EPOS Code style guidelines

Table of contents

File names

- All lowercase with letters, numbers and underscore
- Header prefix: .h
- Source prefix: .cc
- Asm prefix: .s

Naming convention

- Member variables prefixed with one underscore: int _index;
- Access method has the name of the modified variable: int index();
- Modifier method: void index(int i);
- All methods and variables are lowercase, underlines are allowed
- Class names have all words capitalized and are separated by underscores: class NIC_Common;
- Boolean constants are lowercase: like Traits<NIC>::debugged
- Numeric constants are uppercase: like Traits<IP>::ADDRESS

Indentation

- 4 spaces, avoid tabs (configure your editor to insert spaces istead of tab)
- Visibility keywords are not indented (private, public and protected on the same level of class keyword)
- Method signature in one line unless parameter list is too long
- Set/get methods can be one-liners if they are small

There is a Eclipse style on doc/epos-eclipse-style.xml
If you use another editor you can follow the rules bellow to avoid space/tab mix:

- For VIM:

```
set expandtab
set shiftwidth=4
set tabstop=4
set smarttab
```

- EMACS:

```
(setq-default indent-tabs-mode nil)
(setq-default tab-width 4)
```

```
(setq indent-line-function 'insert-tab)
```

- KATE:

```
Settings -> Configure Kate -> Editor -> Editing Settings
+ Insert spaces instead of tab
+ Tab width 4 spaces
```

- GEDIT:

```
Edit -> Preferences -> Editor
+ Insert spaces instead of tab
+ Tab size: 4 spaces
```

- Nano (~/.nanorc):

```
set tabsize 4
set tabtospaces
```

Assembly guidelines

- Labels at 0
- Instructions tabbed at 8
- First operand tabbed at minimum width to line-up things
- Remaining operands spaced
- Remarks tabbed at minimum width multiple of 8 to line-up things
- Macros and assembler directives as instructions

SVN development

Branches

EPOS' SVN repository is structured as follows:

- **epos1** : Legacy EPOS code
- **epos2**: Current, up-to-date EPOS
  - **branches**: Development sub-tree
    - **merge**: Feature additions to be merged back to trunk. Merge branches start as copies from trunk at the first Monday of the month and are merged back at most at the first Monday of the next month. Development must be fully documented via Trac tickets. Any tickets not solved in one month shall be given special attention by LISHA's development team.
    - **rc**: Project-related trees.
    - **usr**: Miscellaneous branches from users. These are sole responsibility of their owners, and become obsolete once the owner leaves the projetc.
  - **tags**: Read-only release history
  - **trunk**: Current, working, polished EPOS code
- **makers**: Projects that don't necessarily consist of EPOS code, but that are closely related to EPOS
- **teaching**: Didactic EPOS versions

Before you commit checklist

Follow these steps before performing a commit to EPOS' SVN repository:

1. Update your working copy and resolve any eventual conflicts. (It might be a good idea to backup your working copy before updating).
2. Make sure your code complies to the **code style**, **compiles, works** and **doesn't break anything else.** If any of these is false, it must not be commited yet.

3. Remove unnecessary comments and initial debugging information from your code.
4. Revert the default traits (unless you have a good reason for changing it).
5. Run "make veryclean" to revert files like include/config.h.
6. Analyse all files you changed, to make sure nothing unnecessary is being commited.

For example:

⬚⬚⬚⬚⬚⬚⬚

```
$ cd <root_of_your_working_copy> $ cp -r . ../working_copy_backup $ svn up $ svn revert
include/system/traits.h $ svn revert include/machine/cortex_m/emote3_traits.h # (Replace with the
machine/model you are using) $ make veryclean $ svn st # Analyse carefully which files will be
commited. Revert any unnecessary ones $ # Be sure your code still compiles and works at this point $
svn commit # Write an informative commit log $ rm -rf ../working_copy_backup
```

Eclipse configuration

Eclipse is the platform recommended to develop EPOS. A common platform allows for a more uniform
development, better documentation, and ultimately a smoother collaboration process.
This howto is based on (but not restricted to) Eclipse Neon 4.6 64 bits on a Ubuntu 14.04 machine.

Install the required Java version

For Eclipse Neon, you need Java JRE 8. You may download from Oracle's website and install. On Ubuntu
14.04, you may alternatively install it from a PPA (info taken from here):

⬚⬚⬚⬚⬚⬚⬚

```
$ sudo add-apt-repository ppa:webupd8team/java $ sudo apt-get update $ sudo apt-get install oracle-
java8-installer $ sudo apt-get install oracle-java8-set-default $ java -version
```

Install Eclipse

Download and install the latest Eclipse C/C++ version from the official download page.

Install Subversive

Subversive is a popular SVN plugin for Eclipse. You can download it directly from Eclipse itself: just click
Help -> Eclipse Marketplace and look for Subversive.

Create an EPOS project

Select File -> New -> Makefile Project with Existing Code, then select your EPOS tree.

Import EPOS' coding style

Select Window -> Preferences -> C/C++ -> Code Style -> Formatter -> Import... and select the file
doc/epos-eclipse-style.xml from any EPOS tree.