Trinity University
Digital Commons @ Trinity

Engineering Senior Design Reports | Engineering Science Department

4-30-2013

Formal Reports: Robotic Mouse Group

K. Gaughan

C. Payne

K. Saiz

D. Villamizar

Follow this and additional works at: https://digitalcommons.trinity.edu/engine_designreports

Repository Citation

Gaughan, K.; Payne, C.; Saiz, K.; and Villamizar, D., "Formal Reports: Robotic Mouse Group" (2013). *Engineering Senior Design Reports*. 92.
https://digitalcommons.trinity.edu/engine_designreports/92

This Restricted Campus Only is brought to you for free and open access by the Engineering Science Department at Digital Commons @ Trinity. It has been accepted for inclusion in Engineering Senior Design Reports by an authorized administrator of Digital Commons @ Trinity. For more information, please contact jcostanz@trinity.edu.

TRINITY UNIVERSITY

Formal Reports

ENGR-4381

4/30/2013

Robotic Mouse Group

K. Gaughan, C. Payne, K. Saiz, D. Villamizar
Dr. Nickels, Dr. Uddin Advisors

The purpose of this report is to describe the design process involved in creating an autonomous robotic mouse that is able to search through a maze, locate a doll, and transport the doll to the beginning of the maze. This project will utilize multiple engineering disciplines studied at Trinity University including: computer programming, electrical and mechanical engineering. In the design process, five prototypes will be made, each building and improving on the previous through testing and implementation. Ultimately, the robot must adhere to the predetermined design criteria and stay within the budget while performing the desired function for the project to be a success.

Executive Summary

The purpose of this project is to design an autonomous robotic mouse that is able to navigate through a maze, locate a doll, and transport the doll to the beginning of the maze. This opportunity will provide students an application in everything they have learned in their electrical, mechanical, and computer engineering courses. The concept for this project was originally taken from the IEEE (Institute of Electrical and Electronics Engineering) robotic mouse competition held annually in Japan, which is a competition in how quickly a robotic mouse can solve a maze. We decided to add a realistic component to the design and simulate a "search and rescue" mission, hence including the need to "rescue" a doll and return it to the start of the maze. This project will also lay a foundation for future Trinity engineers who wish to compete in IEEE Micromouse Competitions or in developing similar projects.

The Micromouse itself is a three-tiered, circular robot (about 10 centimeters to a side) with two wheels on both sides and a single ball bearing located towards the back of the body which provides balance for the robot. The chassis is made of 1/8th inch Plexiglas and is cut into two tiered sheets in order to provide a body for the robot. The major hardware components of our design include sensors, servo motors, DC motors, encoders, wheels, a microcontroller, chassis, battery and a gripper. Three sensors are attached to the front of the robot in order to help locate the walls of the maze and orient the robot. The DC motors are tucked within the body of the robot and are used to power the wheels. The battery is stored within the body of the mouse and is positioned by the Plexiglas columns connecting the three tiers of the chassis. A project control board (PCB) is located in the middle tier and contains all of the electrical components that are necessary to complete our design. Encoders are located on either side of the middle tier and are also used to help the robot determine how far it has travelled in order to navigate the maze. The gripper will be attached to the top tier of the Micromouse and will rotate downwards to pick up the doll when it is within a certain range.

Four prototypes have been developed in order to reach our final design. A wall follower was developed as the first prototype and worked properly. The second prototype failed because the hand soldered board was causing issues when all the components were connected to each other. Prototype #3 included a gripper circuit as well as a robotic circuit that was supposed to navigate the maze. The gripper circuit was developed and worked properly, but similar issues that had occurred with Prototype #2 were apparent within the robotic circuit of Prototype #3. These issues were eventually fixed by including encoders which would help the robot trace how far it has travelled while navigating the maze. Multiple microcontrollers were also destroyed in the process of developing prototype #3. It was determined that by hooking the motors up to the same power source as the microcontroller, a back-EMF voltage was being applied, which was causing the microcontroller to stop working. By isolating the power sources using two individual batteries, this problem was solved.

After testing our final design, the robot was able to navigate the maze, locate the doll, pick up the doll and then return it to the start of the maze within 15 minutes. It was able to this autonomously without leaving any parts behind. Unfortunately, our design was

not as robust as we would have liked. On average, the robot was able to complete the objective perfectly every fifth time it was tested. Because of this, we were not able to complete the objective three times in a row without fail. Nevertheless, our robot was able to accomplish four out of the five objectives that we initially stated in our project charter while adhering to all of the project constraints.

Table of Contents

Table of Figures

1 Introduction

This design project was created in order to give students the opportunity to apply their knowledge obtained at Trinity University in electrical, mechanical, and computer engineering. The purpose of this project is to develop a device that maneuvers through a maze to find a doll and return it to the start of the maze. The objectives that must be completed are as follows:

1) The robot is able to solve the maze.
2) The robot retrieves a troll doll from within the maze and returns to start.
3) The robot completes objective within 15 minutes.
4) The robot completes the objective 3 times in a row without fail (which proves that the robot is relatively robust).

Once the objectives were determined, the constraints were specified for the project and are stated below:

1) The device must be safe for the user and any by-standers.
2) The device must be able to complete multiple times without mechanical failures.
3) IEEE Constraints
    a) The robot must be autonomous.
    b) No parts of the robot can come off at any time and be left behind.
    c) No jumping over walls.
    d) The maze consists of 6 by 10 boxes where each box is 18cm to a side.
    e) The walls of the maze will be painted a reflective color (preferably white).

Given these objectives and constraints, a solution to the design problem was obtained.

2 Design Overview

In order to achieve the desired goals for our robotic mouse, a flow chart was developed to provide an overview of what is necessary for a competent product. This flow chart is shown below in Fig. 1.

