Autotrack

March 30th, 2017

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 405W Design Specification and UI Appendix for an Automated Cart Delivery System Capstone Project

Dear Dr. Rawicz:

The enclosed document, *Design Spec and UI Appendix for an Automated Cart Delivery System*, describe our Capstone project ideas for ENSC440. Our goal is to optimize consumers' shopping experience by delivering shopping bag from one destination to another. By designing an automated cart delivery system, our product will solve the problem of carrying several shopping bags.

The design specification document provides an overview of the mechanical, control system design that will be implemented to meet the initial requirements of the Deliverbot. This document will also provide an explanation of our design process, system specifications, and procedures used to design the product. Finally, this document will also include a test plan to ensure the correct functionality of the Deliverbot.

AutoTrack was founded by three dedicated senior electronics, systems, and computer engineering students from Simon Fraser University. The team members include Eason Tsai, Benjamin Tsai, and JackyTeng. If there are any questions or concerns about our proposal or project ideas, please do not hesitate to contact me at syt8@sfu.ca.

Sincerely,

Benjamin Tsai

AutoTrack Inc.

Design Specification and
UI Interface for
Automated Cart Delivery System

Project Team:

Eason Tsai
Benjamin Tsai
Jacky Teng

Contact Person:

Benjamin Tsai
syt8@sfu.ca

Submitted to:

Dr. Andrew Rawicz - ENSC 440
Steve Whitmore - ENSC 405W
School of Engineering Science
Simon Fraser University

Issue Date:

March 30, 2017

Table of Contents

List of Figures

List of Tables

Glossary

DC	Direct Current
PWM	Pulse Width Modulation
RPM	Rotations Per Minute
C/C++	Programing Language
OpenCV	Programing Language
Python	Programing Language

Design Specification

1. Introduction

Deliverbot is a shopping bag carrier that resembles a four-wheel automated cart. It is able to follow the preinstalled colored line on the floor and send packages to assigned destinations. The Deliverbut project proposed by AutoTrack will address the problem of manually carrying shopping bags in the mall. In addition, it can provide users with a more pleasant ‘hands-free’ experience during shopping. Under the auto mode, the Deliverbot will deliver bags to a designated location automatically. DeliverBot consists of two main systems. The first system is the control system, which includes a Raspberry Pi development board, eight distance sensors, a Raspberry Pi camera module, and a DC motor controller. The second system is the mechanical system, which includes two12V-24V motors, wheel transmission system, and a cart frame with the storage compartment. The design specifications for each system, as well for all components used, will be thoroughly explained in this document.

1.1 Scope

This document outlines the design of DeliverBot, which includes the design approach, design logic, proof of concept design and future modification in development. This document will also discuss the functional requirements that were previously established in the functional specification. Additionally, a test plan with a set of test cases is included for individual components testing as well as system integration to ensure proper functionality of the overall system.

1.2 Intended Audience

This design specification is intended to be used by all engineers of AutoTrack. Throughout the development of DeliverBot, the design team can consult this document as the main guideline for integrating and testing different components of the product. Test engineers can also refer to this document to ensure that all pre-specified functions and design requirements are met.

2. Control System Overview

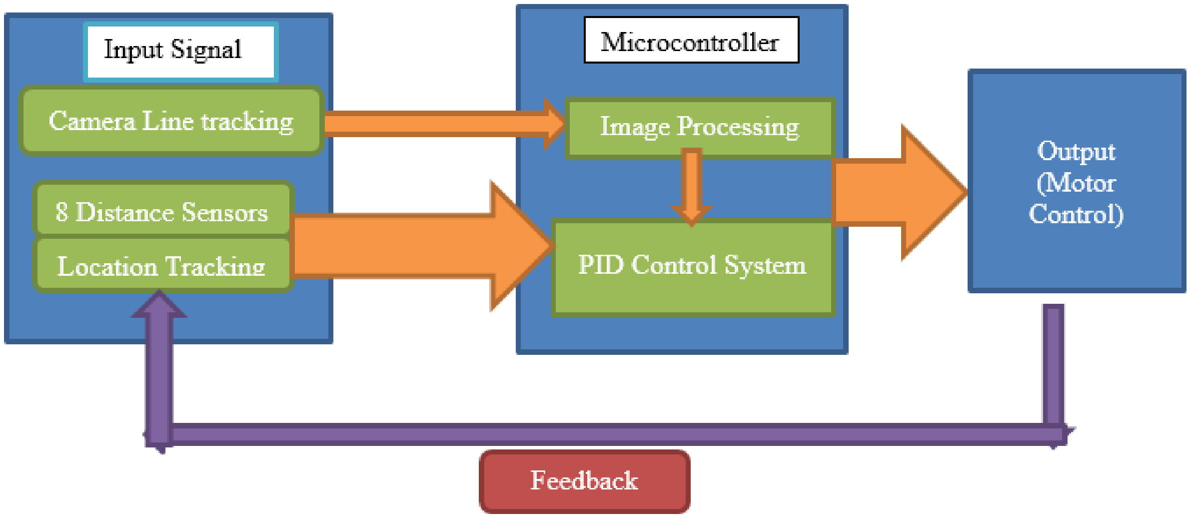

Figure 8 Block Diagram of Control System

The control system of DeliverBot consists of one Raspberry Pi microcontroller, eight distance sensors, a Raspberry Pi camera module, and a DC motor controller. To meet requirement [R47-ii], the microprocessor will have continuous power supply during the whole operation to ensure stability. The input signals from the camera, distances sensors, and location tracking from the view of the camera to the line are analyzed by the microprocessor. Moreover, the PID control feedback system will determine the angle of direction that Deliverbot has to adjust in order to follow the line. The system will track the location of DeliverBot again to calculate the error percentage for PID feedback control after each movement. An overall block diagram of the control system is shown in Figure 1.

2.1 Component Specification

2.1.1 Raspberry Pi B+ Development Board

To implement functions and features of DeliverBot, AutoTrack takes advantages of Raspberry Pi's strong computing power and convenience. Using Raspberry Pi allows engineers in AutoTrack to test the firmware and design concepts in a familiar environment during the proof of concept stage. On the other hand, design engineers in AutoTrack will produce a PCB board with all add-on sockets and circuits to reduce the cost of the final product. In Figure 2, it describes the pin assignment of the Raspberry Pi B+ development board. The GPIO pins on the board are used as input and output for communication with other components.

