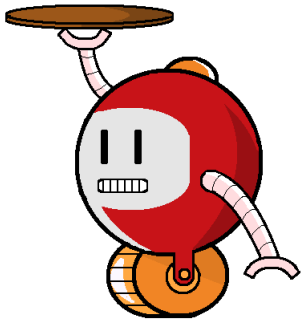

LazyBots

McMaster University

Development Process and Implementation

SE 4GA6 & TRON 4TB6

Group 9

Karim Guirguis	001307668
David Hemms	001309228
Marko Laban	001300989
Curtis Milo	001305877
Keyur Patel	001311559
Alexandra Rahman	001305735

Table of Contents

List of Tables

List of Figures

1 Revisions

Table 1: VIC Table of Revisions

Date	Revision Number	Authors	Comments
November 24th, 2017	Revision 0	Karim Guirguis David Hemms Marko Laban Curtis Milo Keyur Patel Alexandra Rahman	-

2 Purpose

The purpose of this project will be to create an autonomous robot that will navigate to and serve the requested drink to the user who requests a drink. Currently in an office setting, workers must leave their offices to get their own drinks. Also, in restaurants, drinks are served by waiters and waitresses, which hinders them from doing other work at that time. Alfred will be designed to make the serving drinks autonomous.

Alfred will allow users to request drinks. These requests will form a queue which Alfred will serve in order using a FIFO protocol. Alfred will go to the table of each user and pour the drinks ordered from that table. Alfred will also have an administrator user which will be able to call Alfred back and override any action that is being taken at the time.

The following document will outline the overall system components, as well as the overall system behaviour, operation and undesired error handling.

2.1 Scope

The system implemented is one that is meant to automate the dispensing of beverages to customers within a restaurant at the respected customers' table. The customer will be able to order a drink from their table which will be followed by Alfred arriving at their table and dispensing the requested drinks. The staff will be able to request Alfred to come back for charging and refilling when desired.

2.2 Context Diagram

The following is a context diagram of the drink serving robot, Alfred.

2.3 Diagram of Components

The following is a diagram that shows the interaction of components of the drink serving robot system.

3 System Variables

3.1 Monitored and Controlled Variables

The following is a list of variables that will be monitored.

Monitor Name	Monitor Type	Range	Units	Comment(s)
$w_{wheel_{left}}$	Speed	[0, 100]	rad/s	Left Wheel Speed
$w_{wheel_{right}}$	Speed	[0, 100]	rad/s	Right Wheel Speed
$m_{container}$	Mass	[0, 1.0]	Kg	Weight of the storage device
m_{drink}	Mass	[0, 1.0]	Kg	Weight of the drink
$b_{cuptaken}$	Boolean	[0,1]	N/A	If the cup has been taken
$d_{objects}$	Distance[]	[0, 10.0]	m	Set of distances to closest obstacle
V_{batt}	Voltage	[0, 20.0]	m	Voltage levels of batteries

The following is a list of variables that will be controlled

Controlled Name	Controlled Type	Range	Units	Comment(s)
w_{motor}	Speed	[0, 100]	rad/s	Motor Speed
$percent_{dutycycle_{left}}$	Percent	[0,1.0]	%	The duty cycle of the left side of the drive-train
$percent_{dutycycle_{right}}$	Percent	[0, 1.0]	%	The duty cycle of the right side of the drive-train
V_{pump}	Voltage	[0, 5.0]	m	Voltage going to the liquid pump

$errors_{drivetrain}$	Unsigned Byte	[0, 2^8]	N/A	Errors from drive-train
$errors_{pump}$	Unsigned Byte	[0, 2^8]	N/A	Errors from pumping system module
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred
$LED_{drinksignal}$	Boolean	[0, 1]	N/A	Signal of drink that it is ready to be picked up
Q_{pump}	Flow Rate	[0, 100]	(m^3/s)	Flow rate of the pump

3.2 Constants

The following is a list of system constants

Constant Name	Constant Type	Value	Units	Comment(s)
$V_{battMin}$	Voltage	9.0	m	The minimum voltage necessary for drive-train movement
$m_{DrinkMin}$	Mass	0.35	Kg	The minimum weight of the drink to be considered as ready for t
$t_{timeout}$	Time	30	s	The maximum time Alfred or the ordering application will wait f
$t_{pumptimeout}$	Time	5	s	The maximum time Alfred will try to pump without noticing a cl
$t_{maxpump}$	Time	10	s	The maximum time Alfred will try to dispense a drink within.
Q_{pump}	Flow Rate	TBD	(m^3/s)	Flow rate of the pump
$freq_{bauderate}$	Frequency	9600	(Herts)	The rate at which UART communication will be performed at.
$obstruction_{timeout}$	Time	10	s	The amount of time that Alfred will stop movement due to an ob

4 Behaviour Overview

1. **Ordering System Module** - Provided the order information, this module will communicate with the server to "order a drink".
2. **Manager Login Page** - Given the login credentials, will authenticate administrator of the system with the server.
3. **Manager Station Map Software Page** - Will allow administrator to create/modify the map of the area where Alfred will deliver drinks. This map will then be sent and stored on the server.
4. **Manager Station Request Software Page** - Will allow administrator to execute commands for Alfred, as well as view incoming error codes from Alfred. Said commands, will be sent to the server, which will then communicate with Alfred.
5. **Back End Web Service Module** - Will route communication to different components of the system. Also will be responsible for authentication and management of queue.
6. **Alfred Manager Module** - Endpoint for communication with Alfred. Will manage communication with server, as well as send any errors that Alfred is experiencing.
7. **Alfred Drive Train Module** - Responsible for driving and managing the motors based on desired route. Also, will be send errors preventing movement to Alfred Manager Module.
8. **Alfred Pumping Module** - Will control pumping system in regards of when to pour, how long, rate of dispensing, etc. Also, will be send errors of pump/container to Alfred Manager Module.
9. **Image Processing Module** - Will detect any obstacles in the way as well as locate incoming nodes. Will communicate with Alfred Drive Train Module, to determine any required action based on results.

