ds30 Loader
Main manual

Table of contents

Document History

September 9th 2011
New sections: firmware matrix and clients
Revised

Rev E
Added appendix C write times

Rev D
Memory map moved from firmware manual
Minor improvements

Rev C
Added schematics

Rev B
Added boot loader upgrade procedure

Rev A
Initial release

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

Requirements for the different parts of ds30 Loader is found in their respective manual.

Trademarks

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.

Usage

0. Operating system considerations

Windows

N/A

Linux

The user running ds30 Loader should be a member of the dialout group. To add a user in the dialout group run the following command as a super user:

```
useradd –G dialout username
```

Mac OS X

The user running ds30 Loader should have sufficient rights to read and write from the desired port.

1. Modify firmware settings

For detailed information, refer to the firmware manual.

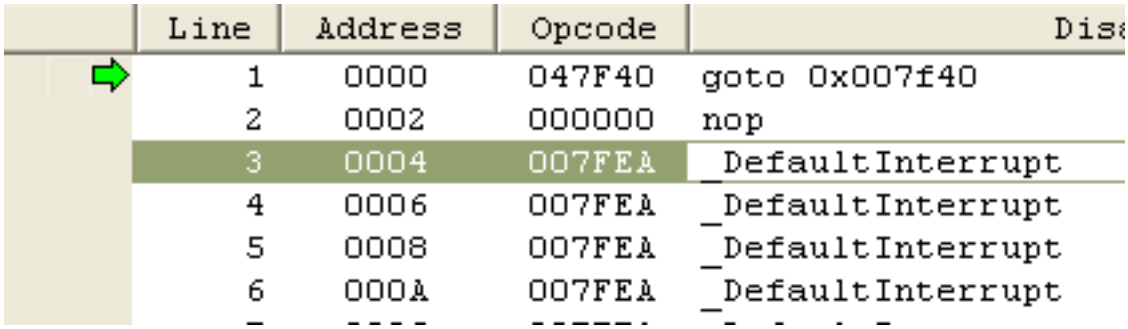
- Start Microchip MPLAB IDE
- Open the firmware by clicking Open on the project menu, then browse to ds30loader.mcp which is located in the firmware directory.
- Open settings.inc by double-clicking it in the project tree to the left
- Modify or verify all values on lines marked with xxx
- Build the firmware by clicking Build all on the project menu
- Notice warnings on the output window
- Correct errors that appears in the output window

2. Write firmware

Write the firmware to your device using your favorite programmer. For detailed information, refer to the firmware manual.

3. Prepare your application

You need to make sure there is a goto placed at location 0x00 in your application. In most cases your linker already does this for you. To check if it does you can pick Progam Memory from the View-menu in MPLAB. It will look something like the screen-shot below. The GUI will let you know if it does not find a goto.



	Line	Address	Opcode	Dis
➡	1	0000	047F40	goto 0x007f40
	2	0002	000000	nop
	3	0004	007FEA	_DefaultInterrupt
	4	0006	007FEA	_DefaultInterrupt
	5	0008	007FEA	_DefaultInterrupt
	6	000A	007FEA	_DefaultInterrupt

4. Download the application to the device

- Start the GUI
- Select the desired communication settings and pick your hex-file. For details, refer to the GUI manual
- Press the download button
- Reset device manually if needed
- Wait for download to complete

Firmware matrix

The following firmwares are available. For detailed information refer to the firmware manual.

	UART	Sw UART	CAN	I^2C	SD card
PIC12	X	X	n/a	X	-
PIC16	X	X	n/a	X	-
PIC18F	X	X	X	X	X
PIC18FJ	X	X	n/a	X	X
PIC24FK	X	X	n/a	X	coming
PIC24FJ	X	X	n/a	X	X
PIC24H	X	X	X	X	coming
PIC24E	X	coming	X	coming	coming
dsPIC30F	X	X	X	X	coming
dsPIC33F	X	X	X	X	X
dsPIC33E	X	coming	X	coming	coming
PIC32	X	coming	X	coming	X

Clients

Console

A command line interface that runs on Windows, Linux, and MAC OS X. For more information refer to the console manual.

GUI

A graphical interface that runs on Windows, Linux, and MAC OS X. For more information refer to the GUI manual.

API

The API can be use to integrate boot loader functionality and in any .NET application. It must be ordered. Contact MG Digital Solutions for more information.

CAN

CAN operation is not available in the free edition of ds30 Loader.

