

**ds30 Loader**  
**GUI manual**

## Table of contents

Table of contents.....	2
Introduction.....	4
ds30 Loader .....	4
Prerequisites and Requirements.....	4
Trademarks.....	4
Getting started .....	5
Installation.....	5
First start .....	5
Window elements .....	6
Menus.....	6
File\Exit .....	6
Options\Debug mode.....	6
Options\Check for new version on start u.....	6
Options\Reset all settings to default.....	6
Commands\Abort .....	6
Commands\Download.....	7
Commands\Check for bootloader.....	7
View\Micro.....	7
View\Always on top .....	7
View\Advanced mode .....	7
View\Output.....	7
Help\Check latest version .....	7
Help\Visit homepage.....	7
Help\About.....	8
Toolbar buttons.....	8
Tab pages.....	8
Basic.....	8
Advanced .....	8
Timing.....	9
Reset.....	9
Activation .....	9
Terminal.....	11
Output textbox .....	11
Graphical hex-file representation .....	11
Operation .....	12
Parsing of hex-file.....	12
1. Validation .....	12
2. Data collection.....	12
3. Check of data that could overwrite the bootloader .....	12
4. Check and move of goto.....	12
5. Counting of data .....	12
6. Set goto to bootloader .....	12

Download .....	13
1. Raise of process priority .....	13
2. Device reset .....	13
3. Find bootloader .....	13
4. Determine bootloader size .....	13
5. Parse hex-file .....	13
6. Download .....	13

## **Introduction**

### ***ds30 Loader***

ds30 Loader is an open source bootloader for PIC18, PIC24 and dsPIC families of MCUs from Micropchip. It supports all devices in each family out of the box (those in production), only minor adjustments need to be done in firmware. The firmware is written in asm/asm30 and comes with a preconfigured MPLAB-project. The GUI is written in C#.

### ***Prerequisites and Requirements***

.NET framework 2.0 or Mono + Mono.WinForms

### ***Trademarks***

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

## Getting started

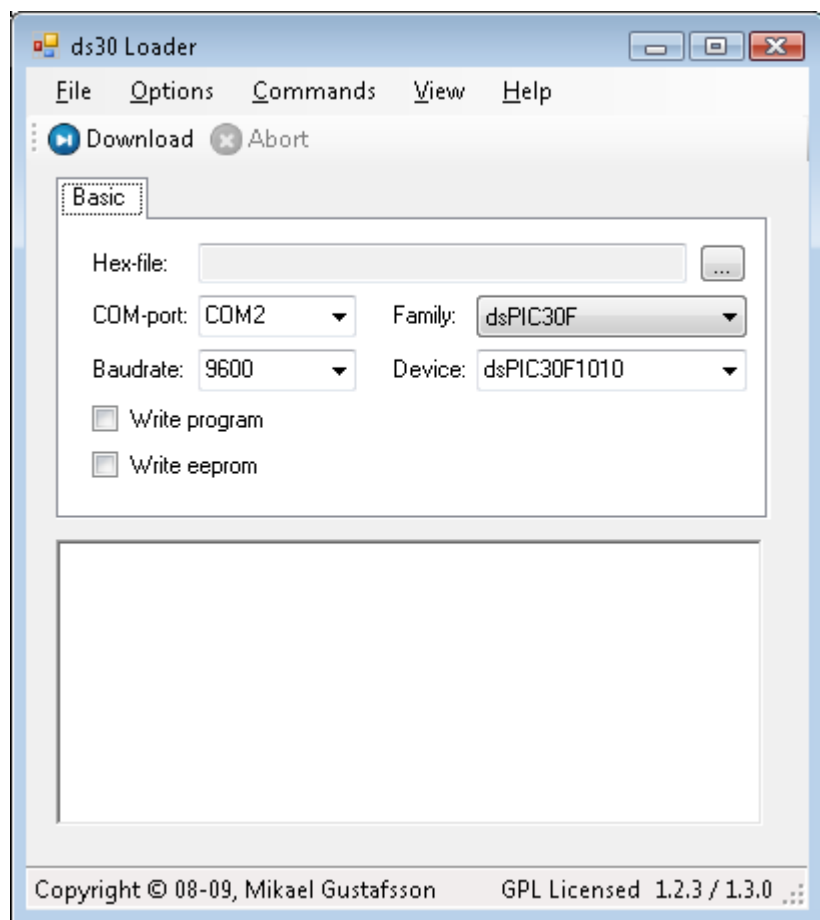
### Installation

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

### First start

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

If you are an experienced bootloader 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 tabpages, the output textbox and the graphical hex-fil 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 download. This option is normally only used when looking for causes of bootloader malfunction. When toggled, the hex-file is reparsed.

