Software Development for the e-Puck Educational Robot : Coders Guidelines

Julien Hubert

April 18, 2006

1 Introduction

This document is intended for developers willing to create new software for the e-Puck or modify existing ones. It explains guidelines to ensure the correct assimilation of the new code inside the existing library.

2 Conventions

The following descriptions are to be followed during the entire project.

Naming The program, either the software or the documentation, should be in English. With the exceptions of the constants and the new data types or structure, all letters should be lower-case and the words should be separated with a "_".

Private Functions *read_ad*. Those functions are not intended to be used by developers. All the letters should be lower-case. All the words should be separated by a "_".

Global Functions *e_read_ad*. Global functions are to be used by developers. They follow the same convention as for private functions but there a "e_" added in front.

Global Variable *e_global_variable*. Global variables can be used by functions external to the libraries. All the letters should be lower-case and the first letters should be "e_" and all the words should be separated by a "_".

Local Variable *local_variable*. Local variables are variables declared inside functions or declared as static for every function of a module. They follow the same convention as for global variables but there is no "e_" in front.

Constants/Define *PI_CONSTANT*. All the letters should be capital letters. All the words should be separated by a "_".

Data Types or Structures *NewDataStructure*. Every word of a new datatype or structure should start with a capital letter and should not be separated.

3 Debugging

The debugging code should be encapsulated inside define statements.

4 Documentation

All documentation should be done using Doxygen. Divide each part of the library into subsections (sound, camera, motors,....)

5 SVN

The svn has three branches :

- Stable which contains a debugged version of the software. It will generally be the oldest but the more reliable of the branches.
- Unstable which contains a fairly well debugged version of the software. It should be quite recent but there are no guaranties that it has been totally debugged.
- Experimental which contains the latest development of the software. This branches contains developers version of the program and is not intended to be used by others than internal developers of the library.

All the uploads should be done in the experimental branches. When developers feel that the code is reliable enough, it can be copied in the unstable branch. When the unstable has not been changed for a few month, the software will be moved to the stable branch. All the bug should be corrected in the experimental branches and then follow the usual schedule up to the other branches, with the exception of critical bugs.

Only the following files are authorized to be uploaded :

- Headers (.h .inc)
- Source (.c .s)
- Compiled libraries (.a)
- Programs (.hex)
- Makefiles (*.in *.am Makefile Doxygen) but only when needed
- Project files (*.mcp) but modified to have only relative path for the directories and the files

No other file extension is allowed.