Using the bootloader with FAT32:

In order to use this bootloader your user application should be located in the program Flash at and above 0x9D00A000.
Almost all the space below this address is used to store the bootoader code.
Thus, add in your user application the following macros:

Code: Select all

```
#define USER_APP_RESRT_ADDR 0x9D004000
#define ConfigMem();  \
        OrgAll(USER_APP_RESRT_ADDR); \
        FuncOrg(__BootStartUp,USER_APP_RESRT_ADDR);
```

In the main function of the user application call the function ConfigMem(). This will ensure that your user application will be linked to the apropriate location in the program Flash.

Actually, this is only all that is needed to prepare the application for being loaded by the bootloader.
Now, since it is a SD bootloader the hex file of the user application should be moved to the SD card.
For some inter-company reasons we have decided to rename it when moved to the SD card to the name BootImg.asr (but it is still the original hex file that has been generated by the MikroC for PIC32 compiler). of course you can change the name to what ever you like, but remember to update the program with the new name.

now burn the bootloader to the mcu (using any programmer like Mikroprog), inser the Sd card to the slot and power up the board (or reset the mcu).

On power up Push the RB0 Button on Fusion V7 board to start the Bootloader. The bootloader then searches for the existence of the hex file (BootImg.asr) on the SD card. If the file is found this means a new version or firmware need to be loaded and it is starting to load it to the mcu. upon end load the bootloader will launch the loaded application and the process is finished.
Now, if you power up or reset the MCU again and the Button is not pressed the bootloader start the application

Attention:

The Bootloader erases all the flash pages that belong to the program flash from the address of 0x9D00A000. this means that if you use some section of this flash to keep your application specific data then **it will be lost** when you use this bootloader to upgrade your firmware. you can adjust the bootloader application to preserve some space that you need.

Check sum

each hex line contains a checksum. Decent application should check this value to verify that the hex lined are received without an error. Unfortunately, although it is such a simple task I have not accomplished it and thus you are encouraged to add it to the bootloader.
Regards,
M. Hooper
(edited by A. Wolf to match recent example Project)