Status : confidential Author: SR Date 13/09/2005
Proofed by :
Date:

New function Boot loader and Monitor V2.3

This description is only valid for Version V2.3

HyNetOS supports such extentions for it's executable files:

HEN (HyNetOS Executable Notcompressed, default now in the system),
HEC (HyNetOS Executable Compressed) and obsolete now HEX (HyNetOS Executable).
HCF extention is NOT supported as it is fully compatible with HEC extention.
It means that if some HCF file will be renamed to HEC file then it will be accepted by the system.

To handle all these extentions we have changed some of our batches.
Switching between extentions is done quite easy. The only thing to do to switch to another extention is to change in SYS_PAR.BAT the setting for EXEC_FORMAT environment variable. Converting from HEX file to HEN or HEC file is done by batches automatically by means of a new tool hyconv.exe.

This tool can also reconvert HEN or HEC files back to HEX if such need will arise but normally in our system files are only converted in one direction - from HEX to HEN or HEC.
These new file formats are better than HEX because their consistency can be checked by means of two check sums contained in the headers of these new files.
This helps preventing of starting corrupted executable files which will hang the system with no possibility to restart it distantly.

Possibility to always have access to the system from distance was seriously kept in mind while creating this release. This means that we have tried to eliminate all this situations when user needs to press reset button on our boards instead of automatically board restarting.

Mostly for this purpose one more new feature was included - so called reserve system.
Previously we had only main system on board represented by PROJECT.HEX file stored either inside System area in flash or in inside flash file system.
If this system became spoiled while, for example, interrupted update of it,
then after reset the board became inaccessible from the network.
Now user can store any of his PROJECT files (HEN, HEC or HEX)
in a special area inside flash called Reserved System area.
This area is placed near Registry area. So now, the consecution of searching the executable to start is following:
a) file inside System area;
b) file with the name PROJECT.HEN, PROJECT.HEX or PROJECT.HEC inside flash file system;
c) file inside Reserve System area.

If any of these files is found and it is consistent, it is started.
It means that if main system file is corrupted, then it will not be started,

but instead reserve system file can be started (if it was previously stored inside Reserve System area).

There are two possibilities to put file inside Reserve System area.
Firstly, by means of hyflash tool.
Such new commands were added to hyflash tool:

a) UPDATERESERVE - writes system file into Reserve System area. If any file was already lying there, it is silently overwritten;

b) ERASERESERVE - cleans Reserve System area (if it contains something) and adds leaned space to the size of free space considered to be inside flash;

c) ERASEREGISTRY - fills Registry area with all zeros;

d) ERASEDISK - frees space in flash occupied by flash file system (if any was there);

e) GETCONFIG - shows in table form on which areas is flash currently partitioned.

Additionally, the meaning of ERASE command is changed: now it is erasing not everything inside flash exept boot loader, but also leaves Registry area untouched.

Secondly, Reserve System area can be updated by means of new systems monitor command UPDATERESERVE.
Such possibility is good in a situation when sending a system file through UDP protocol (hyflash tool uses UDP) is not reliable enough (for example, when the file is send over Internet to some far distance instead of local network). In such a case file can be send to the board via FTP and stored somewhere inside file system (not necessarily in flash but also in ramdisk).
Then connection with systems monitor is established and command like "UPDATERESERVE /ramdisk/PROJECT.HEN" is executed.
This results in the same behaviour as when the command from hyflash tool is executed exept that the file for updating is taken not from Ethernet but from some local disk inside our board. Also monitor now can execute ERASERESERVE and ERASEREGISTRY commands (the same behaviour as for hyflash). To get the same results as for hyflash's GETCONFIG command, inside the monitor "SHOW FLASHCONFIG" command must be executed.

There were also some commands for file/dir operating added in monitor (REN (RENAME), COPY, MD (MKDIR), RD (RMDIR)).

Now at system startup flash disk is checked on consistency. If some errors are found, flash disk is not mounted to prevent system crash during running while attempting to access corrupted flash file system. The only way out of such situation is to rewrite flash disk image into flash by means of hyflash tool. Disk consistency check can also be started from monitor by means of CHKDSK command.
In such a case if some errors will be found during consistency check, disk is not unmounted, only list of found errors is outputed.

Additionally, disk consistency check can be started from users application by means of sync_chkdsk() function. Now we have complete set of sync functions for file/dir access.

hyflash and hyload tools have now protection from trying to operate simultaneously with several boards with the same IP address (this often happens with 10.10.10.10 address).

Writing files to flash via FTP is now much faster because real flashing of written data from memory image of flash disk to flash device itself is done not after each file but after all files in a copied directory.

CompactFlash cards are now supported. When CompactFlash card is inserted it is immediately automatically recognized and a new disk with the name "cflash" appears in the system. Access to this disk either from applications or from monitor or FTP has no any differences than accesses to other system disks

Flash disk now can be operated in read-only mode,this is a possible way to save memory by means of not copying of flash contents in RAM but working directly from flash.

But in such a case it very difficult to do writings to flash. That's why this read-only mode was introduced. To switch on read-only mode for flash disk it is needed to add such line in main() function of MAIN.C file: "FLASH_READ_ONLY = 1;".

Additional blinkings of a red LED to show the stage of boot process.
After our usial first blink comes pause during the board initialises Ethernet chip and waits for possibly incoming load command from hyload or hyflash.
If no command from Ethernet came, board blinks two times with a very short interval between blinks. Then it starts searching inside flash in an above described consecution for an executable system file.
After finishing of this search (no matter was it successful or not), the board again blinks two times with a very short interval between blinks.
These intervals between blinks are so short that it seams sometimes that there where not 4 but 3 blinks.

But I can assure that there are always 4 blinks. Then, if some file was found inside flash, it is started. If the found file is compressed, it is first decompressed. During the decompression process, the LED all the time blinks with rather large intervals between blinks.
It will blink in the same manner when decompressing the downloaded over Ethernet file "on the fly". It must be kept in mind that decompressing process can take several (3 - 7) seconds. After the file is ready to start the LED will very quickly blink 5 times (as it was also in the old boot loaders);