
Installation of ScatterWeb Sensorboards under LINUX (Fedora Core 3)

Note: Some parts (Ports, commands, ...) may differ using other Linux distributions.

In order to install and configure the ScatterWeb development environment do the following steps:

1. Unpack the archiv. The Folder ScatterWeb/ with its subfolders firmware/, apps/, and mspg_toolchain arises.

```
tar xvzf ScatterWeb.tgz
```

2. After downloading and unpacking the ScatterWeb.tgz file change to root in your terminal (Currently, the installation is only executable as root).

```
su
```

3. Change to the directory with the development tools.

```
cd ScatterWeb/mspg_toolchain
```

4. The compiler gcc, some helper tools and the most important libraries for the MSP430 microcontroller platform are translated. After installaiton, all tools and libraries are in the directory /usr/local/msp430/. If you want to change that path, give your own path to the install.sh script as an argument (Note: you will have to set the environment variables by yourself if you change the path).

```
./install.sh
```

4. The rest of the installation can be done with your normal account.

```
exit root and cd ..
```

5. In order to find the tools and libraries by the shell (bash) the path to to the development tools has to be added to the standard path. Additonally, the libraries for the MSP430 must be found by the cross-linker. Therefore, the LD_LIBRARY_PATH must be updated. The setPathes.sh script adds both commands to the .bashrc file in your home directory.

```
./setPathes.sh
```

6. This script starts a proxy (do it in a seperate console, as the proxy must run during the whole development work). The proxy sets up a connection between the parallel port and the TCP/IP network. The

debugger gdb is connected over the internal network with the parallel port. Thereafter, you can flash your programms to the microcontroller and debug them on the fly on the microcontroller. Important: The user must have access to the parallel interface (e.g. /dev/parport0). Add your login name to the group "lp" in /etc/group (to be done as root).

```
./mspProxy.sh
```

7. Change to the directory from the archiv above.

```
cd apps/esb/
```

8. With make the files in firmware/src/ as well as in appa/esb/src are translated and linked. The object and binary files are then in the apps/esb/out/ directory. The .hex file is the file which is exectued on the microprocessor.

```
make
```

9. Load the executable hex file on the microcontroller. After a reset of the sensorboard, the program is running on the node.

```
make flash
```

10. In order to communicate with the sensor open a new console and start minicom with (add you login name to the group uucp in /etc/group, thus you should get access to /dev/ttyS0)
minicom -s
Chose "Serial port setup":
- Set the serial device to /dev/ttyS0
- Set the Bps/Par/Bits to 115200 8N1
- Set Hardware Flow Control to Yes and Software Flow Control to No
- Exit Serial port setup and safe the setup as "dfl".
When running minicom you may chose "CTRL-A e" in order to enable local echo. Thus you can see what you enter in the terminal. Now, if connected to the microprocessor you should be able communicate with it.

10. Communicate
There are a lot of commands available. Some off them are listed below:

```
rid                 //read the id of the sensor
sid 0019            //set the sensors id to 19
RST                 //Reset the sensor
sl{r|y|g} {1|0}     //Enable/disable red, green or yellow led
png x               //ping sensor x, Pong received if x is connected
esr                 //enable sensor readings
dsr                 //disable sensor readings
saf 32              //Set announce flags (e.g. 32 -> serial and radio)
raf                 //Read announce flags
```