T A P R
Tucson Amateur Packet Radio
8987-309 E. Tanque Verde Rd #337
Tucson, Arizona ¥ 85749-9399
Office: (940) 383-0000 ¥ Fax: (940) 566-2544
Non-Profit Research and Development Corporation

The TAPR EVM-RI Operations Guide

DRAFT – Last Revised Saturday, March 06, 1999

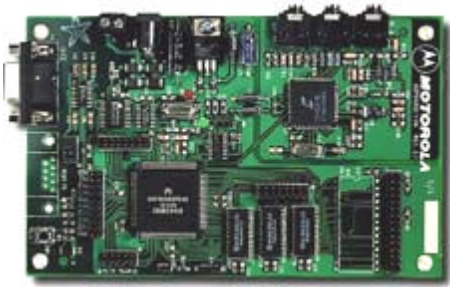

Table of Contents

Appendix A – Documentation and Software
Appendix B – N1OWU Software Distribution Contents
Appendix C – Alef Null Download Program Options

Introduction

Motorola's DSP56002 Evaluation Module (EVM) is an excellent platform for the radio amateur and experimenter alike to implement digital signal processing (DSP) techniques. The EVM is very attractive because of its relatively low cost, excellent flexibility, and the availability of abundant radio application software.

In 1998, a TAPR team designed a radio interface kit with enclosure for the EVM called the EVM-RI. It provides the necessary circuitry to interface the EVM to a transceiver and can be programmed as a KISS TNC. The EVM-RI is based on the radio interface designed by Johan Forrer, KC7WW, described in his ***QEX*** article, *Using the Motorola DSP56002EVM for Amateur Radio DSP Projects.* You can read the article online at http://www.tapr.org/taprf/html/kc7ww.qex_article_8_95.html.

The purpose of this operations guide is to introduce the documentation and software available for the EVM and to get your EVM-RI up and running. This manual applies equally to the Motorola DSP56002EVM and the TAPR EVMRI and can only lay the groundwork for understanding the EVM. There continues to be a great deal of work being done with the EVM. The best way to keep up is to participate in the TAPR Special Interest Group mailing lists DSP and HFSIG (http://www.tapr.org/cgi-bin/lyris.pl) on the Internet and visit the TAPR EVM-RI web pages at http://www.tapr.org/tapr/html/Fevmri.html.

Acknowledgments

TAPR would like to thank the following people for making the EVM-RI Operations Manual possible:

Kaj Wiik, OH6EH, Alef Null International
Jarkko Vuori, OH2LNS, Alef Null International
Johan Forrer, KC7WW
Doug Braun, N1OWU
Steve Bible, N7HPR
Greg Jones, WA5IVD
Mark Hammond, KC4EBR
Paul Williamson, KB5MU
Mike Rudzki, N8MR
Tim Baggett, AA5DF

A Brief History Lesson

One of the first amateur radio groups to start experimenting with DSP was the Finnish group called Alef Null International. They developed several DSP projects, two of which were called the DSP CARD 3 and DSP CARD 4. The DSP CARD 3 was based on the Motorola DSP56001 processor and Texas Instruments TLC32044 voice-band analog

interface circuit. The DSP CARD 4 was also based on the DSP56001 but used the Crystal CS4215 coder/decoder (codec).

Alef Null developed several software packages for the DSP CARDs: narrow bandpass filter, QRM and QRN reduction filter, 1200 AFSK modem, and 9600 bps G3RUH modem. Once the cards were made available to a larger user group, more software programs were developed and contributed to the amateur community. You can read more about these projects at:

DSP CARD 3
ftp://ftp.tapr.org/dsp/dsp56001/dsp_card_3/

DSP CARD 4
ftp://ftp.tapr.org/dsp/dsp56001/dsp_card_4/

In 1994 Motorola introduced the DSP56002EVM, an evaluation module designed to familiarize users with the DSP56002 processor. The EVM was based on the DSP56002 processor and the Crystal CS4215 codec, very similar in design to the Alef Null DSP CARD 4 design.