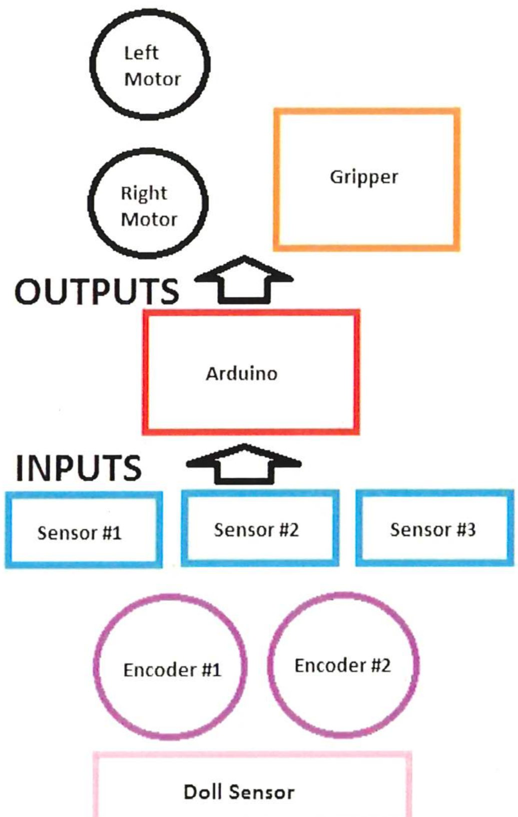


Figure 1: Flow Chart of Robotic Mouse Mechatronic System

The above chart shows the inputs and outputs of our microcontroller. For our robotic design, we are dealing with three IR sensors, two encoders, and two servo motors within the doll system. Based on these seven inputs, an algorithm was developed to control the left and right motors to properly navigate through the maze and trigger the gripper to pick up the doll when it is located. The design for our final robotic mouse is split up into three sections; electronic components, software and robot chassis.

2.1 Electronic Components

The schematics in Appendix A show the basic layout of the circuit design for our robotic mouse. A printed circuit board (PCB) was used for our final design. Within the PCB, there are five sub-circuits which include the power conditioning circuit, the H-bridge circuit, the encoder circuit, the IR sensor circuit as well as the Arduino circuit.

2.1.1 Power Conditioning Circuit

A schematic of the power conditioning circuit can be found in the Appendix as **Figure A1**. This circuit contains two battery sources (14.8V and 9V) which power the system. Two switches are shown which allow the user to turn the circuit on and off. It was necessary to implement two switches since the two power supplies are isolated from each other. The LEDs operate as signals to let the user know when the power supplies are on and off. Two capacitors are included between voltage source and ground in order to help clean up any noise throughout the system.

2.1.2 H-Bridge Circuit

Figure A2 shows the H-Bridge circuit. This circuit amplifies the current in order to supply the DC motors with enough power to operate. It can be seen that Pins 1,2,7,8,9,10,15, and 16 all operate as control inputs from the Arduino Mega. Pins 3, 6, 11, and 14 are outputs that control the motors.

2.1.3 Encoder Circuit

Figure A3 shows the circuit necessary to implement the encoders. This circuit contains pull-up resistors between the power supply and the encoder pins being input to the Arduino. This protects the encoders and ensures that an appropriate current is flowing through them.

2.1.4 IR Sensor Circuit

Figure A4 shows the IR sensor circuit. This is an extremely simple circuit which shows the pins which attach to the power supply, ground, and the signal that the sensors input to the Arduino.

2.1.5 General PCB Circuit

The combination of all of these sub-circuits can be found in Fig. A5. This is known as the general circuit of the electronic mouse and shows how all the sub-components fit together.

2.1.6 Doll Emitter/Receiver Circuits

Circuits that are not included within the PCB are the doll emitter and receiver circuit. The "doll emitter circuit" can be seen in Figure A6. This circuit is powered by a 9V battery which is attached to the doll. This circuit contains a photoemitter diode that will emit a specific frequency that is recognized by the "receiver circuit", as shown in Figure A7. Once the "receiver" registers the frequency, it will initiate a grab sequence that will cause the gripper to rotate downwards and pick-up the doll.

2.2 Software

The robot will be controlled with an Arduino Nano microcontroller. The software is written in a language developed for Arduino that is based on C. A program is developed to control all parts of the robot's movement and logic. At the lower-level, the program will read the sensors, be able to drive the left and right motors, and control the gripper. At the high-level, the code must convert sensor data and use that to find where the robot is in the maze and its orientation. It must also make decisions on where to search and the best way out of the maze based on the search algorithm.

2.3 Chassis

The Micromouse robot has a round chassis and has two tiers where different components are placed. The bottom level holds the motors, battery and sensors. There will be three sensors, which will be positioned in front of the motors. One will face forward, and the other two will be angled at 45° to the right and left of the chassis. The motors (which are cylindrically shaped) will be placed concentrically with the leads facing inward and the shafts outward. The chassis is designed so that the wheels will be approximately in the center of the robot. The wheels are attached to the motor shafts using bushings specifically made for the wheels we selected. The battery is attached adjacent to the motors on the bottom layer on the backside of the robot. In addition, the robot will have a ball bearing attached to the bottom side of the bottom layer in contact with the ground in the front and rear of the chassis to provide extra support in turn and tipping prevention. The middle level will be very thin, just large enough to separate the PCB from the first level and the gripper arm which will be on the top layer. The gripper will be comprised of two main components: the "arm" and the "hand". From the gripper arm purchased only minor modifications need to be made in order for it to be completely compatible with the rest of the robot. The gripper will have a IR sensor that will be able to recognize the doll which will have an IR emitter. Once recognize the gripper should be able to automatically grab and secure the doll and the robot will return to the beginning of the maze.

3 Design Development

3.1 Prototypes

3.1.1 Prototype #1: Wall Follower

The first prototype developed was an autonomous robot that was able to follow a wall (aka a Wall Follower). Two PING sensors were installed on one side of a simple, circular chassis with two PM motors connected to two wheels on the sides and a ball bearing located in the front, center for balance. A breadboard was used to develop the circuitry, and the objective was accomplished on September 31, 2012.

3.1.2 Prototype #2: Maze Follower

For prototype #2, the goal was to develop an autonomous robot with improved sensors, motors, and wheels as well as a new chassis design. The robot would be designed to maneuver through the maze by following simple, coded instructions, such as going straight and turning left and right. Also, the robot was to be self-contained, which means that it could no longer be connected to a breadboard. Instead, a hand-soldered board was constructed which contained all the necessary circuit components to run the motors, sensors and wheels. A 12V battery was also used to power all circuit components. This prototype was unable to be completed by the November 30, 2012 deadline and essentially scrapped.

3.1.3 Prototype #3: Gripper and Advanced Algorithm

For prototype #3, two separate robots were developed. The first was an autonomous robot that is able to navigate the maze using a search algorithm. The robot would be able start in one location and navigate to a specified location within the maze. A new chassis will be included within the design, as well as new batteries, wheels, sensors and motors. A hand soldered board will be used for this prototype as well.

The second robot was the first experience with the gripper component of the robot. This robot was able to recognize when the doll is within reach, grasp the doll, and then lift it up into the air. This will be accomplished by using an infrared emitter and receiver system, where the doll will emit a certain frequency and the robot will receive the emitted signal. This ultimately was completed February 14th, 2013.