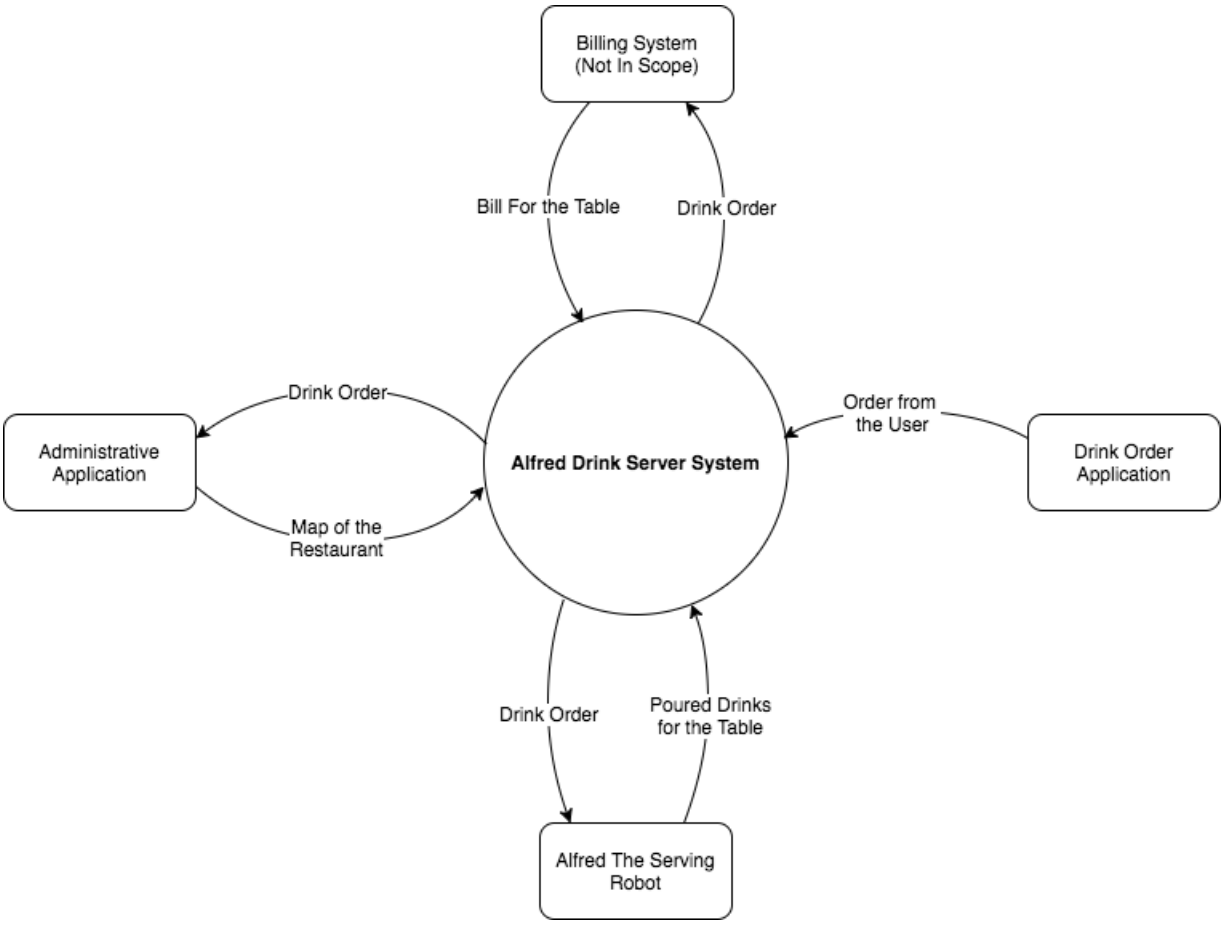

Figure 1: Drink Serving Robot Context Diagram

5 Component Overview

5.1 Ordering System Module

5.1.1 Inputs and Outputs

Inputs: User input defining:

Input Name	Input Type	Range	Units	Comment(s)
$Order_{Num}$	Unsigned Integer User Input	[0,5]	count	Number of Orders
$Order_{Drinks}$	User Input	N/A	N/A	List of Ordered Drinks

Outputs: Packaged Information within HTTP Request for:

Output Name	Output Type	Range	Units	Comment(s)
$Order_{Num}$	Unsigned Integer User Input	[0,5]	count	Number of Orders
$Order_{Drinks}$	User Input	N/A	N/A	List of Ordered Drinks
$Order_{TableNum}$	Unsigned Integer User Input	$[0,2^{16}]$	N/A	Table Number
$Order_{Rid}$	Unsigned Integer User Input	$[0,2^{16}]$	N/A	Identification of the restaurant

5.1.2 Description

This module will be used as the user input to take orders of the client. This user input will be taken in by the mobile application based on different button inputs/ radio button selections. These orders are then packaged by the module to be sent to the server based on an HTTP request.

