RTAI Installation Help

There are quite a few references to help you install RTAI, including the *README.INSTALL* file in the RTAI directory. This is an attempt to give a quick how-to.

1. Download the RTAI that you want to use. You should choose the latest one that will work with a Linux kernel that you want to use. For example, RTAI-24.1.11.tgz.
2. If you are using a non-standard Linux, such as the type that come with Red Hat, SUSE or some other packaged distribution, you will probably have to install a different kernel. The packaged distributions have often been patched and don't work with RTAI. You can get a pristine kernel from kernel.org For the following, assume that we are using linux-2.4.20.
3. Place RTAI-24.1.11.tgz and linux-2.4.20.tar.bz2 in /usr/src.
4. Execute *tar –gzip -xf RTAI-24.1.11.*
5. Execute *tar –bz2 -xf linux-2.4.20.tar.bz2.*
6. Create a link with *ln -s RTAI-24.1.11 rtai.*
7. Delete the *linux-2.4* directory and recreate it with *ln -s linux-2.4.20 linux-2.4.*
8. Change directory to linux-2.4.
9. Copy the config file for you kernel with *cp /boot/config-2.4.??? ./.config.*
10. Execute *make oldconfig* to create a useable config file. You should be asked if you want to enable RTHAL capabilities; say *yes.*
11. Execute *make menuconfig.* In the *modules* configuration turn off the *version on each module* setting. Also, under general configuration, turn off APM power management.
12. Leave make menuconfig.
13. Execute *make dep.*
14. Execute *make.*
15. Execute *make modules.*
16. Execute *make modules_install.*
17. Execute *make install.*
18. Modify your grub.conf or lilo.conf for the new kernel which has been placed in /boot. It isn't wise to remove the old one.
19. Reboot and you should be running your new kernel

Now you need to build RTAI.

1. Go to your /usr/src/rtai directory.
2. Execute *make menuconfig*. You probably don't have to do anything here as the defaults are OK, but you have to do this to save the config file. When you close, be sure that you read the notices that are displayed.
3. Execute *make*.
4. Execute .
   setsched newlxrt.
5. Execute *make modules*
6. Execute *make modules_install*
7. Execute *make /dev*

Finally, you should be done. If you have a problem with make failing on the compile of *usposix*, compile the *watchdog* directory by hand and ignore *usposix* as it probably won't be important.