3.1.4 Prototype #4: Micromouse #1

The purpose of prototype #4 will be to combine the two robots developed in prototype #3. This robot should be able to maneuver through the maze using a search algorithm, located the doll, and pick it up. A project circuit board (PCB) was incorporated into this prototype. This prototype was completed February 28, 2013.

3.1.5 Prototype #5: Final Micromouse

Prototype #5 is a more refined version of prototype #4. It would be able to navigate the maze, locate the doll, pick it up, and return to the start of the maze. Also, components will be analyzed and improvements can be made if necessary.

3.2 Subsystem Design

In order to determine the ideal components for our robotic mouse design, a decision table was developed for each main category. These categories include sensors, microcontroller, chassis, battery, motors, wheels, gripper and algorithm, which are divided between hardware and software.

3.2.1 Hardware

3.2.1.1 Sensors

When selecting a sensor, it was necessary to determine what range the sensor would be able to operate within. The width of the robot was expected to be around 10cm, which would leave 4cm on either side of the robot to the walls (18cm by 18cm boxes). Many sensors have ranges that are much greater than 4cm, which is why careful attention was given to selecting a sensor with an appropriate range. The size of the sensors was also an important factor to take into account. Due to the amount of components and size of each component attached to the robotic mouse, it is necessary to select the smallest option for all. The IR Sharp Sensor was selected for our design. This is due to its size (45mm long and 13mm wide) as well as its range (1cm to 40cm). Four sensors were purchased online through Spark Fun electronics for $13.95 each.

3.2.1.2 Microcontroller

The size of the controller was taken into account when selection occurred. It was also necessary for the controller to be able to interface with the other selected components (encoders, sensors etc.). It needed to have enough I/O pins as well as a pulse width modulator connection which could be programmed with the designated software. The Arduino Nano was selected for our design. This is because it is extremely small (dimensions), it has enough pins for our purposes, and it is compatible with all the selected sensors, encoders and motors which have been selected. The Arduino Nano microcontroller costs $43.

3.2.1.3 Chassis

The chassis is the skeleton of the robot. It must be strong and durable enough to support all the components: motors, battery, wheels, circuit board, etc. The first prototypes' chassis was constructed out of Plexiglas. This material is light, very easy to work with (cutting, drilling into, etc.), doesn't conduct electricity or heat very well (which could prevent overheating or short-circuiting components) and is strong enough to support all of the components mentioned above. In addition, the overall design of the chassis needs to take into consideration the size constraints of the maze, meaning it must be small enough to navigate the maze made up of 18 cm by 18 cm squares. For the final design, the chassis will include at least two levels. One as the bottom base which the motor and battery are secured to. The second level will include all of the electronics. The main challenge to be met in terms of the chassis design will be to incorporate the "gripper" onto it so that the gripper can operate to pick up the target and carry it back. Therefore, the chassis needs to have a design that is well balanced so that tipping does not occur.

3.2.1.4 Battery

The battery chosen must be able to operate the microcontroller, two half h-bridges, two motors, and a gripper. Since this will require a significant amount of power, the group determined that at least a 12V battery should be implemented. The second prototype's chassis size was basically dependent on the battery size and was too big to navigate the maze. To address this issue, a smaller 12V Ni-H battery was used, which allowed the chassis to be smaller and could maneuver throughout the maze much easier. This battery provides 2100mAH. Ultimately, the final design should be smaller than the second prototype. Thus a smaller battery is needed. In addition to that, the gripper will require a lot more power than the prototype models. After researching and investigating other battery alternatives, it was found that a 14.8V Li-Po battery would better for the robot in both voltages supplied and is much smaller than the other battery alternatives. The battery purchased cost approximately $31.

3.2.1.5 Motors

One of the main design goals is to navigate the maze in a timely manner. Therefore, the robot needs to be able to move quickly and must have motors strong enough to drive the robot effectively. After using the preliminary motors in the first two prototypes, it was observed that the final design should have more powerful motors. The motors selected for the final design are parallax 12 V DC motors. These motors have a 4.5V - 12V range (6V Nominal) and outputs 200 RPM which demonstrates flexibility (in terms of input voltage) and sufficient enough power to drive the robot's wheels. In order to properly control the motors, a half h-bridge driver was used. This can be seen within the circuit schematic in Figure A2. The selected motors cost $40.

3.2.1.6 Wheels

In observing the first two prototypes, the wheels were a little over two inches in diameter, which contributed to the size of the robot and its ability to make sharper turns. Therefore, smaller wheels were required. Two sets of wheels (one 1.635 and 1.875 inches in diameter) manufactured by BaneBots were obtained which will be compatible with the new motors. Another factor to consider is that the wheels have an option of tread durometers (hardness). Ideally, harder treads will allowed the robot to move faster because there will be a reduction in frictional losses; softer ones will aid the robot in turning due to increased traction. The final design will implement the better option. The wheels with motor adapters cost $10 total.

3.2.1.7 Gripper

The main objective is to pick up a doll and return it to the beginning. To do so, the gripper implemented on the robot must be able to do just that; secure the target and hold on to it. Therefore, the gripper needs to be strong enough to do so successfully. The gripper also needs to be out of the robot's way when navigating through the maze. Therefore, the robot needs a movable arm design that lifts and lowers the gripper. The selected gripper is the Crustcrawler "Big Grip", which was ordered and will be able to perform all of the criteria mentioned above. The cost of this gripper is $99.

3.2.1.8 Encoders

One of the most important parts of maneuvering and solving the maze is the ability for the robot to know where in the maze it is. This is done with quadrature encoders. This device measures the amount the wheels have moved. The encoders work by shining a light between a rotating disk with slits in it. As the disk rotates the sensor signals every time the light is blocked by a slit. In selecting an encoder, the resolution and size are important factors. The resolution is determined by the number of slits in one rotation of the encoder. The encoders we selected have a resolution of 512. Since they are quadrature encoders, they have two side by side sensors to detect light, so there is actually a signal every time one of the sensors change from low to high, so the actual available resolution is 2048. These encoders cost $40.

3.3 Methods

3.3.1 Design Theory and Criteria

Given these objectives and constraints, a solution to the design problem was obtained. In order for the robot to maneuver throughout the maze, motors, IR sensors, and encoders would be implemented. It also needed IR sensors to accurately guide and navigate the robot. Encoders were used to aid in the programming by relaying the distance traveled to correlate with the robot's position. A rechargeable Lithium polymer battery was needed to supply power to the robot. A gripper and gripper arm were needed to grab the doll. The gripper itself is constructed using two servo-motors and a C bar. A PCB (printed circuit board) will also be used in order to implement the circuit design necessary to operate the motors, sensors and encoders.

