Application Note
0380-0251-10 Rev 2.0

Embedded Performance, Inc.
606 Valley Way, Milpitas, CA 95035
Telephone: (408) 957-0350
FAX: (408) 957-0307
e-mail: support@epitools.com
web: www.epitools.com

Using A MAJIC® Probe With A Linux PC

0380-0251-10 Rev 2.0

January 12, 2005

Introduction

This application explains the steps necessary to configure and use a MAJIC® Intelligent Debug Probe with a PC running Linux. While the *MAJIC® User's Manual* provides complete information on the configuration process and the files that are involved, this application note is an addendum that provides specific information on:

- Installing the EPI Development Tools (EDT) software package on your Linux PC using the Linux Installer,
- Installing the EDT software package from a compressed tarball,
- Installing an Engineering Update Service Pack,
- Connecting to MAJIC® probe with the MONICE command line debugger,
- Connecting to MAJIC® probe with the GNU GDB debugger,
- Performing maintenance updates on the MAJIC® probe.

Note: This is not intended to be a Linux tutorial basic familiarity with the Linux environment is assumed. Nor is it a tutorial on GDB itself; it explains how to connect GDB to the MAJIC® probe, but assumes the reader is familiar with the GDB debug environment.

Additional Documentation

Additional documentation for the EDT software package and MAJIC® probe can be found in the `./manuals` directory of the EDT installation, and the `./manuals` directory on the EDT software package CD. The `edta_doc_index.html` (for ARM and XScale), or `edtm_doc_index.html` (for MIPS), provide easy access to all the on-line documentation. The following documents will prove particularly useful through the course of this application note:

MAJIC® User's Manual	Complete information on configuration and operation of the MAJIC® probe, and the MON command language.
MDI for MAJIC User's Guide	Details on the MDI interface to the MAJIC® probe, and the MDI configuration file.
gdb-readme.txt	Additional technical details on setting up and using GDB and MDI-Server with a MAJIC® probe.

Getting Support

Please do not hesitate to contact our technical support group if you have any questions or need assistance in configuring the MAJIC® probe for your system. We recognize that these issues are complex, and are committed to making sure our tools work well for you.

EPI Development Tools Installation

The EPI Development Tools (EDT) package is distributed on a CD included with the MAJIC® probe. Updates are generally distributed on our web site. Before installing the EDT package, it's a good idea to check the EPI web site to make sure you have the current release.

EPI has made Linux installs easier by providing a graphical based installer, `einstall`. This installer requires that the host be running the X Window System. If your Linux environment does not support the X Window System, or `einstall` is not able to utilize the installed X Window System, please contact EPI technical support for a compressed tarball which you can manually install by following the steps in the *Installing from a Compressed Tarball* section, below.

Using the Linux Installer

An installation key is required in order to install the EDT software package. If you do not have an installation key, use the **`[Get Key]`** button in the main `einstall` window to register the package and request the installation key, which will be sent to the email address you provide. After you receive the installation key, rerun `einstall`, copy the key from the email into the main `einstall` window, and proceed with the installation. If you can not run `einstall` this way, you can also try running it manually.

Installing from CD

If you are installing from a CD, place your CD into your CD-ROM drive. If your installation directory requires root access, then you will need to login as root first. For some Linux hosts, a desktop file browser will open and you can double-click `einstall` to launch the installer. Otherwise, you can run the script from a terminal. For example,

```
/mnt/cdrom/linux/einstall
```

Once you launch the installer, simply follow the directions on the dialog boxes.

Notes:

- If your Linux host does not auto-mount the CD-ROM Drive, please become the root user:

  ```
  su -
  ```

  and mount it with a command similar to:

  ```
  mount /dev/cdrom
  ```

  Before removing the CD, make sure to **`unmount`** it.

- For additional installer information, please refer to the `readme.txt` located in the same directory as `einstall`.

Installing Updates Downloaded from the EPI Web Site

1. Download the appropriate EDT software update file from our Web Site via the link below:

   http://www.epitools.com/support/epi_dev_tools_update.php

2. Untar and uncompress the downloaded file as follows:

   ```
   cd  download_path          // directory in which you saved the downloaded file
   tar -xzf edtXXXXX.tgz      // name of the downloaded file
   ```

   This creates these three files:

   ```
   edtXXXX_tgz.fc             // actual file name is based on release number
   einstall
   readme.txt
   ```

3. Open a file browser to click on the `einstall` executable, or run it from the command shell below. Once you launch the installer, simply follow the directions on the dialog boxes.

