ds30 SD Card Loader manual

DigSol Sweden AB

Table of contents

Introduction

ds30 SD Card Loader

ds30 SD Card Loader is a boot loader supporting 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 C.

Development environment

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

Tool suite requirement

	Tool suite
PIC18F	MPLAB XC8
PIC24F / H / E dsPIC30F dsPIC33F / E	MPLAB XC16
PIC32MX / MZ	MPLAB XC32

Flash space requirement

	Size		**Placement**
	Standard	**Secure**	
PIC18F PIC18FJ	0x8000	0x8000	End of flash
PIC24FK dsPIC30F	TBD	TBD	End of flash
PIC24FJ PIC24H PIC24E dsPIC33F dsPIC33E	0x3800	0x3800	End of flash
PIC32MX	0x7000	0x7000	End of main flash

Trademarks

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

Firmware

The firmware is written in C and is delivered as a ready to use MPLAB IDE project. It is intended to be used with the Microchip C compilers. The core ds30 Loader functionality is encapsulated in a library.

Files

ds30SDCardLoader.mcp

This is the MPLAB IDE project file that should be opened in MPLAB IDE.

lkr*.lkr/gld

This directory usually contains a single linker script for use with the boot loader firmware.

libs

This directory contains the ds30 SD Card Loader libraries. The files ending in debug contains additional debug prints that are enabled by setting the debug setting to 1. The debug setting is a define the board_xxx.h file.

src\board_xxx.c

This file holds the board specific initialization code and the configuration words. The user is responsible add the appropriate initialization code in the board_init() function. Typically clock and I/O need to be configured.

src\main.c

It contains the following functions:

main()
This function is the entry point, it calls initialization functions, sets up the ds30 Loader settings variable, calls ds30 SD Card Loader and branches to the user application. Additional logic may be added in this function, like only do boot loading under certain conditions (e.g. a button is pressed, handle return values differently).

ds30_activity()
This function is called by the ds30 SD Card Loader engine for each byte that is read. It may be customized by the user. It must not be deleted even if it is empty. Typically the watchdog is kicked in this function if enabled.

ds30_exit()
This function is called by the ds30 SD Card Loader engine just before the user application is started. It must not be deleted even if it is empty.

ds30_info()
This function is called by the ds30 SD Card Loader engine with debug information. It may be customized by the user. It must not be deleted even if it is empty.

src\uart_xx.c/h

These files contain UART functionality that is used to output debug information. UART debug output is enabled from the board_xxx.h file.

src\board.h

This file includes the board_xxx.h file. It may need to be modified.

src\board_xxx.h

This file holds the boot loader and communication settings. It should be modified.

src\ds30SdCardLoader.h

This file contains ds30 Loader types and function prototypes. It is usually not modified.

src\misc.h

This file contains function definitions used for text output.

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 user application linker script.

Usage

Firmware

Start by opening the firmware MPLAB 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.

Configuration bits

Use exactly the same configuration bits from the application in the boot loader unless you have a good reason not to.

Add own initialization code

If needed, add initialization and/or exit code in board_xxx.c.

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

Analog 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.

ds30 Loader configuration

The firmware must be configured by editing the board_xxx.h file.

LOG_TO_UART
When this define is not commented all status and debug info is outputted on the selected UART.

LOG_TO_FILE
When this define is not commented all status and debug info is written to a log file if the SD card is not write protected.

USE_LEDS
When this define is not commented status will be indicated by three LEDs.

DEBUG_MODE
When set to 1 and a debug version of the ds30 Loader library is used additional debug information is outputted from the ds30 SD Card Loader engine.
LED_BL
This define should be set to the lat bit for the boot loader indicator led.

LED_BL_TRIS
This define should be set to the tris bit for the boot loader indicator led.

LED_ACT
This define should be set to the lat bit for the activity led.

LED_ACT_TRIS
This define should be set to the tris bit for the activity led.

LED_ERR
This define should be set to the lat bit for the error led.

LED_ERR_TRIS
This define should be set to the tris bit for the error led.

FCY_MHZ
For 16- and 32-bit devices this should be set to the instruction cycle clock speed (nr of instructions per second). For PIC18F it should be set to the oscillator speed. This is only a constant, it does not setup any oscillator settings such has PLL, that has to be done manually.

USE_UARTx
Uncomment to select UART communication.

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. Only for PIC18F.

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 compiling.

USE_SECURE
This setting enables the ds30 Secure SD Card features when uncommented.

USE_XTEA
This setting enables decryption of incoming data when uncommented.

