Joshua Waugh

Title of the Thesis

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor’s Thesis

Date

|  |  |  |
| --- | --- | --- |
| Author  Title  Number of Pages  Date | First name Last name  Title of the Thesis  xx pages + x appendices  21 August 2017 | |
| Degree | Bachelor of Engineering | |
| Degree Programme | Information Technology | |
| Professional Major | Smart Systems | |
| Instructors | First name Last name, Title (for example: Project Manager)  First name Last name, Title (for example: Principal Lecturer) | |
|  | | |
| Keywords | |  |

Contents

List of Abbreviations

[1 Introduction 2](#_Toc491951623)

[2 Chapter Heading 2](#_Toc491951624)

[2.1 Subheading 2](#_Toc491951625)

[2.2 Subheading 2](#_Toc491951626)

[2.2.1 Subheading 2](#_Toc491951627)

[2.2.2 Subheading 2](#_Toc491951628)

[References 2](#_Toc491951629)

Appendices

Appendix 1. Title of the Appendix

Appendix 2. Title of the Appendix

List of Abbreviations

ORM Object-relational mapping. The set of rules for mapping objects in a programming language to records in a relational database, and vice versa.

DBMS Database management system. Software for maintaining, querying and updating data and metadata in a database.

PSoC Programmable System-On-Chip

LED Light Emitting Diode

# Introduction

This report is focused on the project of the first year Smart Systems Course for Information Technology students at Metropolia University or Applied Science. The project involved programming a robot to complete three tasks: sumo wrestling, line following and maze navigation. Each member of the group working on this task has no experience with programming prior to this project. As such, the task was approached largely as a learning exercise.

The project was approached by finding solutions to smaller side projects which would be transferrable to the main task. When discussing the final tasks in this report, it will be as the sum of the solutions to these tasks.

# Materials and Software

### PSoC® Creator™ Integrated Design Environment (IDE)

PSoC Creator is a program created by Cypress Semiconductor for writing programs for and compiling to PSoC hardware. This software was used for all code written during this project.