   /*download_path*/einstall

Notes:

- For additional installer information, please refer to the `readme.txt`.
- The default installation directory is `/opt/edt`*X*, where *X* is an "a" to denote ARM/XScale or "m" to denote MIPS. Normally, the directory `/opt` requires root access to create or update files and directories. So you can either login as root or change the install directory to any directory you have full access permissions for.
- If your Linux environment path variable does not include the current directory, then you will need to use `./` to run the executable. For example:

  ./einstall

Installing Manually

`einstall` will not run successfully on all Linux installations due to shared libraries issues. For this case, you can manually install the tools by following these steps.

1. Paste the registration template below into a new email, fill in the "?" fields and send it to register@epitools.com. Please set the email's subject heading to:

   Register: *S/N, company_name*

   `S/N` is the serial number of your MAJIC probe and `company_name` is the name of your company. If you do not have access to email then you can FAX the needed data to 408-957-0307 (U.S.A.).

```
--- Manual MAJIC Registration (Linux) ---
Company:    ?
First_Name: ?
Last Name:  ?
EMail:      ?
Address:    ?
State:      ?
Zip/Prov:   ?
Country:    ?
Software Release:   EDT 2.2a
Architecture Used:  ?
MAJIC Serial #(s):  ?
Host OS:            Linux ?
Target OS:          ?
Comments:
--------------------------------
```

2. A reply email or FAX will be sent with an installation key. Please allow one business day.
3. Upon receiving your installation key, you can complete the installation with the following commands. Make sure your EDT installation CD is mounted first (see page 2). In double quotes, replace `key` with the installation key. The installation directory defaults to `/opt/edta` or `/opt/edtm` based on the key, unless you pass in an install directory parameter during invocation.

   cd /mnt/cdrom/linux

   einstall_cmd *key [install_directory]*

Notes:

- For additional installer information, please refer to the `readme.txt`.
- If the directory that you intend to install to requires root access, then you will need to login as root first. Alternatively, you can change the install directory to any directory that you have full access permissions for.

Installing from a Compressed Tarball

If your Linux environment does not support the X Window System, or `einstall` was not able to utilize the installed X Window System, please contact EPI technical support for a compressed tarball and follow the manual installation instructions below.

1. To install the EDT software package, you must open a terminal and become the root user with the following command.

   ```
   su -
   ```

2. Create a new directory for the EPI Development tools, then change into the new directory. For ARM users we suggest installing the EDT package in `/opt/edta` and for MIPS users we suggest `/opt/edtm`. For the examples in this application note, we will use the ARM version. MIPS users should simply replace any references to `edta` with `edtm`. The commands for performing these actions will look similar to:

   ```
   mkdir /opt/edta/          /* OR, /opt/edtm/ for MIPS */
   cd /opt/edta
   ```

   Note: If you do not have a directory named `/opt` then you will have to **`mkdir /opt`** first.

3. Extract the EDT files from the gzipped tar file into the directory created above, substituting the path and file name below to reference the update file in the path where you saved it.

   ```
   tar –xzf /path/edtXXXX.tgz   /* using edta or edtm as appropriate */
   ```

   Note: Solaris users should replace **`linux`** with **`solaris`**.

4. Add the `/opt/edtX/bin` directory to your search path. This can be done by editing the proper configuration file, which is usually `.bash_profile`, found in your `/home/username` directory for Red Hat users. Alternatively, the `PATH` can be set for a single console session by running the commands below in that console window.

   ```
   PATH=$PATH:/opt/edtX/bin      /* using edta or edtm as appropriate */
   export PATH
   ```

   Note: When adding a path to the `PATH` variable remember to use a colon (`:`) to separate each additional path.

Installing an Engineering Service Pack

1. If you have not already done so, install the EDT software package as instructed above. Make a note of the root directory where you choose to install the tools.
2. Place the service pack gzipped tar file into the root directory of your EDT software installation and extract the service pack files from there. This is the directory which includes the `bin, ice, targets,` and several other sub-directories. The service pack files will update some of the installed files, and may also install some new files.

   Note: For instruction on how to extract a gzipped tar file, review step three in the *Installing from a Compressed Tarball* section above.
3. If the service pack includes a new version of MAJIC® firmware, make sure to update the firmware in your MAJIC® probe by following the steps in the *MAJIC® Firmware Updates* section, below.

