Micromouse: Designing an Educational Racing-Robot from Scratch
Report

Natalia Poliakova,[1] Autor 2,[2], Autor 3[3], Autor 4, [4] Autor 5, [5] Autor 6 [6]

[1]Department of Informatics, Technical University of Munich
An Unknown Address, Wherever, ST 00000, USA
[2]Another Unknown Address, Palookaville, ST 99999, USA

Our abstract - short overview of the whole report contents? Probably should be left untouched till we finish with the main report body.

1 Introduction

1.1 Micromouse competition

Short description of the competition, maybe a picture?

Fig. 1

1.2 Aims and Objectives

Our general goal in this praktikum (how is it called in proper English btw?..) Basically rephrasing the general Micromouse competition objective but maybe a bit more relaxed. Like ”to present a working robot that can move itself and perceive the environment with the help of sensors”

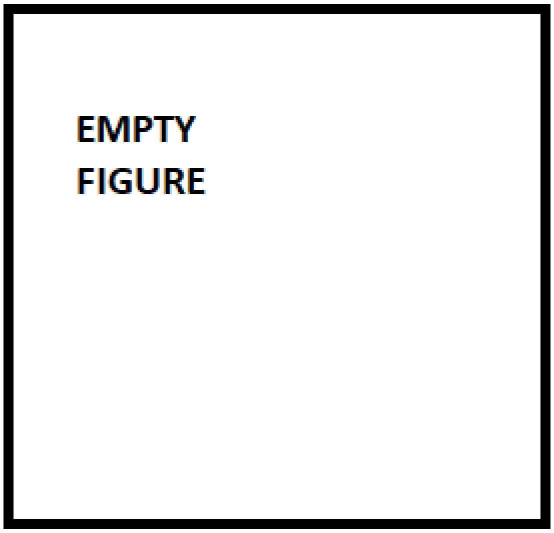

Figure 1: Caption

2 Conceptual design and justification of the design

2.1 Initial design conditions

Here we talk about the size of the labyrinth, explain the size of the mouse (maybe touch on our initial concept of the mouse design or something similar? - later we could extend this topic and motivate our casing design decisions, obviously mentioning the Fusion workshop, but this comes later) Also here should come some innate design ”choices” like using a microcontroller, a number of sensors, why we need those e.t.c Basically the whole logic behind ”how do we build a robot that should drive in the labyrinth and be able to make intelligent decisions (turns) based on the observations (made by sensors) Maybe we could split this section into two subsections:
- given conditions
- and our design decisions based on those conditions (justification)

2.2 Work program and Gantt chart

This is where I'm a bit lost. We could include some ”ideal” version of this chart here, but where exactly should we describe all the changes in planning and organization and why they had to be done at each stage of building the prototype and the final version of the mouse? Should it be described here? Or later in the ”problems and challenges”? Also I think maybe here we should talk about all the initial learning stage we had to go through in the first half of the praktikum (maybe devote a subsection for this here or later in the report)

3 Hardware and software design

Some introductory info

3.1 Hardware design

3.1.1 Board design

The complete description of our schematics design and board design and what we did and why Also including all calculations we've managed to collect throughout the whole course of said design. Pretty pictures and some tables with calculated values would be amazing.

3.1.2 Casing design

The whole journey on the casing design. Pictures of the final 3D model and printed model of course, some calculations and data on the sizes and maybe geometrical values

3.2 Software design

The description of our modular architecture, the work principles of the separate modules and basically ”how we control the peripherals” such as motors, leds, timers, etc. Logic comes a bit later in here. Also all info on DMA and pin remapping and our pretty mapping table are also welcomed here I think

4 Controller Design and Approach

Here goes the whole logic of the mouse movement control (how should it behave when is faced with the wall or on contrary - with the gap in the labyrinth). PID controller design and the logic behind it.

5 Summary of tests

Here goes everything practical we could possibly test - for example the speeding curve of the motor, the temperature conditions of the board (when we'll solder it - whether it is able to work without overheating and such) Of course, the PID working values (such as the error convergence rate) Possible subsections:

- casing tests
- schematic tests
- software logic tests

6 Implementation challenges

Some general words about our journey with the mouse.

6.1 Summary of problems encountered

Well, here go all the problems we faced coupled with our solutions for them. Potentially the longest part in the report. Can be split in parts similarly to the previous sections (talking about software, board and casing)

6.2 Learning experience

What we learned basically? What was especially hard to grasp, e.t.c. We could somehow present it as the short submission from each of us (as well as the previous part) Here we could also put the story of the first half of the semester (at least in the summarized form) and what concepts and practices and stuff we learned from that.

7 Conclusions

Here - short summary of the achieved results, maybe some general words and praises for our final mouse version, just something positive to end the mouse story well.

7.1 Expectations

Slightly controversial part, we could omit it (I would like not to though) or rephrase it somehow. Basically - what did we expect from the course. Not sure if this part actually belongs here, but for now it works.

7.2 Propositions

Our suggestions to maybe somehow improve or better organize some parts of the praktikum or the task or whatever.

Appendix

some extra figures

References

1. One of the equation editors we use, Equation Magic (MicroPress Inc., Forest Hills, NY; http://www.micropress-inc.com/), interprets native TEX source code and generates an equation as an OLE picture object that can then be cut and pasted directly into Word. This editor, however, does not handle LATEX environments (such as {`array`} or {`eqnarray`}); it can interpret only TEX codes. Thus, when there's a choice, we ask that you avoid these LATEX calls in displayed math — for example, that you use the TEX `\matrix` command for ordinary matrices, rather than the LATEX {`array`} environment.