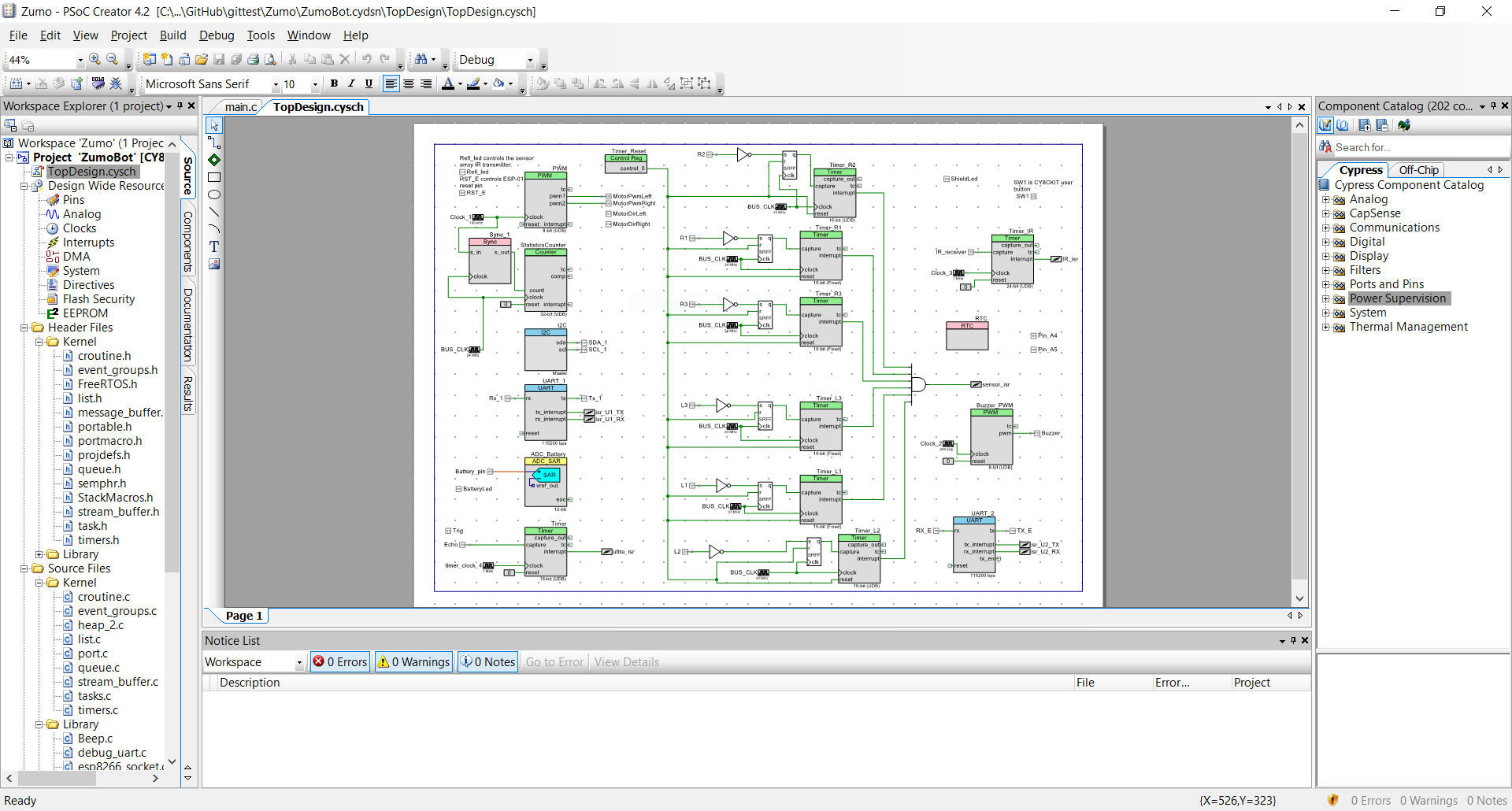


Figure 1

## CY8CKIT-059 PSoC® 5LP Prototyping Kit

The CY8CKIT-059 PSoC 5LP prototyping kit from Cypress Semiconductor was used to control the Zumo robot. Located on the PSoC is a USB connection, an LED, a button to reset the program and a programmable button that can execute programmed commands. Code was compiled and written to the PSoC by connecting it to a computer via USB and using PSoC Creator 4.2.



Figure 2.

## Polulu Zumo Robot

The Zumo robot hardware can be broken down into the following individual components: motors, 6 reflectance sensors, an accelerometer, an ultrasonic sensor, an infrared sensor and a Wi-Fi chip. A router acting as an MQTT broker is used to communicate between our computers and the robot. There is also a programmable button on the PSoC hardware itself. The libraries for controlling these components have been generously provided by Metropolia University of Applied Science.

PuTTY was used to see the output of the code written to the PSoC.

Zotero was used to track any sources used during the course of this project.

The project involved programming the robot so that it could perform 3 different tasks: fight in a sumo arena, follow a line and navigate the maze. Each task can be broken down into components that are common between all 3. These tasks were approached on a weekly basis in side projects. They are as follows: Following a line, collision detection, obstacle detection

# Method

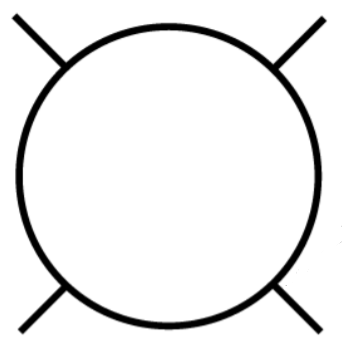
HERE WE OUTLINE THE DIFFERENT THINGS WE WANT THE ROBOT TO DO. Eg line following,

# Implementation

## 3.1 Project Tasks

### Sumo Wrestling

The arena for this task can be seen in the figure below.



The Sumo wrestling competition involves robots being placed at three of the lines perpendicular to the circle. After being placed on the line, the motors of the robot are powered on by pressing a button, after which the robot should drive the edge of the ring. Here it waits for a signal from an infrared remote and then it enters the battle.

The robot must know the time when it has reached the edge of the ring, when it has received a signal from the remote, when it has been struck by an opponent and when it has been powered off.

Initially it was presumed that the robot should react when it has been struck by an opponent. After testing, it was found that too many factors were involved to reliably improve our chances of victory with a simple victory. As such when the robot is in the ring, it will simply drive in a straight line until it sees the edge, turn around for some period of time and then repeat.

The flowchart of the sumo wrestling code looks like this:

### Line Following

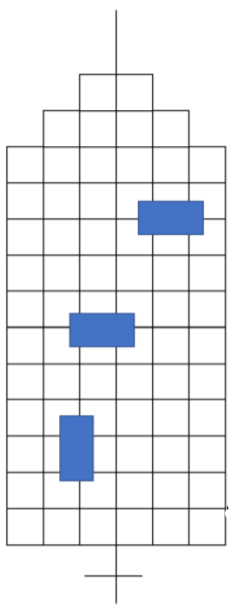
The line following task is a tournament between all teams to see who is the fastest. As such, the robot must know how long it takes to finish the track. To get an accurate time reading, the robot must automatically stop when reaching a goal, as opposed to continuing operation until user input as with sumo wrestling. To achieve this

Following a line – The robot must approach the edge of the arena from one of the 4 lines on the circumference of the arena.

Detecting a horizontal line – The robot must wait at the edge of the arena until it is given a command from the infrared remote.

Detecting a collision – The robot must know it has collided with another robot and it must know on which side the collision occurred.

Detecting an object – The robot must know when an opponent is in front of it and react appropriately.



2.1.2

2.1.3

2.2 The components

2.2.1 Starting and Stopping the Robot

Technically the robot will begin executing its code as soon as it is powered on. To prevent it from immediately executing its motor functions, it can be held in place by holding the code in an infinite loop that can be broken out of by pressing the button or sending an infrared signal via remote.

