Compiling and External Libraries

It is recommended to develop the program on DEV machine. Therefore, you will need a cross-compiler and a linux machine.

Setting up cross-compiler

Check TS8160-4200 wiki page http://wiki.embeddedarm.com/wiki/TS-8160-4200#Cross_Compiling and read the section about cross-compiling. In short, since your dev machine and PacMan have different system architectures, the same compiler won't work. The toolchain you should have downloaded from the page has the same compiler that is used on PacMan. Once you have set up the tool chain, you will want to add that to the path so that you don't need to type the whole directory. In my case, I called it gcc-arm.

To compile my program, I just need to type

```
gcc-arm myprogram.c -o myprogram
```

Compiling and copying binary file

To make it easier to compile the program, I have provided a Makefile. Make changes to the Makefile as needed. Once you have compiled the object, you will want to transfer it to the PacMan. Assuming that you have setup the PacMan correctly and has sshed into the Pacm, first stop the pacman program by typing in the pacman

```
service pm14 stop
```

Then use ./copy.sh on dev machine to transfer the file to the pacman. The script uses scp to copy pm14 binary file to /root/PM14. Then start the program by

```
service pm14 start
```

External Library

The only external library used is libconfig. Information about libconfig can be found here. http://www.hyperrealm.com/libconfig/. The source file of the library is also available in the software backup.

You might find it difficult to install new libraries on Pacman, especially if you want to compile it on a DEV machine but wants to use PacMan library instead. Therefore, it is recommended to make it a static library.

That is why we have libconfig.a which is the static library for libconfig. I also have the header file in /root/PM14/lib folder. Then you just need to link the static library in the makefile.