The optimal design solution includes a 3-layer design. Each layer is made of 0.25" thick Plexiglas. The wheels, (BaneBots wheel 1-7/8" x 0.4", 50A), and motors (12V DC geared motors), are located on the bottom layer with slots cut out for the wheels. The 50A for the wheels indicate the tread durometers (hardness), and 50A is hard and stiff, which are the best for this project. The motors are attached by custom aluminum brackets to the bottom layer. Based on the difficulties of Prototype #2, it was determined that placing the wheels in the center would allow it to maneuver best, even allowing the robot to rotate 360 degrees in a single unit square so it could make 180 degree turns if necessary which Prototype #2 could not do. In addition Prototype #2, also had issues getting the corners of its bottom layer caught up against the walls. The new circular design for the bottom layer would allow it to rotate a complete 360 and prevent it from getting caught on the walls, allowing it to move more efficiently through the maze. After trial and error programming the IR sensors, the setup shown in Figure 2, was chosen as the best,. The sensors are attached to the base by custom Plexiglas mounts.

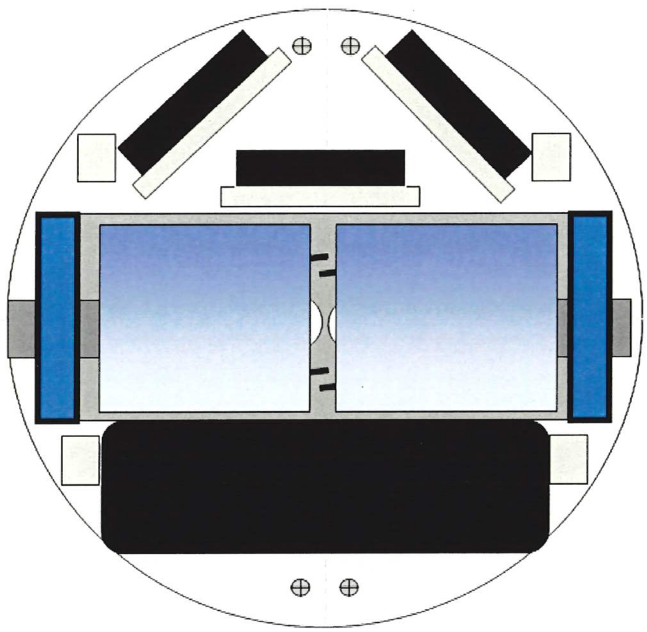


Figure 2: Bottom Layer

The middle layer consisted of the PCB, because it needed to isolated and separated from all the mechanical components to prevent it from malfunctioning or being interfered with. In addition the new This layer for same reason was made circular because it still below the top of the maze walls. Few design adjustments had to be made because of implementation of new components. The original encoders were not sensitive enough to provide the data needed to navigate the maze properly. Instead, shaft encoders were chosen in order to attain this accuracy needed. However, the encoders could not go on the outside of the motor shaft as they typically would, because in would make the robot too wide to maneuver in the maze. Instead, a "gear" mechanism was made right above the wheels (blue). Fig. 3, shows how a second identical wheel (orange), attached to the shaft encoder, was placed right above so that it firmly touches the bottom wheel. Because the top wheel is allowed to freely spin inside the encoder, being the same size as the bottom wheel, it allows to encoder to tell exact how far the bottom wheel has moved indirectly. The encoders were attached to the layer with custom brackets as well. The columns for both the bottom and middle layer were made of 2" 6-32 thread separators, which easily screwed into the layer beneath it and allowed for the layer above it to be screwed into it, the second layer required a ¼" extender to fit the encoder wheel, see Fig. 3

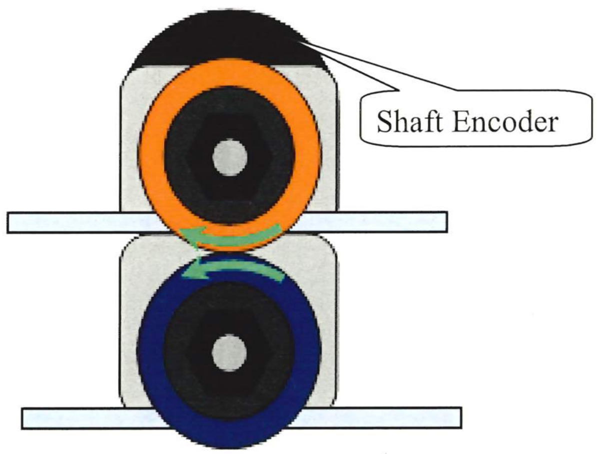


Figure 3: Shaft Encoder and Bottom Wheel

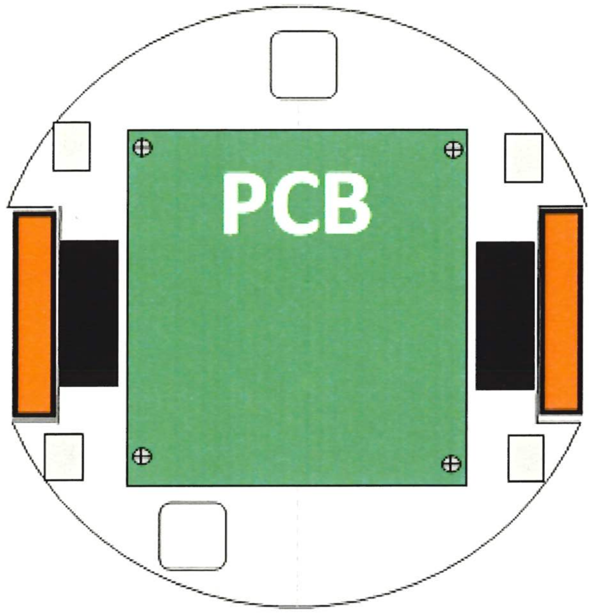


Figure 4: Middle Layer

The top layer consists of the gripper and gripper arm. The gripper is a Crustcrawler Big Grip with an AX-12 Smart Servo. The gripper arm is controlled by a second servo, connecting to the arm piece connecting the two motors and mounted to the base by custom L-shaped brackets. This layer for convenience and aesthetic reasons was cut circularly. However, a section in the middle is cut out so that the gripper arm can rotate all the way up. This can be seen in Fig. 4.