5.1.3 Timing Constraints

Timing Constraints are based on the server sending a success signal within $t_{timeout}$ seconds.

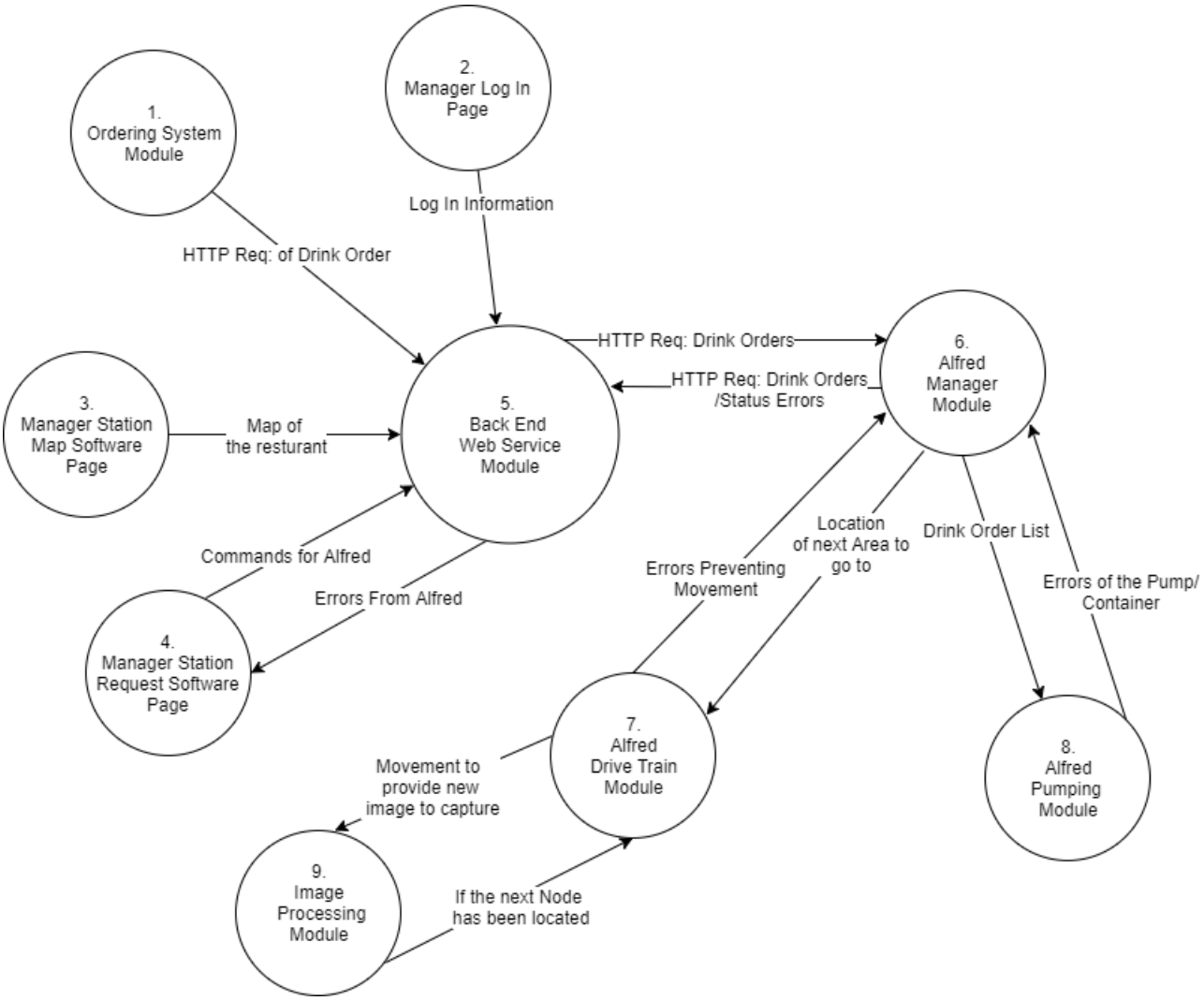

Figure 2: Drink Serving Robot System Component Diagram

5.2.2 Description

This module is a web based application for the managers to be able to log into the management systems.

5.2.3 Timing Constraints

Given optimal networking conditions, the server must respond with $t_{timeout}$ seconds.

5.2.4 Initialization

This will default to a HTML page with no information.

5.3 Manager Station Map Software Module

5.3.1 Inputs and Outputs

Inputs: User input defining:

Input Name	Input Type	Range	Units	Comment(s)
$Resturant_{Width}$	Integer User input	[0,20]	m	
$Resturant_{Length}$	Integer User input	[0,20]	m	
$Resturant_{CanTravel}[][]$	Button[][] User input	N/A	N/A	A set of areas in which Alfred can travel to
$Resturant_{Tables}[][]$	Button[][] User input	N/A	N/A	A set of areas in which contains tables
$Resturant_{CannotTravel}[][]$	Button[][] User input	N/A	N/A	A set of areas in which Alfred cannot travel to
$Resturant_{Home}$	Button User input	N/A	N/A	An area that defines Alfred's home location

Outputs: A text file that contains the following information

Input Name	Input Type	Range	Units	Comment(s)
$Resturant_{Width}$	Integer	[0,20]	m	
$Resturant_{Length}$	Integer	[0,20]	m	
$Resturant_{D}ata$	Char	[H,X,0,T]	N/a	Table Information

5.3.2 Description

This module will be used as the user input to create a map where the manager of the restaurant will be able to define the details about the restaurant that will help Alfred with it's navigation to different tables.