The ds30 Loader supports boot loading on the CAN bus. See table 1 for available firmwares and table 2 for supported CAN adapters.

Device Family	Firmware available	Supported DLC
PIC12	n/a	-
PIC16	n/a	-
PIC18	yes	1
PIC24F	n/a	-
PIC24FJ	n/a	-
PIC24H	yes	1
PIC24E	coming	
dsPIC30F	yes	1-8
dsPIC33FJ	yes	1
dsPIC33E	coming	
PIC32	yes	1-8

Table 1. Available CAN firmwares

Adapter	Supported bit rates	Windows		Linux		Mac OS X	
		x86	x64	x86	x64	x86	x64
IXXAT	10k, 20k, 50k, 100k, 125k, 250k, 500k, 800k, and 1M	X	X	-	-	-	-
Kvaser	50k, 62k, 83k, 100k, 125k, 250k, 500k, and 1M	X	X	-	-	-	-
Vector	100k, 125k, 250k, 500k, and 1M	X	X	-	-	-	-
Peak	5k, 10k, 20k, 50k, 100k, 125k, 250k, 500k, and 1M	X	X	-	-	-	-
Lawicel	10k, 20k, 50k, 100k, 125k, 250k, 500k, 800k, 1M	*	*	X	X	X	X
ESD	Contact						
IntrepidCS							
NI-CAN							
X = supported and recommended		* = supported			- = not supported		

Table 2. Supported CAN adaptors

Peak

On x86 systems, PCANBasic_x86.dll needs to be renamed to PCANBasic.dll
On x64 systems, PCANBasic_x64.dll needs to be renamed to PCANBasic.dll

Lawicel

There may be issues related to Lawicel CAN adapters, see the Trouble shooting-Known issues section in this document.

The Lawicel CAN232 adapter is shipped with the baud rate set to 57600. It can be increased for a performance improvement. It can be done in the configuration window available in ds30 Loader GUI. The serial interface baud rate can not be changed with the ds30 Loader console application.

If the ds30 Loader console application is used, the serial interface baud rate must be set manually in the Lawicel port configuration file to reflect the baud rate the Lawicel adapter is configured for. After a power cycle the CAN232 adapter defaults to 57600. If this is the desired baud rate, no action is necessary.

Adapters with older software does not support single filter mode which means in extended full filtering is not possible, only the 11 most significant bits are used to filter frames.

Boot loader upgrade procedure

This is not recommended and support is not guaranteed.

This guide assumes you have the ds30 Loader installed in the default location. The default location may differ between different firmware versions . Follow these steps carefully. The most critical step is 16: if it fails the boot loader is likely to stop working.

1. Make a copy of boot loader MPLAB project directory.
2. Open the MPLAB project
3. Open settings.inc
4. Disable goto protection if available
5. Enable boot loader protection if available
6. Change boot loader placement, BLPLP (this may be located in ds30loader.s/asm):

Device family	Boot loader placement of temporary boot loader, BLPLP/BLPLR
PIC16F	Upgrade is not supported
PIC18F	BLPLP × 2 + 1
PIC18FJ	BLPLP + 2
PIC24F	BLPLR × 2 + 1
PIC24FJ	BLPLP + 2
PIC24H	BLPLP + 2
dsPIC30F	BLPLR × 2 + 1
dsPIC33FJ	BLPLP + 2

7. Build the project, Menu Project->Make or F10
8. Open ds30 Loader GUI and choose the hex file generated in the previous step
9. Download as you would with any hex file
10. If not already activated, activate advanced mod on the menu View->Advanced mode.
11. Check “Don’t write goto at 0x00” under the advanced tab
12. Check “Custom boot loader” under the advanced tab
13. Put value of BLPLP/BLPLR from firmware in the placement textbox
14. Put value of BLSIZEP/BLSIZE from firmware in the size textbox
15. Choose the hex file of the new boot loader
16. Make sure the temporary boot loader is active, either using time or use a led to indicate that the temporary boot loader is active.
17. Download
18. Boot loader upgrade is finished

Troubleshooting

Known issues

- With the Lawicel CAN adapters read operation is not possible for most combinations of bit rate and serial interface baud rate. Typically any bit rate above 50k will not work for read flash operation. Write operation still works.