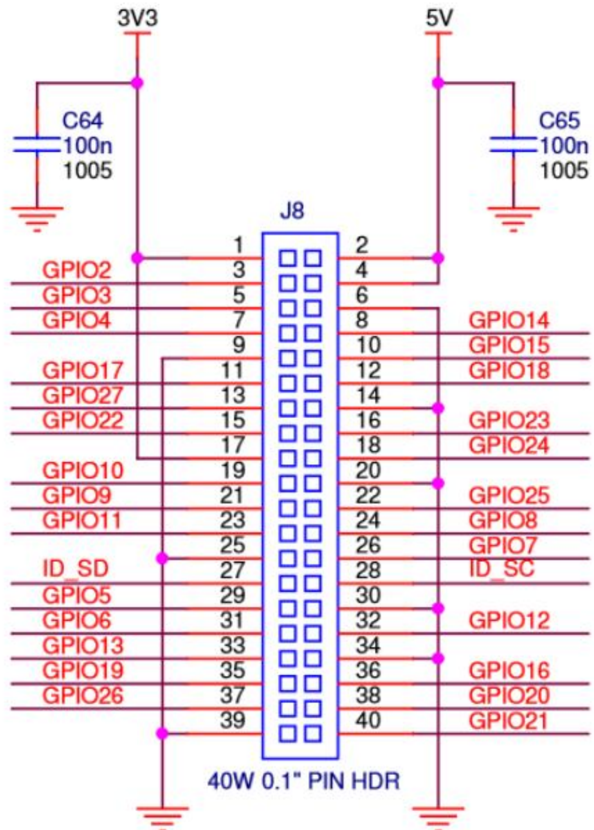

Figure 9 Raspberry Pi GPIO pin layout

2.1.2 Tower Pro Micro Servo SG90 (Proof of Concept)

The servo motor in Figure 3 is used to change directions of the wheels. The power provided from the servo is sufficient to perform desired functions in the proof of concept stage. The parameters show in Table 1 are used for writing the code that controls the servo motor in Raspberry Pi for proof of concept stage.

Figure 3: Tower Pro Micro Servo SG90 motor

Table 1: Electric Parameters of micro servo SG90

Stall Torque	**1.8 kgf * cm**
Operating voltage	**4.8 V**
Operating Speed	**0.1second/60 degree**
Max Bandwidth	**10 us**
Dimension	**22*11*31 mm**

2.1.3 L298N DC Motor Driver (Proof of Concept)

Figure 4: L298N DC motor driver

The DC motor driver allows input signals from DC motor to control the motor speed on wheels. The L298N model is used to control both motors for back wheels in the proof of concept stage. The parameters of the motor driver are shown in the following table 2.

Table 2: Electric Parameters of L298N DC Motor Driver

Driver Power Supply	**5V**
Power Output Voltage	**5V – 7V**
Max Power	**25 W**
Working Temperature	**25C – 130C**
Dimension	**60 * 54 * 7 mm**

2.1.4 Raspberry Pi Camera Module v2.

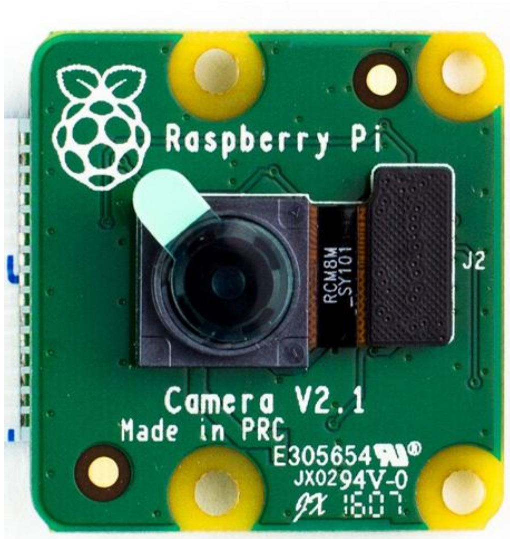

Figure 5 Raspberry Pi Camera Module v2.

Raspberry Pi camera module v2 is used to achieve the line tracking function. It has a GPIO input to receive signals. The detailed specification of the camera is included in table 3.

Table 3: Electric Parameters of Camera Module V2.

Resolution	**8 Megapixels**
Sensor resolution	**3280 x 2464 pixels**
Horizontal field of view	**62.2 degree**
Vertical field of view	**48.8 degree**
Focal ratio	**2.0**
Focal length	**3.04 mm**

2.1.5 Ultrasonic Sensors (Final Product)

To meet the functional requirements [R45-ii], and [R51-ii], regarding the sensor requirements, AutoTrack chose to use the HC-SR04 ultrasonic sensors. The electric parameters of sensors shown in Table 4, will be sufficient to perform the safety functions that the Deliverbot requires.

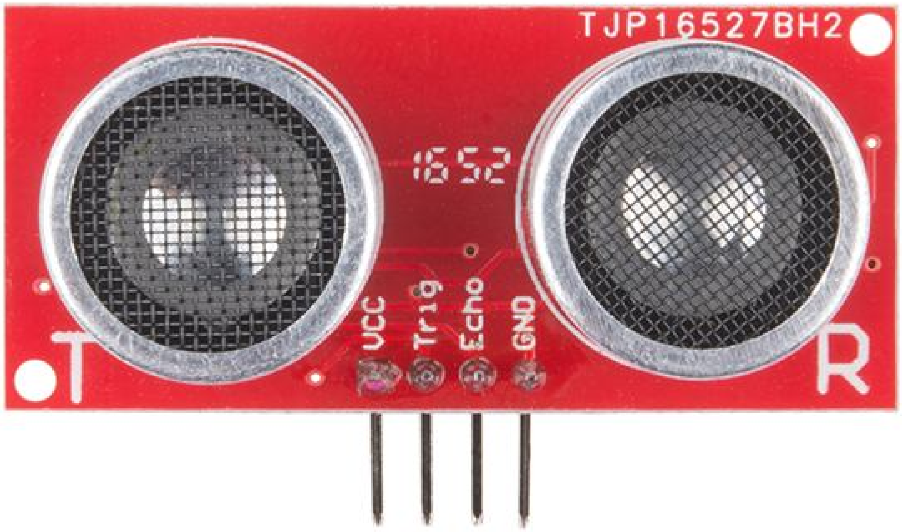

Figure 6 HC-SR04 Ultrasonic Sensor

Table 4: Electric Parameters of HC-SR04

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Since the input pin for the Raspberry Pi B+ is rated at 3.3V [11], we may need to use a voltage divider to convert the 5V output (Echo pin) from the ultrasonic sensor to a workable 3.3V. The voltage divider circuit will be confirmed when the engineers of AutoTrack do a unit test on the HC-SR04 Ultrasonic Sensor.

