

**ds30 Loader free edition  
asm firmware manual**



## Table of contents

Table of contents.....	2
Introduction.....	4
ds30 Loader.....	4
Development environment.....	4
Tool suite requirement.....	4
User application compiler.....	4
Flash space requirement.....	4
Supported communication.....	5
UART.....	5
Software UART.....	5
CAN.....	5
I <sup>2</sup> C.....	5
SPI.....	5
SD Card.....	5
USB.....	5
Trademarks.....	5
MPLAB X IDE project.....	6
Files.....	6
ds30Loader.asm / ds30Loader.s.....	6
devices*.inc.....	6
settings.inc.....	6
uart.inc.....	6
user_code.inc.....	6
xxx.gld.....	6
Usage.....	7
Errata's.....	7
Select device.....	7
Configure boot loader settings.....	7
Configuration bits.....	9
MPLAB IDE + MPASM.....	9
MPLAB X IDE + MPASM.....	9
MPLAB XC8.....	10
MPLAB XC16.....	10
MPLAB IDE.....	10
Add own initialization code.....	10
Analogue pins.....	10
PPS.....	10
Oscillator.....	10
Boot loader linker script.....	10
Build.....	11
Erase device.....	12
Programmer supported by MPLAB IDE.....	13
Programmer not supported by MPLAB IDE.....	13
Write boot loader to PIC.....	13

---

Programmer supported by MPLAB IDE.....	13
Programmer not supported by MPLAB IDE.....	13
Adapt the user application.....	13
Operation.....	14
Initialization.....	14
Wait for hello command from the host application.....	14
Main loop, command handling.....	14
Erase page command.....	14
Write row command.....	14
Write EEPROM word command.....	14
Communication timeout.....	14
Considerations.....	15
Code protection.....	15
Data stored in flash memory.....	15
Application linker script.....	15
Oscillator.....	15
Using different oscillator settings for boot loader and application.....	15
Interrupts.....	15
PIC18 extended instruction set.....	15
PPS.....	15
Register default values.....	15
Watchdog.....	16
Appendix A – Links.....	17

## Introduction

### **ds30 Loader**

ds30 Loader is a boot loader supporting PIC12F, PIC16, PIC18, PIC24, dsPIC, and PIC32 families of MCUs from Microchip. It supports all devices that supports RTSP(run time self programming) and has enough flash and RAM memory. The firmware is written in assembler or C. The host applications run on Windows, Linux, and macOS.

### **Development environment**

The firmware is delivered with both MPLAB IDE and MPLAB X IDE projects.

### **Tool suite requirement**

Device family	Assembler
PIC12F PIC16F PIC18F	MPASM or MPLAB XC8
PIC24F / H dsPIC30 dsPIC33F	MPLAB XC16
PIC24E dsPIC33E dsPIC33C	not supported by free edition
PIC32	not supported by free edition

### **User application compiler**

The application dont need to use the same compiler as the boot loader.

### **Flash space requirement**

The size and placement information may change without notification.

	Size [PCU]	Size [bytes]	Placement
PIC12F PIC16F	0x100	448	end of flash
PIC18F	0x200	512	end of flash
PIC18FJ	0x400	1024	2nd last page
PIC24FK	0x1C0	672	end of flash
dsPIC30	0x240	864	end of flash
PIC24H dsPIC33F	0x400	1536	end of flash
PIC24FJ	0x400	1536	2nd last page
PIC24E dsPIC33E PIC32MX	not supported by free edition		

## ***Supported communication***

ds30 Loader free edition supports UART communication.

### **UART**

The UART boot loader has the following features:

- Configurable baud rate
- Supports alternate i/o (PIC24F and dsPIC30F)
- Auto baud rate detection
- Transmit enable pin for RS-485 operation
- Available for PIC12, PIC16, PIC18, PIC24, dsPIC, and PIC32
- Written in assembler or C

### **Software UART**

See ds30 Loader commercial edition.

### **CAN**

See ds30 Loader commercial edition.

### **I<sup>2</sup>C**

See ds30 Loader commercial edition.

### **SPI**

See ds30 Loader commercial edition.

### **SD Card**

See ds30 Loader commercial edition.

### **USB**

See ds30 Loader commercial edition.

## ***Trademarks***

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

## **MPLAB X IDE project**

### ***Files***

Files marked with \* need to be modified.

#### **ds30Loader.asm / ds30Loader.s**

This is the main file that contains all firmware code (assembler instructions). Normally no changes need to be done in this file.