XTEA_KEY_x
This is the key that is used to decrypt incoming data and also encrypt the IV counter. A key can be generated by ds31 HEX encryptor or can be fetched from an online service such as https://www.grc.com/passwords.htm.

USE_MAC
This setting enables authentication of incoming data when uncommented.

USE_PW
This setting enables password protection when uncommented. When it is enabled all operations require that the password has been entered.

PASSWORD
This defines the password to require when password protection is enabled.

Build

The firmware is built by selecting “Build All” on the Project menu. Make sure to check the output window for errors and warnings when the build is finished.

Write

Write the firmware to the PIC/dsPIC. Refer to the programmer manual how this is done.

SD Card hardware profile

The hardware profile must be modified in the HardwareProfile.h file.

Hex file limitations

The hex file must not contain lines with more than 64 bytes of data. The Microchip tool suites usually output hex file with 16 bytes data on each line. Hex files outputted by the ds33 HEX Rectifier utility are compatible with ds30 SD Card Loader.

SD Card Limitations

- Must be using a FAT16 or a FAT32 file system
- Must not contain file with long file names, only 8.3 is allowed
- Must not contain directories

Hex file preparation

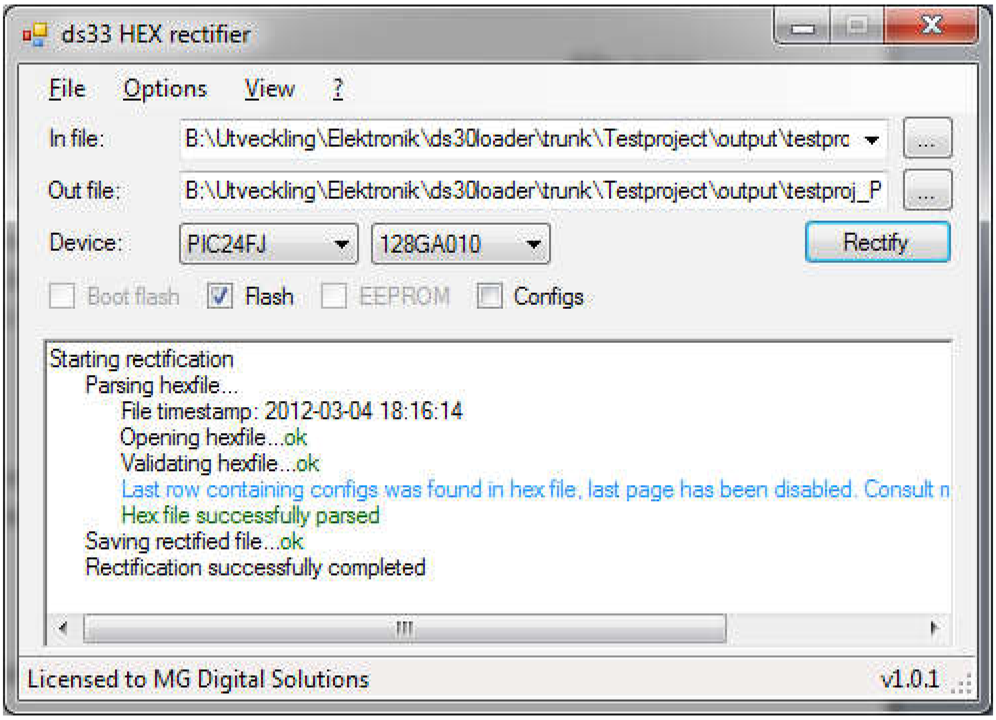
Hex files outputted directly by the Microchip tool suites (such as MPLAB C30) are usually not suitable for usage by the ds30 HEX Loader. This is because it may contain data for an address in multiple places. This cannot be handles by the firmware unless all data is available at a single point in time, which is not feasible. There are multiple solutions to the problem which are follows.

MPLAB X IDE option

MPLAB X IDE has the built-in option "Normalize hex file" which make the outputted hex file usable by ds30 HEX Loader. It is located under Project properties\Building. The project properties are displayed by right clicking the project name in the project tree and selecting properties.

ds33 HEX Rectifier

This is a small utility specifically designed to solve the problem describe above. For PIC18FJ and PIC24FJ the configs checkbox should be unchecked because the configuration bits are located in flash where the boot loader is located.



