MicroPython OOP Pi Pico Mini Project

Kevin Tran

Table of contents

**Contents**

[Research and planning 3](#_Toc206704682)

[UML Class Diagram 3](#_Toc206704683)

[Data Flow Diagrams 4](#_Toc206704684)

[Flowchart 5](#_Toc206704685)

[Wiring diagram 6](#_Toc206704686)

[Material components list 6](#_Toc206704687)

[Power supply calculations 7](#_Toc206704688)

[Online simulations 7](#_Toc206704689)

[Producing and implementing 8](#_Toc206704690)

[Product photos 8](#_Toc206704691)

[Unit testing 9](#_Toc206704692)

[Testing and evaluating 9](#_Toc206704693)

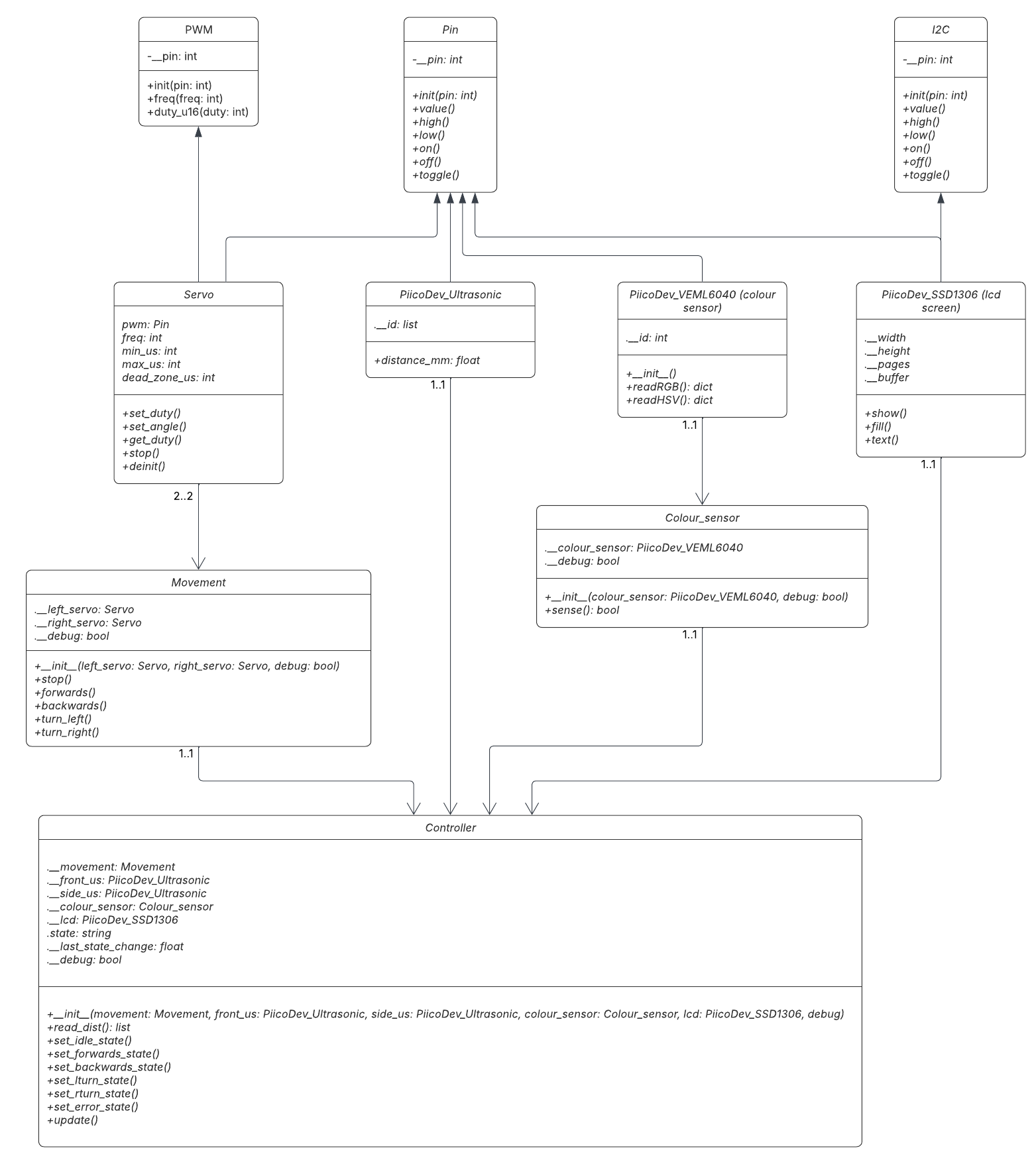


1

# Research and planning

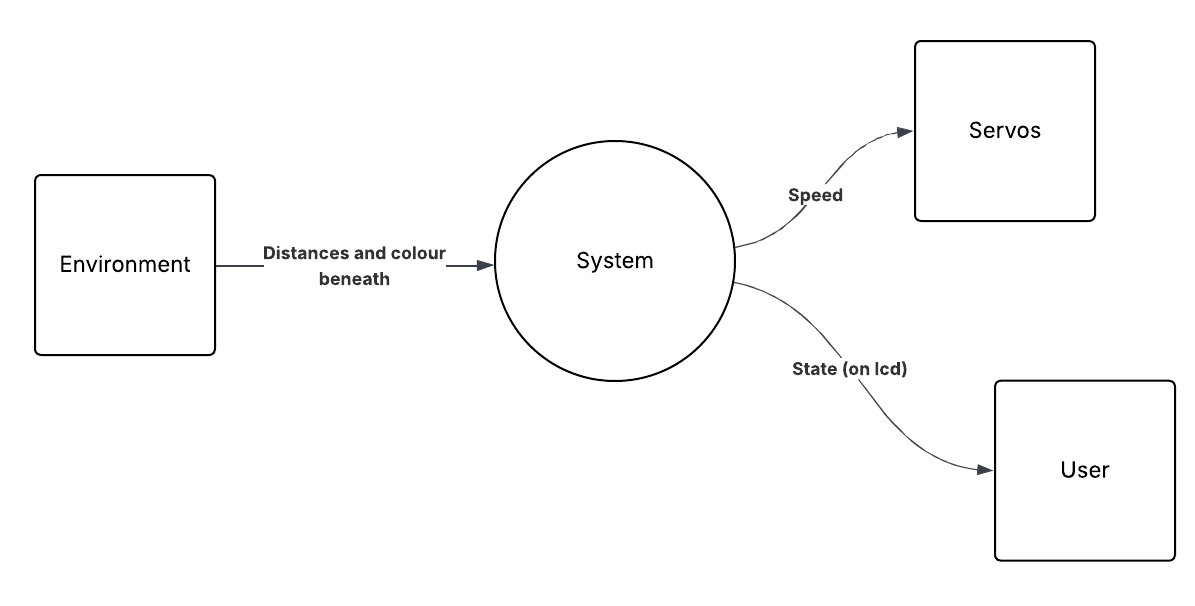
The research and planning phase consists of mostly diagrams, being the UML class diagram, the DFD diagrams, a flowchart, a wiring diagram, and a power supply diagram, along with a material components list.