Error messages

Trouble	Possible cause	Solution
"Verification failed"	Broken chip	Replace chip
	Worn out flash	Replace chip
	Chip is used outside specifications	Read data sheet and make sure all specifications are met
	Code protection is used	If code protection is used the device should be completely erased before the boot loader is written.
"Uart rx buffer not empty as expected (is device sending data?)"	This could happen if the device is not restarted properly and is sending data	Restart the device before pressing download
"Hardware detected a framing error"	This might happen when the baud rate error is just on the limit.	Try a different baud rate, higher or lower. Needs to be changed in both ends.
		Try another com-port
"The hex-file contains code that will overwrite the boot loader"	This occurs when the loaded hex-file contains memory locations that could overwrite the boot loader. It can also occur if the wrong device is selected. Because the application does not know the actual boot loader size prior to communication with the boot loader it assumes a size. This value might be wrong, one can tell the application the correct value under the advanced tab.	Select correct device
		Set custom boot loader properties under the advanced tab to the values from the firmware, usually called BLPLP and BLSIZEP in settings.inc
		Free up program memory if needed
		Reserve boot loader memory space

“Hex-file contains more config locations than the device has”	If you use the export function from MPLAB this might happen.	Use the hex-file produced by the assembler or linker.
“Last row containing configs was found in hex file, last page has been disabled. Consult manual for more information.”	On PIC18FJ and PIC24FJ devices the configuration words are stored as the last words in the ordinary flash address space. The configuration words are vital to the PIC operation. To avoid corrupting the configuration words or changing them which could lead to the boot loader not functioning properly the last page in flash will not get written.	To write the last page in flash, check the write configs checkbox under the advanced tab. It is unlikely that the last page will contain any code, therefore it is usually safe to not write the last page and this is also recommended.

HW schematics

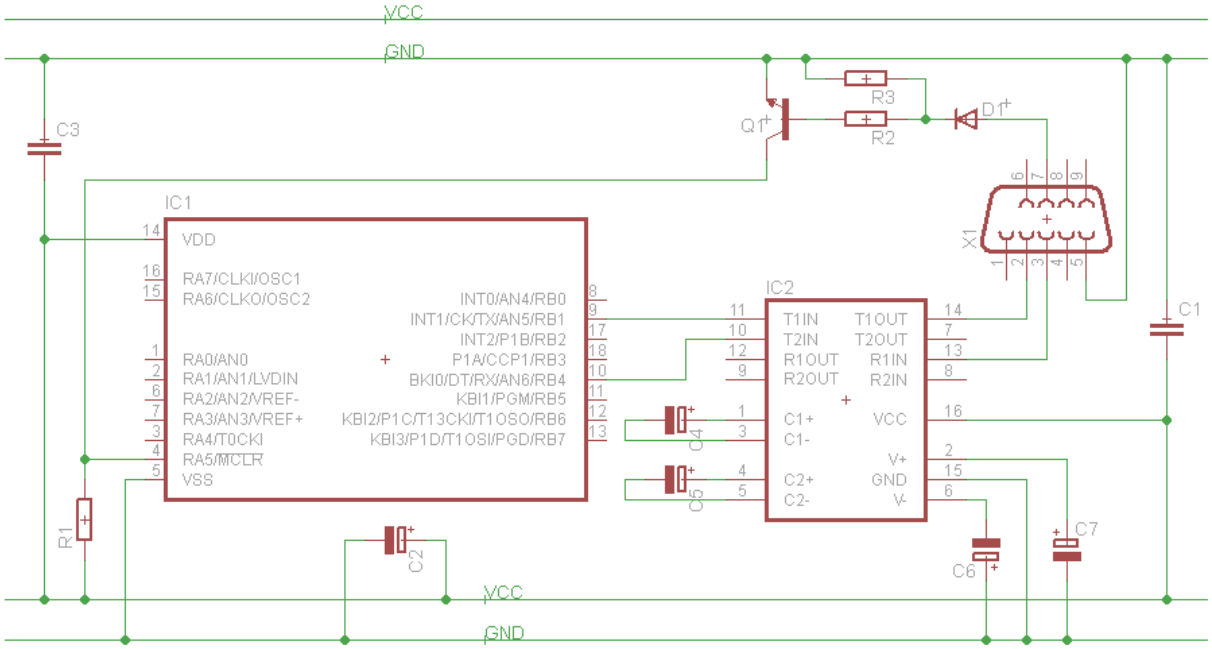
UART / software UART

The schematic comes without warranty. Read datasheets and adapt the schematic to fit your specific application.

