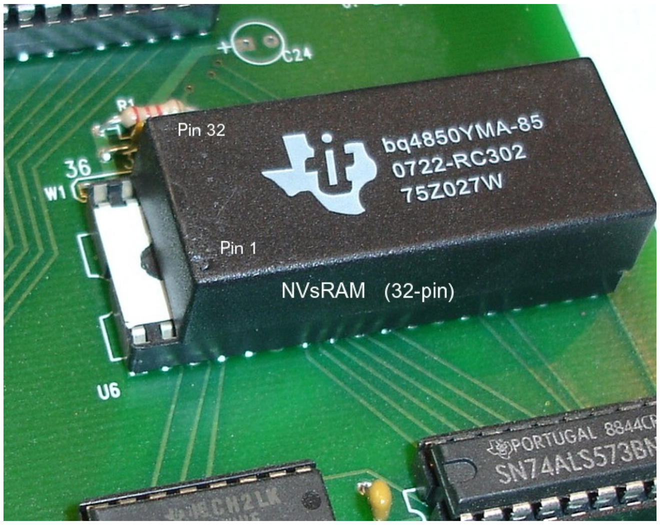
Z-100 LifeLine
Since 1989

March 2022

#WEB

~~~~~~



IDE NVsRAM Reprogramming

by Steven W. Vagts
Editor, "Z-100 LifeLine"

IDE NVsRAM Reprogramming

Every system has been shipped with a preprogrammed NVsRAM for the type of system expected, that is, with or without an MFM hard drive system installed. The NVsRAM is bootable and is designed to be the primary boot device. However, if you have just obtained this IDE Controller from another source to install in a different Z-100 system, or should you desire to change the programming, or if the NVsRAM should lose its programming for some reason, it can be easily reprogrammed.

The NVsRAM chosen for distribution was the Texas Instruments BQ4850YMA, a 512Kb NVsRAM (non-volatile static RAM) model with real time clock. The Dallas DS1647Y model has similar capabilities and is a direct replacement, if necessary. While these models use only 32 pins, the 36-pin socket was installed to allow using the larger 1024Kb models from both companies.

However, if these 1024Kb models include real time clocks, additional hardware and software is required to process this capability. Some clock models include alarms and wakeup capability!

CAUTION: When installing the 32-pin NVsRAMs, ALWAYS leave the four empty pins of the socket to the LEFT! A jumper, located just above the NVsRAM socket, is also required to be modified and jumpered to accommodate the 36-pin units.

Note: Another 2-pin jumper is located just above the NVsRAM that can be cut to ensure write protection to the NVsRAM. When the trace between the pins is cut on the back of the board and a jumper is NOT installed, the NVsRAM will be WRITE-PROTECTED.

The easiest means of reprogramming the NVsRAM is to reload the original programming, using the utility EPWRFILE and a disk image file, EPTEST .DAT. For those individuals who ordered IDE devices with their IDE controller boards, the EPTEST.DAT file was already created by me and saved in two directories on the IDE device:

\EPROM0 (No MFM Drives), and
\EPROM1 (With MFM Drives attached)

For those installing their own IDE devices, I recommended in the installation procedures that you do the same using the procedures provided later.

Initial Programming Procedures

As the NVsRAM is 'READ ONLY' once it is programmed, we must discuss the procedures for programming the NVsRAM.

When we interrupt the boot process by pressing any key at the CONFIG.SYS question, the computer will replace the message with:

Select CONFIG.SYS option (A-Z) ->_

Then we can press a key to choose a configuration option. To choose to enable programming the NVsRAM, we need to press **{P}**.
~~~~~~

~~~~~~

CAUTION:

When the NVsRAM is prepared for programming, the write protection is removed from the NVsRAM. The checksum thus becomes changed and, unless you run CHKSUMEP before rebooting or shutting down the computer, the NVsRAM will be reported as **corrupt** on the next bootup! You will have to reboot to another device and run CHKSUMEP before booting again to the NVsRAM.

~~~~~~

Note: The Bootup process and explanation of DRIVECFG and CONFIG.SYS are fully discussed in the full article, "**IDE_ControllerART.pdf**" also available on the "*Z-100 LifeLine*" Website.

The computer will load the EPROMDSK.SYS driver and update the drive list in the IO.SYS boot screen to reflect new K: and L: drives:

```
    K:  512K EPROMDSK/CLOCK (programming)
    L: ST3290      FAT16    261MB LIDEp 0
```

and below the window, the boot session is completed when we see the DOS prompts:

```
    J:\>ECHO OFF
    J:\>_
```

Note: Depending upon your configuration of DRIVECFG, your drive letters will be different.

