SunSPOT Install Notes

Install

1. Install the latest JDK
2. install ant (comes with Eclipse)
    - add the `C:\Programs\eclipse\plugins\org.apache.ant_1.7.0.v200803061910\bin` to the path *(version of ant may be different)*.
3. add to `PATH` env the path to the bin in the jdk (by default it will add the jre only)
    - this will make javac work on a prompt
4. add a `JAVA_HOME` variable with the dir of the jdk
    - this will make ant work on the prompt (needs the `tools.jar` from the jdk)
5. install the SpotManager
    - go to http://www.sunspotworld.com/spotmanager/
        - *remember to restart the browser so env variables take effect*

(the following steps are described in `InstallationInstructions.pdf` in the doc folder)

1. install the USB driver (this has been installed) as part of the SpotManager installation
    - plug the Base Station and go for automatic
        - it is recommended to start with the BS
    - plug the other Spots as well, as they are recognized differently
2. upgrade Spots
    - in a directory with a build.xml for the spots (the BounceDemo for example) do:
      ```
      ant upgrade
      ```
    - after that for the base station do
      ```
      ant selectbasestation
      ```

Serial port classes

They should be available from the installation, but may need to be added to path see #Errors.

IDE setup

See: forum posting

> Doing Sun SPOT development in Eclipse is fairly straightforward, and some members of the team use that IDE - we just don't have pre-packaged modules for it. Here's a rough sketch of setting up a SPOT project. First, copy an existing Sun SPOT app from the examples. Then do File/New Project in Eclipse, then select Java project. In the next dialog, give the project a name, click "Create project from existing source" and give it the path to your copied app. In the next dialog, click on the libraries tag, remove the JRE System library, and click "Add external Jars". Navigate to the lib subfolder of your SDK installation, and pick these jars:

> - `multihoplib_rt.jar`
> - `transducerlib_rt.jar`
> - `spotlib_common.jar`
> - `spotlib_device.jar`
> - `squawk_rt.jar`
> - `spotworld_ext.jar`

You should remove the JRE system libray. And change the settings for the Java compiler to match a *"Compliance Level"* of 1.4.

> Click "Finish" and you have a project. To attach source, first open the project's properties dialog. Select the Libraries tag, expand each jar, double click the source attachment, click "External file" and navigate to the matching source jar in the src folder within your SDK installation. `spotlib_source.jar` covers both the spotlib jars, and there is no source shipped for squawk_rt.jar. To attach javadoc, double click the javadoc location for each jar and use the browse button to go to the `doc/javadoc` folder inside your SDK installation. This location is good for all the jars. For a host app, the process is much the same, except the jar list is now
>
> - `multihoplib_rt.jar`
> - `spotlib_common.jar`
> - `spotlib_host.jar`
> - `squawk_classes.jar`
>
> and the javadoc location is `doc/hostjavadoc`. And for a host app, you should **not** remove the JRE System library.

If using the SpotClientCommands to issue commands to the spot, read its library, etc, you also need to include the:

- `spotclient_host.jar`
  - src: `src/spotclient_source.jar`
  - documentation: `doc/hostjavadoc/`

Sharing SunSpots (key issue)

From the SunSpot `developer's guide.pdf`

> If you want to share Sun SPOTs between two or more SDK installations or users, you have to ensure that the SDK installations and users share the same key-pair. To do this, start by installing each SDK as normal. Then, copy the key-pair from one "master" user to each of the others. You can do this by copying the file `sdk.key` from the `sunspotkeystore` sub-directory of the "master" user's home directory and replacing the corresponding file in each of the other user's `sunspotkeystore` directories.
> You then have to force the master's public key onto each of the Sun SPOTs associated

> with the other installations. The simplest way to do this is to re-deploy the application via USB
>
> ```
> ant deploy
> ```
>
> for each Sun SPOT.

Errors:

- no `rxtxSerial` in `java.library.path`

Add the `-Djava.library.path=C:\Develop\SunSpot\sdk\lib` to the VM args in the running configuration Or add the `LD_LIBRARY_PATH` to the running environment with the path to the lib in SunSpot

- `SERIAL_PORT` property must be set to access the basestation

Add the `-DSERIAL_PORT=COM7` to the VM args in the running config

Windows 64 bits

- need to install Java 64 bits
- eclipse is also available in 64 bits
  - go to downloads and follow the latest version, choosing 64 bits version
- there is a RxTx java 64 bit version from CloudHopper
- when connecting a SunSpot some problems may occur with the drivers
  - see the forum's thread

Fedora 64 bits

Some more things are needed for Fedora 64 bits (libs for 32 in 64 system), but two main points troubled our deployment (the 1st is more or less known but it has similar solution to the 2nd):

- Permissions for the `/dev/ttyACM*`: either set the permissions of `/dev/ttyACM*` to 777 or add the developer/java compiler user to the `dialout` group. The `/dev/ttyACM*` should already have rw permissions for that group's members.
- Permissions errors for `/var/lock`: same as the above, but this time for the group `lock`. The `/var/lock` should also be `chgrp`ed to `lock`.
- ant build complains about missing `hal-device`: fedora has removed HAL from the distribution. The build process can live without it, but the shared basestation and the usage of the base station was not possible until we installed the hal rpm from alternative source.

Last Update: 2014/02/12 (GMT)