#### **devices\*.inc**

This file contains device specific constants such as size of the eeprom memory and the number of UARTs available.

#### **settings.inc**

This file contains all common user customizations such as communication module assignment, baud rate, device and more. This file needs to be modified in order to make the boot loader work for each different hardware setup.

#### **uart.inc**

This file contains UART functionality.

#### **user\_code.inc**

This file holds user code that should execute before and after boot loader operation.

#### **xxx.gld**

This is the device specific linker script needed by the linker for 16-bit PIC/dsPIC. This is part of the XC16 compiler and does not come with the ds30 Loader.

## Usage

Start by opening the firmware MPLAB X IDE project located in the firmware\_XXX directory.

### **Errata's**

No device specific errata workarounds are implemented. The user must read the device errata sheet carefully to make sure there are no problems that could interfere with boot loader operation.

### **Select device**

Select project properties from the File menu and select the correct device from the drop down list.

### **Configure boot loader settings**

Most if not all settings are located in the file settings.inc (assembler firmwares) or board\_XXX.h (C firmwares). All lines commented with XXX needs to be verified/changed. Not all settings are available in any firmware. Here follows a description of all available settings.

**.equ**        **\_30F4011, 1**

Set to your device name. This setting is only valid for the PIC24 and dsPIC firmware.

**LIST**       **P=18F2550**

Set to your device name. This setting is only valid for the PIC12, PIC16, and PIC18 firmwares.

### **FCY**

Set to instruction cycle clock speed (nr of instructions per second). This is only a constant it does not setup any oscillator settings such as PLL, that has to be done manually. This setting is only valid for the PIC24 and dsPIC firmware.

### **OSCF / FOSC**

Set to oscillator frequency. This is only a constant it does not setup any oscillator settings such as PLL, which has to be done manually. This setting is only valid for the PIC12, PIC16, and PIC18 firmwares.

### **BLINIT**

This is the receive timeout in milliseconds for the first hello command sent from the PC client. Decreasing the value means the application will start sooner on power-up. There is an upper limit which depends on the oscillator frequency.

### **HELLOTRIES**

This is how many non-hello commands that are discarded on start-up before the boot loader is aborted and the user application is loaded. If a reset command is used this

should be set to the length of the reset command + 1. If no reset command is used this should be set to a low value higher than 0.

#### **BLTIME**

This is the communication receive timeout in milliseconds.

#### **USE\_UARTx**

Uncomment the line matching the uart you are using.

#### **USE\_ALTIO**

Uncomment to use alternative I/O for UART1. This setting is only valid for dsPIC30F devices. More information about the USE\_ALTIO setting is available in the device datasheet.

#### **BAUDRATE**

Set to the desired UART baud rate, the brg value is automatically calculated. If the error of the chosen baud rate exceeds 2.5% an error message will be displayed when assembling.

#### **USE\_ABAUD**

Uncomment to use auto baud rate detection. Please read errata first to make sure there are no problems when using auto baud rate detection.

#### **USE\_BRG16**

Uncomment to use 16-bit baud rate register. Please read errata first to make sure there are no problems when using BRG16=1. More information about the BRG16 settings is available in the device datasheet.

#### **USE\_BRGH**

Uncomment to use high baud rates. Please read errata first to make sure there is no problems when using BRGH=1. More information about the BRGH settings is available in the device datasheet.

#### **USE\_TXENABLE**

Uncomment to use a transmit enable pin allowing RS485 communication.

#### **TXE\_DELAY**

Time in  $\mu\text{s}$  to wait before transmitting after pulling the tx enable pin high.

#### **TRISR\_TXE**

Set to tris register of transmit enable pin.

#### **LATR\_TXE**

Set to lat register of transmit enable pin.

#### **TRISB\_TXE**

Set to bit in tris register of transmit enable pin.

### **LATB\_TXE**

Set to bit in lat register of transmit enable pin.

### **KICK\_WD**

If the watchdog is enablea. Uncomment this line to enable kick of the watchdog in the receive loop. If the watchdog is not enabled this line must be disabled.

### **USE\_READ**

Uncomment to enable read of flash and eeprom contents. This option is only available in the commercial version.

### **PROT\_GOTO**

Comment to disable protection of the goto at 0x00. It is recommended to not disable goto protection. If the goto gets corrupted the boot loader will not be called on start-up.