EPROMPGM.BAT Batch File

The batch file, **EPROMPGM.BAT**, is used to actually program the NVsRAM on the IDE controller card. EPROMPGM.BAT copies all the files that you wish to transfer to the NVsRAM.

For my case, I created an \EPROMPGM directory on my MFM hard drive and copied all the files that I wanted on the NVsRAM to that directory. These could also be copied to a floppy drive, just change the batch file to use the correct source location.

The files that I wanted were:

```
  COMMAND.COM   IO.SYS        MSDOS.SYS
  AUTOEXEC.BAT  EPROMPGM.BAT  CONFIG.SYS
  EPROMDSK.SYS  LLIDEHD.EXE   CHKSUMEP.COM
  EPRDFILE.COM  EPCPFILE.COM  EPWRFILE.COM
```

Next, copy any additional files as you may desire into the Root directory on the disk.

```
  ASGNPART.COM  DETECT.COM    DSKCOPY4.COM
  EDLIN.COM     FLAGS.COM     FORMAT.COM
  LOOK.COM      PART.COM      PREP.COM
  SYS.COM       ZDIR.COM      ZFMT207.COM
```

CAUTION: If you are using a large capacity floppy, remember to leave at least 525Kb of space for the EPTEST.DAT data file generated by EPRDFILE during the NVsRAM programming.

Modify PATH in AUTOEXEC.BAT and modify EPROMPGM .BAT to load the files from the floppy drive.

The EPROMPGM.BAT file is already loaded on any IDE device that I prepared, usually in the EPROM0 (without MFM hard drive configuration) or EPROM1 (with MFM hard drive configuration) directories. It is also located on the Z-DOS v4.06 CD-ROM. A simplified version is also listed here:

EPROMPGM.BAT

```
ECHO.
ECHO Also Remember: Any time that the EPROM
     is placed in Program
ECHO Mode, the present checksum is destroyed
     and the EPROM is no
ECHO longer bootable, whether the EPROM was
     programmed or not.
ECHO.
ECHO       Once again, ensure the EPROM is
          drive  p %1  q.
ECHO Type {CTRL}-{C} to exit or
PAUSE
ECHO Copying IO.SYS, MSDOS.SYS, & COMMAND.COM
COPY IO.SYS %1/v
COPY MSDOS.SYS %1/v
COPY COMMAND.COM %1/v
ECHO Copying CONFIG.SYS, EPROMDSK.SYS, and
     LLIDEHD.EXE
COPY CONFIG.SYS %1/v
COPY EPROMDSK.SYS %1/v
COPY LLIDEHD.EXE %1/v
ECHO Copying AUTOEXEC.BAT, SYS.COM, &
     DRIVECFG.COM
COPY AUTOEXEC.BAT %1/v
COPY SYS.COM %1/v
COPY DRIVECFG.COM %1/v
ECHO Copying ASGNPART, DATETIME, DEBUG,
     EDLIN, FC, & FLAGS
COPY ASGNPART.COM %1/v
COPY DATETIME.COM %1/v
COPY DEBUG.COM %1/v
COPY EDLIN.COM %1/v
COPY FC.EXE %1/v
COPY FLAGS.COM %1/v
ECHO Copying FORMAT, LOOK, SETLPS, XCOPY, & ZDIR
COPY FORMAT.COM %1/v
COPY LOOK.COM %1/v
COPY SETLPS.COM %1/v
COPY XCOPY.EXE %1/v
COPY ZDIR.COM %1/v
ECHO Copying CHKSUMEP and EPROMPGM.BAT
COPY CHKSUMEP.COM %1/v
COPY EPROMPGM.BAT %1/v
ECHO Copying SHARE and ICLK??? utilities
COPY SHARE.EXE %1/v
COPY ICLK????.COM %1/v
ECHO Copying EPRDFILE, EPCPFILE and EPWRFILE
COPY EP??FILE.COM %1/v
CHKSUMEP
ECHO  EPROM Programming Completed.
GOTO Y
:X
ECHO This Batch File requires a drive letter
     filespec be added
ECHO for the EPROM...  For example: EPROMPGM K:
ECHO.
ECHO             p EPROM programming NOT
     completed.  q
:Y
ECHO.
ECHO Remember to run CHKSUMEP after making
     any other EPROM changes.
```

Description:

EPROMPGM must be invoked with a drive letter (colon not needed) for the NvsRAM. If the drive letter is not included, you will receive the error at "Y".