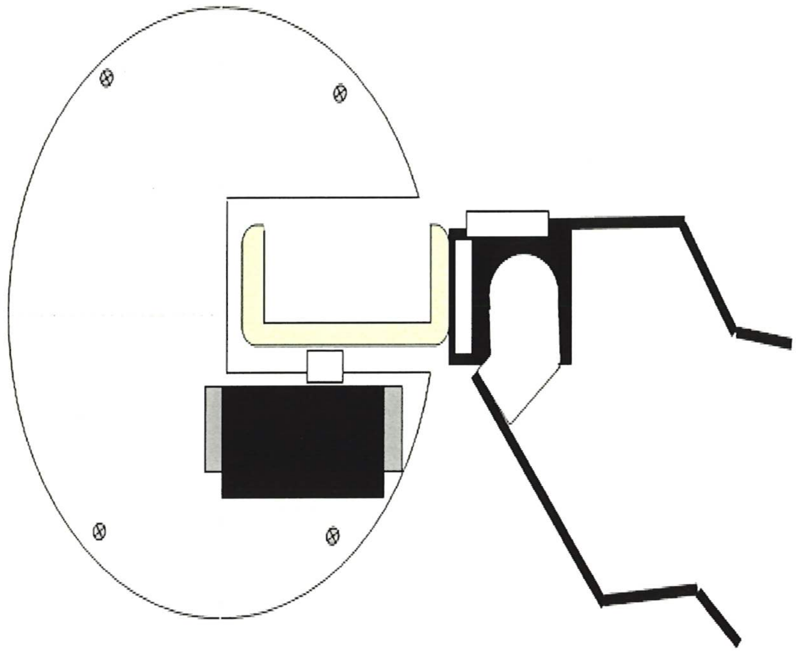


Figure 5: Top Layer

The purpose of Prototype #3 was to make two separate robots and combine them to make Prototype #4. The gripper robot, which included the gripper and gripper arm for Prototype #3 was tested and performed just as expected. Once the receiver diode was close enough to the emitter diode, the gripper sequence was activated so that the doll was picked up. The final product incorporated this into its design.

3.3.2 Coding and Programming

After turning on the Arduino circuit and the motors, the microcontroller takes a couple of seconds to initialize and set up all the pins and functions within the code. After this process is done, the Arduino enters and remains in the main loop, where basic routines are performed. These basic routines include reading data from the IR sensors and the encoders, and performing calculations to determine the current robot positioning (in an imaginary x and y axis) and the angle it is facing. This leads to the robot determining what box its currently in, and what the associated function with the box is. Every time there is movement within the robot, these routines are looped and the resulting values recalculated.

The mouse is intended to aimlessly roam through the maze, collecting data about each box and its surroundings. For instance, if the second box the mouse encounters has two walls on opposite sides, the mouse will recognize that the only useful command for this box is to go forward instead of turning against the walls. The Arduino will keep itself in this loop until the doll is sensed by the phototransistor receiver. At this point, the routine will shift from exploring the maze to looking at the doll, moving forward until sufficiently close, and activating the gripper function (which moves the hand down and grabs the doll). Once this routine is complete, the robot will enter its third and final routine.

The third routine consists of calculating the fastest route back to the start. During the first routine, the robot will update a chronological list of all the boxes it was in. At the end of the second routine, this list is essentially the way out of the maze, though the software will discard 'repeated' boxes and all those boxes within. For example, if the robot stores the box sequence A->B->C->D->F->G->C->H, where A is the beginning and H is where the doll is, the robot will declare A->B->C->H as the fastest way to solve the maze (the above sequence implies that C connects with B, D, and H), and it will go back in the order of H->C->B->A. These routines mark the basic way in which the Arduino code will work and execute robot movement.

3.4 Results

3.4.1 Design Results

The primary tests were to have the robot successfully:

1. Complete the maze.
2. Retrieve the doll from within the maze and return it to the start.
3. Complete the assigned tasks within 15 minutes.
4. Complete the assigned tasks 3 times in a row without fail.

Although our final design was not extremely robust, it was able to complete every other task reasonably well. The robot was able to navigate the maze effectively using the sensors and encoders. It was also able to locate the doll using the emitter/receiver circuits and pick it up. Once this was accomplished, the doll was transported back to the beginning of the maze. All tasks were completed well within 15 minutes.

The final design was autonomous and did not jump over walls. No parts of the robot were left behind or came off of the robot at any time. The size of the maze is 6x10 squares where each square is 18cm to a side. Except for robustness of our design, our Micromouse robot accomplished all the objectives and met all designated constraints.

3.4.2 Experimental Results

3.4.2.1 Doll Emitter Circuit

In order to implement the "Doll Emitter Circuit", it was first necessary to create two dummy circuits of the emitter and receiver diodes to simulate the situation. These circuits were constructed and an oscilloscope was hooked up to the positive node of the collector diode. The expectation was that as the emitter diode approached the collector diode, this would result in the voltage at the node to drop to zero volts. Figure 7 shows the experimental results of our simulation.