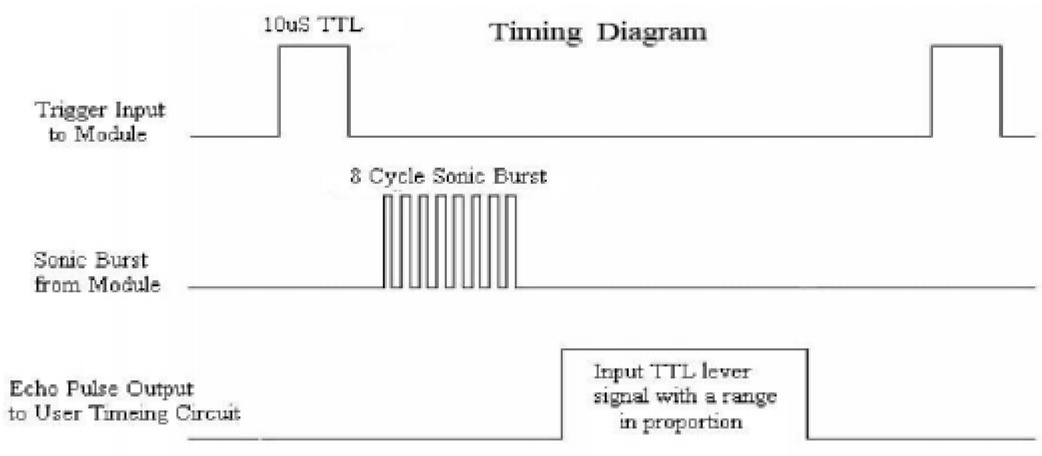

Figure 7: HC-SR04 Operation Timing Diagram

For the purposes of distance sensing, the ultrasonic sensors can be used in the following way. The Trigger pin of the sensor receives a 10us pulse to start the ranging and then the module will send out an 8-cycle burst of ultrasound at 40kHz and raise its echo. The range can be calculated by using the time interval between sending the trigger signal and receiving the echo signal. The distance between the cart and the detected obstacle equals to the time between sending trigger signal and receiving echo signal divided by 2 then multiplied by the speed of sound (343 m/s).

2.1.6 Sabertooth Dual 12A 6V-24V Motor Driver (Final Product)

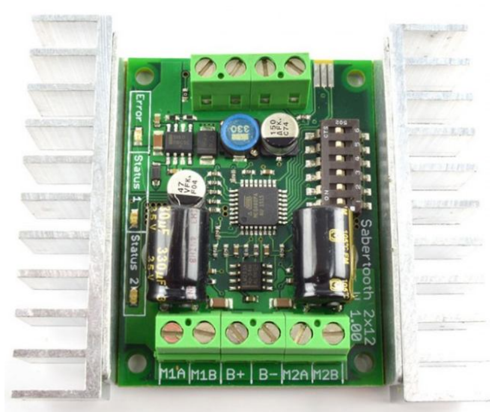

Figure 8 Sabertooth 2x12 Motor Driver

The two motors on Deliverbot are controlled by the Sabertooth 2x12A DC motor driver. Each motor is controlled by signal input S1 and S2 respectively. For forward motor operation, S1 and S2 inputs are set to above 2.5V with 5V at highest forward RPM.

2.2 Tracking system algorithm

Our line tracking feature utilizes the Raspberry Pi camera module v2 to capture the path direction in front of DeliverBot as specified in [R46-i]. The image is captured by the camera module and analyzed by OpenCV program. In addition, the information from the image will be sent to a motor control system, which will make an adjustment of DeliverBot's direction. To increase the efficiency of image processing time, we decided to use OpenCV programming language rather than C++ or C programming language.

To analyze the image and identify the line, we need to find the region of interest from the image. Figure 9 is an example of the camera view of the line. However, for DeliverBot to follow the line precisely, we only need to consider the area enclosed by the blue rectangle. The level of the region can vary, but the turning and timing of the motor control should also be taken into the calculation.

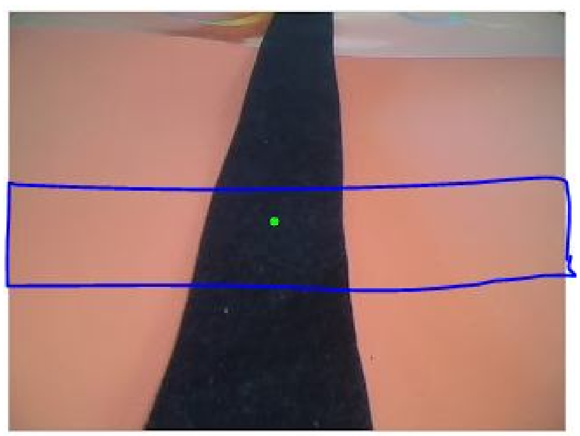

Figure 9 Camera View of Tracking Line

By reducing the image with morphological operations and threshold the image, we turn the image to grayscale as demonstrated in Figure 10.

Figure 10 Image Processing Steps of Line Tracking

We can identify the contour of the line (the left arc) from reducing the image to grayscale image. We can calculate the position of the center that will be used to make DeliverBot turn certain degrees from this contour.

2.3 PID control System

PID control system is implemented in our design to make sure DeliverBot can follow the colored path and stay within the range of the camera view of the line. To avoid the robot overturning and losing track of the colored line, the PID system will be used in the Deliverbot tracking system. It is because PID system considers the overshoot percentage, settling time, and final position to make sure DeliverBot is able to correct the error when it goes off the track.