WARNING: Ensure the drive letter is correct, or you may erase the wrong drive!!

The **COPY** commands copy the system files to the NVsRAM, followed by all the files from the \EPROMPGM directory on the IDE drive. Adjust the COPY files as necessary for your needs.

Note: The usable size of the NVsRAM is presently limited to 507K. So ensure the files in the \EPROMPGM directory do not exceed this amount, less the size of the system files.

CHKSUMEP is automatically run to calculate the checksum and place that on the NvsRAM.

Once completed, additional files can be copied to the NVsRAM manually, but remember to run CHKSUMEP again and last to validate the programming.

I also recommend running EPRDFILE last to save a data file EPTEST.DAT on the default drive (if it has sufficient space) to save the programming in a backup data file. Upon rebooting, the NVsRAM is ready for use.

To program the NVsRAM from the IDE drive L:, we need to change to the \EPROM directory on the L: drive and then run our EPROMPGM.BAT file that we created earlier. The series of commands are:

```
    J:\>L:
    L:\>CD EPROM{RETURN}
    L:\EPROM>EPROMPGM{RETURN}
```

To program the NVsRAM from the 3-1/2" floppy drive (preparing this backup disk is explained shortly), we do not need an EPROM directory, instead we have our EPROMPGM.BAT file in the root directory and additional files in the \EPROMPGM directory. Booting to the floppy, using the same procedures given above, the command sequence is simply:

```
    B:\>EPROMPGM{RETURN}
```

In either case, the batch file is run and a new CHECKSUM is generated and also placed on the NVsRAM.

Note: EPROMPGM.BAT does not need to be run for minor changes. If you just wanted to add a file, the NVsRAM is treated as any other drive letter. Put the NVsRAM in programming mode, then COPY the new file to the K: drive and, when all the changes are completed, run CHKSUMEP to generate a new Checksum.

Alternate NVsRAM Programming

Once the NVsRAM has been programmed using the above procedures, there is a set of 3 utilities that are of tremendous assistance in maintaining the status of the NVsRAM chip.

EPRDFILE.COM - Reads all the data stored in the NVsRAM and stores it in the same directory as EPTEST.DAT.

When finished running, it reports:

```
    File Successfully Written from EPROM.
```

and EPTEST.DAT is written to the current directory. The file is 525Kb so the EPTEST.DAT command cannot be run from the NVsRAM or 5-1/4" floppy and will require a healthy chunk of a 3-1/2" floppy!

For that reason, the set of three files are placed in the \EPROM directory on the IDE drive.

EPCPFILE.COM - This utility compares the previously saved EPTEST.DAT file with the NVsRAM again to ensure they are the same.

When complete, it will hopefully report:

```
    End of Compare EPROM and File.
```

If there are any differences, each is reported as a separate line. For example:

```
    Found in File 00h in EPROM A0h for
    sector   0 at Offset  645
```

If the files are completely different, these reports will scroll seemingly forever on the screen. Press **{CTRL}-{C}** to exit.

EPWRFILE.COM - This utility makes recovery for a corrupt NVsRAM quick and easy. It copies the data file, EPTEST.DAT, back into the NVsRAM in the event the NVsRAM becomes corrupt for some reason other than internal failure. The NVsRAM does **NOT** require to be in programming mode.

Note: For any of these commands, you can change the name of the NVsRAM image file by adding the new file name as an argument to the command. For example, using the command:

```
    EPRDFILE EPTEST2.DAT{RETURN}
```

will create an image file, EPTEST2.DAT. And the command:

```
    EPWRFILE EPTEST2.DAT{RETURN}
```

will program the NVsRAM from that file.

Note: There is a write enable jumper located beside the NVsRAM on the IDE Controller Card. The pins are not installed and it is presently enabled by a trace between the two pin holes. If you wish to disable ALL writes to the NVsRAM, including EPWRFILE, cut the trace and install pins to enable writing when necessary.

Creating a Backup Floppy

In spite of all the precautions that we have built into the IDE Controller Card and NVsRAM programming, the IDE Card and NVsRAM are critical to the booting process of the Z-100. If for some reason, the NVsRAM is erased, or you have to change it out, it is best to have a bootable backup disk to be able to reprogram the NVsRAM.

