Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2022

Runtime Terror
Turing Board

Sahaj Amatya
Sarker Nadir Afridi Azmi
Kendall Buchanan
Keaton Koehler
Happy Ndikumnaa
Lydia Sarver

Revision History

Revision	Date	Author(s)	Description
0.1	1.01.2016	GH	document creation
0.2	1.05.2016	AT, GH	complete draft
0.3	1.12.2016	AT, GH	release candidate 1
1.0	1.20.2016	AT, GH, CB	official release
1.1	1.31.2016	AL	added design review requests

CONTENTS

List of Figures

List of Tables

1 Introduction

Your introduction should provide a brief overview of the product concept and a reference to the requirement specification and architectural design documents in 1 or 2 paragraphs. The purpose is to provide the reader with the location of relevant background material that lead to the design details presented in this document.

2 System Overview

This section should reintroduce the full data flow diagram from the architectural specification, and discuss at a high level the purpose of each layer. You do not need to include a subsection for each layer, a 1 - 2 paragraph recap is sufficient.

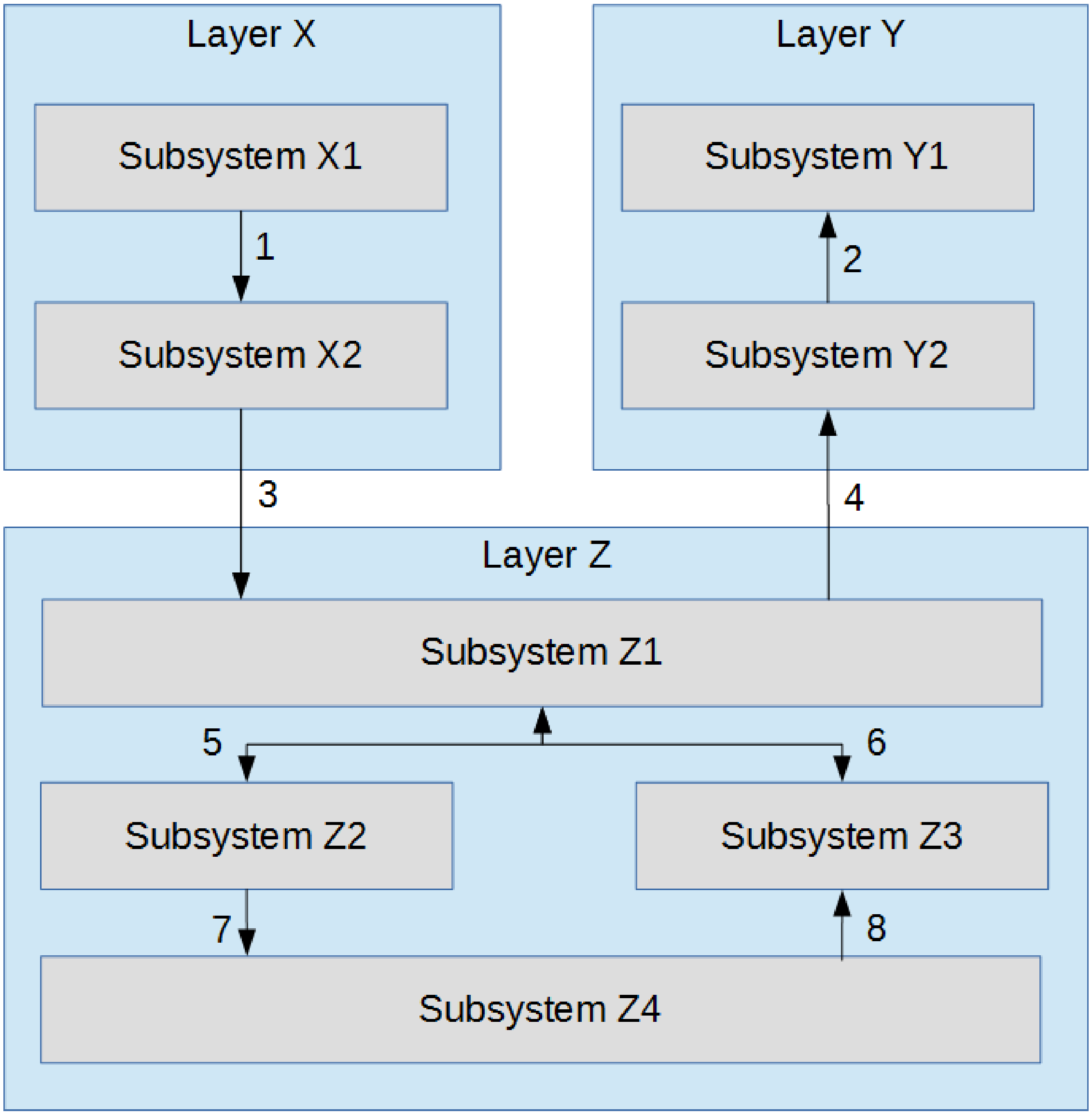

Figure 1: System Architecture

3 X Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be omitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

3.1 Layer Hardware

A description of any involved hardware components for the layer. For example, if each subsystem is a software process running on an embedded computer, discuss the specifics of that device here. Do not list a hardware component that only exists at the subsystem level (include it in the following sections).