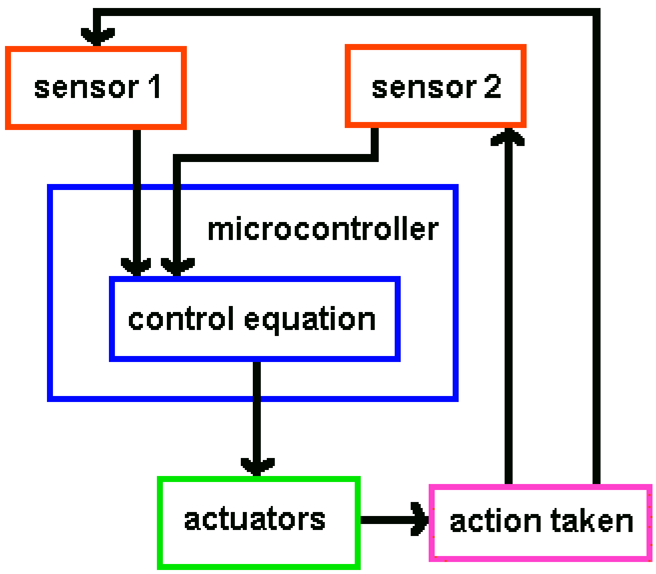

Figure 11 PID Control System Block Diagram

Figure 11 shows the block diagram of the PID control system on Deliverbot. Sensor 1 is the line tracking system which DeliverBot has to follow. On the other hand, sensor 2 considers the current position of the robot and the location of the line. By incorporating the two pieces of information into the PID control equation, the actuator adjusts the required angle that the wheels have to turn in order to pull Deliverbot back on track. By using two sensors for PID control, Deliverbot can follow the line closely and will not go off the track.

2.4 Motor Control System

The micro servo is connected to the chassis of the front wheels and it is used to shift the directions of the wheels. The micro servo is preferred in the proof of concept stage since the weights of the wheels are not heavy and the torque generated from the servo is sufficient. On the other hand, the weight of the chassis and wheels in the prototype will be considerably heavier compared to that in the proof of concept stage, we must attach a motor to every wheel and adjust the speed of the motor on each wheel to perform the task of turning in the prototype of DeliverBot.

The micro servo allows the angle adjustment based on the pulse width from PWM signal in GPIO pin. The servo will turn 180 degrees when the pulse width equals to 1 millisecond and it will turn 90 degrees when the pulse width is 1.5 milliseconds, and 0 degrees corresponds to 2 milliseconds. Therefore, we can determine a linear relationship between the angle of turning and the duty cycle.

The period of the micro servo can be calculated by the following equation:

$$1/50Hz = 0.02\ seconds$$

In addition, we can adjust the pulse width by changing the duty cycle from GPIO command as the equation shown below:

$$DutyCycle = PulseWidth/Period$$

Overall, to obtain the 0-degree position, the pulse width should be 0.001 seconds, and therefore the duty cycle is equal to 0.001/0.02 = 5%. Through the same calculation, we found that, for the wheels to turn 180 degrees and 90 degrees, the duty cycle should be 10% and 7.5% respectively. However, there is an uncertainty on each servo motor. To obtain a 0-degree position for the servo, the duty cycle value has to be 2.65%, and the duty cycle has to be 10.5% for 180 degrees. With this adjustment in mind, we can determine the linear relationship in the plot below.

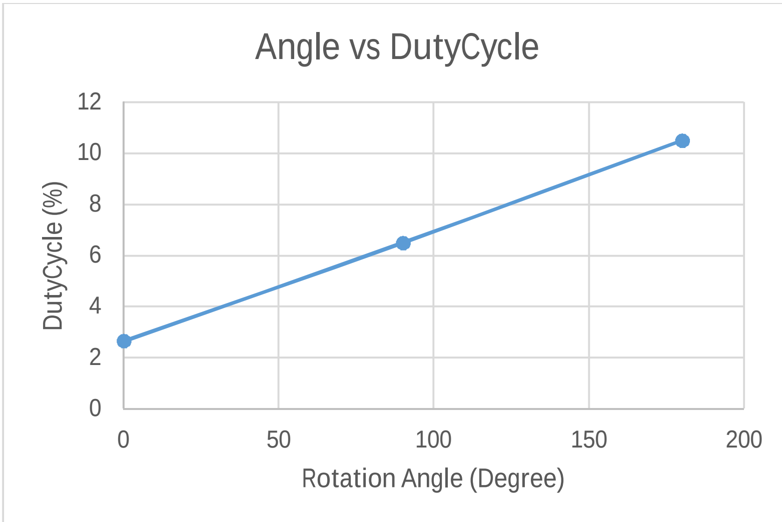

Figure 10 Angle vs DutyCycle Plot

The duty cycle must be 6.5% at 90 degrees by calculating the slope of the plot.

Since the robot can only turn a maximum 45 degrees from the neutral position (90 degrees), the rotating angle will stay within 45 to 135 degree due to the robotic performance and hardware limitation. We set the speed of the motor to a constant value since the direction control is at the front wheel in the proof-of-concept stage. Therefore, the back wheels only output a constant speed or complete stop.

The servo control will be removed due to the cost consideration in our prototype design. Since we only require the wheel to turn in low speed and the price of obtaining a servo motor controller and servo motor are too high, we decide to use another algorithm to control the turning of Deliverbot. To implement the turning algorithm with only two DC motors, the speed outputted from the left motor and right motor must be different. As shown in figure 13, to turn DeliverBot towards the left, the right wheels have to spin faster than the left pairs; and to turn right, the left wheels need to spin faster than the right wheels. However, at this point of our project, we do not have the accurate speed and angle relationship since we are implementing our motor design with servo control. We will calculate the accurate speed and angle relationship when we build the prototype of Deliverbot.

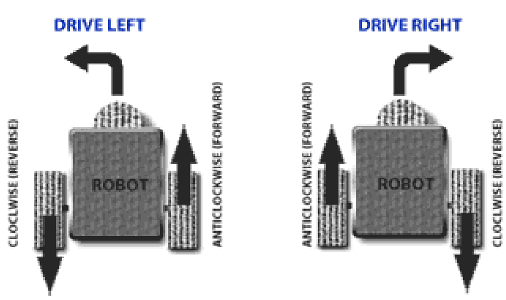

Figure 13 Wheel Control Theory

3. Mechanical System Overview

The mechanical system grants Deliverbot the ability to move by motor power. This section of the document will outline the design of the mechanical system. The specifications of mechanical components, transmission system, and design in mounting and wiring will be discussed.

The mechanical system section will mainly focus on the prototype and final product stage. In addition, the mechanical system in the proof-of-concept (POC) stage will be briefly described. The mechanical system in the POC stage of Deliverbot is concentrated on the two DC motors, one tower pro micro servo SG90 motor, two driven wheels, two 3.7V rechargeable battery, and two active wheels. The two DC motors control the two wheels and two rechargeable battery are used to provide the power to the toy car. In addition, the servo motor is used to adjust the turning angle of the active wheels.