Components

Name	Value		Comment
IC1	PIC	Required	PIC microcontroller
IC2	MAX202	Required	RS232 level converter
X1	DB9	Optional	Connector
R1	10kΩ	Required	Master clear pull-up
C1	100n	Recommended	Decoupling capacitor
C2	10µF	Recommended	Filter capacitor
C3	100nF	Recommended	Decoupling capacitor
C4	100nF	Required	Charge pump capacitor
C5	100nF	Required	Charge pump capacitor
C6	100nF	Required	Charge pump capacitor
C7	100nF	Required	Charge pump capacitor
R2	10kΩ	Optional	Reset by RTS, base current limiter
R3	10kΩ	Optional	Reset by RTS, base pull-down
D1	1N4148	Optional	Reset by RTS, RTS signal rectifier
Q1	BC547	Optional	Reset by RTS, RTS signal inverter

Schematic



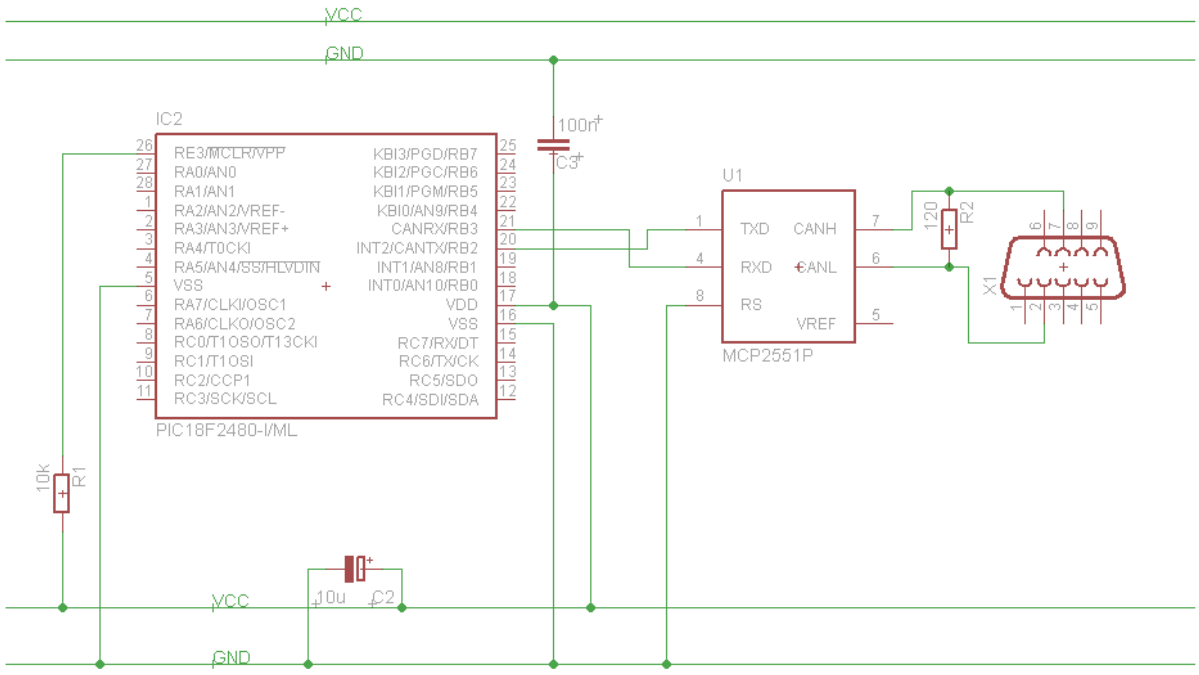
CAN

The schematic comes without warranty. Read datasheets and adapt the schematic to fit your specific application.

Components

Name	Value		Comment
IC1	PIC	Required	PIC microcontroller
U1	MCP2551	Required	CAN transceiver
X1	DB9	Optional	Connector
R1	10kΩ	Required	Master clear pull-up
R2	120Ω	Optional	Bus termination
C2	10µF	Recommended	Filter capacitor
C3	100nF	Recommended	Decoupling capacitor