3.2 Layer Operating System

A description of any operating systems required by the layer.

3.3 Layer Software Dependencies

A description of any software dependencies (libraries, frameworks, etc) required by the layer.

3.4 Subsystem 1

Describe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subsystem and not a repeat of anything discussed above for the overall layer.

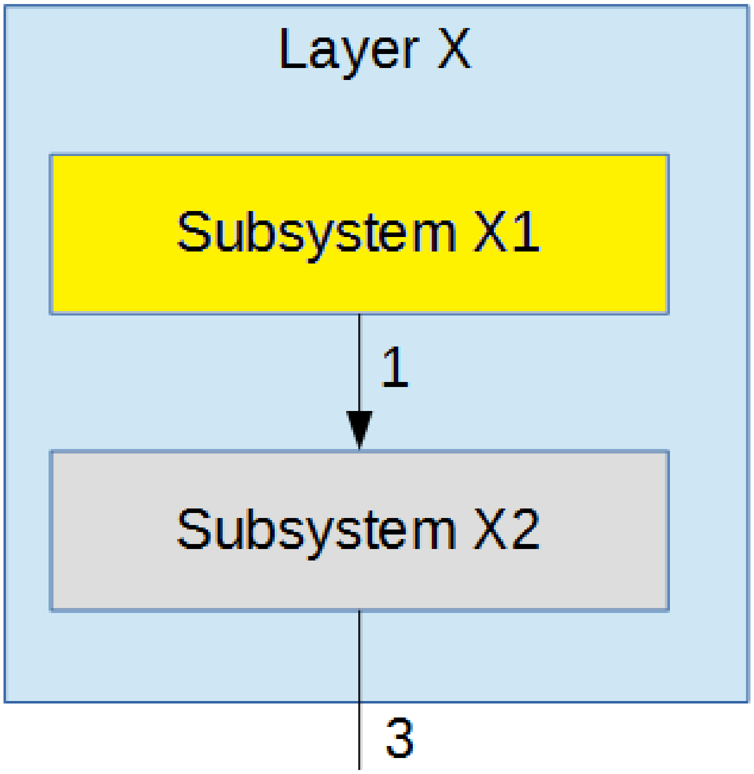

Figure 2: Example subsystem description diagram

3.4.1 Subsystem Hardware

A description of any involved hardware components for the subsystem.

3.4.2 Subsystem Operating System

A description of any operating systems required by the subsystem.

3.4.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

3.4.4 Subsystem Programming Languages

A description of any programming languages used by the subsystem.

3.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

3.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

4 Y Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be omitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

4.1 Layer Hardware

A description of any involved hardware components for the layer. For example, if each subsystem is a software process running on an embedded computer, discuss the specifics of that device here. Do not list a hardware component that only exists at the subsystem level (include it in the following sections).

4.2 Layer Operating System

A description of any operating systems required by the layer.

4.3 Layer Software Dependencies

A description of any software dependencies (libraries, frameworks, etc) required by the layer.

4.4 Subsystem 1

Describe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subsystem and not a repeat of anything discussed above for the overall layer.

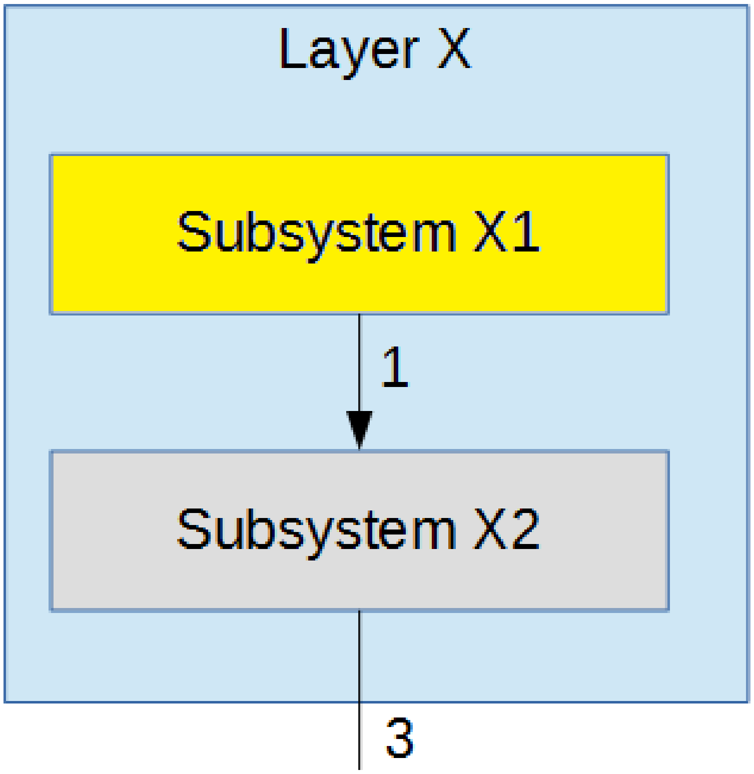

Figure 3: Example subsystem description diagram

4.4.1 Subsystem Hardware

A description of any involved hardware components for the subsystem.