The mechanical system in the prototype and the final product stage will concentrate on the cart frame, transmission system, battery, and locking shopping cart wheel.

3.1 Component Specifications

3.1.1 M27-150-P Motors

Deliverbot uses two 24V, 150W rated Electric DC motor for movement. Motor's performance chart is shown in Figure 14.

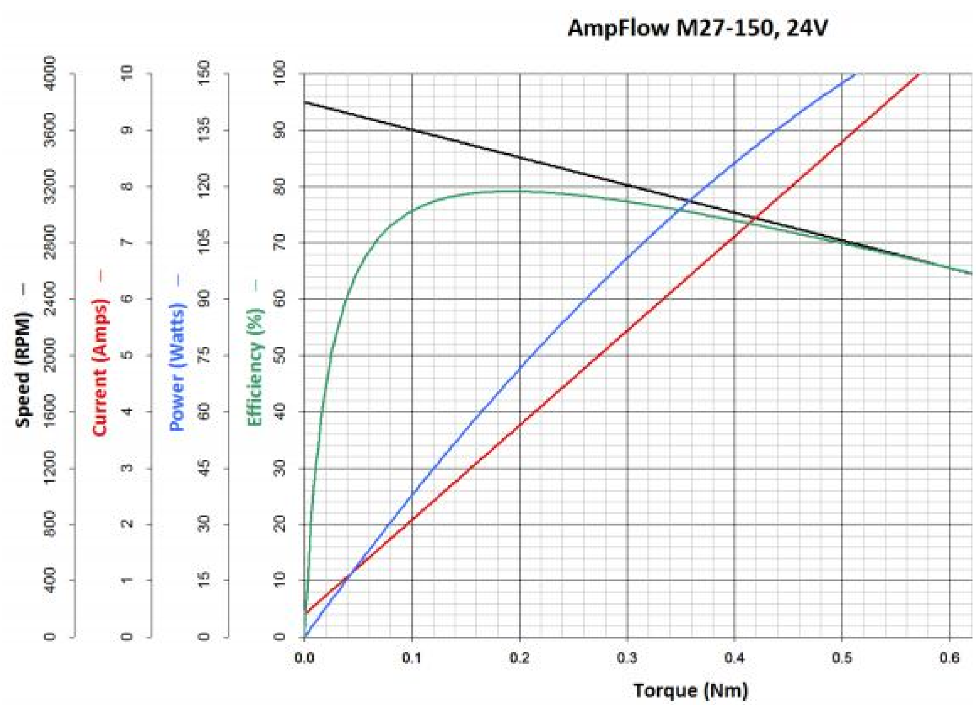

Figure 11 Performance Chart of the M27-150-P Motor

The following table shows the detail parameter of M27-150-P motor.

Watts- Continuous	**150**
Diameter (inches)	**2.7**
Length (inches)	**3.9**
Efficiency(%)	**79%**
Voltage(V)	**24**
No- Load RPM@ 24V	**3800**
Pounds	**3.0**

Table 5: M27-150-P Motor Specifications

3.1.2 12V Battery

According to [R43-i] and [R44-i], the Deliverbot will use two 12V, 5Ah lead-acid batteries to power the electrical system. The two batteries are connected in series to provide the 24V power required by the motors.

Figure 12: Rechargeable Sealed Lead Acid (VRLA) Battery

The battery requires a continuous current with a minimum voltage of 14.40 V for recharging. The recharge voltage specification is shown in Figure 16.

Cyclic Application Recharge Voltage (77°F / 25°C)			
Minimum	**Recommended**	Maximum	
14.40	14.55	14.70	Volts D.C.
2.40	2.425	2.45	Per Cell
Temperature Coefficient: -2.8mV / °F / Cell (- 5mV / °C / Cell)			
Standby Application Recharge Voltage (77°F / 25°C)			
Minimum	**Recommended**	Maximum	
13.50	13.65	13.80	Volts D.C.
2.25	2.275	2.30	Per Cell
Temperature Coefficient: -1.7mV / °F / Cell (- 3mV / °C / Cell)			

Figure 16: Battery Recharging Specification

3.1.3 Cart Frame

The cart frame of the Deliverbot prototype is based on 01746 Double Tier Metal Deck Wagon in Figure 17. The reason that the original cart frame is used on the Deliverbot prototype design is because the cart frame is similar to the shopping cart and the back wheels are fixed on the chassis.

Figure 17: Original Cart Frame

3.2 Transmission system

Deliverbot uses a V-Belt pulley system as its transmission system. The transmission ratio of this system is 1:5, which reduces the speed and increases the torque of Deliverbot, as shown in Figure 18.

Another transmission system being considered has a transmission ratio of 1:3, shown in Figure 19. Design engineer of Deliverbot will decide which transmission ratio will be used on the prototype of the Deliverbot after testing each transmission system.

