

ds30 Loader
Firmware manual

Table of contents

Table of contents.....	2
Document History	4
RevN	4
RevM.....	4
Rev L	4
Rev K.....	4
Rev J.....	4
Rev I	4
Rev H.....	4
Rev G	4
Rev F	4
Rev E	4
Rev D.....	4
Introduction.....	6
ds30 Loader	6
Prerequisites and Requirements.....	6
Trademarks.....	6
The basics	7
Why use a boot loader	7
Different firmware versions	7
The MPLAB project.....	7
ds30Loader.asm / ds30Loader.s	7
settings.inc	7
devices.inc	7
xxx.lkr / xxx.gld	8
Boot loader placement.....	8
Usage	9
0. Errata	9
1. Select device.....	9
2. Customize settings.inc.....	9
3. Add own initialization code	11
4. Build.....	12
5. Write boot loader to PIC	13
5.1 Programmer supported by MPLAB IDE	13
5.2 Programmer NOT supported by MPLAB IDE	13
Considerations.....	14
Default values.....	14
Data stored in flash memory.....	14
Oscillator	14
Using different configs for boot loader and application.....	14
Unplanned download of different oscillator setup.....	14
Linker script	14

Interrupts.....	15
Watchdog	15
User application	15
PPS	15

Document History

RevN

Added PIC12
Revised

RevM

Revised

Rev L

Added PPS to considerations section.

Rev K

Updated the C30 integration instructions.

Rev J

Moved memory map to the main manual
PIC16, PCLAT modification is not longer necessary
Minor improvements

Rev I

Added new append, memory map

Rev H

Added USE_ABAUD and USE_BRGH options

Rev G

More PIC16 fixes

Rev F

Added PIC16F information

Rev E

Added additional details in appendix B

Rev D

Added integration section.
Fixed incorrect page size values in appendix B

Fixed incorrect data in appendix C

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

Depending on which firmware is used, MPLAB ASM30 or MPASM assembler is needed. Although not required, MPLAB IDE is recommended and is used during development.

Trademarks

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

The basics

Why use a boot loader

A boot loader usually allows software upgrade with cheap or generally available equipment such as an RS232 port, as opposed to specialized and expensive equipment such as a PIC programmer. Write time might also be lower with a boot loader.

Drawbacks of using a boot loader include added boot-up time and increase memory usage.

Different firmware versions

The ds30 Loader firmware comes in several different versions. Each designed to work with a specific family of PIC devices. The following firmware's are available:

- PIC16F
- PIC18F
- PIC18FJ
- PIC24F
- PIC24FJ
- PIC24H
- dsPIC30F
- dsPIC33FJ

The MPLAB project

The firmware MPLAB project typically consists of four files:

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.

settings.inc

This file contains all common user customizations such as uart assignment, baudrate, device and more. This file needs to be modified in order to make the boot loader work for each different hardware setup. This file is included by ds30Loader.asm/ds30Loader.s.

devices.inc

This file contains device specific constants such as size of eeprom and number of uarts available. This file is included by ds30Loader.asm/settings.inc.

xxx.lkr / xxx.gld

This is the device specific linker script need by the linker. This does not come with the ds30 Loader; it comes with the Microchip language tool suite.

Boot loader placement

The boot loader is normally placed at the very end of flash memory. This way there is usually no need to reserve space for the boot loader in the linker script. Some device families place configs at the end of flash memory. In those cases, the boot loader is placed so that it ends in the second last page.

	Size	Placement
PIC16F	192 words*	End of memory
PIC18F	7 pages*	End of memory
PIC18FJ	1 page	2nd last page
PIC24F	4 rows*	End of memory
PIC24FJ	1 page	2nd last page
PIC24H	1 page	End of memory
dsPIC30F	8 rows*	End of memory
dsPIC33FJ	1 page	End of memory

* May differ for different firmware versions

Usage

Start by opening the firmware MPLAB IDE project.

0. Errata

No device specific errata workarounds are implemented. You should read the errata's carefully to make sure there are no problems that could interfere with boot loader operation.

1. Select device

Select correct device on the menu *Configure->Select Device...*

2. Customize settings.inc

All lines commented with xxx needs to verified/changed. A short description follows of the most common settings. Not all settings are available in any firmware.

DEV_MODE

Used during development, delete or comment this line.

.equ **__30F4011, 1**

Simply set to your device name. Only valid for PIC24 and dsPIC.

LIST **P=18F2550**

Simply set to your device name. Only valid for PIC16 and PIC18.

FCY

Set to instruction cycle clock speed (nr of instructions per second). Only valid for PIC24 and dsPIC.

OSCF / FOSC

Set to oscillator frequency. Only valid for PIC16 and PIC18.

BLINIT

This is the receive timeout in milliseconds for the first hello command from the PC client. This could be set lower to decrease start-up time.

HELLOTRIES

This is how many non hello commands that is discarded on start-up before the user application is loaded.

BLTIME

This is the communications receive timeout in milliseconds.

USE_UARTx

Uncomment the line matching the uart you are using.

