Conceptual Design Review
Team: Ugly Blues

Raul Venegas - Controls

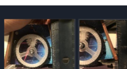
Objective: Design a system that allows the robot to balance as it navigates the maze

Research

Overwhelmingly, the method of choice for a self balancing robot is to use an IMU sensor to output gyroscope and accelerometer data.

- Requires motors to be very responsive
- Chassis are built with the balancing concept in mind
- In order to achieve translational motion, center of mass is allowed to swing.
- If the robot below is going to move forward, it must first sharply roll backwards.

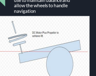
https://www.instructables.com/Arduino-Self-Balancing-Robot-1/



- 3Dot Chassis by comparison has significantly less room for maneuverability.
- Further increases requirement for responsive motors
- The center of mass would have to be compensated for since the original intent of the design is to use a castor.

Solution currently Under Review

-Use the same hardware drones use to maintain balance and allow the wheels to handle navigation

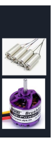


Two Primary Types of Motors for Drone Market:

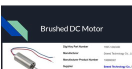
Brushed Barrel type DC motor:
-Generally less efficient
-Less torque
-High RPM
- Options for 3.7V

Brushless Disc Type Motors:
-Higher efficiency
-Higher torque
-Significantly lower RPM
-No Options for Anything Under 2S(7.4V)

1S = 1 cell in series x 3.7 V = 3.7 V
2S = 2 cells in series x 3.7 V = 7.4 V
3S = 3 cells in series x 3.7 V = 11.1 V
4S = 4 cells in series x 3.7 V = 14.8 V
5S = 5 cells in series x 3.7 V = 18.5 V
6S = 6 cells in series x 3.7 V = 22.2 V



Brushed DC Motor



Digi-Key Part Number	1597-1202-ND
Manufacturer	Seeed Technology Co., Ltd
Manufacturer Product Number	108990001
Supplier	Seeed Technology Co., Ltd
Description	STANDARD MOTOR 50000 RPM 3.7V

Previously mentioned characteristics are based off of ability to provide flight which we do not need:

- Higher RPM means a smaller size propeller can be used (3-4 inches)
- No need to add a second battery
- No need for Electronic Speed Control Unit.

Current assumption is that the struggle will be for space and power consumption. Brushless motors are significantly more stable but as mentioned they require the addition of at least one more battery and their lower RPM operation requires the use of larger propellers(5-7 inches). A trade-off will be conducted to determine which motor delivers the most suitable performance for our application.

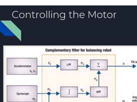
Brushless DC Motor Standard Chart

Frame Size	Prop Size	Motor Size	KV
150mm or smaller	3" or smaller	1306 or smaller	3000KV or higher
180mm	4"	1806	2600KV
210mm	5"	2204-2206	2300KV-2600KV
250mm	6"	2204-2208	2000KV-2300KV
350mm	7"	2208	1600KV
450mm	8", 9", 10"	2212 or larger	1000KV or lower

Brushless motors come standardized by motor size which has significantly simplified the search. In addition to the Brushed DC motors, Brushless motors in the 1306 and 1806 category have also been acquired and will be used to test variances in performance for our application.

- Motors in the 1306 category can still run on a 1s LiPo battery. While having less RPM their efficiency may be preferred if our robot is to spend a significant amount of time mapping and navigating the maze.

Controlling the Motor



https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2015/dc686_nn233_hz263/final_project_webpage_v2/dc686_nn233_hz263/index.html

The use of an IMU requires two different sets of data averaged together:

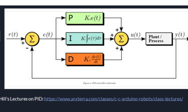
- Accelerometer can measure tilt reliably in the short term but not long term
- Gyroscope can reliably measure long term angular velocity but not reliable for the short term.
- Complexity is high and there is only on degree of freedom.
- Opted for using a sensor that can measure distance from the floor to the chassis and relay that information to the controller.

Using an ToF sensor to track distance from the floor.

-VL6180X

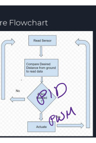
Operating voltage:	
Functional range	2.6 to 3.0 V
Optimum range(4)	2.7 to 2.9 V

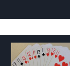
- Fast, accurate distance ranging
  - Measures absolute range from 0 to above 10 cm



Professor Hill's Lectures on PID: https://www.arxterra.com/classes/c-c-arduino-robots/class-lectures/

Software Flowchart





Name: Justin Chieng

Role: Card Reader Design Engineer

Objective: Designing and testing the hardware and software needed to detect, identify, and record data encoded on the markers/cards placed on the paper maze.