In 1995, Johan Forrer, KC7WW, discovered that the EVM was very similar to the DSP CARD 4. With the help of the Alef Null group, he made the necessary changes to the DSP CARD 4 software so that it would run on the EVM. Johan created a package of software programs for the EVM and designed a simple radio interface for microphone push-to-talk (PTT) and frequency UP/DOWN adjustment. He wrote about his adventures in the August 1995 issue of ***QEX***, *Using the Motorola DSP56002EVM for Amateur Radio DSP Projects*. The article can be read online at http://www.tapr.org/taprf/html/kc7ww.qex_article_8_95.html and the software package is available at ftp://ftp.tapr.org/dsp/dsp56002/evm56k/kc7ww/.

The ***QEX*** article created a following for the EVM. Several more modems were programmed for the EVM, including many new and exotic HF modems such as PSK31. The EVM become the DSP platform of choice for software development and experimentation.

In 1996, TAPR was approached by Motorola and made a purchase of 200 EVMs. They sold out in a matter of weeks at a price less than retail. This put even more EVMs into the hands of the amateur community. Though it is no longer available from TAPR, the EVM continues to be sold by Motorola distributors today. Here's a partial list:

Monterey Tools http://www.montereytools.com
Arrow Electronics 1-800-777-2776 http://www.arrow.com
Newark 1-800-4-NEWARK http://www.newark.com
Wyle 1-800-943-7446 http://www.wyle.com

In 1997, Doug Braun, N1OWU, modified and improved Johan and Alef Null's software package for the EVM. Doug released his version 1.1 in February 1997.

In 1998, a TAPR team designed a radio interface kit with enclosure for the EVM called the EVM-RI containing parts necessary to interface the EVM to a transceiver. The design of the EVM-RI was based on Johan's radio interface and Doug's software modifications. The EVM-RI is compatible with many software programs written for the EVM. Some slight changes may be necessary. This operations guide will get you up and running with the basic modems, 1200 bps AFSK, 9600 bps G3RUH, and 1200 bps PSK. For other programs that run on the EVM-RI, refer to the installation and operation instructions that come with it.

In order to get your EVM running the following steps are required:
1. Verify that the hardware is operational through software included with your EVM.
2. Download and unzip the assembler, BIOS, and application programs.
3. Load the EVM with the BIOS and application programs of choice.

Getting Started

– Ready!

The DSP56002EVM Evaluation Module is fully assembled and tested. It comes with documentation for the DSP56002 processor, CS4215 codec, and development software. You will need to supply a 7-9 volt AC or DC power at 700 mA. The Radio Shack 7.5VAC adapter (273-1655) is a good choice for a power adapter.

[ ] Power supply for the EVM-RI (7-9 Volts AC or DC at 700 mA).

A serial cable will be needed to connect your computer to the EVM. The EVM has two serial ports. The OnCE port is use for programming and debugging and the Host port is used for serial communications with the program the EVM is executing, such as a 1200 bps AFSK KISS TNC. The DOS version of the Domain Technologies Debugger that is packaged with the EVM will only work on COM1 or COM2.

[ ] Serial cable (connect to computer's COM1 or COM2 port and EVM OnCE port).

The software supplied with the EVM is Motorola's DSP5600x cross assembler, Domain Technologies' debugger, installation instructions, user notes, demo programs, and self-test programs.

Web sites of interest:

Motorola Semiconductor Products Sector – Digital Signal Processing:
http://www.mot.com/SPS/DSP/

DSP56002 Evaluation Module overview:
http://www.mot.com/SPS/DSP/products/DSP56002EVM.html

Specifications on the EVM:
http://www.mot.com/pub/SPS/DSP/LIBRARY/56002/56002EVM/002EVMPB.PDF

– Set!

Motorola supplies a software package diskette for the EVM labeled "Debug – EVM." The latest version as of February 6, 1997 is 2.8. This package includes a DOS version of the Domain Technologies debugger. Installation instructions are in the ***Quick Start Guide*** packaged with your EVM or you can download it from http://www.mot.com/pub/SPS/DSP/LIBRARY/TOOLSDOC/56002EVM/QS/QUICKS.PDF

[ ] Install "Debug – EVM" diskette using the instructions in the ***Quick Start Guide***.
or
[ ] Install `EVM28.EXE` (see instructions below).

You may also download the software package from
http://www.mot.com/pub/SPS/DSP/helpline/evm/evm28.exe.