4.4.2 Subsystem Operating System

A description of any operating systems required by the subsystem.

4.4.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

4.4.4 Subsystem Programming Languages

A description of any programming languages used by the subsystem.

4.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

4.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

5 Controls Layer Subsystem

The controls layer is responsible for binding all modules of the Turing Board to be part of the same system. All data coming in is intercepted by this layer and forwarded to the respective modules which are responsible for processing the forwarded data.

5.1 Layer Hardware

The hardware involved in this layer consists of the following:

- Nvidia Jetson TX2 responsible for running the controls code.
- VESC responsible for the speed control of the wheels.
- Intel real sense camera responsible for capturing real time footage for the follow-me feature and autonomous navigation.
- Texas Instruments Tiva C Series microcontroller which controls the turning mechanism.

5.2 Layer Operating System

The Layer makes use of the Linux operating system to run applications such as the control code.

5.3 Layer Software Dependencies

- Python Dependencies
  - PyVESC
  - Pyrebase
  - Pyserial
  - Pycrc
  - Threading

5.4 Controls Subsystem

There are three main components of project which calls for this piece of software which must all be non-blocking in nature to ensure the entire system stays responsive.

- Reading data from the microcontroller.
- Forwarding data to the microcontroller.
- Fetching data from the Firebase Real-time database.

5.4.1 Subsystem Hardware

The controls subsystem does not consist of explicit hardware. All work is done through software to communicate information.

5.4.2 Subsystem Operating System

The controls subsystem does not make use of an underlying operating system. The layer makes use of the Linux operating system.

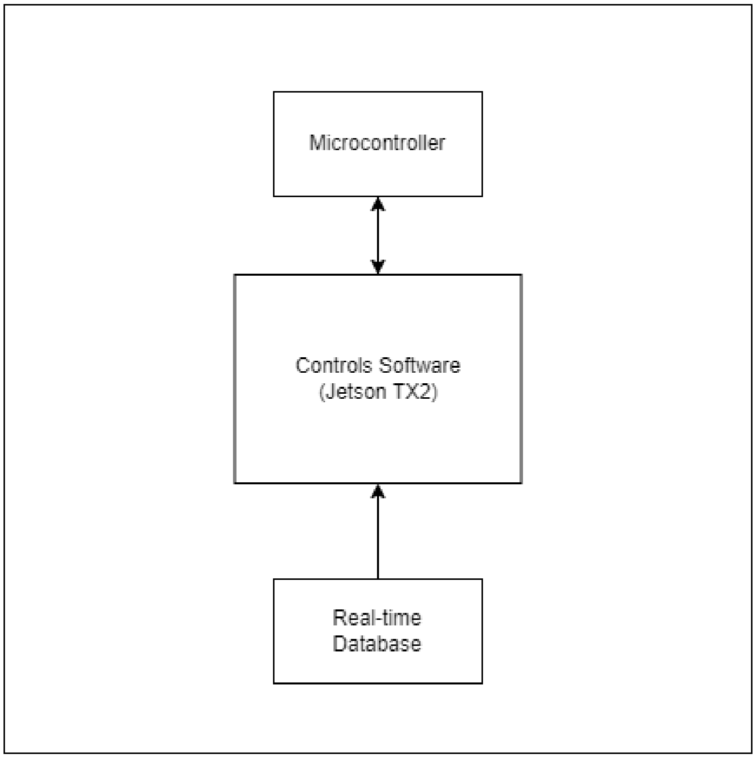

Figure 4: Example subsystem description diagram

5.4.3 Subsystem Software Dependencies

- Python Dependencies
  - PyVESC
  - Pyrebase
  - Pyserial
  - Pycrc
  - Threading

5.4.4 Subsystem Programming Languages

- Python
  - Used for the main controls system.
- C
  - Used to program the microcontroller.

5.4.5 Subsystem Data Structures

- Classes
  - Controls
    - ⋆ Responsible for running the controls code.
  - SerialCommunication
    - ⋆ Establishes connection with the microcontroller.

5.4.6 Subsystem Data Processing

After fetching data from the database, the controls software will process the data first to an extent. Since data coming in will be floating point values, it first needs to be translated into value which the microcontroller can understand. So, the entire range of data from the remote control app is taken and the required data is mapped from 0-255 which is then forwarded to the microcontroller which causes the wheels to change speed. As part of the same data packet, angle data from the controls code is also sent to the microcontroller which aids in turning the turning mechanism to a specific angle with respect to the turning mechanisms origin. Any data such as weight values if someone is standing on the long board (an integral part of the design so that the software knows when to turn off the turning mechanism) is received back in the same data format (0-255) which gets translated to weight values inside of the controls code.

6 Appendix A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

References