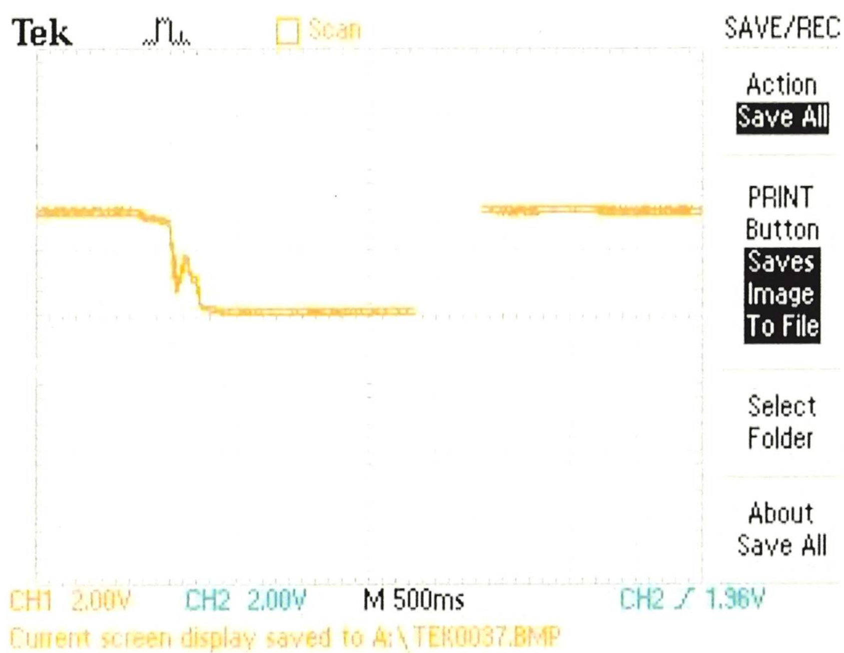


Figure 6: Doll Emitter/Receiver Circuit Testing

It can be seen from Figure 6 that originally the voltage at the node was close to 3V. As the emitter diode approached the receiver diode, it can be seen that the voltage quickly dropped from 3V to 0V. This occurred as the emitter came within one foot of the receiver. When the emitter was moved away from the receiver diode, the voltage returned to its original 3V value. It was determined that as the angle of the emitter to the receiver diode was increased, this allowed the receiver to approach the receiver to a closer distance before it short circuited to ground. By facing the emitter diode 60° away from the receiver diode as it approached it, this allowed the emitter diode to reach within six inches of the receiver before it short-circuited to ground.

3.4.2.2 Arduino Microcontroller

During the development of Prototype #3, three Arduino Nano's and 1 Mega were blown up due to unknown reasons. After analysis of our circuit, it was determined that an electromotive force from the motors was causing rapid fluctuations of voltage throughout the circuit. As the robot speeds up and slows down, this results in a large voltage that fluctuates rapidly. Since the Arduino microcontrollers and the motors were connected to the same power supply, these large fluctuations were pulsing through the Arduino. Figure 7 provides an illustration of what is occurring within the circuit. Since the Arduino microcontrollers are quite sensitive, these rapid fluctuations in voltage resulted in their blowing up.

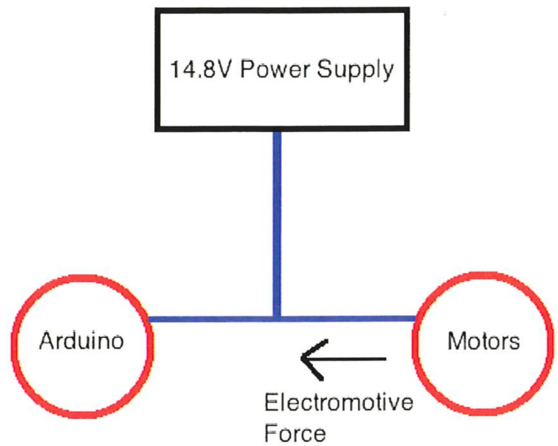


Figure 7: Reason for blowing up Arduino Microcontroller

In order to rectify the problem, we added a 9V battery to the robotic circuit. The original 14.8V battery would continue powering the DC motors. The 9V battery would now power the microcontroller, the IR sensors, the encoders, receiver diode and the servo-motors. By implementing a second power supply, this isolated the rest of the circuit from the back voltage of the motors.

By isolating the power supplies, this also helped to eliminate a huge portion of noise within the IR sensor and encoder readings. The noise generated by the fluctuation in electromotive force was now eliminated. In order to help reduce the noise further, capacitors were connected between voltage supply and ground to act as filters.

3.4.3 Performance Analysis

The robot was tested to see how well it accomplished the objectives and goals of the design. The robot was able to move through the maze, pick up the doll and return to the start under 15 minutes. It was able to do this without any human intervention so the robot was deemed autonomous. Also, it did not break other constraints such as leaving parts behind, being safe for the user and spectators, and not jumping over walls. Unfortunately the final design was not able to repeat this operation successfully three times in a row without failure. The design was not as robust as it should be according to the project goals. The design was not successful at solving the maze on its own. This was due to several programming problems as well as too large of error on the encoders. This error made it impossible for the robot to know where exactly it is in the maze.

3.5 Conclusions and Recommendations

The final design is able to navigate and maneuver a maze. It is also able to locate, detect and pick up a doll. Once it has performed this task it is also able to return to its origin, all while being fully autonomous. The robot was constructed in a safe manner and to the specifications which prevent the doll from being able to jump walls or drop pieces behind. The final design was not able to meet all the original design specifications. Mainly, it was not able to solve the maze. This was a software/programming issue that may have been resolved with more testing. In addition, a financial constraint of $1200 was placed on our project by the Trinity University Engineering Department. The Micromouse Robot's net expenses for the entire project came out to be $1137.

In terms of design components, the biggest key for future designs is investing in a solid microcontroller. The problems with the microcontroller really propagated and slowed down progress. Better selection of a microcontroller would include one that is able to handle multiple components and signals, both analog and digital. Since speed did not turn out to be a huge factor, the powerful 12V geared motor may not have been necessary. A motor that is sufficient enough to move the robot through the maze that is easier to program may be a better option, perhaps one with an internal encoder or incorporate a better tested, more accurate encoder system. This will greatly increase the chances of the robot being able to know where it is in the maze and be able to implement the solving algorithm. Making the robot more compact would make it easier for the robot to maneuver through the maze and not get stuck. In addition, the quality of the sensors could be increased; this would also reduce errors and make maneuvering easier, yet it more costly. Finally, a more aesthetically pleasing design would be beneficial to the design; the implementation of the PCB would greatly improve this.

Bibliography

1. Alley, Rachel. "Robockey." Report. Trinity University, 2011. Print.

2. "Robockey." *University of Pennsylvania*. N.p., 12 2011. Web. 20 Nov 2012.

3. "GP2D120 Optoelectronic Device." 2006. <http://www.sharpsma.com/webfm_send/1205>.

4. "SN754410." *Quadruple Half-H Driver*. 2008. <http://www.ti.com/lit/ds/symlink/sn754410.pdf>.