Installation is slightly different if you downloaded the software package. `EVM28.EXE` is a self-extracting file. Create a directory on your hard drive and give it a suitable name such as `C:\EVM`. Copy `EVM28.EXE` into the directory and execute it. There will be a number of files extracted. The main ones are:

`ASM56000.EXE` – Motorola's DSP5600x cross assembler
`DOS4GW.EXE` – Memory manager used by the assembler
`EVM56K.EXE` – Domain Technologies debugger
`README.lST` – DSP56000EVM User Information File
`CLDLOD.EXE` – *.CLD to *.LOD conversion utility
`DEMO.BAT` – 60 Hz demo batch file
`EVMTEST.EXE` – EVM Performance Analysis Program

– Go!

If you are anxious to see if your EVM works, there are demo (`DEMO.BAT`) and test (`EVMTEST.EXE`) programs. Run the 60 Hz demo as explained in the DSP56002EVM ***Quick Start Guide***. Feel free to work the remaining demos. This is a great way to familiarize yourself with the EVM.

[ ] Run demo program using the instructions in the ***Quick Start Guide*** (optional).

NOTE: The Domain Technologies Debugger (DOS version) runs only on COM1 or COM2.

The test program `EVMTEST.EXE` will conduct a performance test. There is no documentation on how to run `EVMTEST.EXE` except for the source code in `EVMTEST.C`.

[ ] Run EVM Test program using the instructions below.

When you run the program it will prompt you to

```
INSTALL LOOPBACK CABLE between Audio IN and Audio OUT
Press any key to continue…
```

[ ] Install a cable with mini stereo connectors on both ends (Radio Shack 42-2387) into jack J9 (IN) and J14 (OUT) on the EVM.

When you "`Press any key to continue…`" the Domain Technologies debugger will start. There will be an error "`file not found`" but it does not seem to affect the program in any way. If you connect headphones to the HDPHNE jack J16, you will hear a series of tones.

After the test is complete, the test asks for the EVM serial number that can be found on the circuit board. Type in the serial number and press the return key. This will create a log file using the serial number with the `*.TST` extension. The program will finish and print a summary of results.

The test program will output to the screen the results of the analog circuitry and memory tests. The analog circuitry results show the DC offset, the noise level, and the response of the analog circuitry to the sequence of tones listed in the first column. The second two columns contain the raw data received by the DSP56002 from the left and right channels of the analog circuitry. The raw data is evaluated in decibels relative to the maximum value and placed in the next two columns, labeled `--dB below MAX--`.

The last column shows the acceptable responses, in dB, of the analog circuitry for the various tones. The dB levels for each channel are compared to the acceptable responses to determine if the DSP56002EVM passed the analog circuitry test. The dB levels must be below the acceptable responses for DC offset, noise, 24 kHz, 23 Hz, 12 Hz, and 6 Hz and above the acceptable responses for 6 kHz, 1.5 kHz, 375 Hz, and 94 Hz for the DSP56002EVM to pass the analog circuitry test. If either channel's dB levels do not satisfy the acceptable responses, an asterisk will be located at the end of the row corresponding to the test that did not pass the analog circuitry test. For example, in Figure 1, the left channel did not satisfy the acceptable response for 6 kHz, the right channel did not satisfy the acceptable response for 1.5 kHz, and both channels did not satisfy the acceptable response for 375 Hz. If the DSP56002EVM passed the analog circuitry test, no asterisks will be seen at the end of the rows, as in Figure 2.

```
 -- DSP56002EVM Performance Analysis    Ver.1.2 --
 EVM Serial No.: 10585             Tue Feb 11 21:53:43  1997

                 LEFT       RIGHT    --dB below MAX--
DC Offset :      65535      65535       6.02     6.02       10.00
NOISE     :          0          0       0.00     0.00       10.00
24     kHz:       3086       2773     -20.52   -21.45      -20.00
 6     kHz:         20      24014     -64.29    -2.70       -6.00 *
 1.5   kHz:      24422         22      -2.55   -63.46       -6.00 *
   375  Hz:      20 19     -64.29     -64.73    -6.00           * *
    94  Hz:      22772      22553      -3.16    -3.24       -6.00
    23  Hz:      15563      15218      -6.47    -6.66        0.00
    12  Hz:       8627       8401     -11.59   -11.82        0.00
     6  Hz:       2701       2567     -21.68   -22.12      -20.00

 Pass:    12      Address:   X:8200  Y:8200
                  Expected:  000000  000000
                  Received:  000000  FF0000
 MEMORY ERRORS FOUND!  (check J12 is in 16k position)
 *** FAIL *** FAIL *** FAIL *** FAIL ***
```