EDT Directory Hierarchy

After successful installation of the EDT package, you will find the following subdirectories beneath the installation directory (`/opt/edtX`):

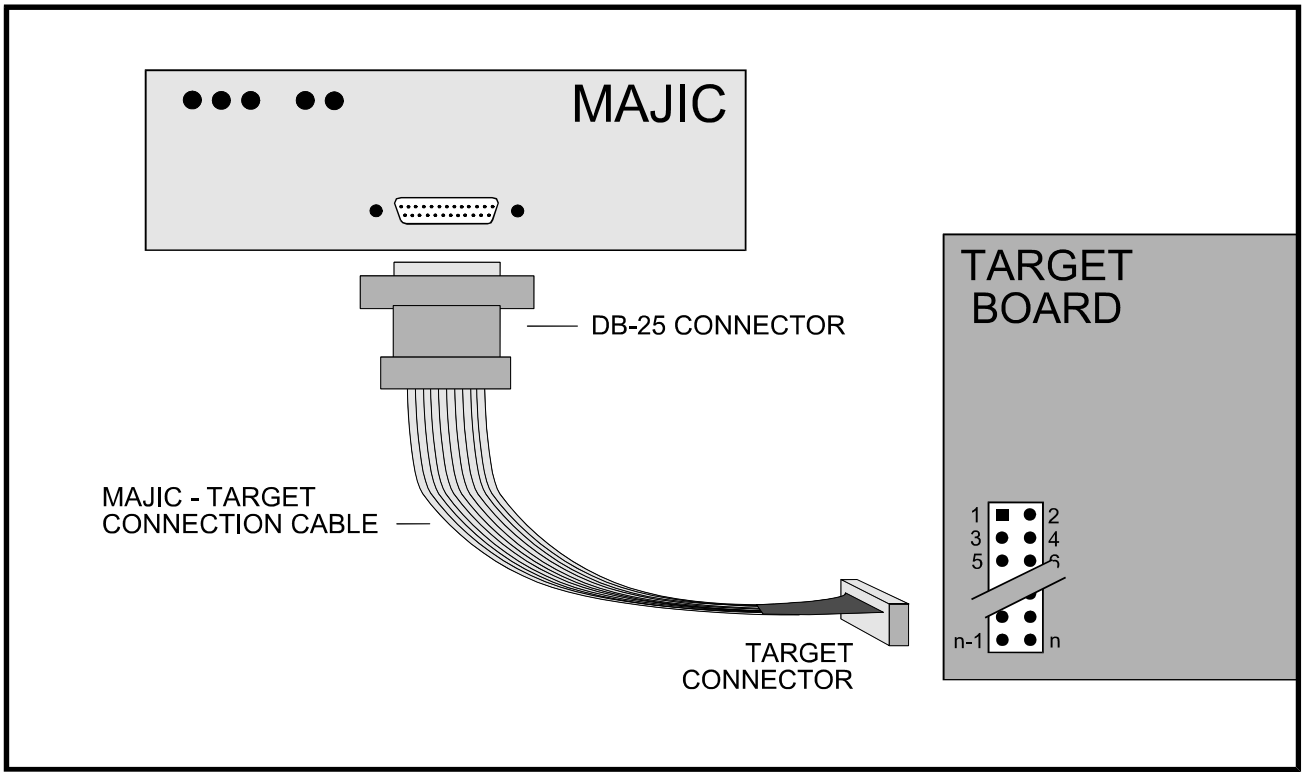
`./bin`	Executable programs and associated files.
`./ice`	MAJIC® firmware and MAJIC$^{PLUS®}$ trace control PLD files. New users can ignore this for now, but if you are installing an update, you should check the `MAJIC_RelNotes.html` file for the current MAJIC® firmware version number and revision information. Instructions for updating the MAJIC® firmware and MAJIC$^{PLUS®}$ trace control PLD are provided at the end of this application note.
`./manuals`	Installed documentation. The `edta_doc_index.html` (for ARM and XScale), or `edtm_doc_index.html` (for MIPS), provide easy access to all the installed documentation. The support page of the EPI web site may also have new or updated application notes.
`./mdi`	MDI back-end files for the MAJIC® probe.
`./samples`	Sample programs, including the flash programming utility. Source code is in `./samples`, and pre-built executables are in the `be/` (big endian) and `le/` (little endian) sub-directories. For ARM and XScale, lower level subdirectories are provided for different link addresses. For MIPS, the program executable file names end in `1` and `0` for kseg1 and kseg0, respectively.
`./targets`	Preconfigured start-up files for many common reference platforms. These can be used as is, if you have one of these platforms, and they also serve as a template for creating custom initialization files for your own target hardware.

Hardware Connections

Chapter 2 of the *MAJIC® User's Manual* provides full instructions for setting up the MAJIC® hardware, but the key points are summarized in the following sections as well.