Schematic



Appendix A – Links

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

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

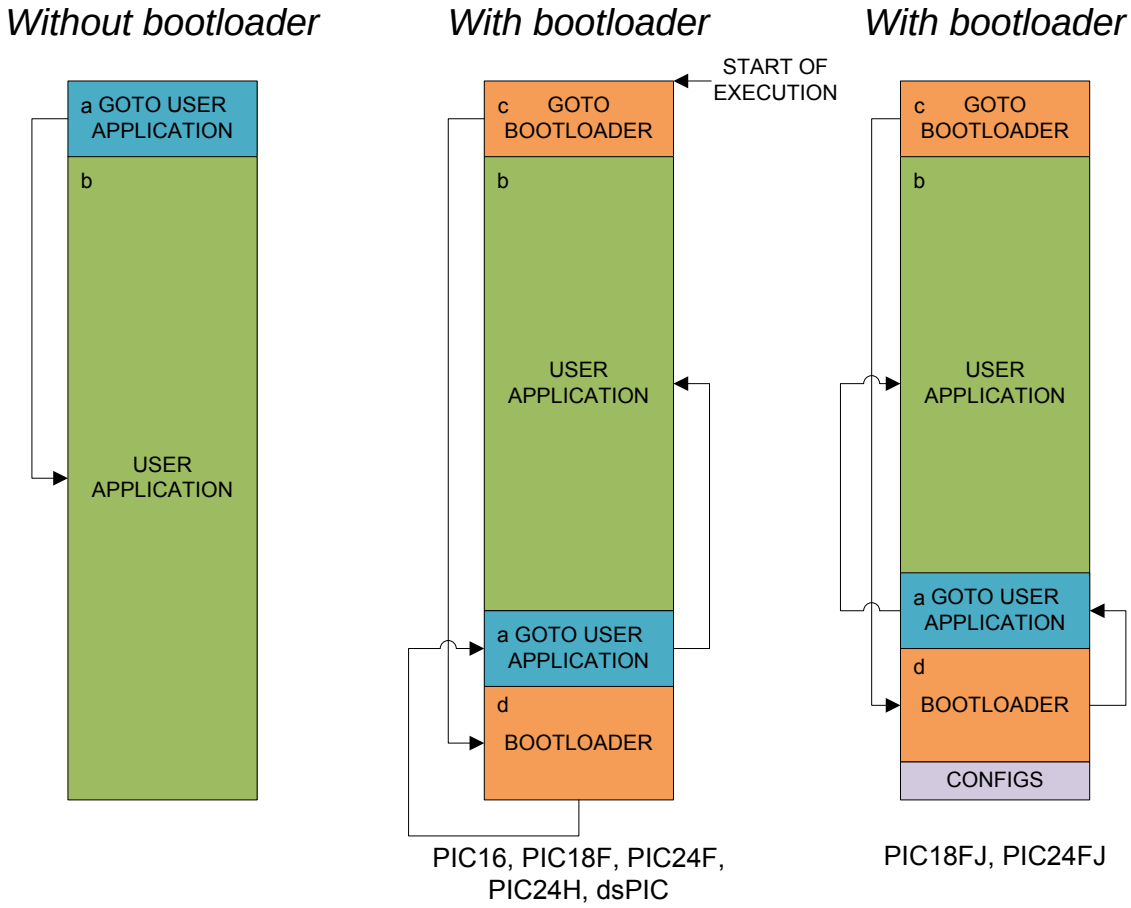
Microsoft Visual C# express edition
http://www.microsoft.com/express/vcsharp/

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

Eagle
http://www.cadsoft.de

Appendix B – memory map

a. The goto application word at 0x00 is normally created automatically by the compiler. The GUI moves the goto to just before the boot loader code.
b. The user application is not affected by the boot loader
c. A new goto at 0x00 pointing to the boot loader code is created by the GUI.
d. The boot loader code is normally placed at the end of the memory.



Appendix C – Write time samples

These are sample times and should not be seen as a warranty, performance may be different in your application. Data is actual data that ends up in the flash memory.

UART	data [kB]	baud rate	time [s]	kB/s	Device
PIC12F					
PIC16F	14,0	115 200	16,5	0,8	PIC16F1936
PIC18F	31,5	115 200	8,3	3,8	PIC18F2550
PIC18FJ	14,0	115 200	2,8	5,0	PIC18F24J11
PIC24F	15,5	115 200	2,3	6,7	PIC24F16KA102
PIC24FJ	125,0	115 200	15,5	8,1	PIC24FJ128GA010
PIC24HJ	126,5	115 200	15,8	8,0	PIC24HJ128GP504
PIC24E					
dsPIC30F	47,3	115 200	7,7	6,1	dsPIC30F4011
dsPIC33FJ	254,5	115 200	31,9	8,0	dsPIC33FJ256GP710
dsPIC33E	509,0	115 200	76,4	7,2	dsPIC33EP512MU810
PIC32MX	508,0	115200	69,4	7,3	PIC32MX795F512L