5.3.3 Timing Constraints

Timing Constraints are based on the server sending a success signal within $t_{timeout}$ seconds.

5.3.4 Initialization

At startup/new order of the application, it will load in the users map that is associated with their profile. If this is the first time using the application then it will default to a 1x1 map.

5.4 Manager Station Request Software Module

5.4.1 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
req_{Home}	Boolean User input	[0,1]	N/A	Calling back Alfred
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred

Outputs: To be displayed to the User

Output Name	Output Type	Range	Units	Comment(s)
$warn_{LowLiquid}$	Boolean	[0, 1]	N/A	Low liquid levels
$error_{LiquidLeak}$	Boolean	[0, 1]	N/A	Leaking of liquid error
$error_{NotPumping}$	Boolean	[0, 1]	N/A	Not pumping error
$error_{NoMovement}$	Boolean	[0, 1]	N/A	Not able to move error
$error_{LowBatt}$	Boolean	[0, 1]	N/A	Low battery Error

5.4.2 Description

This module is used for the manager to determine the state of Alfred from an office as well as being able to override the robot to come back to the home base.

5.4.3 Timing Constraints

One timing constraints are based on the server sending a success signal within $t_{timeout}$ seconds. Another constraint is that Alfred will return within the time of w_{wheel}*distance.

5.4.4 Initialization

At startup this interface should pull the last status of the robot and display it to the user. If no previous data is found then it will return that there are currently no errors.

5.5 Back End Web Service Module

5.5.1 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$HttpReq_{Orders}$	HTTP Request	N/A	N/A	HTTP request package with the information for drink orders.
$Map_{textfile}$ of the restaurant	Text File	N/A	N/A	A text file with a map
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred
$UserName$	String	10 chars	N/A	
$Password$	HashCode	N/A	N/A	Success result for authenticated results.

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$HttpReq_{Orders}$	HTTP Request	N/A	N/A	HTTP request package with the information for drink orders.
$Map_{textfile}$ of the restaurant	Text File	N/A	N/A	A text file with a map
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred
$Login_{Status}$	String	10 chars	N/A	Login Success or Fail
$Http_{Result}$	String	10 chars	N/A	

5.5.2 Description

This module is a server component that will hold account information for different restaurants. This will perform authorize different users to be able to accept different drink requests from the ordering system. It will return the next drink order for the restaurants Alfred robot.

5.5.3 Timing Constraints

Given optimal networking conditions, the server must respond with $t_{timeout}$ seconds.

5.5.4 Initialization

This server will be initialized by a database with one restaurant which will be used for the purposes of testing the system

5.6 Alfred Manager Module

5.6.1 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
Map of the restaurant	Graph	N/A	N/A	Graph of the text file with a map
$Orders_{drinks}$	Order[]	N/A	N/A	A list of drink orders ordered by table.
$b_{requestHome}$	Boolean	[0, 1]	N/A	A request to come back to the base
$errors_{drivetrain}$	Unsigned Byte	[0, 2^8]	N/A	Errors from drive-train
$errors_{pump}$	Unsigned Byte	[0, 2^8]	N/A	Errors from pumping system module

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
NextOrder	Order	N/A	N/A	Next Drink Order.
NextNode	Node	[(0,0), (length(*Map*),width(*Map*)]	N/A	The next node to travel to.
$errors_{Alfred}$	Unsigned Byte	[0, 2^8]	N/A	Errors from Alfred

5.6.2 Description

This module acts as the manager to the different components of Alfred. It will run on a raspberry pi to command the drive-train to move to different nodes of the map. The manager will also communicate through UART to the different on board micro-controller for commanding the next drink to the pumping system.

5.6.3 Timing Constraints

Given optimal networking conditions, the system must receive a response within $t_{timeout}$ seconds. The communication with the pumping system must be done at the specified $freq_{bauderate}$.

5.6.4 Initialization

This will assume that there is no requests at startup. It will request from the server asking for the next drink order. If there is no map previously defined within a text file then it will request this as well.

5.6.5 Alfred Communication Circuit Diagram

The following is a circuit diagram showing the communication between the Raspberry Pi and the pumping system.

5.6.6 Raspberry PI Specifications

Manufacturer: Raspberry PI
Processor: Broadcom BCM2387 chipset.
Memory: 1GB
Power:Micro USB socket 5V1, 2.5A
GPIO: 17 pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector: 15-pin MIPI Camera Serial Interface (CSI-2)
Memory Card Slot: Push/pull Micro SDIO