JTAG Port

The MAJIC® probe is connected to the JTAG debug connector on your target board using an EPI Cable Kit. The details of each cable kit depend on the MAJIC® model, and the type of debug connector provided on your board. Each cable kit includes an application note with specific information on how to install it, but a typical example is shown below:



Notes:

1. **Be Careful!** Make sure you have the cable installed on the right connector, and in the right orientation. Remember that some boards may have more than one JTAG connector for different purposes—you want the one that is connected to the processor's debug interface.
2. There are many different "standard" JTAG connectors, and using the wrong one may damage the MAJIC® probe, target board, or both. If the cable does not fit the target board's debug connector, OR the pin-out of your debug connector does not match the *MAJIC® Interface Specifications* application note, please contact EPI technical support to see if there is a more suitable cable kit for your board.
3. Check the documentation for the board to see if there are any jumpers or switches that must be set in order to enable and use the JTAG debug interface.

Communication Ports

The MAJIC® probe may be connected to the host computer with a high speed RS-232C serial interface or through an Ethernet network connection. For serial connection, simply connect the serial cable provided from the RS-232 port on the rear panel of the MAJIC® probe to the serial port on your computer. Make a note of which serial port you connected to, as the debugger will need to know this.

In order to use the MAJIC® probe with an Ethernet connection, it must have an IP address. As described in Chapter 2 of the *MAJIC® User's Manual*, there are three ways for the MAJIC® probe to obtain its IP address. The easiest way is to program a static IP address into the MAJIC® probe using the serial port, so that it always comes up knowing its address. The procedure for programming the static IP address is provided below:

1. Assign (or obtain from your network administrator) a static IP address for the probe, and optionally a host name. The network administrator may want the Ethernet (MAC) address of the MAJIC® probe, which is available on the serial number tag on the bottom, and may also want to review Appendix A of the *MAJIC® User's Manual*.
2. Connect the serial cable provided from the RS-232 port on the rear panel of the MAJIC® probe to the serial port on your computer. Leave the JTAG cable disconnected for now.
3. Open a terminal window, and start MONICE to connect to the MAJIC® probe as shown below, replacing the port name command as appropriate. Several informational messages should scroll by when MONICE starts.

   ```
   monice -d /dev/ttyS0 -7
   ```

   Note: Remember that almost everything in Linux is case sensitive, so **`/dev/ttyS0`** is not the same as **`/dev/ttys0`**.
4. Set the `tv_ip_address` (`tia`) configuration option using the Enter Option (`eo`) command, filling in the blanks with the IP address for your MAJIC® probe.

   ```
   MON> eo tia = ___.___.___.___
   ```

   After a few seconds, MONICE will report that the IP address has been programmed.
5. Quit MONICE with the **`Q`** command, then cycle power on the MAJIC® box or press its reset button.
6. Ping the MAJIC® probe's IP address as shown below, filling in the blanks with the IP address you just programmed, to make sure the computer can communicate with the MAJIC® probe.

   ```
   ping ___.___.___.___
   ```

Notes:

- To remove a static IP address, follow the procedure above, and set the **`tia = 0.0.0.0`**.
- If your MAJIC® probe and computer are located on separate subnets, you may need to set the `tv_ip_netmask` (`tin`) and `tv_ip_gateway` (`tig`) options in step #4, above. See Appendix A of the *MAJIC® User's Manual* for additional information.

The Debug Environment

Chapter 3 of the *MAJIC® User's Manual* explains the configuration process and the files that are involved in detail. The following sections of this application note outline the basic steps necessary to configure the debug environment on Linux, first using MONICE, and then with GDB.

Configuration Files

Every target system is different, so the MAJIC® probe needs information about the target system design in order to operate correctly. The primary file, called `startice.cmd`, configures the JTAG interface and certain capabilities of the MAJIC® probe. In some cases `startice.cmd` will call in a board initialization file to initialize the target hardware, and declare the memory map of the target board.

1. Create a new, empty directory for staging the configuration files, then **`cd`** into that directory. For example:

```
mkdir  majic
cd  majic
```

2. The EDT software package includes pre-validated start-up files for many common reference platforms, in the `./targets` directory of your EDT installation. If you are using a standard reference platform, you can simply copy the right files into the MAJIC® configuration directory you created above. If your board is similar to a standard reference platform, you may need to adjust it to account for the hardware differences by editing it with your preferred text editor.

```
cp  /opt/edtX/targets/_______/*.cmd  .     /* using edta or edtm as appropriate */
```

   If your board is your own design, you will need to create suitable start-up files, using the files from the `./targets` directory for the board that is most similar to yours as a template.