5. "Schematic-How H-Bridges Work", *Society of Robots*. 2012. <http://www.societyofrobots.com/schematics_h-bridgedes.shtml>

Appendix A: Circuit Figures

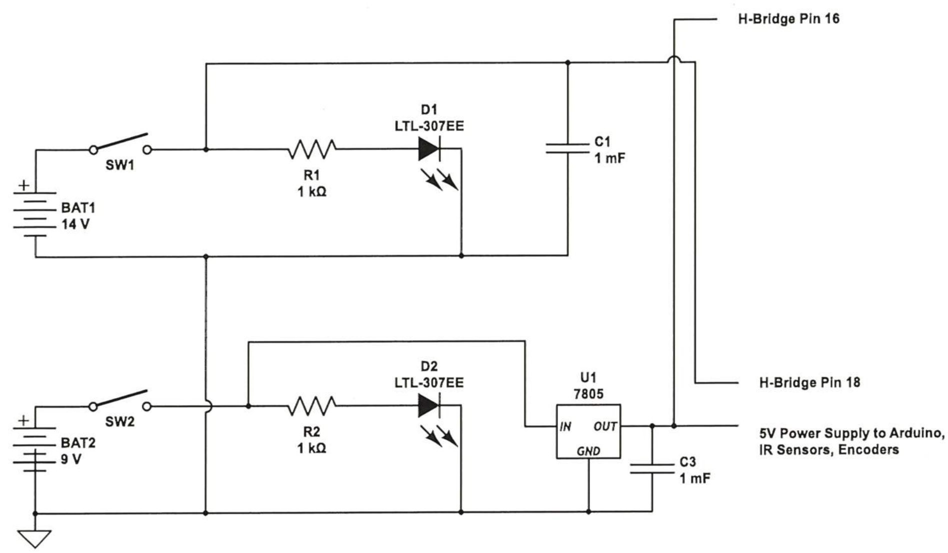


Figure A1: Power Conditioning Circuit

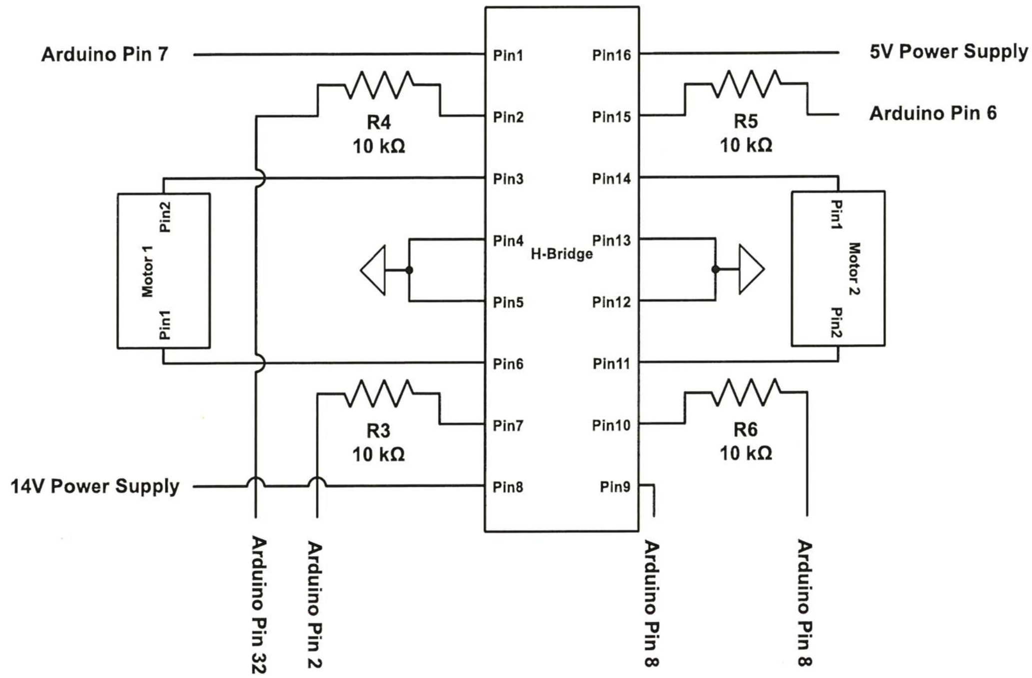
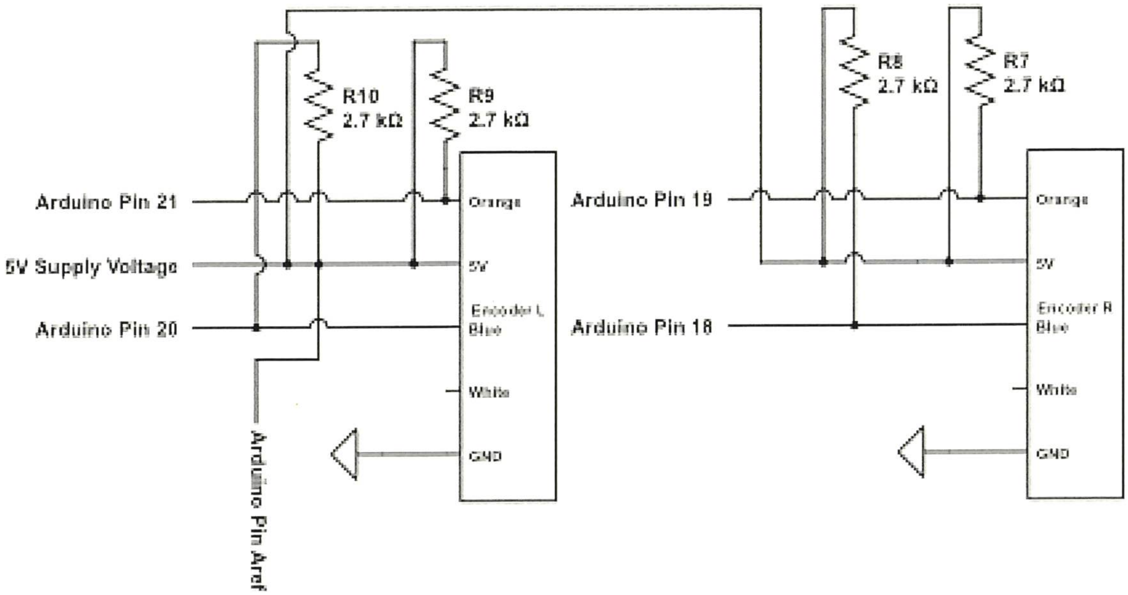