USE_ALTIO

Uncomment to use alternative i/o for uart 1. An error message will be displayed if alternative i/o is not available or if uart 1 is not chosen. Only valid for dsPIC30F.

BAUDRATE

Set to uart baudrate, the brg value is automatically calculated. If the error of the chosen baudrate 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 is no problems when using BRG16=1.

USE_BRGH

Uncomment to use high baud rates. Please read errata first to make sure there is no problems when using BRGH=1.

USE_TXENABLE

Uncomment to use a tx enable pin allowing RS485 communication.

TRISR_TXE

Set to tris register of tx enable pin.

LATR_TXE

Set to lat register of tx enable pin.

TRISB_TXE

Set to bit in tris register of tx enable pin.

LATB_TXE

Set to bit in lat register of tx enable pin.

USE_CANx

Uncomment to select CAN port.

ID_PIC

CAN id/node number for this device.

ID_GUI

CAN id/node number of the ds30 Loader GUI

CAN_BRP, CAN_PROP, CAN_SEG1, CAN_SEG2, CAN_SJW
CAN timing settings.

PROT_GOTO

Comment to disable 0x00 goto protection. 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.

BLSIZE

Size of boot loader, used by boot loader protection

config xxx

Set desired configuration bits, this is a required and vital step to make the boot loader work. Also see the config considerations chapter later in this manual.

The configurations can also be set in MPLAB IDE instead, they are found on the menu Configure->Configuration bits...

3. Add own initialization code

If needed, add init/exit code at designated areas in ds30loader.asm/ds30loader.s. In some firmware's, the space available for user code is restricted to a few instructions. See table below for details. The exact number depends on firmware version and which features are enabled.

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

- A/D configuration, on some PICs, i/o-pins are configured as analog on start up
- Internal oscillator, if you use the internal oscillator, you may need to configure it for a higher frequency

	Words free to use for user code* (varies for different fw versions)
PIC16F	~15
PIC18F	~30
PIC18FJ	>100
PIC24F	~10
PIC24FJ	>100

PIC24H	>100
dsPIC30F	~10
dsPIC33FJ	>100

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.

4. Build

On the menu Project->Build All (Ctrl+F10).

Notice any warning. ds30 Loader itself may generate errors, see the table below. For other errors, consult the Microchip documentation.

Error	Solution
Unknown device specified	The selected device may be not supported, contact the author to get device support.
Do you need to configura uart/can pins to be digital? If not, remove this line	Some pins on some devices are configured as analog on start-up. If they are to be used by the UART or CAN module, the need to be configured as digital. This must be done manually.
PPS restoration is not configured.	Devices
UART/CAN pps is not configured. Read datasheet and configure pps.	
Both UART and CAN is specified	Self explanatory
Neither UART nor CAN is specified	Self explanatory
Fcy specified is out of range	Self explanatory
Both CAN ports are specified	Self explanatory
CAN is specified for a device that don't have CAN	Self explanatory
CAN2 specified for a device that only has CAN1	Self explanatory
Both uarts are specified	Self explanatory
UART2 specified for a device that only has uart1	Self explanatory
Baudrate error is more than 2.5%. Remove this check or try another baudrate and/or clockspeed.	Self explanatory
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	
No communication is specified	Self explanatory
CanBus specified for a device that only has uart	
UART1 and Canbus specified	Self explanatory
UART2 and Canbus specified	Self explanatory
TX enable is not available for CAN	Self explanatory
Both uarts are specified	Self explanatory
UART2 specified for a device that only has uart1	Self explanatory
spbrg_value_ is out of range	Oscillator frequency and baud rate may be incompatible
spbrg_value_ might be out of range	Oscillator frequency and baud rate may be incompatible
No communication is specified	Self explanatory

5. Write boot loader to PIC

5.1 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.

5.2 Programmer NOT supported by MPLAB IDE

Consult the programmer manual.

Considerations

Default values

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

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 download.

Oscillator

It is 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 configs 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.

Unplanned download of different oscillator setup

If one needs to download a different oscillator setup and the boot loader does not already have clock switching code, great care must be taken to make sure that the boot loader will still be operable with the new oscillator setup. There are only a few ways to do this.

The simplest solution is to use the uart command reset method. That way, one can add clock switching code prior to loading the boot loader. It could look something like this in pseudo code:

```
if ( ReceivedBlResetCommand )  
    SwitchToBoot_loaderOscillatorSetup()  
    GotoBoot_loader()  
end if
```

Linker script

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

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.

Interrupts

There are no considerations for interrupts.

Watchdog

A ClrWdt instruction is placed in the receive loop.