Figure 1 -- DSP56002EVM Test Sample Output—Fail

```
 -- DSP56002EVM Performance Analysis    Ver.1.2 --
 EVM Serial No.: 10585             Tue Feb 11 21:55:39 1997

                 LEFT       RIGHT    --dB below MAX--
DC Offset :      65126        153       5.97   -46.61       10.00
NOISE     :       5240        731       3.99     5.75       10.00
24     kHz:       2643       2492     -21.87   -22.38      -20.00
 6     kHz:      24048      24426      -2.69    -2.55       -6.00
 1.5   kHz:      24142      24528      -2.65    -2.52       -6.00
   375  Hz:      23952      24342      -2.72    -2.58       -6.00
    94  Hz:      22744      23169      -3.17    -3.01       -6.00
    23  Hz:      14750      15050      -6.93    -6.76        0.00
    12  Hz:       7098       7284     -13.29   -13.06        0.00
     6  Hz:        615        704     -34.53   -33.36      -20.00

 Pass:    12      Address:   X:0000  Y:0000
                  Expected:  000000  000000
                  Received:  000000  000000
 -------- PASS ------
```

Figure 2 -- DSP56002EVM Test Sample Output—Pass

The results of the memory test are shown below the results of the analog circuitry test. These results tell how many passes of the external memory test were run and if the EVM passed the test. If all the values in the Address, Expected, and Received fields are zero,

as in Figure 2, the EVM passed the external memory test. If the EVM failed the external memory test, these fields will tell which memory location caused the failure, the value that was expected to be read, and the value that was actually read from that memory location, as in Figure 1.

The last line of the diagnostics tells if the EVM passed or failed the test. If the last line says `*** FAIL ***`, as in Figure 1, double-check the jumpers and the power, serial connection (only COM1 and COM2 are supported), and stereo connections and repeat the test. If the last line says `---PASS---`, as in Figure 2, then the EVM passed the test and is ready for use. Now it's time to make your EVM-RI do some real work.

N1OWU's DSP56002EVM Software Distribution Release 1.1

On February 10, 1997, Doug Braun, N1OWU, released his modified and improved software package for the DSP56002EVM. It is available from http://www.erols.com/dougbert/.

[ ] Download `N1OWU_11.ZIP`.

[ ] Download `BIOS2.ASM`.

NOTE: This is an updated version of BIOS2 than is packaged with `N1OWU_11.ZIP`.

The contents of the `N1OWU_11.ZIP` package are:

`DL.ZIP`	N1OWU version of Alef Null's Download Program.
`EVMCODE.ZIP`	DSP56002EVM `*.asm` files for the EVM.
`PMP11K.ZIP`	KISS-mode version of the Poor Man's Packet (PMP) program for use with the EVM running an AFSK modem.
`PMPS11K.ZIP`	Source code for Poor Man's Packet program.
`SPY.ZIP`	N1OWU version of the Alef Null SPY program.
`N1OWU_C.ZIP`	Tools for using the GNU G56K compiler.

[ ] Unzip the `EVMCODE.ZIP` file into the directory created by the "Debug – EVM" diskette. The contents and description of `EVMCODE.ZIP` are given in Appendix B.

> **CAUTION!** Do not overwrite the existing `ASM.BAT` in the directory with the `ASM.BAT` in `EVMCODE.ZIP`.

[ ] Move the `BIOS2.ASM` file you downloaded into the directory created by the "Debug – EVM" diskette and replace `BIOS2.ASM` in the directory.

1200 bps AFSK Modem

`FSK1200.ASM` is a 1200 bps AFSK modem that is the common modulation used in terrestrial packet radio. It was written by Pawel Jalocha, SP9VRC, of the Alef Null

group. The serial interface is KISS; thus you will need a program that can interface with a KISS TNC. Phil Karn's, KA9Q Network Operating System, NOS, or N8KEI's (KISS version by N1OWU) Poor Man's Packet will both work.