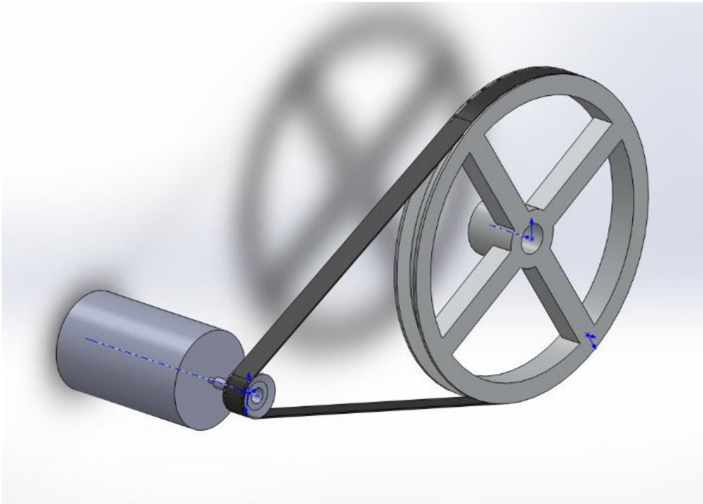

Figure 13: V-Belt transmission system ratio 1:5

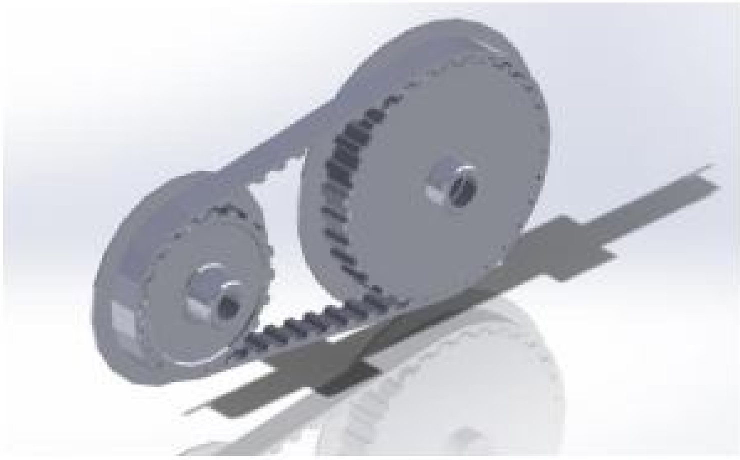

Figure 14: V-Belt transmission system ratio 1:3

3.3 Security system

The smart wheel is an anti-theft wheel on Deliverbot, which is used to prevent Deliverbot from being stolen. The company that produces the smart wheel does not specify the range that is considered abnormal, and thus we do not know at what distance between the cart and its designated track the alarm will be triggered. The company that makes the smart wheel does not provide any product specification details. Therefore, the designers in AutoTrack will try to contact the manufacturer of the smart wheel and obtain more information prior to building the prototype of Deliverbot.

Figure 15 Locking shopping cart wheel

The secure box will be used to contain shopping bags in Deliverbot. There is a key-lock system, which is used for security purpose. The box is mad by aluminum, which has a lighter weight than steel. Therefore, designers are going to use the box shown in Figure 21 to build the prototype of Deliverbot.

Figure 21 Secure box for carrying shopping bags

3.4 Wiring and Mounting

For safety purpose, all components and wires should be mounted onto the Deliverbot. The bottom of the Deliverbot is covered by a wooden board with holes for screws. All the components and wires will be contained in the wooden board. The designers of AutoTrack will focus on wiring and mounting of mechanical components when building the prototype of Deliverbot.

4. Test Plan

The following test plan is created to ensure that our final product meets the requirements of our function specifications. Engineers in AutoTrack are going to test Deliverbot with various type of testing.

4.1 Unit Testing

4.1.1 Raspberry Pi Board

Raspberry Pi Functionality Testing

Testing Method: Run a sample code on the board and connect a multimeter to the analog pins in order to measure the pin values.

Expecting Outcome: The code should be able to run correctly. The voltage measured from the GPIO output pins should be between 0V and 3.3V.

4.1.2 Motor Control

Motor Controller Functionality Testing

Testing Method: Connect the micro servo and motor driver to Raspberry Pi. Then run a simple code to change the duty cycle of PWM for micro servo and turn on the motor driver by outputting GPIO pin to 1.

Expecting Outcome: The micro servo should be able to shift at a certain angle depending on the value of duty cycle, and the motor should turn on when GPIO pin is 1.

4.1.3 Ultrasonic Sensors

Ultrasonic Sensors accuracy and detection range testing:

Testing Method: Move an object in front of the sensor between the ranges of 2m to 4m.

Expecting Outcome: The sensor should have a reading between 52cm to 4m with an error within the range of +/- 5cm from the datasheet of sensors.

Interference testing between sensors:

Testing Method: Put two sensors besides each other and place an object far from one of the sensors and another object close to the other sensor.

Expecting Outcome: Both sensors should give correct readings and the ultrasonic waves from the sensors should not interfere with one and another.

4.1.4 DC Motor

Motor power testing:

Testing Method: Apply a load weighing 30kg on the cart and attempt to move at a constant speed.

Expecting Outcome: As required mechanical specification, the cart should manage to move at a constant speed of 0.6m/s with 30kg of weight on it.

Breaking testing:

Testing Method: Move the cart at speed of 0.6m/s and suddenly turn off the power to the motor.

Expecting Outcome: Based on the requirement of [R27-ii] in the functional specification, the cart should reach the full stop less than 0.5 seconds.

4.1.5 Battery

Battery life testing method: Connect the full charged battery to the final integrated system and let the system run.

Expecting Outcome: The system should run for than 5 hours without the need of external power as required by [R8-ii] [2].

Battery charging speed testing method: Charge the 12V 5Ah battery with 13.65V and 2.275A.

Expecting outcome: Based on the requirement of [R7-ii] in the functional specification, the battery should be full charge in 5 hours.

4.2 Integration System Testing

4.2.1 Ling Following System

Line Following System Testing: Straight line

Testing Method: Place the module at one point where there are clearly traced straight line on the floor, and make sure the camera view includes the line on the floor

Expecting Outcome: As required in [R46-ii] and [R49-ii], DeliverBot should be able to follow the path traced on the floor traveling within the maximum speed of 10km/hr.

Line Following System: Angle turning

Testing Method: Place the module at one point where the traced line is about to turn to an angle, and make sure the traced line is within the camera view.

Expecting Outcome: The module should recognize the turning angle and adjust the motor to perform turning without losing the track as required in [R46-ii]. If DeliverBot loses the line, it should stop at the point where it is lost and send a signal to the user as required in [R52-ii].

5. Conclusion

This document provides the design solutions to the requirements and functional specifications and addresses all the technical aspects of each solution.

In the component specification section and system overview section, the document provides a brief description of how all the components are put together to establish a complete product.

Within the system, software and hardware requirements sections further explain how each feature and component is designed to work according to the electrical, physical, functional and performance requirements.

The test plan will include the steps to ensure the integrity of the device and will keep track of whether the requirements are met or if there are needs for improvement. Overall, the document establishes goals for the development of the product to its best potential.