Starting MONICE

There are two reasons to run MONICE first, even if you intend to use GDB normally. First, to establish a baseline operation with your EDT installation and MAJIC® configuration files. Second, to create the `epimdi.cfg` file required when using GDB with the MAJIC® probe.

To start MONICE, **`cd`** into the MAJIC® configuration directory that you created and populated above, then run **`monice`** with suitable **`-d`**, **`-v`**, and **`-l`** switches. For full information on the MONICE invocation line, please refer to Appendix C of the *MAJIC® User's Manual*.

Examples:

```
monice  -vh                                  // Show the supported CPU version (-v) switches
monice  -vARM926EJS  -d  /dev/ttyS0 -7 -l    // RS232 (-d ____), 115k (-7), little endian (-l)
monice  -v5Kc  -d  205.158.243.236:e         // Ethernet (-d ____:e), big endian (no -l)
```

Note: A space is required between the **`-d`** switch and the port name or IP address, but a space is prohibited in the **`-v`** switch. The little endian (**`-l`**) switch is a lower case L, not a number one.

When MONICE starts, it displays a number of messages with version and configuration information, then it displays a `MON>` prompt. At this point, you should be able to peek and poke both registers and memory on your target. You can even download, step through, and run a program. Chapter 4 of the *MAJIC® User's Manual* provides numerous examples of using the MON command language, and Chapter 5 details the syntax of each command and parameter type. A hierarchical help system is available with the **`H`** command.

Starting MDI-Server

GDB communicates to the MAJIC® probe through a separate program named MDI-Server. GDB sends "remote protocol" debug requests to MDI-Server which then sends corresponding debug requests to the MAJIC® probe. Therefore, the MDI-Server program must be running before GDB can connect.

MDI-Server, by way of the MDI shared library, is responsible for managing the details of the MAJIC® configuration. It uses the same `startice.cmd` and (possibly) board initialization command file as MONICE. However, CPU information and communication parameters are set via the `epimdi.cfg` configuration file instead of using command line switches. The easiest way to create `epimdi.cfg` is to start MONICE as described above, and then ask MONICE to create the file for you. The following command creates an `epimdi.cfg` file in your current working directory that matches the configuration in use by MONICE.

```
MON>  fw  mdi  epimdi.cfg
```

Note: The `epimdi.cfg` file created in this way supports just one specific MAJIC® probe and target. If you have more than one MAJIC® probe and/or target system, you can edit the `epimdi.cfg` file to define multiple "devices" and "controllers", and then pick the desired configuration for each debug session. Please refer to the *MDI User's Guide* for details on creating `epimdi.cfg` files that support multiple MAJIC® probes and/or targets, and *MDI-Server Options* (below) for information on choosing the active configuration.

Once you have your MAJIC® configuration files and `epimdi.cfg` file prepared, you are ready to start MDI-Server. You should always start MDI-Server from the directory containing the `epimdi.cfg` file. Following is the minimum command for launching MDI-Server and linking in the requisite MDI shared library:

```
mdi-server  -l  /opt/edtX/bin/mdi.so   /* using edta or edtm as appropriate */
```

Note: The load (**`-l`**) switch is a lower case L, not a number one.

MDI-Server Options

In some cases, additional start-up options beyond the minimum invocation line above may be desired or even required. Running MDI-Server with the **`-h`** switch shows the MDI-Server version number and usage prompt. Some of the commonly used options are explained below, and additional information is available in the `./manuals/gdb-readme.txt` file.

If you have created an `epimdi.cfg` file with multiple "DevName" definitions (see *MDI User's Guide* for details), then you may include a **`-d`** switch to identify which you intend to use. If your `epimdi.cfg` file has multiple "DevName" definitions but **`-d`** is omitted, then MDI-Server will display a list of configurations available and prompt the user to choose one each time GDB connects, which may cause some GDB versions to timeout.

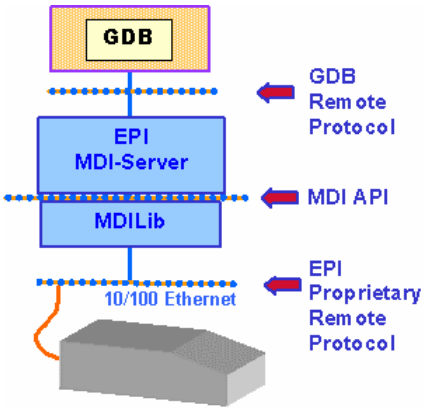
There are several options relating to how GDB represents the number of registers, their order, and size. ARM and XScale users can generally ignore this issue, but there are significant differences between various MIPS targets, so additional MDI-Server switches may be necessary to accommodate different GDB builds.