In the steps that follow, you will assemble the `BIOS2.ASM` and `FSK1200.ASM` programs and load them into the EVM. You will run Poor Man's Packet on your computer as a KISS terminal program to communicate with the EVM.

Assemble Programs

[ ] Assemble `BIOS2.ASM`. At a DOS prompt in the directory containing the Motorola assembler and N1OWU programs, enter:

```
asm bios2
```

The resulting output will look like:

```
C:\EVM>asm56000 –a –b –l bios2
Motorola DSP56000 Assembler  Version 6.1.0
Copyright Motorola, Inc. 1987-1996.  All rights reserved.

C:\EVM>
```

The `BIOS2.ASM` file will be assembled into `BIOS2.CLD`. The `*.CLD` format is used by the Domain Technologies Debugger to load files to the EVM.

[ ] Assemble `FSK1200.ASM`. At a DOS prompt, enter:

```
asm fsk1200
```

The output will be similar to that shown above for assembling `BIOS2.ASM`.

You should now have two new files that you assembled: `bios2.cld` and `fsk1200.cld`.

Loading Programs into the EVM

The next step is to load `bios2.cld` and `fsk1200.cld` into the EVM using the Domain Technologies debugger.

[ ] Make certain that the serial cable is connected to the computer's COM1 or COM2 serial port.

[ ] Make certain that the serial cable is connected to the EVM OnCE port.

[ ] Power on the EVM.

NOTE: Then EVM needs to be powered on prior to running the Domain Technologies debugger. The debugger searches for the com port the EVM is connected to.

[ ] Start the Domain Technologies debugger by entering at a DOS prompt:

```
evm56k
```

The red LED on the EVM should light indicating that the EVM is in the debug mode. The debugger should load and display three windows. An `EVM>` prompt will be displayed in the lower left window.

[ ] At the `EVM>` prompt, enter

```
load bios2
```

A dialog box should pop-up saying "`Loading file : bios2.CLD`" that graphically displays loading progress. After BIOS2.CLD completes loading, you will see the `EVM>` prompt.

[ ] At the `EVM>` prompt, enter

```
load fsk1200
```

A dialog box should display the loading progress of the `FSK1200.CLD` file.

Now both files are loaded in the EVM.

[ ] To start `FSK1200` running, enter at the `EVM>` prompt:

```
go 0
```

[ ] Exit out of the debugger by entering at the `EVM>` prompt:

`ALT-X` and answer YES

[ ] Move the serial cable from the OnCE to the Host port on the EVM.

Install Poor Man's Packet

[ ] Create a directory to hold the PMP program such as `C:\PMP`

[ ] Unzip `pmp11k.zip` to the directory.

[ ] Edit `pmp.cfg` file to match your system and callsign.

```
Mycall =
Comport =
Baud Rate = 19200
```

[ ] Start Poor Man's Packet by entering at a DOS prompt:

```
pmp
```

A welcome screen greets you.

[ ] Hit any key to continue. Monitor 2 meter packet.

Alt-H will bring up a help screen

9600 bps G3RUH Modem

`g3ruh.asm`
`filtr.asm`
`filtx.asm` The OH2LNS software, slightly customized for my hardware. I use this daily. It works very well! The files filtr.asm and filtx.asm are needed by g3ruh.asm.

1200 bps PSK Modem

`psk.asm` Patched version of the SP9VRC BPSK PACSAT modem. This version has a rewritten transmitter section and other changes. I am still working on improving the DCD algorithm, but I still use it daily.

Programming the Flash EPROM

To really make the EVM useful, load the Flash EPROM with your choice of default program. Use the `FLSHBOOT.ASM` program to program the Flash EPROM. The following steps assume you have installed a Flash EPROM in the EVM.

WARNING! Use the `FLSHBOOT.ASM` that comes in `N1OWU_11.ZIP` or Motorola distribution version 2.8 or later. There is an error in earlier versions that it can't program all locations in the flash memory.

[ ] Assemble `FLSHBOOT.ASM`. At a DOS prompt, enter:

```
asm flshboot
```

[ ] Connect the serial cable to the EVM OnCE port.