6. Reference

[1] Raspberry Pi Pinout, PiPinout [Online] Available: https://pinout.xyz/#

[2] Controlling a servo, (June 2, 2015) Technology Tuturials [Online] Available: http://www.toptechboy.com/raspberry-pi/raspberry-pi-lesson-28-controlling-a-servo-on-raspberry-pi-with-python/

[3] PID Control, Society of Robots [Online] Available: http://www.societyofrobots.com/programming_PID.shtml

[4] Signs Reading with OpenCV (Feb 10, 2014), Electronics and Robotics [Online] Available: http://roboticssamy.blogspot.ca/

[5] Ampflow. M27-150-P [Online]. http://www.ampflow.com/M27- 150_Chart.gif

[6] Infinity Battery. IT5-12 RECHARGEABLE SEALED LEAD ACID (VRLA) BATTERY [Online]. Available:http://www.infinitybattery.com/files/I242-IT5-12_0901.pdf

[7] Millside Industries Inc. Double Tier Metal Deck Wagon [Online]. http://millside.ca/details/50/01746

[8] Gatekeeper systems [Online]. http://www.gatekeepersystems.com/sup_cc_cc_smartwheel_maintenance.php

[9] Miscellaneous performance box [Online]. http://www.machine.co.nz/miscellaneous-WD/tool-box-30-performance-site-box-32955

[10] Sparkfun HC-SR04 Ultrasonic Sensor [Online] https://www.sparkfun.com/products/13959

[11] Adafruit Industry.Introducing the Raspberry Pi Model B+ [Online]. https://learn.adafruit.com/downloads/pdf/introducing-theraspberry-pi-model-b-plus-plus-differences-vs-model-b.pdf

[12] Dimension Engineering Sabertooth 2x12 User's Guide [Online]. http://www.robotshop.com/media/files/pdf/user-guide-sabertooth-2-12.pdf

User Interface Design Appendix

1. Introduction

The goal of the user interface design is to make the user's interaction with the system as simple and efficient as possible. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself, in terms of accomplishing user goals [1].

Our Product, Deliverbot is a shopping bag carrier that resembles a four-wheel automated cart. It is able to follow the costumed color line on the floor and send packages to assigned destinations. In this document, we will use the User Interface Design to analysis the Deliverbot.

1.1 Scope

The scope of this user interface design specifies the details of analysis through both user analysis and technical analysis. Additionally, this document also includes the test plan from analytical usability to empirical usability. For the analytical usability testing, we careful analysis the requirements of the UI in terms of meeting user needs; while for the empirical usability testing, we will ask some potential users of the system/device to try using it.

1.2 Intended Audience

This user interface design is intended for all staffs in AutoTrack Inc. The develop team can use this document as a guideline for designing and improving Deliverbot. The test team can use this document as a test plan for unit, integrated, and system tests.

2. User Analysis

Our product, Deliverbot, is designed for future shopping malls where customers can leave the purchased goods at the stores and have them sent to a designated location. To ensure the security of the purchased goods during travel, the cart is designed to be covered with stainless steel and a secure lock. A key will be required to access the products inside the cart. Therefore, users need to keep the key safe and know how to use the key.

In addition, users can choose different operating modes when they use the Deliverbot. There are two operating modes in Deliverbot, Auto, and Manual. For Auto model, the cart will deliver items from stores to pick up locations automatically with its dynamic tracking systems. To ensure safe and efficient operation, a user manual will be displayed on the cart user interface so that the users can fully understand how to use the cart properly under the auto mode. On the other hand, if the power system of the cart is off, it can be used as a normal shopping cart and the user needs to push the cart to mobilize it.

When under the auto mode, the Deliverbot will allow the users to choose a pickup location in the mall for their purchased items to be delivered. If the user is not familiar with the layout of the mall, the Deliverbot is also equipped to display a map for the user.

3. Technical Analysis

The seven-stage model of the action cycle can be a valuable design tool, for it provides a basic checklist of questions to ask. In general, each stage of action requires its own special design strategies and, in turn, provides its own opportunity for disaster [2]. Figure 22 summarizes the questions:

1. What do I want to accomplish?

2. What are the alternative action sequences?

3. What action can I do now?

4. How do I do it?

5. What happened?

6. What does it mean?

7. Is this okay? Have I accomplished my goal?

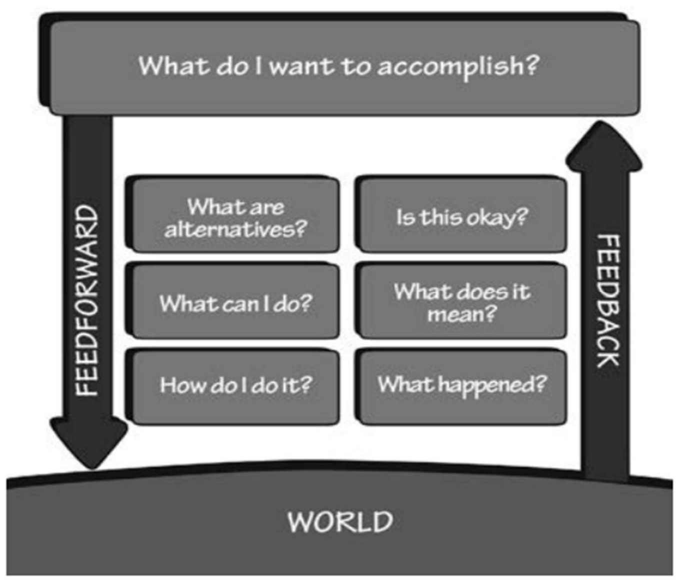

Figure 22: The Seven Stages of Action as Design Aids

The insights from the seven stages of action lead us to seven fundamental principles of design, and we use those principles for our Deliverbot:

i) **Discoverability**: It is possible to determine what actions are possible and the current state of the device.

   a) Moving: There are two states during cart moving; during the auto mode, the cart will move automatically. During the manual mode, the cart operates as a normal shopping cart and the user pushes the cart to move.
   b) Stop: There are three states during which the cart will stop moving automatically, including when the cart detected obstacles when the cart arrives the destination, and when the cart is operating under the manual mode.

ii) **Affordances**: Affordance is a Relationship. Affordances are the possible interactions between people and the environment. Some are perceivable, others are not.