Figure A2: H-Bridge Circuit



A3: Encoder Circuit

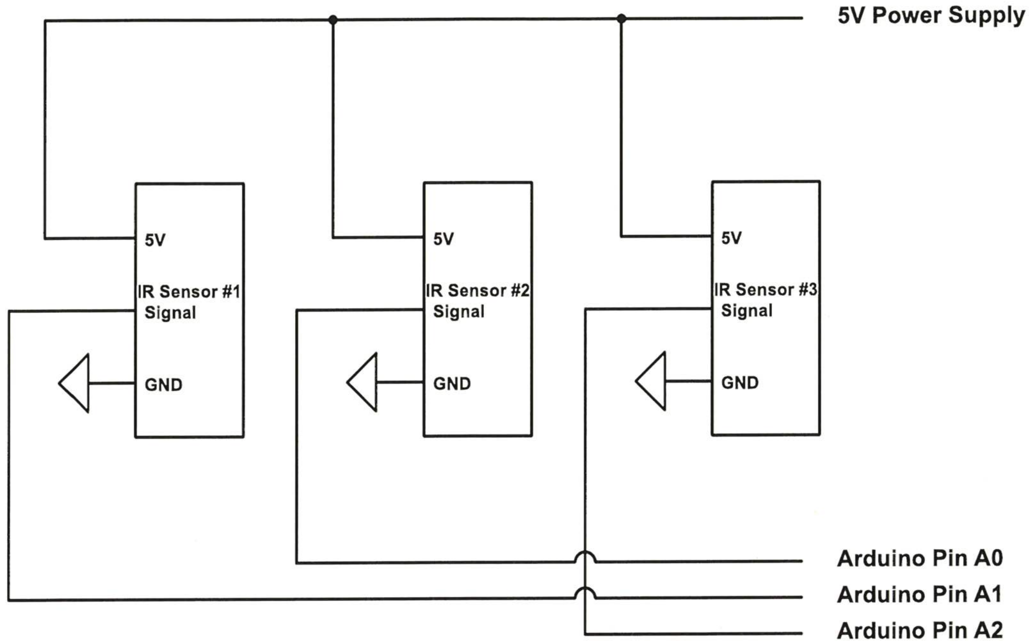


Figure A4: IR Sensor Circuit

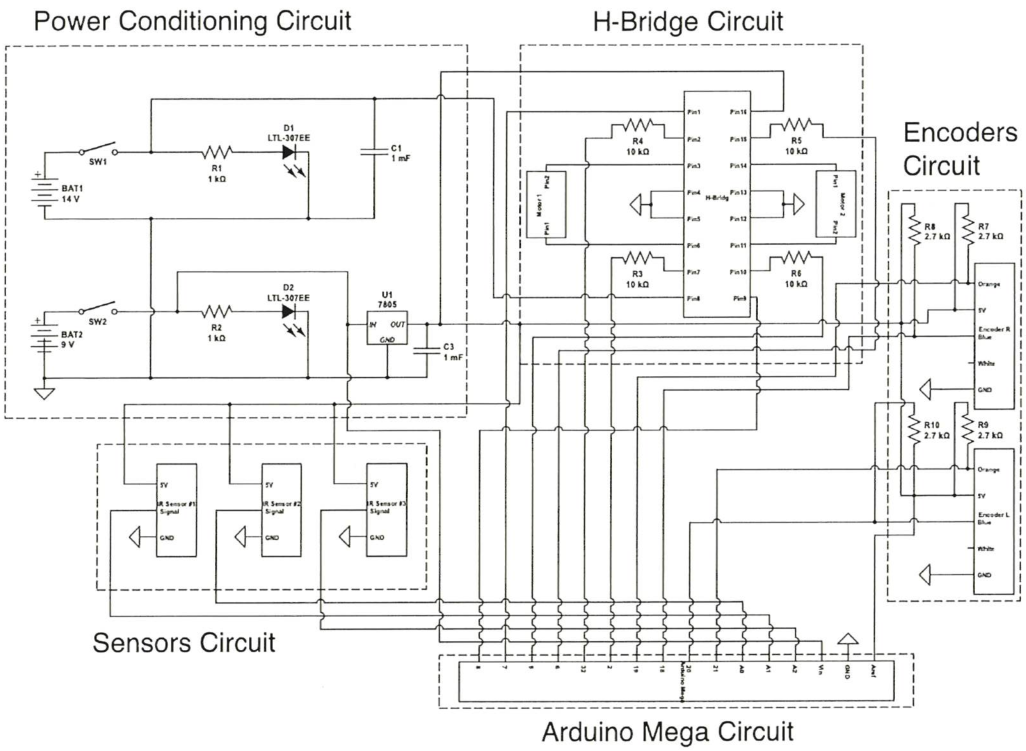


Figure A5: General Micromouse Circuit

Figure A6: Doll Emitter Circuit

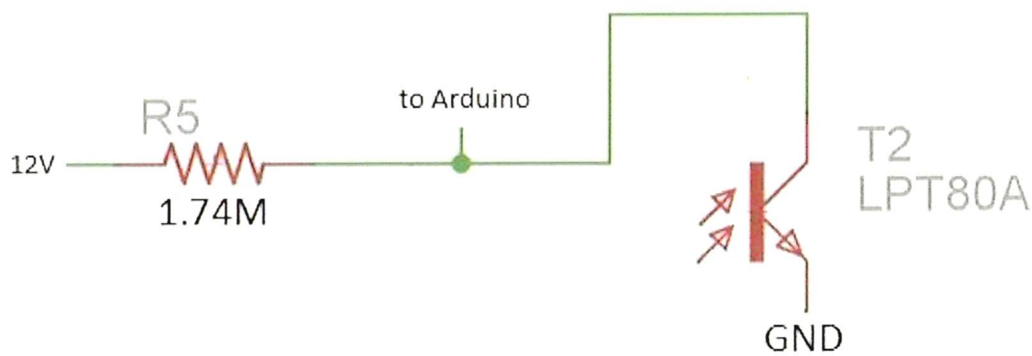


Figure A7: Doll Receiver Circuit

Appendix B: Final Robot Psuedocode

Walls = array [60]*16; // array with all values of 16 (as byte = B00001111) where last 4 bit are n,s,w,e walls, 1 being there 0 no wall

```
SearchForDoll:
        WHILE(still searching):
                Find position with encoders and output x, y, and angle
                Use position to determine which box robot is in
                Save how far this box is from start
                Read front IR
                IF(front IR < 300):
                        Delete front wall
                Way = DecideWay()  // n,s,e,w random
                IF (way != heading);
                        TurnToWay()
                ELSE IF(doll is close):
                        End loop and Switch to approaching doll routine
                ELSE():
                        Go to next box and register side walls
ApproachDoll:
        WHILE(still approaching):
                Read center IR
                IF(center IR< 200):
                        Stop loop and swich to grab routine
                ELSE():
                        Go straight
Grab:
                Lower gripper
                Close gripper
                Raise gripper
```

```
        Turn 180 degress
        Switch to find way back routine
FindWayBack:
    WHILE(still finding):
        Calculate position
        Use position to calculate which box it's in
        IF(at start):
            Stop program
        ELSE:
            Use walls data to find which ways it can go
            Go to available box that is closest to the start
        IF(at start):
            Stop program
```