[ ] Start the Domain Technologies debugger.

[ ] Load the `*.CLD` files in this order:

- BIOS2.CLD
- the application of your choice
- FLSHBOOT.CLD

[ ] Start the execution of FLSHBOOT by entering at the EVM> prompt

```
go $3400
```

Wait for the program to finish. The lower right hand corner of the debugger indicates the state of the EVM.

[ ] Exit from the debugger (ALT-X and YES)

[ ] Reset the EVM by pressing the reset button or powering down and then back up.

[ ] Move the serial cable to the EVM Host port.

The EVM will start with the application you programmed into the Flash EPROM. You do not have to continually load a modem program into the EVM each time you want to use it.

Using the Download Program DL

Above you learned how to assemble and load programs into the EVM. There is a much fast way to load programs into the EVM without running the Domain Technologies debugger each time. The Alef Null "DL" program is designed to download programs to the EVM. The combination of DL and BIOS2 makes it very easy to run different modem or audio applications on your EVM. DL was written by Jarkko Vuori, OH2LNS, Rob Janssen, PE1CHL, and modified by N1OWU.

Install the N1OWU bios2.asm into the EEPROM of your EVM.

[ ] Set the environment variable DSPPORT to the COM port number connected to the EVM, either 1 or 2. In the AUTOEXEC.BAT file, type

```
SET DSPPORT=2
```

NOTE: You will need to reboot your computer for the environment variable to take affect.

DL and BIOS2 assume they will communicate with the EVM at 19200 baud.

For each application you run, generate a *.LOD file with the Motorola-supplied CLDLOD.EXE utility. At a DOS prompt, type:

```
cldlod fsk1200.cld > fsk1200.lod
```

To upload and start the application, type:

```
dl fsk1200
```

The various options for the `DL` program are listed in Appendix C.

Appendix A – Documentation and Software

Motorola Semiconductor Products Sector – Digital Signal Processing
http://www.mot.com/SPS/DSP/

DSP56002 Evaluation Module overview
http://www.mot.com/SPS/DSP/products/DSP56002EVM.html

DSP56002EVM Product Brief:
http://www.mot.com/pub/SPS/DSP/LIBRARY/56002/56002EVM/002EVMPB.PDF

DSP56002EVM Quick Start Guide:
http://www.mot.com/pub/SPS/DSP/LIBRARY/TOOLSDOC/56002EVM/QS/QUICKS.PDF

DSP56002EVM Schematics:
http://www.mot.com/pub/SPS/DSP/LIBRARY/TOOLSDOC/56002EVM/SCHEM/

Software package for the DSP56002EVM Version 2.8, February 6, 1997. Package includes a DOS version of the Domain Technologies debugger.
http://www.mot.com/pub/SPS/DSP/helpline/evm/evm28.exe

Alef Null International – Signal Processing Division
DSP CARD 3
ftp://ftp.tapr.org/dsp/dsp56001/dsp_card_3/

DSP CARD 4
ftp://ftp.tapr.org/dsp/dsp56001/dsp_card_4/

Johan Forrer, KC7WW, Using the Motorola DSP56002EVM for Amateur Radio DSP Projects, ***QEX***, The ARRL Experimenter's Exchange, Issue No. 162, August 1995, pp. 14-20.
http://www.tapr.org/taprf/html/kc7ww.qex_article_8_95.html

Software Distribution Version 3
ftp://ftp.tapr.org/dsp/dsp56002/evm56k/kc7ww/

Doug Braun, N1OWU
Software Distribution Version 1.1
http://www.erols.com/dougbert/

TAPR EVM Radio Interface
http://www.tapr.org/tapr/html/Fevmri.html

TAPR Special Interests Groups

http://www.tapr.org/cgi-bin/lyris.pl

Domain Technologies, Inc.
http://www.domaintec.com/

DSP56002 Debugger *EVM56K-Win* (MS-Windows 3.1) Version 3.07 (09/21/98):
ftp://ftp.domaintec.com/domtech/e002_307.zip

Crystal Semiconductor Corporation (now Cirrus Logic Crystal Product Solutions)
http://www.crystal.com

CS4215 CrystalClear 16-Bit Multimedia Audio Codec
Overview: http://www.cirrus.com/products/overviews/cs4215.html
Data Sheet: http://www.cirrus.com/ftp/pubs/4215.pdf

Appendix B – N1OWU Software Package contents of evmcode.zip.