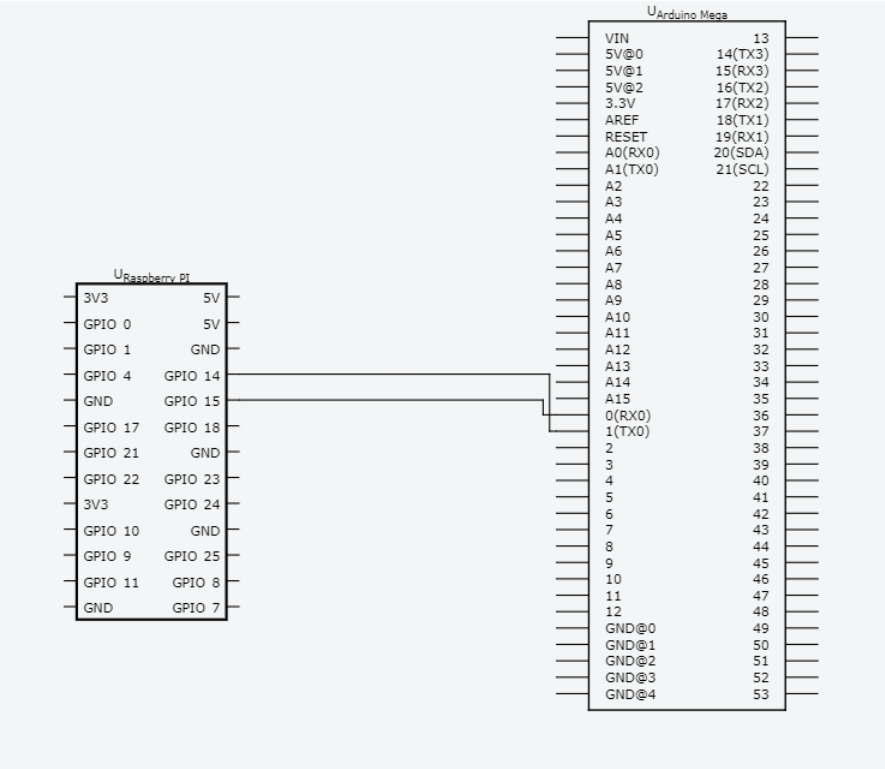

Figure 3: Alfred Communication Circuit Diagram

5.7 Alfred Drive-train Module

5.7.1 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
Map of the restaurant	Graph	N/A	N/A	Graph of the text file with a map
$NextNode$	Node	[(0,0), (length(Map),width(Map)]	N/A	The next node to travel to.
$w_{wheel_{left}}$	Float	[0, 100]	rad/s	Left Encoder Input
$w_{wheel_{right}}$	Float	[0, 100]	rad/s	Right Encoder Input
$d_{objects}$	Float[]	[0, 10.0]	m	
$Marker_{PosX}$	Unsigned Integer	[0,2^{16}]	Pixels	
$Marker_{PosY}$	Unsigned Integer	[0,2^{16}]	Pixels	
$b_{NextMarkerFound}$	Boolean	[0,1]	N/A	
V_{batt}	Float	[0, 20.0]	m	

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$percent_{dutycycle_{left}}$	Float	[0,1.0]	%	
$percent_{dutycycle_{right}}$	Float	[0, 1.0]	%	
$errors_{drivetrain}$	Unsigned Byte	[0, 2^8]	N/A	

5.7.2 Description

This module will provide power to the drive-train of Alfred. It will use the feedback of the left and right encoders and take the error to perform PI control on them. This PI control output will then be translated into a duty cycle for each side to be able to power the DC motors with pulse width modulation. This module will communicate to the image recognition software to receive the position of the next next marker and use this information of where it currently is on its path. This module will use ultrasonic sensors to get a set of distances ($d_{0,...,n}$) to the nearest object to determine if it is safe to continue moving

