

SIXNET IPm Application Development Kit (IADK)

Usage

The SIXNET IPm Application Development Kit (IADK) is an i386 cross PowerPC-Linux GCC toolchain. This lets you build programs for the IPm on a much faster and more powerful desktop.

The IADK has been compiled on both a standard i386 Red Hat 7.0 installation and a standard i386 Windows installation. It should be compatible with Red Hat 7.0 and higher. Red Hat 6.2 has an older version of glibc. If you have updated glibc in Red Hat 6.2 you may be able to use the IADK, but this is unsupported.

If you need to run the IADK on an unsupported version of Linux/Unix, the source files and build instructions can be provided upon request.

Using the IADK

Step 1:

If you have not set the PATH environment variable, see [iadc_install_guide.pdf](#) before attempting to compile. If '/opt/sixnet/bin' is missing from the path, set it now.

using bash: `export PATH=/opt/sixnet/bin:${PATH}`

using csh: `setenv PATH /opt/sixnet/bin:${PATH}`

Step 2:

Compile the program:

To compile a program that doesn't use SIXNET I/O libraries, just use 'powerpc-linux-' prepended to the normal gcc commands.

Example:

```
powerpc-linux-gcc file.c -o my_program      -- turns 'file.c' source code into 'my_program'
powerpc-linux-objdump -d my_program        -- dumps to the screen the assembly code for
                                           'my_program'
```

You can also use makefiles in this fashion. You can find tutorials for makefiles on the Internet.

If your program needs SIXNET I/O libraries:

1) The header file 'iodb.h' must be included. This file is provided in /opt/sixnet/powerpc-linux/include

Add the following statement to your program:

```
#include <iodb.h>
```

2) You must link against libsxio.a and libsxiodb.so. These libraries are provided in /opt/sixnet/powerpc-linux/lib

Example: to compile 'file.c' which needs SIXNET libraries, into 'my_program' which runs on the IPm:

```
powerpc-linux-gcc file.c -lsxiodb -lddio -lsxio -o my_program
```

Step 3:

Load the program (oem_iodb) to the IPm:

The compiled program can be loaded to the IPm station using either ftp or the SIXNET Tool Kit.

SIXNET IPm Application Development Kit (IADK)

Using FTP and Telnet:

Note: FTP and Telnet must first be enabled in the IPm station. There are “Enable FTP protocol” and “Enable Telnet protocol” checkbox selections in the “Advanced” view of the IPm Configuration window within the SIXNET I/O Tool Kit. You’ll need to check the two boxes and reload the configuration into your IPm station.

Now run your favorite FTP client (or just type “ftp” at the command prompt) and open a connection to the IPm (see example below). Login is “anonymous”. No password is necessary. (Hit <enter>.)

Example: if your IPm's IP address is 10.1.0.1, you would type:

```
ftp 10.1.0.1 [enter]
anonymous      [enter]
[enter]
```

Next, 'cd' to the existing location you want to place 'my_program' (typically /usr/local/bin) and use 'put' in binary mode to upload your file:

```
bin [enter]
cd /usr/local/bin [enter]
put my_program [enter]
quit [enter]
```

Optionally, you can use the mkdir command and create a folder to upload your file into:

```
mkdir my_folder [enter]
bin [enter]
cd /my_folder [enter]
put my_program [enter]
quit [enter]
```

Using the SIXNET I/O Tool Kit:

Copy the compiled program into a folder on your Windows computer. Run the SIXNET I/O Tool Kit and open your project file. Click in any of the fields for your IPm station. Select Operations → File Operations. In the File Operations window, select the location you want to place your program. Then click the Load File button. In the file selection window, locate the compiled program file and click the “Open” button.

Step 4:

Start your program (my_program) running in the IPm.

To start the program running, make a Telnet connection to the IPm. (Note: Telnet must be enabled in the IPm. See “Using FTP and Telnet” above.) The default login/password is 'root'. There is no password. Then 'cd' to the directory where you placed the program:

```
telnet 10.1.0.1 [enter]
root [enter]
cd /usr/local/bin [enter]
```

Note: You may need to set the execute flag for the file:

```
chmod +x my_program
```

To run the program, type:

```
./my_program
```

SIXNET IPm Application Development Kit (IADK)

Configuring the IPm station to start your program automatically

Your program, and the appropriate path to it, will need to be listed in a startup script file (.sh file) located in the /etc/user.d folder in the IPm station. Any startup script files found in the /etc/user.d folder will be launched automatically by the IPm station after powerup. For more information on the behavior of startup scripts, refer to the User Script Management technical note (user_script_management.pdf). The latest IPm technical notes are available for download from http://www.sixnetio.com/html_files/ipm_tech_notes/ipm_technotes.htm.

Here is an example of a startup script file:

```
#!/bin/sh
/usr/local/bin/my_program
```

Note: Your startup script file must be saved or resaved in such a manner that each line does not end with a carriage return. If you create this file using the text editor provided in the File Operations window of the SIXNET I/O Tool Kit (click the "Edit Text File" button), the file will be saved in the proper format. Otherwise, you can assure that the file is in the correct format by running Telnet and issuing the following commands on your file (called "myfile.sh" in this example):

```
cd /etc/user.d
dos2unix -u myfile.sh
chmod 755 myfile.sh
```

If you experience problems such as:

```
bash: ./my_program: Permission denied
Use 'chmod +x my_program' to make the file executable.
```

```
bash: ./my_program: cannot execute binary file
The file is probably compiled for a desktop, not the IPm. Make sure you didn't use 'gcc' instead of
'powerpc-linux-gcc'
```

```
many 'undefined reference to' messages when compiling.
Make sure you have -lsxiodb -lsxio after the .c file (file.c)
```