#### Options\Check for new version on start up

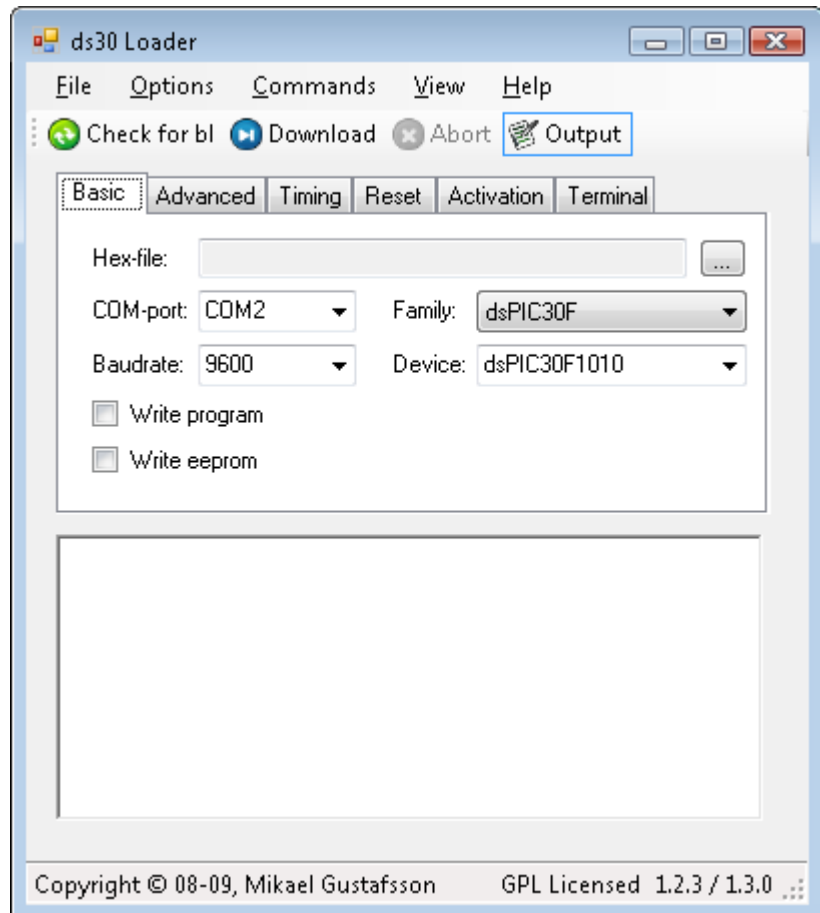
When enabled, the application retrieves an xml-file from the homepage containing information about the latest version publicly available. If a new version is available a message box is displayed.

#### Options\Reset all settings to default

When clicked, all settings are set to their default values. Textboxes and drop down boxes retain their values.

#### Commands\Abort

Aborts the current download or check for bootloader operation. This command is also available as toolbar button.



### **Commands\Download**

Initiates a download. 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 download.

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

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

### **Commands\Check for bootloader**

Sends the hello command to the bootloader and displays firmware and device information if it is received properly. A download can not be issued without first resetting the device first. But the user application is still loaded after the time out time.

### **View\Micro**

Switches to a minimalistic view.

### **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\Output**

Toggles visibility of the output textbox.

### **Help\Check latest version...**

Opens a new window that contains

### **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 downloaded. Pressing the "..."-button opens up open file dialog.

##### **Com-port**

Port used to communicate with the bootloader. If the port is not in the drop down list it is possible to manually type in the port name.

##### **Family & device**

Chosen device and family are used to parse the hex-file correctly. When device or family is changed, the hex-file is reparsed.

##### **Write program**

When checked, the program memory is written during download. 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 download. 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 at 0x00**

When checked, the first 2 instruction are not replaced with a goto to the bootloader. This is useful when downloading data or part of an application.

##### **Allow overwrite of bootloader**

When checked, download of program overwriting the bootloader is allowed.



### **Write configs**

When checked, the config memory is written during download. This is disabled if the chosen hex-file does not contain any config memory locations. Writing configs are only possible once per power on cycle.

### **Custom bootloader**

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

## **Timing**

### **Poll time**

The interval that the hello command is sent to the bootloader.

### **Timeout**

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

## **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 download. The value in the reset time textbox specifies the time the rts pin is held high.

### **DTR**

When checked, the dtr pin is used to reset the device by holding it high prior to download. The value in the reset time textbox specifies the time the dtr pin is held high.

### **Command**

When checked, a command is sent to the device to request reset or loading of bootloader. The format of the command in the textbox is hexval1;hexval2.... For example, 0;11;f;ab;3e

The value in the reset time textbox specifies the time to wait after the command is sent before trying to communicate with the bootloader.

The baudrate to use to send the reset command is specified in the baudrate 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 download.

**DTR**

When checked, the dtr pin is used to activate the device by holding it high during download.

## Terminal

### Baudrate

The baudrate used when opening the uart.

### Switch to after download

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

### Open

Opens the port specified on the basic page.

### Close

Closes the port

### Clear Rx

Clears the receive textbox

### Char

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

### Hex

Transmit of data, type the data and press enter to send. The format of the data is hexval1;hexval2.... Example, 0;11;f;ab;3e

## ***Output textbox***

Information, warnings and errors are outputted to this textbox. It can be hidden from the menu *View\Output*.

## ***Graphical hex-file representation***

This bar represents the entire device flash memory. Eeprom and configs memory are not shown. Colours:

- Green – user application
- Orange – bootloader
- Red – colliding user application and bootloader

## Operation

### ***Parsing of hex-file***

The specified hex-file is reparsed on the following events:

- Change of filename
- Change of device family
- Change of device
- During download if needed
- When window gets activated and the file time stamp has changed
- Toggling of debug mode

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 bootloader**

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

#### **4. Check and move of goto**

If no goto is found at address 0x00 the bootloader do not know how to load the user application and the Write program checkbox gets disabled.

If it is found, it is moved to the two words just before the bootloader.

#### **5. Counting of data**

The data in the buffers are counted for presentation.

#### **6. Set goto to bootloader**

A goto to the bootloader is inserted in the first words beginning at 0x00.

## **Download**

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

### **1. Raise of process priority**

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

### **2. Device reset**

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

### **3. Find bootloader**

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

### **4. Determine bootloader size**

The bootloader size is determined based on the firmware version.

### **5. Parse hex-file**

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

### **6. Download**

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