In file: B:\Utveckling\Elektronik\ds30loader\trunk\Testproject\output\testprc

Out file: B:\Utveckling\Elektronik\ds30loader\trunk\Testproject\output\testproj_P

Device: PIC24FJ 128GA010

□ Boot flash ☑ Flash □ EEPROM □ Configs

```
Starting rectification
    Parsing hexfile...
        File timestamp: 2012-03-04 18:16:14
        Opening hexfile...ok
        Validating hexfile...ok
        Last row containing configs was found in hex file, last page has been disabled. Consult n
        Hex file successfully parsed
    Saving rectified file...ok
    Rectification successfully completed
```

Licensed to MG Digital Solutions v1.0.1

ds33 HEX Rectifier Console can be invoked automatically when building the application in MPLAB IDE or MPLAB X IDE. See the ds33 HEX Rectifier Console manual for information about arguments. A command can also be generated from ds33 HEX Rectifier GUI.

MPLAB X IDE

Right click the project name in the project tree and select properties. In the dialog window select "Building" and check "Execute this command after build". Enter the desired ds33 HEX Rectifier command in the text box below the checkbox.

Sample ds33 HEX Rectifier console command in MPLAB X IDE:

```
ds33HexRectifierConsole.exe  -i=${ImagePath} -o=${ImageDir}\rectified.hex
-d=${Device} --flash
```

MPLAB IDE

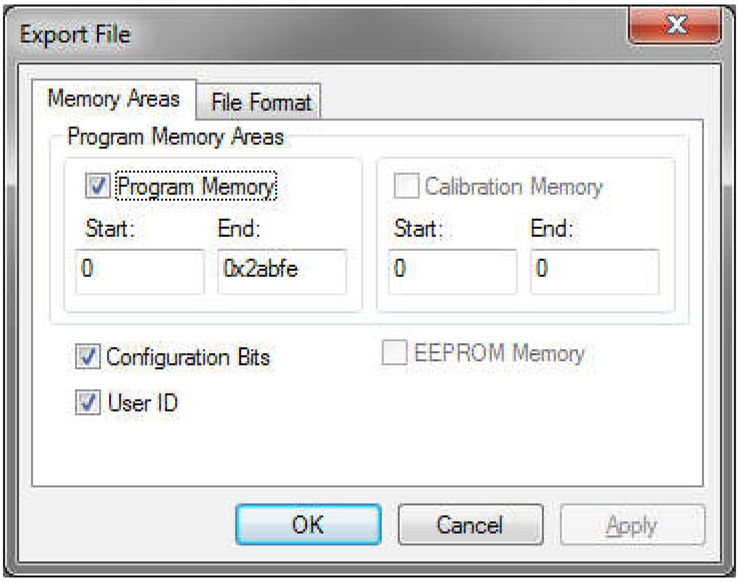
Click menu Project\Build Options…\Project. In the dialog select “Custom Build”. Check “Post-Build Step” and enter the desired ds33 HEX Rectifier command in the text box below the checkbox.

Sample ds33 HEX Rectifier console command in MPLAB IDE:

```
ds33HexRectifierConsole.exe  -i=e:\app.hex -o=e:\app_rectified.hex -
d=dsPIC30F6014A --flash
```

MPLAB IDE Export

On the file menu in MPLAB IDE there is an export option. Clicking it will present a window from which one can export a hex file. The end address should be set to a value that is lower than the boot loader start address. The outputted file is suitable to use with ds30 HEX Loader.



☑ Program Memory
Start: 0
End: 0x2abfe

□ Calibration Memory
Start: 0
End: 0

☑ Configuration Bits

□ EEPROM Memory

☑ User ID

Operation

Hardware initialization

- The user supplied board initialization code is executed.
- The SD card related I/O is configured according to the configuration supplied in the HardwareProfile.h file.
- If logging to UART is enabled the UART is configured according to the settings in the board_xxx.h file.
- If status LED's are enabled the LED I/O pings are configured as outputs.

Setup of boot loader settings

The bootloader settings variable is setup according to the settings in the board_xxx.h file.

File system initialization

If a card is detected the file system is initialized. If the initialization fails and UART logging is enabled an error code is outputted on the UART. If file logging is enabled and the SD card is not write protected the log file is opened in append mode.

Boot loading

- The hex file is opened, if the file does not exist the user application is loaded.
- The hex file is read line by line and the data is stored in a row buffer.
- When a row boundary is crossed the row buffer is written to the flash. If it is also a page boundary the flash page is erased first. The boot loader will protect itself from being overwritten. If it is the first row the goto at 0x00 is saved in a buffer and is replaced with a goto pointing at the boot loader. For the secure version the data is decrypted and the MAC is verified.
- The written data is verified against the RAM buffer. If the verification fails the upgrade operation is aborted.
- When the end of the hex file is reached the saved goto that is pointing to the user application is written.
- The hex file is closed.