Therefore, once the NVsRAM has been programmed and is working to your satisfaction, create a bootable, high capacity (the standard 360Kb, 48 tpi, 5" drive will not work) backup disk.

This floppy disk would contain the files necessary to reprogram the NVsRAM using either method described above (i.e., using EPROMPGM.BAT or using the image file, EPTEST.DAT).

CAUTION: Whatever disk you use, remember to leave at least 525Kb of space for the EPTEST.DAT data file generated by EPRDFILE after the NVsRAM programming. Obviously, it will not fit on the 360K, 48 tpi normal floppy.

Note: A 360K 5-1/4" floppy disk will have enough space for the critical files and could be used to reprogram the NVsRAM using the EPROMPGM.BAT batch file method.

CAUTION: When reprogramming the NVsRAM using EPTEST.DAT, the file on an IDE Device (CF card or IDE drive) partition can **NOT** be used. You must use another boot device, such as an MFM or SCSI hard drive, or floppy drive to run EPWRFILE.

If you attempt to run EPWRFILE from an IDE partition, the file will appear to reprogram the NVsRAM, and will even report "*EPROM successfully written from image file*". However, if you check the programming using EPCPFILE, you will find that the programming was NOT changed. AND if you attempt to boot from it, the NVsRAM will still report being corrupt!

Note: Even booting to a Z-DOS v4 floppy and changing to an IDE partition to run EPWRFILE will not work.

This places those with only a 5" 48 tpi floppy drive with no easy way to reprogram the NVsRAM, because the EPTEST.DAT file is too large to fit! They could still use the Backup Floppy (but without the EPTEST.DAT file) and would need to use the manual reprogramming method (using EPROMPGM.BAT) as described above.

Only those who have installed a 96 tpi 5" drive, an 8" drive, or a 3.5" drive would have the room necessary to copy the NVsRAM utilities and the 524 Kbyte EPTEST.DAT file on a floppy disk.

If you have large capacity floppy capability, I highly recommend that you create an emergency backup bootable floppy disk that contains the NVsRAM utilities and the EPTEST.DAT file for NVsRAM reprogramming purposes. I recommend creating this Work Disk, even if you are using another hard drive system, such as the normal MFM Winchester.

Every purchase of the LLIDE Controller board included a backup floppy disk that included all the files necessary for most NVsRAM functions, except the EPTEST.DAT files. These were placed on any IDE device that I may have programmed as a service to new purchasers of the Controller. If you did not receive any IDE device from me, you can still create your own EEPROM/NVsRAM Working Disk, as follows:

Create a spare bootable disk for whatever floppy disk system that you will be using; 8", 96 tpi 5", or 3". Run the command; **FORMAT B:/s/v**, which will install IO.SYS, MSDOS.SYS, and COMMAND.COM on the floppy to make it bootable.

Locate the bootable 5" floppy disk, "*Z-DOS 4.06 IDE Controller Utilities*" (or similar) and copy all the following files to your new IDE Work Disk:

```
AUTOEXEC.BAT   CHKSUMEP.COM
COMMAND.COM    CONFIG.SYS
DRIVECFG.COM   EDLIN.COM
EPCPFILE.COM   EPRDFILE.COM
EPWRFILE.COM   EPROMPGM.BAT
FC.EXE         FLAGS.COM
FORMAT.COM     LOOK.COM
ICLKCAL.COM    ICLKREG.COM
ICLKSET.COM    ICLKTST.COM
LLIDEHD.EXE    SETLPS.COM
IO.SYS         MSDOS.SYS
README0.DAT    SHARE.EXE
XCOPY.EXE      ZDIR.COM
```

Locate the directories, EEPROM0 (for NO MFM drives) or EEPROM1 (with MFM hard drives) that I included on any IDE device that I programmed for you and copy the appropriate EPTEST.DAT file to the new Work Disk.

Note: If you do not have these files, or cannot locate them, boot to the new Work Disk, or change the default drive to the drive your Work Disk is in, and run the command EPRDFILE. A new EPTEST.DAT image file will be created from the current NVsRAM programming and saved to your Work Disk for future use.

To check the programming of the NVsRAM, just run the command **EPCPFILE EPTEST.DAT.**

To reprogram the NVsRAM, simply run the command **EPWRFILE EPTEST.DAT.**

Finally, reboot to the floppy and reprogram the NVsRAM to ensure the floppy disk works as intended. Label the disk and store it in a safe location.

If you have any questions or comments, please email me at:
z100lifeline@swvagts.com

Cheers,
Steven W. Vagts

Steve