   a) The handle provides the affordance for the user to mobilize the cart under manual mode
   b) The locker compartment provides the affordance for secure storage of purchased items
   c) The UI display screen on the cart allows interaction between the user and the system

iii) **Signifiers**: Signifier refers to any mark or sound, any perceivable indicator that communicates appropriate behavior to a person

   a) Different colored tapes on the ground indicate different routes for the cart to follow
   b) A safety sign on the locker reminds user to secure the storage compartment before leaving the cart
   c) An alarm will be triggered when the cart is in an unexpected condition, such as when someone tries to steal the cart

iv) **Mapping**: Relationship between the elements of two sets of things

   a) Steering a car: rotating clockwise causes the car to turn right and rotating counterclockwise causes the car to turn left
   b) Different operating mode: when in auto mode, the cart will deliver items from the stores to the pickup locations automatically; while in manual mode, the power system of the cart is off and it can be used as a normal shopping cart
   c) The camera on the cart will keep searching the designated colored tapes for the cart to follow when the cart operates in auto mode
   d) When the sensors on the cart detect any obstacles, the cart will stop to prevent crashing into the obstacles

v) **Constraints**: Constraints are powerful clues limiting the set of possible actions

a) The speed of the cart is around 5-10 km/h for safety purposes
b) The locked compartment on the cart can only be opened with designated key

vi) **Feedback**: Some way of letting users know that the system is working on the request

a) Steering a car: the cart turns right or left
b) Different operating modes: when in auto mode, the cart will automatically move without user pushing; while in manual mode, the cart will stop and wait for the user to push.
c) The cart will stop when obstacles are detected on its designated path

vii) **Conceptual model**: A Conceptual Model is an explanation, usually highly simplified of how something works.

a) A user manual will be on the cart display for the user to read to fully understand how to use the cart
b) A map section will be on the display in case of users are not familiar with the mall

4. Engineering Standards

Engineering standards apply to the proposed user interface. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself [1]. Therefore, graphic design and the choice of choosing correct word printed on the device is important to support its usability. In the design progress, we want to integrate the technical functionality and visual elements to build a device that is easy to operate.

There are 4 principles to follow:

- *The simplicity principle*: The design should make simple, common tasks easy, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures [3].
- *The visibility principle*: The design should make all needed options and materials for a given task visible without distracting the user with extraneous or redundant information. Good designs don't overwhelm users with alternatives or unneeded information [3].
- *The feedback principle*: The design should keep users informed of actions or interpretations, changes of state or condition, and errors or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users [3].

Autotrack

- *The reuse principle*: The design should reuse internal and external components and behaviors, maintaining consistency with purpose rather than merely arbitrary consistency, thus reducing the need for users to rethink and remember [3].

Engineering standards:

1) IEC 102290 (Graphic Standard)

2) IEC 102287 (Word Choice)

3) IEC 129090 (Guidance on Usability)

IEC Stand for (IEEE International Committee, n.d)

5. Analytical Usability Testing

For the analytical usability testing, we proposed several tests to be performed after the prototype is completed:

a) Test the cart on different routes to see if the cart will follow the taps on the ground.
b) Test the cart on different colored tapes to see if the cart will find and follow the desired color.
c) Test the cart at the intersection of different colored tapes to see if the cart can find the correct way to go.
d) Test in different environments to see if the camera will detect the right color regardless of interference such as reflective surface, shadows, etc.
e) Test the cart to see if it will stop when encountering obstacles ahead. The cart's response and the time it takes to stop will also be assessed.
f) Test the locker on the cart to see if it is safe to keep user's products. Difficulty to break into the locked compartment will be assessed.
g) Test the cart speed to make sure it is traveling maximum 10km/hr in a straight line.

6. Empirical Usability Testing

For the empirical usability testing, we plan to find a small shopping mall and contact the administrators to ask for any pieces of advice on our Deliverbot system. We can also conduct a survey to better assess the needs and specific requirements of the customers. To identify the population that is most likely to use our product, we will ask people of different age groups to gauge their interest. This process will start in next semester when the prototype of Deliverbot is built.

Some potential errors and safety issues may occur in the design, and we propose the solutions below:

a) We will set up Deliverbot to move at a constant low speed to prevent the cart from causing any damage to the people and objects around it.
b) When obstacles are detected, the cart should stop and continuously detect if the obstacles are still present. It will move only if all obstacles are cleared.
c) When someone tries to steal the cart or to deviate the cart from its designated path, an alarm will be triggered
d) When the cart can't find the desired colored tape on the ground to follow, it will stop and send error notifications and current location to the administrators.
e) When the cart is in manual mode, the firmware should not send any output signal to the DC motors or lock the wheels.

7. Conclusion

Currently, Deliverbot is under development to prove that our concepts are achievable. Our goal is to integrate automation into shopping malls so that customers are not required to carry bags of products during their shopping trip and can enjoy a more pleasant shopping experience. In this semester, we will use a toy car to prove that our line-tracking system is feasible, and the car is shown below:

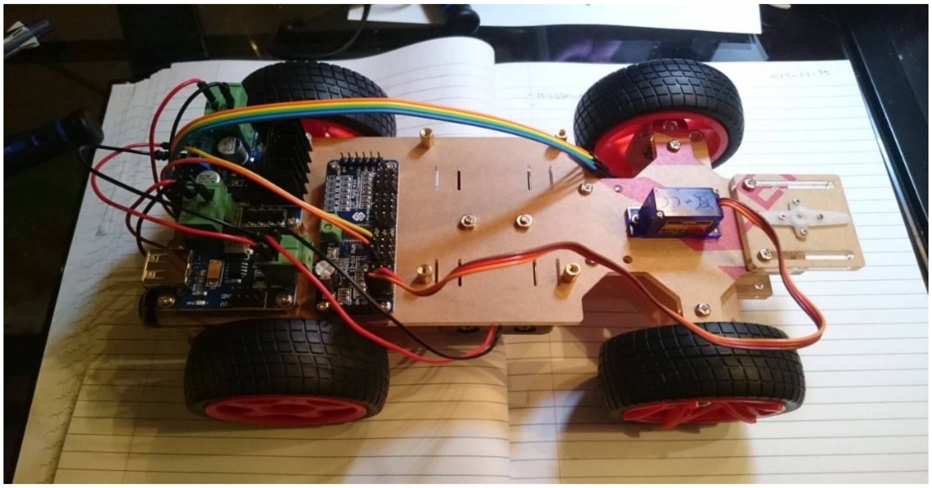

Figure 23: Toy car for proof of concept stage

As of now, we have finished assembling the car and we are currently working on the raspberry pi to control the camera and the motor. The conceptual prototype including all basic functions and will be developed by April 4, 2017.

Autotrack

8. Reference

[1] Norman, D. A. (2002). "Emotion & Design: Attractive things work better". *Interactions Magazine, ix (4)*. pp. 36–42. Retrieved 26 March 2017.

[2] Norman, D. A. (2013). *The design of everyday things*. pp. 60–73. Retrieved 26 March 2017. London: MIT Press.

[3] Principles of user interface design. (2016, December 30). Retrieved March 26, 2017, from https://en.wikipedia.org/wiki/Principles_of_user_interface_design