### **PROT\_BL**

Comment to disable boot loader protection. It is not recommended to disable boot loader protection.

### **BLPL**

Placement of the boot loader in the PIC flash memory, pages/rows from the end. When this is changed by the user the custom bootloader option ds30 Loader GUI must be enabled and the new values entered in the textboxes.

### **BLSIZE**

The size of the boot loader, it is used by boot loader protection. When this is changed by the user the custom bootloader option ds30 Loader GUI must be enabled and the new values entered in the textboxes.

### **config xxx**

See the next section.

## ***Configuration bits***

Setting the configuration bits is a required and vital step to make the boot loader work. Also see the oscillator considerations chapter later in this manual. Information about the configuration bits is found in the device datasheet.

There is a template to use in the last section of the file settings.inc. All available configuration bits can be found at the end of the devices include file. Default include file locations:

### **MPLAB IDE + MPASM**

c:\Program Files (x86)\Microchip\MPASM Suite

### **MPLAB X IDE + MPASM**

c:\Program Files (x86)\Microchip\MPLABX\v5.35\mpasmx

### **MPLAB XC8**

c:\Program Files\Microchip\xc8\v2.31\docs\chips

### **MPLAB XC16**

c:\Program Files\Microchip\xc16\v1.61\support\xxx\inc

### **MPLAB IDE**

When using MPLAB IDE (without X). Make sure to check the checkbox labeled “Configuration Bits set in code” on the menu “Configure->Configuration bits...”.

For PIC18 there is also configuration bits documentation available in MPLAB IDE. Click menu Help->Topics then choose “PIC18 Config Settings” and click OK.

### ***Add own initialization code***

If needed, add initialization and/or exit code in user\_code.inc. On PIC12 and PIC16 devices, do not forget to the correct bank for each register access. Use the BANKSEL macro.

The amount of flash space available for user core varies between firmwares and versions. If more space is need the boot loader size and placement needs to be changed in settings.inc. In the GUI you need to check custom boot loader under the advanced tab and enter the details of the new boot loader properties.

Here are the most common things that may need initialization that is not covered automatically by ds30 Loader:

#### **Analogue pins**

Pins that can be used by the A/D are many times configured as analog on startup. If any of those pins that are to be used by the communication module they need to be configured to be digital. Read more about this in the device datasheet, sections I/O Ports and A/D module.

#### **PPS**

On PICs/dsPICs with the peripheral pin select feature it must be configured manually. There is a template available in user\_code.s. More information about PPS is found in the I/O Ports section of the device datasheet.

#### **Oscillator**

If the internal oscillator is to be used it may need to be configured it for a higher frequency. It is often not set for to maximum frequency on startup.

#### ***Boot loader linker script***

There is usually no need to alter the linker script for ds30 Loader firmware.

### **Build**

- Select "Release" in menu "Project->Build Configuration"
- Start build by clicking menu "Project->Build All"
- Notice any warnings.
- Fix errors. ds30 Loader itself may generate errors, see the table below. For other errors, consult the Microchip tool suite documentation.

<b>Error</b>	<b>Description</b>	<b>Solution</b>
Unknown device specified	The selected device may be not supported	Contact the author to get device support.
Do you need to configure communication pins to be digital? If not, remove this line	This is just a reminder.	Configure A/D if needed and then remove the line that generates the error.
You need to configure PPS	This is just a reminder.	Configure PPS then remove the line that generates the error.
Both UART and CAN is specified	ds30 Loader can only operate with one communication module	Select only one communication module in settings.inc
Neither UART nor CAN is specified	Exactly one communication module must be selected in settings.inc	Select a communication module in settings.inc
Fcy specified is out of range		Change Fcy to be within the devices maximum.
Both CAN ports are specified	ds30 Loader can only operate with one communication module	Select only one communication module in settings.inc
CAN is specified for a device that don't have CAN		Select a communication module that is available for the selected device
CAN2 specified for a device that only has CAN1		Select a communication module that is available for the selected device
Both uarts are specified		Select only one communication module in settings.inc
UART2 specified for a device that only has uart1		Select a communication module that is available for the selected device
Baud rate error is more than 2.5%. Remove this check or try another baud rate and/or clock speed.		Try a different baud rate or oscillator frequency.
overflow in delay calculation	Oscillator frequency and timings may be	