CAN	data [kB]	bit rate	time [s]	kB/s	Device
PIC18FJ					PIC18F24J11
PIC24HJ	126,5	250 000	40,0	3,0	PIC24HJ128GP504
PIC24E					
dsPIC30F	47,0	250 000	5,9	8,0	dsPIC30F4011
dsPIC33FJ	253,0	250 000	84,0	3,0	dsPIC33FJ256GP710
dsPIC33E					dsPIC33EP512MU810
PIC32MX	508,0	250 000	51,0	10,0	PIC32MX795F512L

Appendix D - Files

Package content

File	Description
\copying.txt	The license for ds30 Loader
\bin	**Contains binaries**
\bin\canlibCLSNET.dll	Use by the Kvaser CAN port plug-in
\bin\ds30 Loader.dll	Contains actual boot loader functionality Requires: GHelper.dll Requires: devices.xml
\bin\ds30LoaderPortIXXAT.dll	IXXAT CAN port plug-in Requires: vcinet.dll Requires: ixxatBitRates.xml
\bin\ds30LoaderPortKvaser.dll	Kvaser CAN port plug-in Requires: canlibCLSNET.dll
\bin\ds30LoaderPortSerial.dll	Serial port plug-in
\bin\ds30LoaderPortVector.dll	Vector CAN port plug-in Requires: vxlapi.dll Requires: vxlapi_NET20.dll Requires: vectorBitRates.xml
\bin\ds30LoaderPortPCAN.dll	PCAN CAN port plug-in Requires: PCANBasic.dll
\bin\GHelper.dll	Contains helper functions used by GUI and console
\bin\PCANBasic.dll	Used by the PCAN CAN port plug-in
\bin\PCANBasic_x64.dll	Used by the PCAN CAN port plug-in, this needs to be manually renamed to PCANBasic.dll on x64 systems.
\bin\PCANBasic_x86.dll	Used by the PCAN CAN port plug-in, this needs to be manually renamed to PCANBasic.dll on x86 systems.
\bin\vcinet2.dll	Used by the IXXAT CAN port plug-in
\bin\vxlapi.dll	Used by the Vector CAN port plug-in
\bin\vxlapi64.dll	Used by the Vector CAN port plug-in
\bin\vxlapi_NET20.dll	Used by the Vector CAN port plug-in Requires: vxlapi.dll
\bin\ds30 Loader GUI.exe	Graphical user interface for ds30 Loader Requires: ds30 Loader.dll Requires: GHelper.dll
\bin\ds30LoaderConsole.exe	Console application for ds30 Loader. Requires: ds30 Loader.dll Requires: GHelper.dll
\bin\devices.xml	Contains information about supported devices
\bin\ixxatBitRates.xml	Contains CAN timings used by IXXAT CAN port plug-in for different bit rates

\bin\vectorBitRates.xml	Contains CAN timings used by IXXAT CAN port plug-in for different baud rates
\documentation	**Contains all documentation for the ds30 Loader**
\firmware xxx	**Contains a MPLAB IDE project for a single Microchip device family**
\firmware xxx\ds30loader.mcp	MPLAB IDE project file
\firmware xxx\src\devices.inc	Contains device constants
\firmware xxx\src\settings.inc	Contains boot loader settings that should be checked/changed before download
\firmware xxx\src\user_init.inc	User initialization and exit code
\firmware xxx\src\ds30loader.s/asm	Contains the boot loader code
\links	**Contains links to web pages that are relevant to the ds30 Loader**
\source	**Contains a Visual Studio 2008 solution**
\tools	**Contains a few tools that may be useful when determining what settings one should use settings.inc**
\tools\dsPIC33 PLL settings.xls	An Excel sheet to help setting up the PLL on dsPIC33 devices. Provided by Leo Bodnar
\tools\UART error calculator	Helpful to calculate uart settings
\tools\CAN timing calculator	Helpful to calculate CAN settings
\tools\bldelay calculator.xls	Calculates timing constants

Files created by ds30 Loader

Created files are usually placed in user home directory\ .ds30Loader

File	Description
settingsPortLawicel.xml	Settings saved by Lawicel CAN port plug-in
settingsPortSerial.xml	Settings saved by serial port plug-in
settingsPortVector.xml	Settings saved by Vector CAN port plug-in
settings.xml	GUI settings
recentfile*.xml	GUI settings for recent files