- **`-f`** Specifies whether a MIPS-targeted GDB is configured to expect floating point registers to be passed as single or double precision in the remote protocol.
- **`-n`** Specifies how many target registers GDB knows about.
- **`-r`** Specifies the register size passed in the remote protocol.

You may be able to determine the correct settings for these options from the GDB source code and build options. If you are using the GNU tools distributed by MIPS Technologies (configured as mipsisa32), the default values for **`-f`** and **`-n`** are correct but you need to specify **`-r 8`**. Otherwise, you will need to take several steps to figure out these options should be set for your GDB build. The procedure for this is provided in the `./manuals/gdb-readme.txt` file.

The GDB Debugger

GDB is a standard source-level debugger used under the Linux and Solaris Operating Systems. GDB communicates with the MAJIC® probe through an open Application Programming Interface (API) called the Meta Debug Interface (MDI). The MDI API is implemented through a program called MDI-Server, which works in conjunction with a shared library called MDILib. MDI-Server translates GDB "remote protocol" debug service requests into EPI proprietary remote protocol by making calls to the functions exported by the MDILib.



With this connection method you don't need to build a special GDB for the MAJIC® probe, but can use an off-the-shelf GDB (or any of the GUI front-ends available for GDB) and connect to MDI-Server with the "target remote" command. MDI-Server then reads the `epimdi.cfg` file created above, initializes the MAJIC® probe and target, and then begins servicing debug requests from GDB.

Building GDB

If you do not already have a cross-GDB built and installed you can create one by downloading the latest source for GDB available from `http://www.gnu.org/software/gdb/gdb.html` and building it using the appropriate `-target=xxxx` setting. For example the version of GDB used to validate these instructions was configured as

```
---target=arm-elf  --prefix=/opt
```

Running GDB

Before starting GDB, you must start the MDI-Server program as described above so that it will be ready to complete the interface between GDB and the MAJIC® probe. Once MDI-Server is ready, open a separate terminal window and start the GDB debugger version that is intended for cross development of your target processor (i.e. not the default Linux GDB).

Note: Not all GDB's are created equal. The commands and variables supported by a particular GDB depend on its version and on how it was configured when it was built. The commands and variables mentioned here are valid in GDB 4.18 (target xscale-elf) and 5.0 (target mipsisa32-elf).

From the `(gdb)` prompt enter the following commands to load your file and connect to the target and begin debug. Note that these commands can be placed in a GDB startup script so that they are run automatically at execution.

```
(gdb) set heur 0                          /* MIPS only—see note below */
(gdb) set remoteti 10                     /* Remote timeout—see note below */
(gdb) file yourfile                       /* Open your program file */
(gdb) target remote localhost:2345        /* Open MDI connection to the MAJIC® probe */
(gdb) load                                /* Download your program image */
```

Notes:

- GDB has an internal variable named `heuristic-fence-post`. It controls how far back from the current PC GDB will scan memory looking for a function start. This variable should default to 0, but it may not be in all GDB builds. If it is not 0, GDB may try to access invalid memory locations below the PC. This is primarily a concern for MIPS targets, where the initial PC value when connecting is usually 0xbfc00000 (the reset vector), and addresses in the 0xbfbxxxxx range are invalid.
- When GDB first connects, it can take a few seconds for MDI-Server to open the MDI connection to the MAJIC® probe and process the `startice.cmd` startup command file. Setting the internal GDB variable named `remotetimeout` prevents most GDB builds from timing out prematurely.

Now you are ready to set breakpoints and run or single step through your program. Users new to GDB may want to visit http://www.gnu.org/manual/gdb-4.17/html_chapter/gdb_toc.html for information on using GDB.

MON Commands

The GDB `monitor` command can be used to pass MON commands through to the MDILib, as shown in the examples below. This allows access to MAJIC® features that are not available in GDB, such as trace display, and access to all CPU registers (and user defined registers) rather than the limited set that GDB knows about. Appendix C of the *MAJIC® User's Manual* provides a table showing which MON commands are supported by MDILib, and Chapter 5 documents the MON command language in detail.

```
(gdb) mon di                    /* Pass  di  command to MDILib and display response */
(gdb) mon h                     /* Pass  h  command to MDILib to display help on MON commands */
(gdb) mon h h                   /* Display help on using MON help */
(gdb) mon h d                   /* Display help on MON Display commands */
(gdb) mon h dw                  /* Display help on MON Display Word command */
(gdb) mon fr c cmdfile          /* Read and execute a MON command script file */
```

Note: If you want MON commands to have access to symbols in your program, either add an **LS** (Load Symbols) command to the `startice.cmd` or just enter it directly with the **mon** command. If you download your program with the MON Load (**L**) command, as described below, then the **LS** command is superfluous.