Previous Work:

With the maze project research assignment, I researched all the possible solutions that can be used for the card reader. After having gone through all possible solutions, I settled at six. The six solutions are the RFID tags, e-ink, camera, Infrared sensors, Color Sensors, and Barcode/QR code sensors.

Sensor Comparison:

L1.1 The robot shall distinguish between the different types of cards that can be interacted with and store the information for the Software to utilize as an input. The different types of cards will be sorted into 3 sections, monsters, weapons, and treasure. Moreover, these sections will also be sorted by their grade, ranging from 1 to 3.

L1.2 The robot shall scan the cards without requiring the robot to stop. This will allow the robot to maintain a competitive edge in accomplishing the mission objectives.

L1.3 The robot shall display the scanned barcode once it has successfully accounted for it.

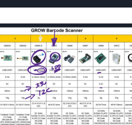
Conceptual Design Solution: Barcode/QR Code Scanner

My proposed conceptual design solution is a barcode/qr code sensor. In addition, the type of markers (QR and Barcode) will be easily discernible to the sensor. The markers will have three different barcode/qr codes and will have three different subsets depending on the grade of the marker.

Arduino MKR1000
(Reference)

Specifications

- WINC1500 low power 2.4GHz IEEE® 802.11 b/g/n Wi-Fi
- ECC508 CryptoAuthentication
- Li-Po charging circuit
- Digital I/O pins 8
- Flash Memory 256 KB
- Length 6.15 cm
- Width 2.5 cm
- Weight 32 grams
- Cost ($36.80)

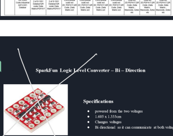


GM 65 1D 2D Barcode Reading Board QR Code Scanner Reader

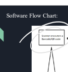
Specifications

- Dimensions 5.33 cm x 1.35 cm x 1.18 cm
- USB Interface
- Weight 7 grams
- Buzzer to remind of successful scanning
- Scanning angle: 34 Degrees
- Elevation Angle : 26 Degrees
- Reading Distance 2.5 - 25 cm
- Cost ($37.71)

GROW Barcode Scanner



Nextion NX3224T024 - Generic 2.4" TFT LCD Touch Display



Specifications

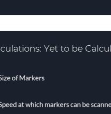
- LCD Touch Display (Liquid crystal display uses a backlight to light up crystals)
- One Serial Port
- Visual Area: 3.672 cm(L)×4.896 cm(W)
- 4M Flash memory for User Application Code and Data
- Cost ($16.40)

SparkFun Logic Level Converter – Bi – Direction

Specifications

- powered from the two voltages
- 1.605 x 1.333cm
- Changes voltages
- Bi directional so it can communicate at both voltages

Software Flow Chart:



Marker Flow Chart:

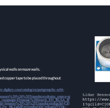
Calculations: Yet to be Calculated

- Size of Markers
- Speed at which markers can be scanned
- Angle of deviation of scanner

Sample Pseudo Code: PHP Script to Get Data from Barcode to Scanner

Source: https://create.arduino.cc/projecthub/sanruskmv/wireless-product-tracking-arduino-mkr-barcode-scanner-3708d9

Sample Pseudo Code: Sending Data to a server for storage



Source: https://create.arduino.cc/projecthub/sanruskmv/wireless-product-tracking-arduino-mkr-barcode-scanner-3708d9

Name: Kerin Bermudez

Role: Navigation Engineer (Hardware & Software)

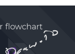
Objective: Map out and navigate maze to complete the game.

Research

First options:

1. Lidar Sensor
2. Radar Sensor
3. Sonar Sensor
   a. Restraint: Physical walls on maze walls.
4. Inductive Sensor
   a. Restraint: Need copper tape to be placed throughout maze.

