TABLE OF CONTENTS

1.0 EXECUTIVE SUMMARY

2.0 MECHANICAL DESIGN

3.0 ELECTRONIC DESIGN

4.0 SOFTWARE DESIGN

5.0 STRATEGIES AND INNOVATION

6.0 SYSTEM INTEGRATION AND TESTING

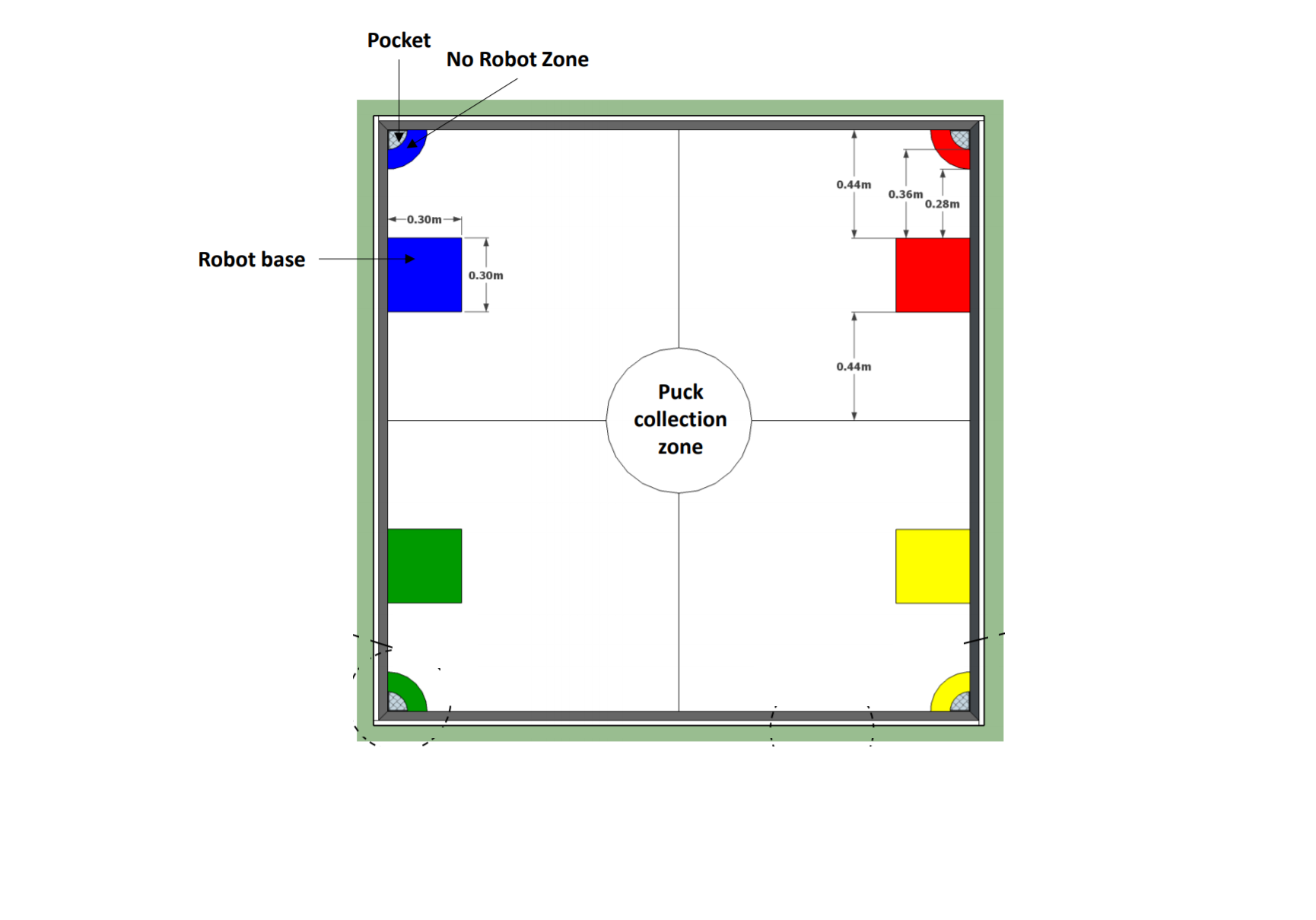
7.0 PROJECT MANAGEMENT

8.0 RECOMMENDATIONS

9.0 CONCLUSION

**1.0 EXECUTIVE SUMMARY**

The ECE3091 Project requires students to work in teams of 3 to build a microprocessor controlled autonomous robot that will compete in a competition resembling a game of carroms with colored pucks. The competition is held within an arena shown below in Figure 1 and 2 robots will compete at any given time. The robot is required to move towards, detect and retrieve one of 2 colored pucks (red and blue) assigned to it at the beginning of the demonstration. The robot is to start moving from one of four bases (red, green, blue and yellow) that is also assigned randomly. Once the puck is retrieved, the robot is to fire the puck into the slot of the corresponding base at the corner of the arena without entering the ‘No Robot Zone’. Every successful shot into the slot will be followed up with a chance to retrieve and fire the sole green puck into the slot. The robots must compete in rounds of 7 minutes.The round is won by winning the most points by scoring shots. The green puck carries 3 points while the red and blue pucks each carry 1 point.