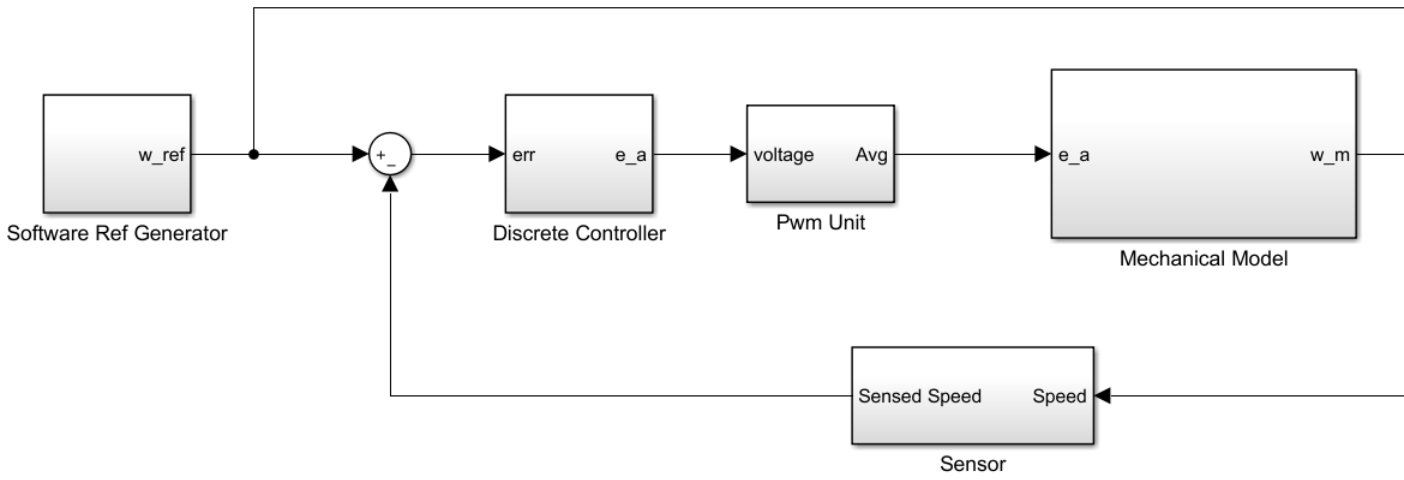

Figure 4: Top level of the dc motor control system

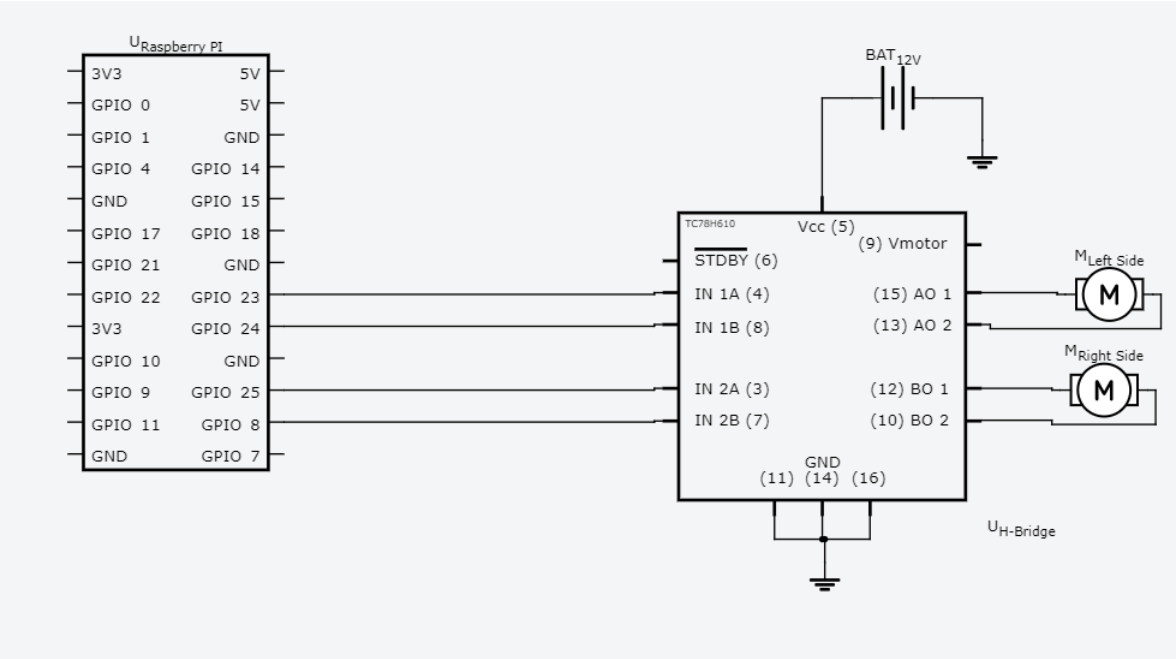

Figure 5: Weight Detection Circuit Diagram

5.7.7 DC Motor Specifications

5.7.8 H-bridge Specifications

5.7.9 Encoder Specifications

5.7.10 Battery Specifications

5.8 Alfred Pumping System Module

5.8.1 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
$b_{cuptaken}$	Boolean	[0,1]	N/A	
$m_{container}$	float	[0, 1.0]	Kg	
m_{drink}	float	[0, 1.0]	Kg	
$Order_{drinks}$	Order[]	N/A	N/A	List of Drinks

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$LED_{drinksignal}$	Boolean	[0,1]	N/A	
V_{pump}	float	[0, 5.0]	V	
$errors_{pump}$	Unsigned Byte	[0, 2^8]	N/A	

5.8.2 Description

This module consists of an Arduino Mega which will receive information from the Alfred manager module through UART to receive the next drink order. This will then begin to dispense the specific drink until it has reached $m_{containerMin}$. It will then show the customer that the drink has been completed by turning on: $LED_{drinksignal}$. It will then read $b_{cuptaken}$ from a light sensor to determine if the cup has been taken at which point it will then wait for the next drink cup to get in place and begin pouring. If it is not able to pump liquid, or it is losing fluid when not dispensing, then it will send the appropriate errors through UART back to the manager.

5.8.3 Timing Constraints

This module will have to dispense the drink within $t_{maxpump}$

5.8.4 Initialization

This module start by waiting for the manager to send the drink information for the system to pour. All errors within the system will start as false until they have been triggered.

5.8.5 Diagram of DC Pump Control System

The following is a diagram that shows the top level of the dc pump control system.

5.8.6 DC Pump Circuit Diagram

The following is a diagram that shows the DC Pump Circuit.