`acars.asm`	A not-yet-working attempt to make an ACARS decoder.
`apt.asm`	A very nice WESAT APT demodulator w/ synchronous detection, designed by N1OWU.
`asm.bat`	A batch file that shows the correct assembler arguments.
`bios2.asm`	N1OWU modified version of the Alef Null Leonid `bios.asm`.
`feedthru.asm`	A trivial application to feed audio input to the output.
`firmacs.asm`	A collection of macros for generating FIR filter coefficients. Used by several of these programs.
`flshboot.asm`	A program to program a Flash ROM in the EVM, with important bug fixes. Meant to work with bios2.asm.
`fsk1200.asm`	The SP9VRC software, slightly customized for `BIOS2`.

`g3ruh.asm`
`filtr.asm`
`filtx.asm`
	`G3RUH 9600 bps FSK modem written by` OH2LNS. Files `filtr.asm` and `filtx.asm` are sourced by `g3ruh.asm`.

`ioequlc.asm`	Motorola-supplied header file with I/O port definitions for 56000-family processors.
`intequlc.asm`	Motorola-supplied header file. Has interrupt vector definitions for 56000-family processors.
`leonid.asm`	Matching header file for `bios2.asm`. Every application sources this file.
`pocsag.asm`	Patched version of PE1CHL's software. Not working yet.
`newqpsk.asm`	A very sophisticated HF modem program by SP9VRC.
`psk.asm`	Patched version of the SP9VRC BPSK pacsat modem. This version has a rewritten transmitter section and other changes.
`qrmqrn.asm`	The OH2LNS audio noise/tone reducer program, heavily modified and slightly improved by N1OWU.
`rtcoeff.asm`	A adjustable-bandwidth audio filter.
`scope1.asm` `scope2.asm`	

`scop9600.asm`	Simple programs that let you look at the audio input with the SPY program. Written by ZS6AWK.
`slowbpsk.asm`	The SP9VRC program with numerous additions by N1OWU. Works quite well.
`tone.asm`	A simple tone-generator program. Works best with N1OWU enhanced `DL.EXE` program, which lets you specify the frequency, level, and channel at run time.
`uo11.asm`	A UO-11 modem, adapted by N1OWU from `fsk1200.asm`. Works fine for receiving UO-11 (satellite) data.
`wefax.asm`	Basically the same as SP9VRC's `fskiface.asm`.
`wesat2.asm`	A simple demodulator for WESAT APT transmissions. `apt.asm` above is much better.
`wesat3.asm`	A WESAT APT demodulator that attempts to work better with "normal" receivers whose IF bandwidths are too narrow and mess up the white values of the signal. It needs input from the FM discriminator (like 9600 BPS modems). Works so-so. Needs tweaking.

Appendix C – Alef Null Download Program Options

1. The program is smart about the application name. You can say things like:

```
dl ..\..\evm\g3ruh
or
dl g3ruh.lod
```

2. To specify a different COM port, use `-p`. Then you must also use `-g` before the application name:

```
dl -p2 -g g3ruh
```

3. To reset the EVM, enter:

```
dl -x
```

4. You may patch any labeled EVM memory location with an integer of fractional value. For example, to start up N1OWU's audio signal generator `TONE.ASM` for a given frequency and amplitude, run:

```
dl tone f=1200 rms=0.12
```

`DL` reads the EVM program's symbol table from the `*.lod` file to determine the actual memory locations to be patched. But keep in mind that there is no integer/fractional type checking. If you give a numeric value with a decimal point, it is assumed to be fractional. For example, if you run:

```
dl tone f=.234 rms=99
```

you will get strange results.

Entering a decimal number greater than +/- 1.000 will give an error message. If a value of exactly 1.0 is given, it is replaced by the largest possible fractional number (.99999something).

If the symbol name does not represent a X, Y, or P memory location, you will get an error message.

No spaces are allowed between the symbol name, the equal sign, and the numeric value.

If the given symbol cannot be located, a search is made for a G56K (C compiler) global symbol with the matching name (e.g. the C variable blah is labeled Fblah by the compiler).