Lab

Exercise 02 „Platforms for Embedded Systems“ Prof. Cochlovius

- Installing and testing the Eclipse IDE -

Goal:
During this exercise you will step-by-step establish an Eclipse-based integrated development environment (IDE) to be used for upcoming exercises.

Note: only for explanation – Do NOT execute on VMimage provided

Exercise 2.1: Installing the „correct“ version of Eclipse
If already installed, deinstall the default version of Eclipse using:

```
sudo apt-get remove eclipse
```

Then install the IDE for C/C++ Developers – specifically the Luna version SR2. Additional information (also valid for Luna) may be found here: *http://akovid.blogspot.de/2012/08/installing-eclipse-juno-42-in-ubuntu.html*

Note: you may copy the binary

```
eclipse-cpp-luna-SR2-linux-gtk-x86_64.tar.gz
```

from here http://webuser.hs-furtwangen.de/~coe/LabPMS/Res/Fortbildung into your download folder ***/home/ubuntu/Downloads***

Unzip / unpack it using ***tar*** *and then move it in the* ***/opt****-folder with*

```
sudo mv ./eclipse /opt
```

Then install the standard Java Runtime using

```
sudo apt-get install default-jre
```

Exercise 2.2: Fire-up Eclipse
Using **/opt/eclipse/eclipse,** we are able to start the IDE, and then we jump into the „workbench“

Note:
For more convenient startup of Eclipse in the future, you can create a softlink in the folder **/usr/local/bin** using

```
sudo ln –s /opt/eclipse/eclipse
```

This makes sure that the binary is part of your regular search path.

In addition, you can create an Eclipse icon for your desktop (see web-link mentioned above)

Exercise 2.3: Hello-World on the Host (Run and Debug)

Now create in Eclipse under New→ Project → C/C++ Project a Managed-Make project with a pre-fabricated hello-world example. During the definition of the project you select both toolchains Linux GCC and(!) Cross GCC. To avoid clutter, make sure that only the two debug configurations are checked.

Now activate the local platform-configuration (Linux GCC), build the software and execute the resulting program using a local Run-Configuration.

Now extend your hello_world program with a for-loop to write integer values from 0 to 99 to your console. Now Debug your program by setting a breakpoint in the loop. Monitor the value of the loop-variable. Change its value directly during a debugging session. Now modify the breakpoint by setting an additional ignore-count. Then modify the breakpoint by setting a logical condition to gather more experience with the debugger.

Note:
Within the scope of this exercise, we execute the program only on the host.