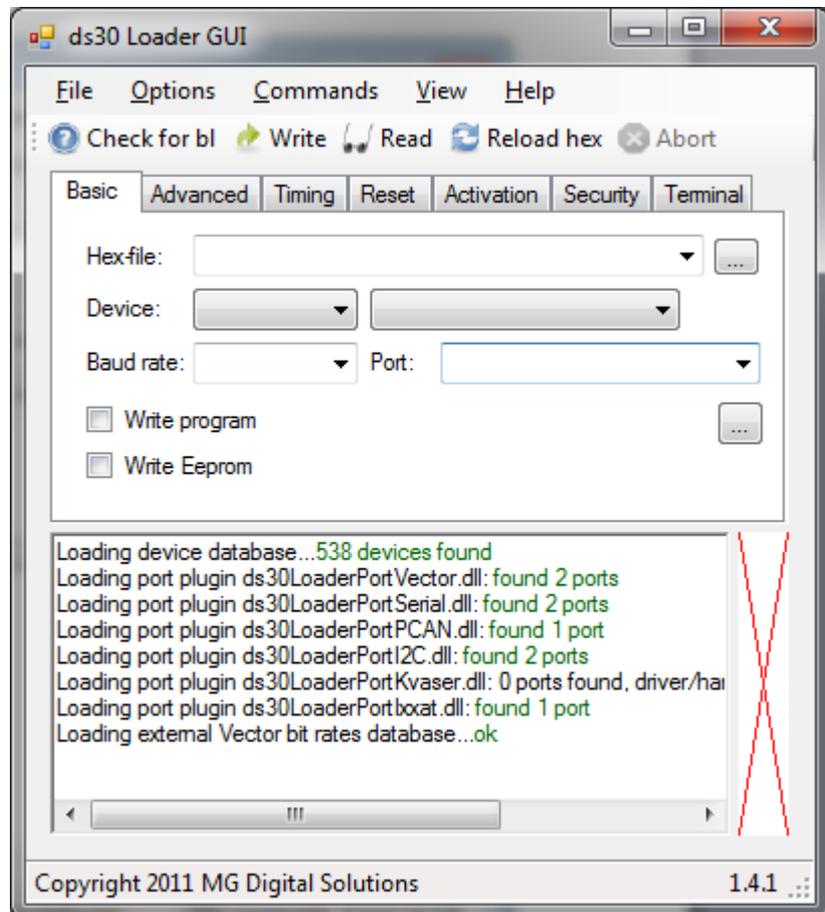
Commands\Write

Initiates a write operation. If write program, write eeprom and write configs are all unchecked a message box will appear saying "Nothing to do". This command is also available as toolbar button.

If poll time is set to lower than 100ms, the applications process priority is raised to above normal. The priority is restored after write.

If "Switch to after write" is checked in the Terminal tab, the terminal tab is shown after a successful write and the port is opened.

After write, successful or not, the user application is loaded after the time out time specified in firmware.

Commands\Check for boot loader

Sends the hello command to the boot loader and displays firmware and device information if it is received properly. A write cannot be issued without first resetting the device first. But the user application is still loaded after the time out time.

Commands\Reparse hex-file

Reload the current hex-file.

Commands\Read

Reads the flash and eeprom memory contents and exports them to a hex file.

View\Always on top

When checked, the window stays on top of other windows even when not active.

View\Advanced mode

When enabled all available options are made visible.

View\Info window

Open a window that displays information about the selected device and hex file.

View\Settings directory

Opens the directory where the setting files are stored.

Help\Visit homepage

Opens the ds30 Loader homepage in the default web browser.

Help\About…

Opens the about window.

Toolbar buttons

The toolbar buttons have the same function as the menu items with the same name.

Tab pages

Basic
This page contains all settings that are required for proper operation

Hex-file
This is the file that is to be written. Pressing the "…"-button opens up open an open file dialogue.