Download Performance

By default, GDB generally downloads program code and data in small chunks (a few hundred bytes) that are not necessarily a multiple of four bytes in length. This causes program download times to be slower than necessary, especially with ARM targets. There are two GDB internal variables (whose names depend on the GDB version) that affect this. To improve GDB download performance, you should set the download write size to a binary value like 4096 or 8192, and the memory write packet size to a larger value to allow for packet overhead (+100 bytes is plenty). For example, to download 4KB at a time:

```
(gdb) set remote memory-write-packet-size fixed      /* Using GDB 5.x */
(gdb) set remote memory-write-packet-size 4200
(gdb) set download-write-size 4096

(gdb) set remotewritesize fixed                      /* Using GDB 4.x */
(gdb) set remotewritesize 4200
(gdb) set download-write-size 4096
```

An even better way to get the fastest possible program download speed is to tell MDI to do the download via MON's Load (**L**) command. The first example below downloads the whole program, the second example downloads everything except the .bss section, which is even faster, but requires that your boot code zero-fill the whole .bss section.

```
(gdb) mon l progfile            /* Download your program file via MDI */
(gdb) mon l -no b progfile      /* Download your program file via MDI, excluding .bss section */
```

Since this avoids the overhead of the GDB remote protocol, which is not very efficient even with the tweaks mentioned above, using the MON Load command is the recommended method for large programs.

MAJIC® Firmware Updates

The MAJIC® probe has internal firmware that can be updated by the customer. This is referred to as a "Firmware Update" and should be done whenever moving to a new EDT release, or when an EDT Service Pack is provided by EPI. It is a good idea to check the support page of the EPI web site periodically to check for updates and revision information. Service Pack releases are most often associated with new device additions to currently supported families, and occur between major EDT releases. Check with sales@epitools.com if you are interested in a device that is not yet shown on the EPI web site.

If you are updating to a new EDT Software release, install the new software as described earlier in this application note. The new MAJIC® firmware is located in the `./ice/majic` directory of your new installation, and revision information is provided in `MAJIC_RelNotes.html`. If you are performing a special firmware update from an EDT Service Pack, then the firmware update files will usually be in a separate sub-directory such as `./ice/majic.xxx`.

Firmware update

MONICE, the EPI command line debugger, is used to perform the firmware update. If you perform the update via the serial port, the update may take several minutes. With Ethernet is should take less than 15 seconds.

First **`cd`** into the directory containing the desired firmware update, then use MONICE to update the firmware with a command such as follows, substituting the communication port name or IP address as appropriate:

`monice -d` *`/dev/ttyS0`* **`-7 fwupdate.cmd`** // RS232 (**`-d`** ____), 115k (**`-7`**)

`monice -d` *`205.158.243.236`***`:e fwupdate.cmd`** // Ethernet (**`-d`** ____**`:e`**)

Notes:

- **It is important** to start MONICE in the directory containing the desired `fwupdate.cmd` and `majic.abs` files.
- MONICE is located in the `./bin` directory of your EDT installation, which should be in your search path. If not, then you can simply prepend the appropriate path to the MONICE program name.
- The CPU version displayed by MONICE is irrelevant, because the MAJIC® probe will not connect to the target during the update process.

When the update process begins, you should see basic connection information pass by on the screen, and then the following update information:

```
Reading commands from C:\Pkg\EPITools\edtm20b\ice\majic\startice.cmd
MON> /*-------------------------------------------------------------------*/
MON> /* startice.cmd:  startup command file for F/W update.               */
MON> /*-------------------------------------------------------------------*/
MON> //
MON> eo Ice_Power_Sense = Off   /* leave disconnected during F/W update */
MON> // <eof>
MON> (closing C:\Pkg\EPITools\edtm20b\ice\majic\startice.cmd)
Reading commands from C:\Pkg\EPITools\edtm20b\ice\majic\fwupdate.cmd
MON> +Q
  loading memory image file...

Validating download and programming EEPROMs
Flash update completed successfully
Please cycle emulator power so new firmware can take effect
```

Note: Check to make sure that the Status LED on the front panel has turned off, which is the indication that the firmware update was successful. Even if error messages are displayed by MONICE, an extinguished status light means that the firmware update was completed.