The robot can similarly be stopped by these two inputs, however for several tasks during the project it was more sensible to designate a goal and to break the motor function loop when the goal condition is met. The below figure illustrates how the robot can recognize when it has reached its goal.

### Following a line

Lines are followed by using the Zumo robot’s infrared reflectance sensors. The sensors can return a RAW number or a digital number that is flipped depending on some threshold. An if/else statement was constructed that influenced the speed of the motors based on the sensor readings. The below figure illustrates the conditions considered for this task and the reactions to them.

2.2.3 Detecting objects

The robot can detect objects ahead of it using an ultrasonic sensor. Careful consideration was made when using the sensor as its reliability rapidly decreases as objects get very near and very far. It is necessary to detect objects in the maze task so that the robot can navigate around 4 wooden blocks placed on the grid. The below figure outlines when the robot is asked to search for obstacles and how it avoids them.

Obstacle detection was initially considered for the Sumo Wrestling task also; however it was ultimately decided that it provided no benefit.

2.2.4 Detecting changes in acceleration

The robot can detect changes in acceleration using an accelerometer, most notably from a collision with an obstacle or another robot. Initially it was presumed that the robot should change its behavior during Sumo Wrestling if a collision was recorded. However, this seemed unlikely to improve the robot’s chance at victory. As such, the robot knows when it has been hit, but its only reaction is printing a message that it has been hit.

There must always be text or a new subheading below each heading. Do not place a figure or table below a heading with no text in between.

Label each figure and table appropriately. Provide a number, title and reference (if needed) below each figure and above each table. Make sure to mention all figures and tables in the text. Each figure and table must be explained in the text and referred to by its number (… as figure 1 illustrates. /as summarized in table 1.).

Apply the Figure style for each image. This is necessary in order to prevent a page break from occurring between the figure and its caption. The Figure caption style is applied for the figure’s caption. This causes the figures to be numbered automatically.

1. Virtual studies completed by Metropolia students in the academic year 2009-2010.

There must always be text between a figure or table and a new heading.

## Subheading

### Subheading

There must always be text or a new subheading below each heading. Do not place a figure or table below a heading with no text in between.

1. Virtual studies completed by Metropolia students in the academic year 2009-2010.

|  |  |
| --- | --- |
| Field of study | Studies completed, ECTS |
| Culture | 131 |
| Technology, Communication and Transport | 552 |
| Health Care and Social Services | 175 |
| Business and Administration | 52 |
| Not bound to a field of study | 18 |
| Metropolia total | 928 |

There must always be text between a figure or table and a new figure or table or a new heading.

### Subheading

There must always be text or a new subheading below each heading.

Use the Quotation style for an indented quotation. For the last paragraph immediately before the quotation, use the Body Text before Quotation or List style.

If a direct quotation is several lines long, indent the quotation and use single (1.0) line spacing. No quotation marks are used then. Always provide a reference to the source. If the direct quotation is shorter than two lines, include it in the body of the text in quotation marks, and provide a reference to the source.

After an indented direct quotation, continue the text at the left margin using the Body text style.

Use the Bulleted list style for an in-text list:

* This is the first list item.
* The second item of the list contains a long text that spans more than one row. The left margin will be automatically justified.
* This is the third list item.
* This is the fourth list item.

The items on the bulleted list begin with a capital letter. An item ends in a full stop if each item on the list is a full sentence.

The list items begin with a lower-case letter if the list items are not sentences. The last item is followed by a full stop. Thus, a thesis consists of

* words
* sentences
* paragraphs
* sections.

After the list, the text continues from the left margin in the Body text style.

You can insert numberChapter Heading

References

Details of the references are given here. Use the referencing system required in your degree programme or as agreed with your supervisor.

Layout of this page in the author-date (Harvard) referencing system:

Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

Layout of this page in the number (Vancouver) referencing system:

1. Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.
2. Details of the reference Details of the reference Details of the reference Details of the reference Details of the reference.

**Title of the Appendix**

The contents of the appendix are placed here. Below are the instructions for removing and adding appendices in a way that maintains the headers and footers in their correct form.

Instructions for removing an unwanted appendix:

1. Select the entire page(s) that form the appendix and delete the contents by hitting the Delete key.
2. As you are in the beginning of the empty appendix page (see the image below), double-click the header of the empty page and press Link to Previous button in the ribbon. The following dialogue window opens:



Click Yes.

1. If necessary, make hidden format information visible by pressing .
2. Delete the section break immediately before the appendix to be removed (see image below).



Instructions for adding a new appendix:

1. Place the cursor at the end of the last appendix.
2. Select Page Layout from the menu bar. From the ribbon, select Breaks/Section Breaks/Next Page. This causes a new appendix to appear, but the appendix number in the header is not yet correct.
3. Double click the header of the new appendix with the wrong appendix number. If the option “Link to Previous” is selected, click the corresponding button to deselect it.
4. Replace the appendix number with the correct number.

Note that the appendices need to be updated in the table of contents manually.

**Title of the appendix**

The contents of the appendix are placed here.