Family & device
Chosen device and family are used to parse the hex-file correctly. When device or family is changed, the hex-file is re parsed.

Baud-/bit rate
If the desired rate isn't available, it may be possible to specify a custom rate depending on which port type is selected.

Remote ID
This setting is only available for CAN and I^{2}C ports, it specifies destination CAN ID.

Local ID
This setting is only available for CAN ports. It specifies local CAN ID.

Port
Port used to communicate with the boot loader. If the desired serial port is not available in the drop down list it is possible to manually type in the port name.

"…"-button
When pressed a configuration window is opened for the selected port. Not all ports have a configuration window, when that is the case the button is disabled.

Write program
When checked, the program memory is written during a write operation. This is disabled if the chosen hex-file does not contain any program memory locations.

Write eeprom
When checked, the eeprom memory is written during a write operation. This is disabled if the chosen hex-file does not contain any eeprom memory locations.

Advanced
This page contains advanced settings that are normally not used.

Don't write goto to the user application
When checked, the first instructions will not be moved to just before the boot loader. This can be useful when writing part of an application or only data to flash.

Allow overwrite of boot loader
When checked, write of program that will overwrite the boot loader is allowed. This does not bypass the firmwares own overwrite protection if any.

Write configs
When checked, the config memory is written during a write operation. This is disabled if the chosen hex-file does not contain any config memory locations or if the hex-file contains more config locations than the selected device has. Writing configs are only possible once per power on cycle.

Custom boot loader
Check to allow write to a customized boot loader. Great care must be taken to write correct custom placement and size. If a write is made with the wrong custom parameters, the boot loader may be broken because to GOTO at 0x00 is pointing to the wrong location.

Auto baud rate
When checked, an auto baud rate synchronization character is sent prior to write, read and check for boot loader operations. The firmware must be set-up for auto baud rate detection. Not all firmwares come with this feature.

Echo verification
If the device hardware or firmware is set-up to echo all received data. Select this option to verify all sent data with the echo from the firmware.

Put CRC before goto
Calculates CRC16 value of the program memory and stores the crc value just before the user application goto that in turn is placed just before the boot loader.

Don't write empty pages
When this is checked pages that only contain 0xff will not be written. This may decrease write time and may also resolve the "The hex-file contains code that will overwrite the boot loader" message.

Timing

Hello timeout
Time to send hello command before giving up when no response is received.

Poll time
The interval that the hello command is sent to the boot loader.

Time-out
Communication timeout

Reset

Manual
When checked, the user is responsible to get the device into boot loading state.

RTS
When checked, the rts pin is used to reset the device by holding it high prior to a write operation. The value in the reset time text box specifies the time the rts pin is held high. Hardware support must be present.

DTR
When checked, the dtr pin is used to reset the device by holding it high prior to a write operation. The value in the reset time text box specifies the time the dtr pin is held high. Hardware support must be present.

Command
When checked, a command is sent to the device to request reset or loading of boot loader. The syntax of the reset command in the text box is different between serial and CAN:
Serial: see Terminal Tx hex later in this document for syntax and examples
CAN: id hexval0 hexval1…. For example, 7ff 11 0f ab 3e

The value in the reset time text box specifies the time to wait after the command is sent before trying to communicate with the boot loader.

The baud rate to use to send the reset command is specified in the baud rate drop down list.

Activation

Manual
When checked, the user is responsible to activate the device.

RTS
When checked, the rts pin is used to activate the device by holding it high during a write operation. Hardware support must be present.

DTR
When checked, the dtr pin is used to activate the device by holding it high during a write operation. Hardware support must be present.

Security
Encrypted hex file
This is checked if the loaded hex file contains encrypted data.

Signed hex file

This is checked if the loaded hex file contains signed data.

IV Counter
If the hex file contains encrypted data the iv counter is displayed

Password
The password will be sent before any boot loader operation is performed.

Save password
Check this if you want the password to be remembered. It is not stored in a safe way.