DO NOT cycle power while the status LED is RED! Once the status LED has gone out, cycle power on the MAJIC® probe so that the new firmware can take affect. This is very important, as the new firmware will not be run until you reboot the MAJIC® probe. After rebooting the MAJIC® probe, connect to your target with MONICE again to verify the new version number. Rerun MONICE as above, with only the connection (**-d**) switch. For example:

```
monice  -d  /dev/ttyS0  -7                    // RS232 (-d  ____), 115k (-7)
monice  -d  205.158.243.236:e                 // Ethernet (-d  ___:e)
```

When you connect to the MAJIC® probe via MONICE the system information will be displayed automatically, but if it scrolled off the screen, you can display it again with the **di** command. Look for the line that displays the firmware version number:

```
Target System:  EPI Majic Probe, Version: 3.4.5, S/N 0208G030...
```

MAJIC$^{PLUS®}$ Trace Control PLD

The MAJIC$^{PLUS®}$ probe can support both ARM Embedded Trace Macrocell (ETM) and MIPS EJTAG PCTrace. However, the Trace Control PLD must be programmed with the right version for the trace interface you are using. Normally each MAJIC$^{PLUS®}$ probe is preconfigured with the right PLD, based on which cable kit or active probe you ordered, but if you want to switch from one trace interface to another then you must reprogram the Trace Control PLD.

To check which PLD version is installed in your MAJIC$^{PLUS®}$ probe, run MONICE as described above. When MONICE starts, the system information will be displayed automatically, but if it scrolled off the screen, you can display it again with the **di** command. Look for the line that displays the hardware revision information (shown below). The last field in this line indicates which PLD image is presently installed.

```
Hardware Rev:   90:33:41:60
```

20	MIPS/PCTrace.
40	Original ARM/ETM version suitable for the CKP_ARM series of passive probes.
60	Current ARM/ETM version supporting both the CKP_ARM series of passive probes, and AP_ARM series of active probes.

Note: The hardware revision display was recently changed. The table shown above assumes you have installed MAJIC® firmware version 3.3.2 or later. If you are using firmware version 3.3.2 or later, and the fourth field is not shown in the table above, then please contact EPI technical support for assistance.

Hardware Update Procedure

When switching from one trace interface to another, you must reprogram the Trace Control PLD. This is accomplished by performing a hardware update procedure that is very similar to updating the MAJIC® firmware.

First, **cd** into the directory containing the hardware update files:

- For ARM/ETM, the files are located in the `./ice/majicplus_v6_etm.pld` directory of your EDTA software installation.
- For MIPS/PCTrace, the files are located in the `./ice/majicplus_v2_pctrace.pld` directory of your EDTM software installation.

Next, initiate the hardware update process by running MONICE as shown in the following examples, substituting the communication port name or IP address as appropriate:

```
monice -d /dev/ttyS0 -7 hwupdate.cmd          // RS232 (-d ___), 115k (-7)
monice -d 205.158.243.236:e hwupdate.cmd      // Ethernet (-d ___:e)
```

Notes:

- **It is important** to start MONICE in the directory containing the desired `hwupdate.cmd` and `*.xsv` files.
- MONICE is located in the `./bin` directory of your EDT installation, which should be in your search path. If not, then you can simply prepend the appropriate path to the MONICE program name.
- The CPU version displayed by MONICE is irrelevant, because the MAJIC® probe will not connect to the target during the update process.

When the update process begins, you should see basic connection information pass by on the screen, and then the following update information (ARM example):

```
MON> /*-------------------------------------------------------------------------*/
MON> /* startice.cmd:  startup command file for H/W update.                     */
MON> /*-------------------------------------------------------------------------*/
MON> //
MON> eo Ice_Power_Sense = Off   /* leave disconnected during H/W update */
MON> // <eof>
MON> (closing C:\Pkg\EPITools\edta20b\ice\majicplus_v6_etm.pld\startice.cmd)
Reading commands from C:\Pkg\EPITools\edta20b\ice\majicplus_v6_etm.pld\hwupdate.
cmd
MON> +Q
  loading memory image file...

Programming trace control PLD for ...
Please cycle MAJIC power so the update can take effect
```

Note: Check to make sure that the Status LED on the front panel has turned off, which is the indication that the update was successful. Even if error messages are displayed by MONICE, an extinguished status light means that the update was completed.

DO NOT cycle power while the status LED is RED! Once the status LED has gone out, cycle power on the MAJIC® probe so that the update can take affect. After rebooting the MAJIC® probe, rerun MONICE to verify the PLD version number, as described above.