Some guidelines from the template

1. Please follow the style for references outlined at our author help site and embodied in recent issues of *Science*. Each citation number should refer to a single reference; please do not concatenate several references under a single number.

2. Please cite your references and notes in text *only* using the standard LaTeX `\cite` command, not another command driven by outside macros.

3. Please separate multiple citations within a single `\cite` command using commas only; there should be *no space* between reference keynames. That is, if you are citing two papers whose bibliography keys are `keyname1` and `keyname2`, the in-text cite should read `\cite{keyname1,keyname2}`, *not* `\cite{keyname1, keyname2}`.

Failure to follow these guidelines could lead to the omission of the references in an accepted paper when the source file is translated to Word via HTML.

Handling Math, Tables, and Figures

Following are a few things to keep in mind in coding equations, tables, and figures for submission to *Science*.

In-line math. The utility that we use for converting from LaTeX to HTML handles in-line math relatively well. It is best to avoid using built-up fractions in in-line equations, and going for the more boring "slash" presentation whenever possible — that is, for `$a/b$` (which comes out as a/b) rather than `$\frac{a}{b}$` (which compiles as $\frac{a}{b}$). Likewise, HTML isn't tooled to handle certain overaccented special characters in-line; for $\hat{\alpha}$ (coded `$\hat{\alpha}$`), for example, the HTML translation code will return [^(α)]. Don't drive yourself crazy — but if it's possible to avoid such constructs, please do so. Please do not code arrays or matrices as in-line math; display them instead. And please keep your coding as TeX-y as possible — avoid using specialized math macro packages like `amstex.sty`.

Displayed math. Our HTML converter sets up TeX displayed equations using nested HTML tables. That works well for an HTML presentation, but Word chokes when it comes across a nested table in an HTML file. We surmount that problem by simply cutting the displayed equations out of the HTML before it's imported into Word, and then replacing them in the Word document using either images or equations generated by a Word equation editor. Strictly speaking, this procedure doesn't bear on how you should prepare your manuscript — although, for reasons best consigned to a note (*1*), we'd prefer that you use native TeX commands within displayed-math environments, rather than LaTeX sub-environments.

Tables. The HTML converter that we use seems to handle reasonably well simple tables generated using the LaTeX {`tabular`} environment. For very complicated tables, you may want to consider generating them in a word processing program and including them as a separate file.

Figures. Figure callouts within the text should not be in the form of LaTeX references, but should simply be typed in — that is, `(Fig. 1)` rather than `\ref{fig1}`. For the figures themselves, treatment can differ depending on whether the manuscript is an initial submission or a final revision for acceptance and publication. For an initial submission and review copy, you can use the LaTeX `{figure}` environment and the `\includegraphics` command to include your PostScript figures at the end of the compiled PostScript file. For the final revision, however, the `{figure}` environment should *not* be used; instead, the figure captions themselves should be typed in as regular text at the end of the source file (an example is included here), and the figures should be uploaded separately according to the Art Department's instructions.

What to Send In

What you should send to *Science* will depend on the stage your manuscript is in:

- **Important:** If you're sending in the initial submission of your manuscript (that is, the copy for evaluation and peer review), please send in *only* a PostScript or PDF version of the compiled file (including figures). Please do not send in the TeX source, `.sty`, `.bbl`, or other associated files with your initial submission. (For more information, please see the instructions at our Web submission site, http://www.submit2science.org/ .)

- When the time comes for you to send in your revised final manuscript (i.e., after peer review), we require that you include all source files and generated files in your upload. Thus, if the name of your main source document is `ltxfile.tex`, you need to include:

  - `ltxfile.tex`.
  - `ltxfile.aux`, the auxilliary file generated by the compilation.
  - A PostScript file (compiled using `dvips` or some other driver) of the `.dvi` file generated from `ltxfile.tex`, or a PDF file distilled from that PostScript. You do not need to include the actual `.dvi` file in your upload.

- From BIBTEX users, your bibliography (`.bib`) file, *and* the generated file `ltxfile.bbl` created when you run BIBTEX.

- Any additional `.sty` and `.bst` files called by the source code (though, for reasons noted earlier, we *strongly* discourage the use of such files beyond those mentioned in this document).

References

1. One of the equation editors we use, Equation Magic (MicroPress Inc., Forest Hills, NY; http://www.micropress-inc.com/), interprets native TEX source code and generates an equation as an OLE picture object that can then be cut and pasted directly into Word. This editor, however, does not handle LATEX environments (such as {`array`} or {`eqnarray`}); it can interpret only TEX codes. Thus, when there's a choice, we ask that you avoid these LATEX calls in displayed math — for example, that you use the TEX `\matrix` command for ordinary matrices, rather than the LATEX {`array`} environment.

Fig. 1. Please do not use figure environments to set up your figures in the final (post-peer-review) draft, do not include graphics in your source code, and do not cite figures in the text using LaTeX `\ref` commands. Instead, simply refer to the figure numbers in the text per *Science* style, and include the list of captions at the end of the document, coded as ordinary paragraphs as shown in the `scifile.tex` template file. Your actual figure files should be submitted separately.