Terminal

Baud rate
The baud rate used to communicate with the firmware. If auto baud rate is not checked, the baud rate must match the baud rate set in firmware.

Open
Opens the port specified on the basic page.

Close
Closes the port

RTS
Sets/clears the RTS pin. This option is only enabled for serial ports.

DTR
Sets/clears the DTR pin. This option is only enabled for serial ports.

Tx text
Transmit of text, type the text and press enter to send.

Tx hex
Transmit of data, type the data and press enter to send. The format of the data is hexval1;'character1';"string"….
Example hex only: 0;11;f;ab;3e
Example characters only: 'H';'e';'l';'l';'o'
Example strings only: "Hello";"Reset"
Example mixed: a5;'R';"eset"

Clear Rx
Clears the receive text box

Switch to after write

When checked, the terminal page is showed and the port is opened after a successful write operation.

Output text box

Information, warnings and errors are outputted to this text box.

Graphical hex-file representation

This bar represents the entire device flash memory. Eeprom and configs memory are not shown. Because the boot loader size is unknown until communication has been established it may be partly incorrect. Colours:

- Green – user application
- Orange – boot loader
- Red – colliding user application and boot loader

Operation

Parsing of hex-file

The specified hex-file is re-parsed on the following events:
- Change of file name
- Change of device family
- Change of device
- Change of custom boot loader setting
- During write operation if needed
- When window gets activated and the file time stamp has changed
- Toggling of debug mode
- Toggling of "Put CRC before goto"

Here follows a list of operations that are done during parse.

1. Validation

The contents of the hex-file are controlled making sure it seems OK. Three things are checked, checksums, file format and file completeness. If any error is detected, the parsing is aborted.

2. Data collection

All data found in the hex-file that fits in the selected devices memory area are stored in memory buffers.

3. Check of data that could overwrite the boot loader

If data found in the hex-file belong to the same memory space as the boot loader, a warning is displayed and the Write program check box gets disabled.

4. Check and move of GOTO

This step is not performed if the hex-file is encrypted. If no GOTO is found at address 0x00 the boot loader does not know how to load the user application and the Write program check box gets disabled. If a GOTO at 0x00 is found, it is moved to the two words just before the boot loader.

5. Counting of data

The data in the buffers are counted for presentation.

6. Set GOTO to boot loader

A GOTO to the boot loader is inserted in the first words beginning at 0x00.

Write

Here follows a list of operations that are done during a write operation.

* Raise of process priority

If the poll time is set lower than 100ms, the GUI process priority is raised to above normal.

* Device reset

If activated in the reset tab, the device is reset.

* Auto baud rate

If auto baud rate is checked, the auto baud rate synchronization character (0x55) is sent using poll time until time out is reached or the boot loader responds.

* Find boot loader

The hello command is sent using poll time until time out is reached or the boot loader responds. The boot loader responds with device id and firmware version. The received device id is checked against the selected device in the GUI.

* Determine boot loader size

The boot loader size is determined based on the firmware version.

* Parse hex-file

The hex-file is parsed if needed depending on if the determined boot loader size is different from the one used during last parse.

* Write

Program, eeprom and configs are sent for write by the firmware. If checksum error is detected by the firmware, the GUI retries 3 times. If all 3 tries fail, the write oepration is aborted.

Distribution settings

The settings file contains a section called DefaultStartSettings, the values in this section can be manually modified to make restrictions of the settings in the GUI. This is useful when GUI is distributed to customers. The settings.xml should be placed together with the GUI binary.

Appendix A – Links

ds30 Loader homepage
http://www.ds30loader.com

ds30 Loader free edition homepage
http://mrmackey.no-ip.org/elektronik/ds30loader/

Microsoft .NET Framework Version 2.0 Redistributable Package (x86)
http://www.microsoft.com/download/en/details.aspx?id=19

Microsoft .NET Framework Version 2.0 Redistributable Package (x64)
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=6523

Mono / MonoDevelop
http://www.mono-project.com