User application

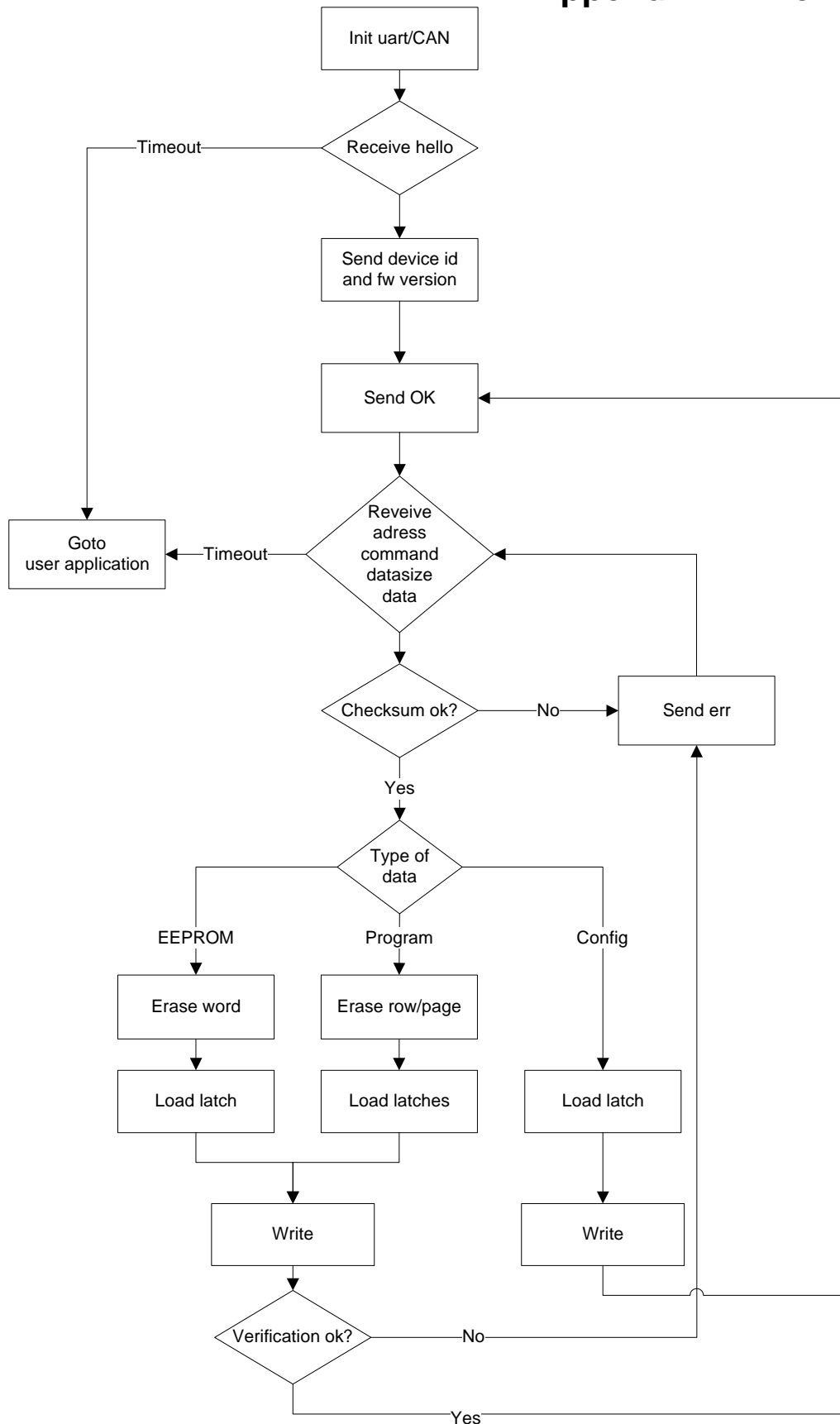
If the boot loader is activated by resetting the device, there is usually no need to adapt the user application.

If the boot loader is called from the user application, interrupts should be disabled prior to calling the boot loader.

PPS

For families with PPS, the firmware has a template for configuring UART/CAN pins. The PPS configuration registers are not locked by ds30 Loader.

Appendix A – flowchart



Appendix B – PIC families details

	Pagesize [words]	Rowsize [words]	Wordsize [bits]	Flash- range	EE- range	Config- range
PIC12F	16/32	16/32	14	0x0 0x1000	0xF000 0xF100	0x8007- 0x8008
PIC16F	1/4/16/ 32/64	1/4/8/ 16/32	14	0x0 0x2000	0x2100 0x2200 0xF000 0xF100	0x2007 0x2008 0x8007 0x8008
PIC18F	64/32	4/8/16/ 32/64	16	0x0 0x20000	0xF00000 0xF00400	0x300000 0x30000D
PIC18FJ	512	32	16	0x0 0x20000	-	-
PIC24F	n/a	32	24	0x0 0x2C00	0x7FFE00 0x800000	0xF80000 0xF8000E
PIC24FJ	512	64	24	0x0 0x2AC00	-	-
PIC24H	512	64	24	0x0 0x2AC00	-	0xF80000 0xF8000E
dsPIC30F	n/a	32	24	0x0 0x18000	0x7FF000 0x800000	0xF80000 0xF8000E
dsPIC33FJ	512	64	24	0x0 0x2AC00	-	0xF80000 0xF8000E
PIC32	1024	128	32	0x1D000000 0x1D080000	-	0x1FC02FF0 0x1FC02FFF