Sonar Sensor https://www.digikey.com/catalog/en/partgroup/hc-sr04-ultrasonic-sonar-distance-sensors/82267



Lidar Sensor https://www.adafruit.com/product/4441?gclid=Cj0KCQiAvvKBBhCXARIsACTePW-NSMrJRyGPNj9411B8Dd1rggt_mHLJ_A06GUGx3j90_2mw29eTPTkaAvw7EALw_wcB

Sensor Solution - Decided to go with inductive sensors

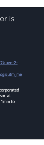
Sensor	Voltage Range:	Current Range:	Size:	Range:	cost:	Data Transfer:
LDC 1612	2.7 - 3.6 V	Typ 21 mA	2 ⅜ in x ¾ in	< 7.8 in	$14.95	I2C
MP 2017	Up to 30 V	N/A	1 in x .243 in	.15 -.5 in	$ 2.36	TBD
IC Station DIY	3 - 5 V	< 40 mA	3.4 in x 2.4 in	< 2.36 in	$ 2.09	TBD

Requirements: Sensor - copper tape would need to be in reading range
Size - Must be reasonably sized to fit on 3Dot Chassis Cost - reasonably priced (<$20)

LDC 1612
https://www.seeedstudio.com/Grove-2-Channel-Inductive-Sensor-LDC1612.html?utm_source=blog&utm_medium=blog

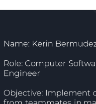
MP2017
http://www.farnell.com/datasheets/2791108.pdf

DIY
http://www.icstation.com/metal-detector-60mm-contact-sensor-electronic-kits-security-test-p-13558.html



Req: How much current could 3dot supply?

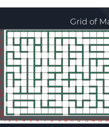
Sensor flowchart



Requirement

- Stay within bounds of 'hallway' of maze by implementing Inductance line following
- Be able to communicate to Game software about available actions at any given time

Given code for LDC 1612 determining if sensor is located



Source:

https://www.seeedstudio.com/Grove-2-Channel-Inductive-Sensor-LDC1612.html?utm_source=blog&utm_medium=blog

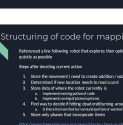
Also referenced a code that incorporated distance of a coin from the sensor at multiple ranges ranging from <1mm to roughly 20mm

Name: Kerin Bermudez

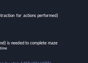
Role: Computer Software Engineer

Objective: Implement data from teammates in maze solving algorithm

Grid of Maze



Decided that going with maze data given to us would be easier not considering the exit points of the maze as added column and rows



Research: Movement throughout Maze

Decided that the whole robot exploring algorithm should be based upon where the robot is in the maze. For movement in maze needed to think about

- To determine position in maze created a grid. (given)
- For each point in grid need to determine possible actions based on square in grid.
- Determined list of possible actions to be
  - Go forward
    - Both motors on
  - Turn left
  - Turn Right
  - Turn Around
    - Need one motor going CW and other going CCW to turn
    - Amount of time in this action TBD for each respective action
- Need to consider points in grid with 2 or more possible actions

Still need to consider how to cue card reading sensor to read cards and how to implement data - Requirement

Other Requirement: algorithm for exploring maze with available actions while fulfilling roles of teammates as well.

Structuring of code for mapping out the maze

Referenced a line following robot that explores then optimizes the route to finish the maze as quickly as possible

Steps after deciding current action

1. Store the movement (need to create addition / subtraction for actions performed)
2. Determined if new location needs to read a card
3. Store data of where the robot currently is
   a. Implement moving portion of code
   b. Implement saving of picked up items
4. Find way to decide if hitting dead end(turning around) is needed to complete maze
   a. Is there item on that turn around portion or wasted time
5. Store only phases that incorporate items

https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/

Structure of code: Optimizing maze solving

From an Email from chris the idea of a tree traversal algorithm was touched upon briefly

Could maybe incorporate the tree traversal for lists of actions taken and discard branches

Req: The robot would be able to discard routes where a wall is placed during mapping phase if there is no accomplishments in that path.

Idea from video Alpha Beta pruning: used to evaluate and decrease number of branches in the tree traversal (https://www.youtube.com/watch?v=U4ogK0MIzqk)

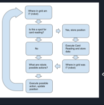
The final path would be found and executed based on only essential branches of tree traversal

Making motors go at maximum speed while maintaining balance and being able to take in card data in the optimized execution to create fastest time.

Still need to:

- Determine where data about the optimized movement will be stored in 3dot
- Determine how to cue card reading
- Determine how to implement alpha beta pruning since not a two player game

Flow Chart for position of maze



Requirements

- Be able to 'communicate' a cue possible so that the card reading algorithm could be issued.
- While executing possible actions need to ensure that balance is maintained at all time.
  - Would need another loop to ensure that throughout the entire exploring process balance is being maintained.