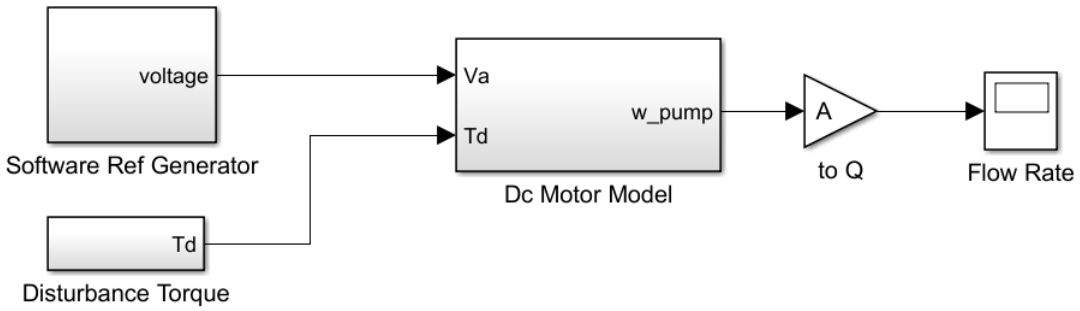

Figure 6: Top level of the dc motor control system

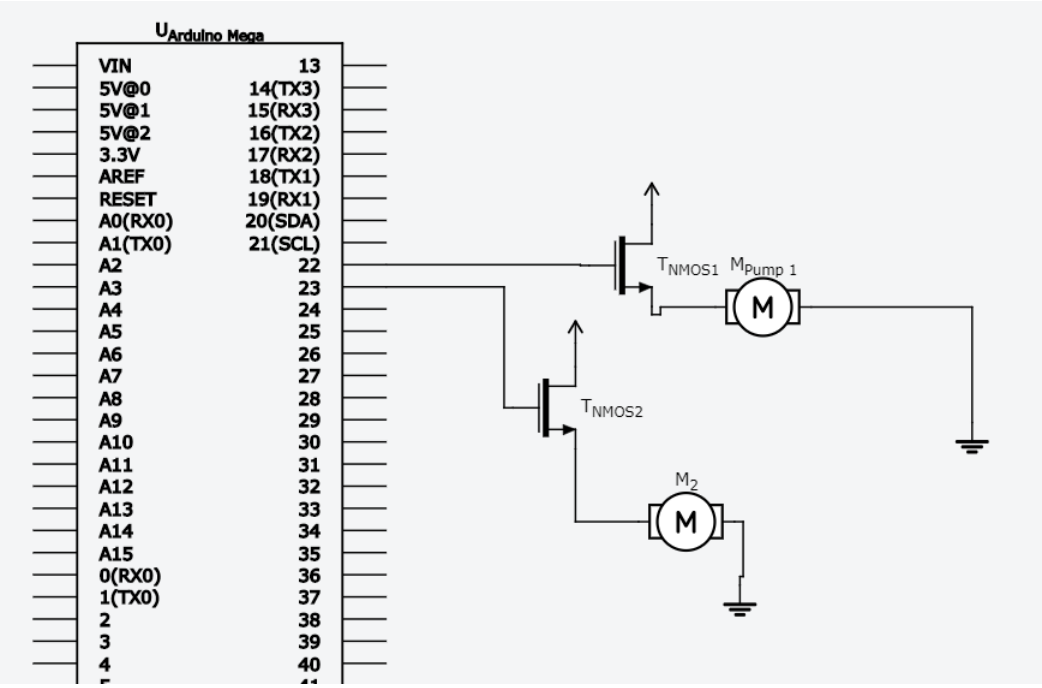

Figure 7: DC Pump Circuit Diagram

5.8.7 Liquid Temperature Circuit Diagram

The following is a circuit diagram for sensing the temperature of the liquid.

5.8.8 Weight Detection Circuit Diagram

The following is a circuit diagram for sensing the weight of the storage containers.

5.8.9 DC Pump Specifications

Manufacture : Yosoo
DC Voltage: 3-6V
Flow rate: 80-120L/H
Material: engineering plastic