## UML Class Diagram

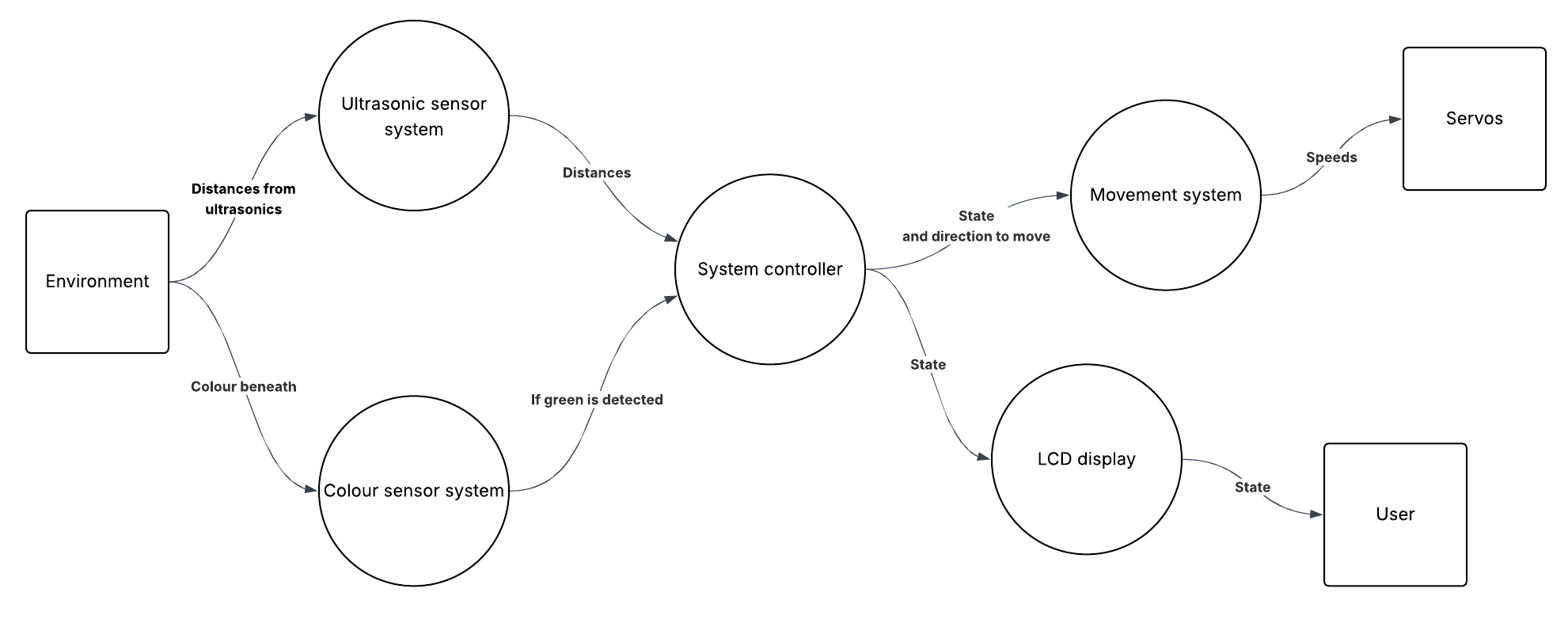


## Data Flow Diagrams

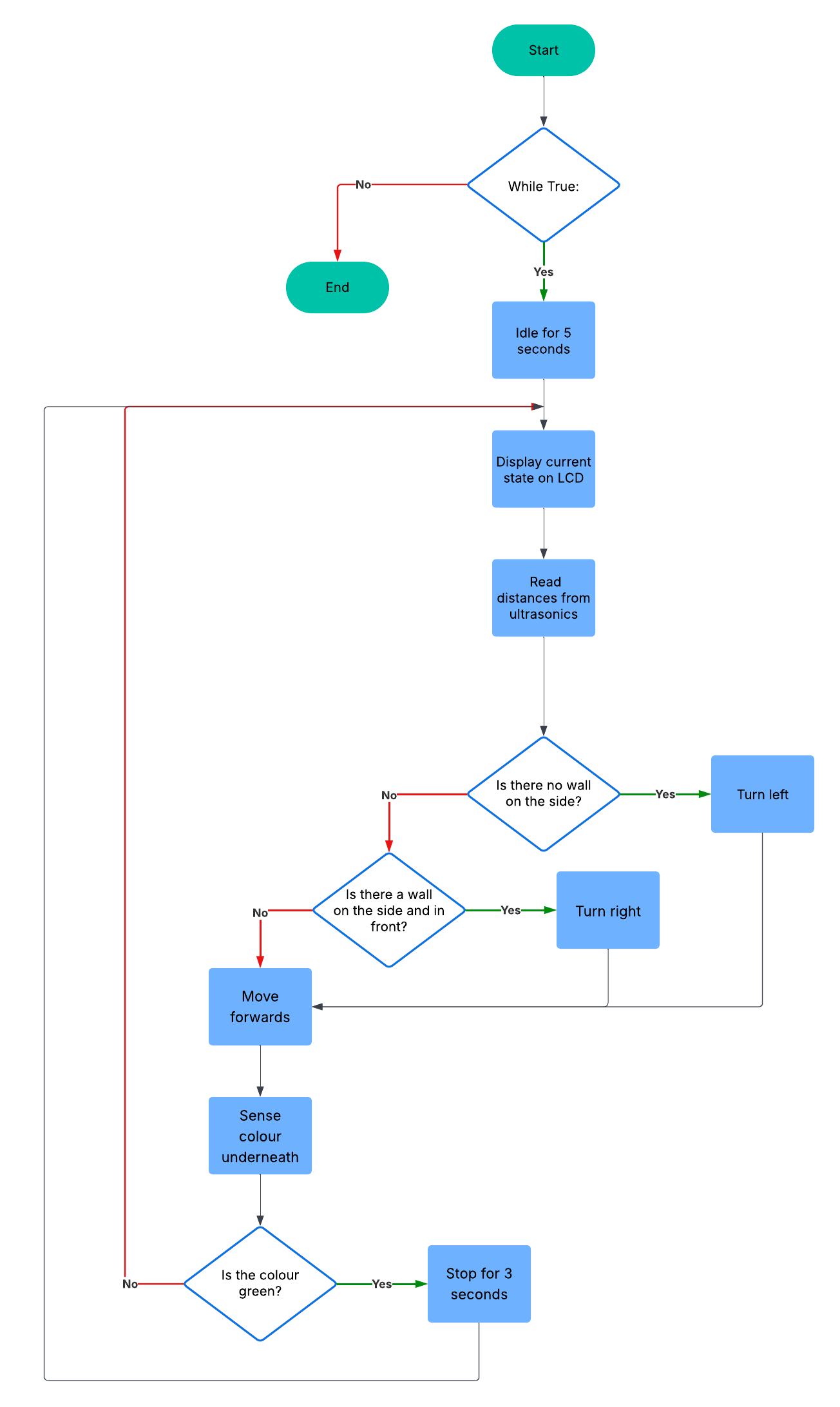
DFD 0:



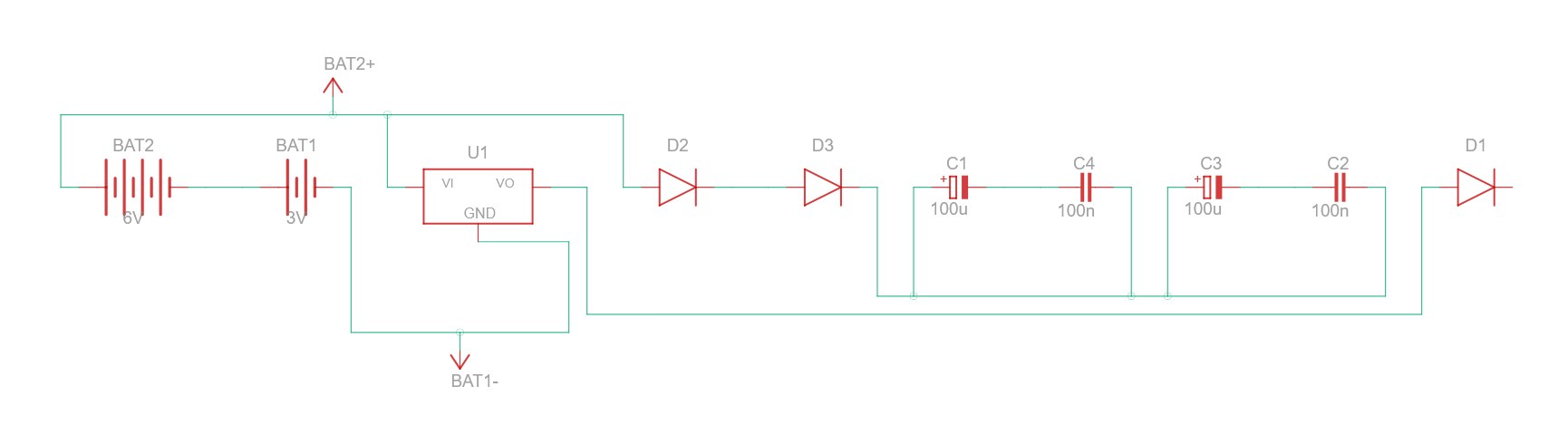
DFD 1:



## Flowchart



## Wiring diagram



## Material components list

The materials and components that I used in my prototype were:

Materials:

* Wooden chassis
* 2x wheels
* 1x omnidirectional wheel
* 1x battery pack
* 4x male to male wires
* 4x male to female wires

Components:

* 1x Raspberry Pi Pico 2
* 2x 3.7V batteries
* 2x DFrobot DF15RSMG servos
* 2x ultrasonic sensors
* 1x colour sensor
* 1x OLED screen
* 3x diodes
* 1x fuse
* 1x de amplifier
* 2x capacitors
* 2x polarized capacitors

## Power supply calculations

|  |  |
| --- | --- |
| **Battery pack** (2x 3.7V):  7.4V | **2 capacitors** (step down of ~0.5V each)  7.4V -> 6.4V |
| **Servos**  5V received | **De amplifier** (down to 5V)  6.4V -> 5V |

## Online simulations

😬

2

# Producing and implementing

## Product photos

( put them here )

3

# Testing and evaluating

## Unit testing

## Justification of techniques

The object-oriented programming techniques that I have used in this project were used to increase the efficiency, maintainability, and readability of my code if I were to edit it in the future. Each technique used has its own benefits, such as encapsulation creating modular code and improving security by having private variables, like in the code snippet of the