Successful upgrade

- If file logging is enabled the log file is closed.
- If UART logging is enabled the UART is uninitialized.
- If status LED's are enabled the boot loader indicator led is turned off and the activity led is turned on.
- The user application is loaded

Failed upgrade

- If file logging is enabled the log file is closed.
- If status LED's are enabled the activity led is turned off and the error led is turned on. The boot loader indicator led remains turned on.

Considerations

The secure version clears the entire RAM when it exits to the appication.

Known limitations

PIC18

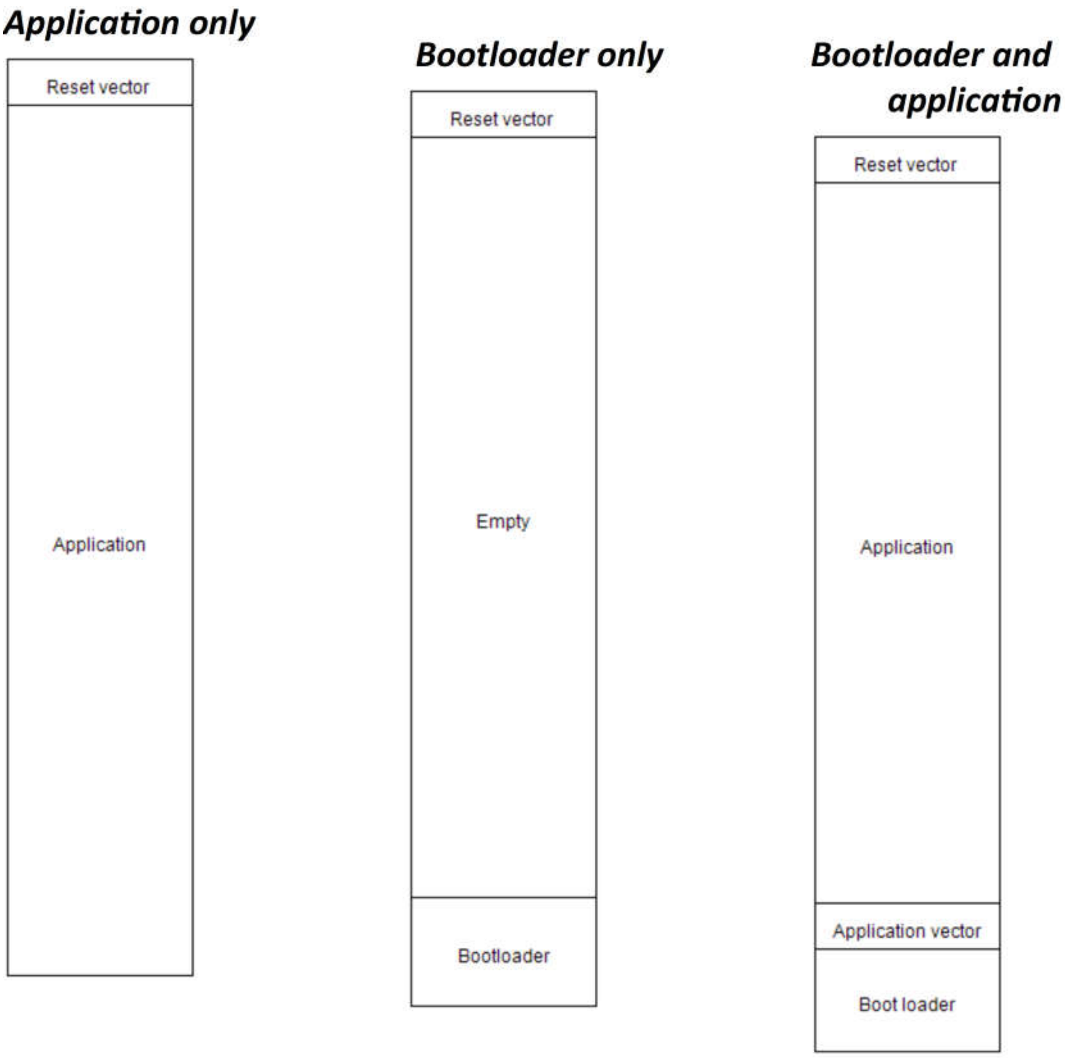
Watchdog Time-out Flag bit

The TO bit in the RCON register will not indicate correctly.

Appendix A – Links

ds30 SD Card Loader website
https://ds30loader.com

Appendix B – memory map


Application only
Reset vector
Application
Bootloader only
Reset vector
Empty
Bootloader
Bootloader and application
Reset vector
Application
Application vector
Boot loader