	incompatible	
BLSTART_ is out of range	Oscillator frequency and timings may be incompatible	
BLSTART_ might be out of range	Oscillator frequency and timings may be incompatible	
BLDELAY_ is out of range	Oscillator frequency and timings may be incompatible	
BLDELAY_ might be out of range	Oscillator frequency and timings may be incompatible	
You need to configure PPS	See 4.2	Configure PPS then remove the line that generates the error.
No communication is specified		Select exactly one communication module in settings.inc
CanBus specified for a device that only has uart		Select a communication module that is available for the selected device
UART1 and Canbus specified		Select only one communication module in settings.inc
UART2 and Canbus specified		Select only one communication module in settings.inc
TX enable is not available for CAN		Disable tx enable
UART2 specified for a device that only has uart1		Select a communication module that is available for the selected device
spbrg_value_ is out of range	Oscillator frequency and baud rate may be incompatible	Try a different baud rate or oscillator frequency
spbrg_value_ might be out of range	Oscillator frequency and baud rate may be incompatible	Try a different baud rate or oscillator frequency

### **Erase device**

If code protection is used the device should be erased completely. This may be essential to correct boot loader operation if code protection is used.

**Programmer supported by MPLAB IDE**

On the menu Programmer->Erase Flash Device

**Programmer not supported by MPLAB IDE**

Consult the programmer manual.

***Write boot loader to PIC***

**Programmer supported by MPLAB IDE**

On the menu Programmer->Program

Notice that this step requires an ordinary programmer such as the ICD2. The boot loader itself cannot be used to write the boot loader.

**Programmer not supported by MPLAB IDE**

Consult the programmer manual.

***Adapt the user application***

The user application does not need any adaptation.

## **Operation**

### ***Initialization***

The communication module is initialized.

### ***Wait for hello command from the host application***

The boot loader waits for a command that starts the main boot loader loop. If the hello command is not received within the configured timeout the boot loader branches to the application. If no application exists; the boot loader will perform a software reset.

### ***Main loop, command handling***

The boot loader receives read/erase/write commands from the host application. For each command the packet checksum is verified.

### **Erase page command**

This command sets a flag and the subsequent write row command performs the actual erase.

### **Write row command**

After writing the row the written data is verified.

### **Write EEPROM word command**

After writing the word the written data is verified.

### ***Communication timeout***

When communication timeout eventually occurs; the boot loader branches to the application. If no application exist the boot loader performs a software reset.

## **Considerations**

### ***Code protection***

Depending on configuration, write verification and read operation may not function. Write verification may need to be disabled in this case.

### ***Data stored in flash memory***

If the user application stores data in flash memory, this data must be placed in a separate page/row that does not contain any actual code or it will be overwritten on the next write.

### ***Application linker script***

In some cases when using large data arrays, the linker or assembler may place these in the same place as the boot loader. One way to solve this is to reserve the boot loader addresses in the linker script. Another solution is to place the data array at a specific address that does not interfere with the boot loader memory space.

### ***Oscillator***

It is strongly recommended to use the same oscillator setup for both the boot loader and the user application. If you have code to setup your oscillator and/or pll, it is recommended to move that code to the boot loader.

### ***Using different oscillator settings for boot loader and application***

If the user application is to be run on a battery powered device, the oscillator may be running at very low speed. To still achieve low boot loader write time, one might want to have different oscillator setups for boot loader and application. The solution is to add clock switching/pll initialization code in the boot loader firmware.

### ***Interrupts***

The boot loader does not use interrupt but some interrupt flags will be set. Always clear respective interrupt flag prior to enabling an interrupt in your application.

### ***PIC18 extended instruction set***

ds30 Loader does not rely on the extended instruction set, however the boot loader and the application should use the same setting when compiled to ensure correct operation.

### ***PPS***

The PPS configuration registers are not locked by ds30 Loader.

### ***Register default values***

Some register values are not restored when download is complete. For details, examine the code.

### ***User application***

If the boot loader is activated by resetting the device, there is usually no need to adapt the user application. Performing a device reset is preferred to using call, goto or branch because the boot loader may assume reset values of some registers.

If the boot loader is called, “gotoed”, or branched to from the user application, interrupts should be disabled prior to calling the boot loader.

### ***Watchdog***

A ClrWdt instruction is placed in the receive loop. Depending on configuration this may not be enough. In this case the watchdog should be disabled during boot loader operation.

## Appendix A – Links

ds30 Loader website

<https://ds30loader.com>

ds30 Loader free edition website

<https://picbootloader.com>