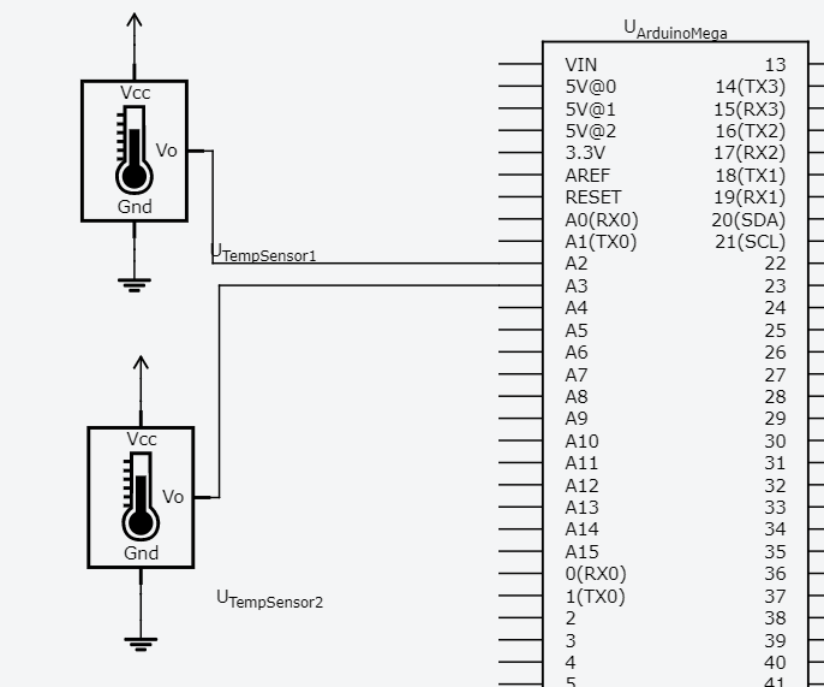

Figure 8: Top level of the dc motor control system

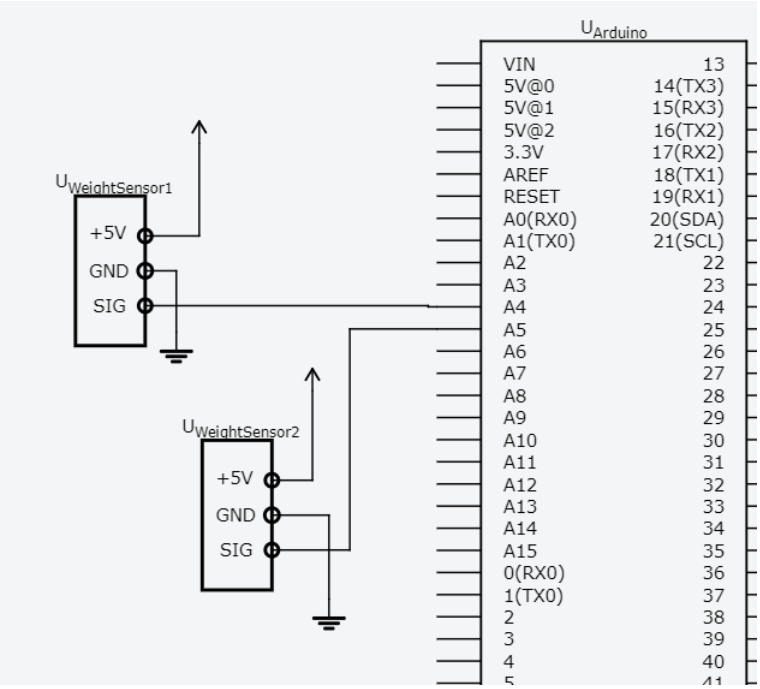

Figure 9: Weight Detection Circuit Diagram

Diameter: 24.5mm by 46mm
Outside diameter: 0.3"

5.8.10 Arduino Mega Specifications

Microcontroller: ATmega1280
Operating Voltage: 5V
Digital I/O Pins: 54 (of which 15 provide PWM output)
Analog Input Pins: 16
DC Current per I/O Pin: 40 mA
DC Current for 3.3V Pin: 50 mA
Flash Memory: 128 KB
Clock Speed: 16MHz

5.8.11 Container Specifications

Manufacture : Rubbermaid
Dimensions: 10 1/2" x 9" x 4" Capacity: 18 Cups/4.2 L

5.8.12 Tubing Specifications

Manufacture : Plumb Craft
Diameter: 1/4"ID Food Grade Tubing will be used.

5.9 Alfred Image Processing Module Module

5.9.1 Inputs and Outputs

Inputs:

Input Name	Input Type	Range	Units	Comment(s)
	Bitmap	N/A	N/A	Image of the ceiling

Outputs:

Output Name	Output Type	Range	Units	Comment(s)
$Marker_{PosX}$	Unsigned Integer	[0,2^{16}]	Pixels	
$Marker_{PosY}$	Unsigned Integer	[0,2^{16}]	Pixels	
$b_{NextMarkerFound}$	Boolean	[0,1]	N/A	

5.9.2 Description

This module will receive information from a camera while the drive-train is in motion. This camera will be looking at the ceiling looking for positions of circles that will denote one meter distances set up within a grid around the ceiling of the restaurant. This module will determine if there is a new maker within the image and if so where is the X and Y position of that marker to provide to the drive-train.

5.9.3 Timing Constraints

This information must process in time for the drive train to be able to navigate based off of it.

5.9.4 Initialization

This module start by the drive-train application.

5.9.5 Raspberry PI Camera Specifications

Manufacturer: Raspberry PI
Resolution: 1080p30, 720p60 and 640 ÃŮ 480p60/90
Field of View (FOV): 62.2 degrees by 48.8 degrees

6 Normal Operation

Alfred is a mostly autonomous robot, only requiring human intervention in the event of an error or warning. Alfred will be able to navigate the restaurant by itself, and will serve drinks to tables. Customers will be able to place orders via a mobile application, which will be sent to a server. Orders to serve will be sent to Alfred using a FIFO protocol. Once Alfred has finished with an order, it will be able to request for a new order to serve. Management will be able to recall Alfred to the kitchen at any point using the admin console. In the event of a recall, Alfred will finish the current job and will return to the kitchen afterwards. Management will also be able to create a map of the restaurant and upload it to Alfred, which will give Alfred the means to navigate the restaurant.

7 Undesired Event Handling

Alfred will be able to detect undesired behaviours and conditions such as low liquid levels below threshold $m_{containerMin}$, any issues with pumping liquid, low battery level, leaking liquids, and any blockages in the current path once Alfred has been blocked for a time greater than $obstruction_{timeout}$. In the event of any error condition, Alfred will send an error code to the kitchen, to alert the staff of the issue. Wherever possible, Alfred will return to the kitchen in an error condition to request a fix. Otherwise, if movement is not possible, the kitchen staff will have to pick Alfred up from the dining room. Along with alerting management of any current errors, Alfred will also be able to indicate whether it is returning to the kitchen or requires pickup.

8 References

Raspberry pi data sheet

[1]"Exploring Raspberry Pi", 2017. [Online]. Available: http://docs-europe.electrocomponents.com/webdoc

Arduino Mega Information

[2] "Arduino Mega". [Online]. Available:
https://www.arduino.cc/en/Main/ArduinoBoardMega. [Accessed: 23- Dec- 2017].

DC Pump Information

[3] "3-6V Mini Micro Submersible Pumps DC Motor Pump Water Pump - 80-120L/H". [Online]. Available:
https://www.amazon.ca/gp/product/B01LWQCXEL/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1. [Accessed: 23-

Pump Container Information

[4] "Rubbermaid 1856059 Modular Cereal Keeper". [Online]. Available: https://www.amazon.ca/gp/product/B

Raspberry camera pi data sheet

[4] "CAMERA MODULE". [Online]. Available:
https://www.raspberrypi.org/documentation/hardware/camera/. [Accessed: 23- Dec- 2017].