Design Specifications

* Must be fully autonomous and robust.
* Start from base towards the centre of the arena (the puck collection zone).
* Correctly detect the color of the pucks consistently.
* Pick up green and assigned color pucks.
* Launch the pucks successfully into the slots.
* Detect and avoid obstacles (includes the opponent’s robot and arena walls).
* Develop a power system to support the robot to function a full 7 minutes.

Design Restrictions

* The robot base must not exceed 30x30cm.
* Must be programmed using PSoC kit only.
* The robot must not manipulate pucks without touching .
* Must carry only one puck at a time.
* The power to the robot must be only from alkaline or Ni-MH batteries.
* Project budget should not exceed RM 320

The robot is made up of 7 subsystems: the locomotion subsystem to move the robot around, the power subsystem to provide energy to the robot, the gripper for picking up the puck, the firing mechanism to launch the puck into the slot, the color sensor to detect the puck, the collision detectors to avoid collision with the walls and other robots, and lastly the PSOC microcontroller to coordinate the actions of the subsystems to completes its objective.

At the conclusion of the project, it is hoped that the resulting robot will be efficient in performing its tasks and in it’s use of energy. The robot will also be expected to win most if not all of the round it competes in.

In this endeavour, there are issues that do present itself. Firstly, a budget is set for the team to use it to acquire components with, it is our aim to spend it wisely. the budget is limited so it is a must to keep within it. To mitigate costs it is imperative that the electronics and mechanisms of the robot is handled with the utmost care as to avoid spending money for replacements. Careful planning of what items are needed is also required in order to avoid buying unnecessary or overly abundant components.

Secondly, the software of the robot is required to be effective in completing its tasks and in responding to various stimuli. Development of the software is a difficult and very important aspect that needs much of the time spent on the project. Due to its autonomous nature, it is important that the robot be independent in completing its objective without any problems. To alleviate the problems in this part of the project, proper documentation of all code and algorithms will be needed to troubleshoot errors in the code and logic. Thorough brainstorming in a group will be necessary to ensure that the best methods in completing the robot’s tasks are used.

Thirdly, the power subsystem must be efficient and have batteries with enough ampacity to supply the robot. If there is not enough power provided to the various subsystems, they may underperform. Furthermore, if the power consumed is not utilized properly, the robot’s components may get damaged. To ensure that robot will not be underpowered, proper evaluation of the robot’s energy needs by examining the characteristics of its components will be required as to get the proper batteries. Proper protection with the use of devices such diodes and voltage regulators will be used in order to protect the robot’s electronics.

Fourthly, the electronics of the robot are required to perform the actions of the robot. They need to be uncomplicated in design to ease troubleshooting while being energy efficient. It is is easy to complicate the design of electronic subsystems due to the difficulty in integrating them. To counter this problem, simple circuit designs must be applied. An uncomplicated design will also lead to a more energy efficient robot due to fewer components. For implementation, the wires connecting to various components must be colored coded to easily test and debug the robot.

Lastly, the mechanical parts of the robot must be robust yet light enough to support the components especially when moving. If the right materials aren’t used the robot may be too flimsy or too heavy. For this reason, the mechanical parts will be 3D printed using PLA filament. This will also give us the advantage of customizing the parts to fit our needs.

Some of the key innovations included in the final design in order to improve the robot’s performance are as follows:

* An Infrared Sensor was used instead of an ultrasonic sensor as it was easier to implement into the software and also it provided greater precision in puck detection.
* A pre made color sensor was mounted at the bottom just beside the IR sensor. A pre made color sensor was far better and accurate in distinguishing colors and it did not require any calibration.
* To improve power efficiency a buck converter was used instead of a linear voltage regulator.
* The gripper design itself is one of our key innovations. The final design uses a linear solenoid actuator attached to the gripper along with a striking plate which makes the gripping and flicking efficient and much easier to control.
* A funnel made of acrylic sheet was constructed and placed on the bottom base so that the detection of puck was more accurate.
* The use of two microswitches at the back of the robot aids in aligning the robot 90 degrees to the wall.
* The use of a simple yet effective puck searching algorithm which complies with the available hardware of the robot.

**Overview of the Robot Design**

In order to achieve the finest design, we adopted some key innovations. The final design of the robot is composed of a gripper which has a very innovative firing mechanism integrated into it and a crane to lower and raise the gripper to pick up the puck. The gripper has a linear solenoid actuator attached to the striking plate of the gripper which makes very efficient in launching the pucks. The initial design was thought to have a three wheel mobility system but later it was changed to 4 wheel mobility system. The locomotion system consists of two caster wheels on the front of the base of the robot, and two encoder motors connected to gearboxes on each side of the robot. This robot is rear wheel powered. Another innovation which made the puck detection easier and successful is the puck guider on the bottom base. The battery/power subsystem sits on the bottom base whereas the PSoC sits on the top base. Initially a design that includes combining the color sensor with the gripper was considered, but was scrapped because this would result in a very complex structure. To keep the design simple and effective the color sensor and infrared sensor is placed on the bottom base for puck detection and color sensing. Two ultrasonic sensors sit on the front to detect obstacles and walls. The CAD drawings of the final design are given below.

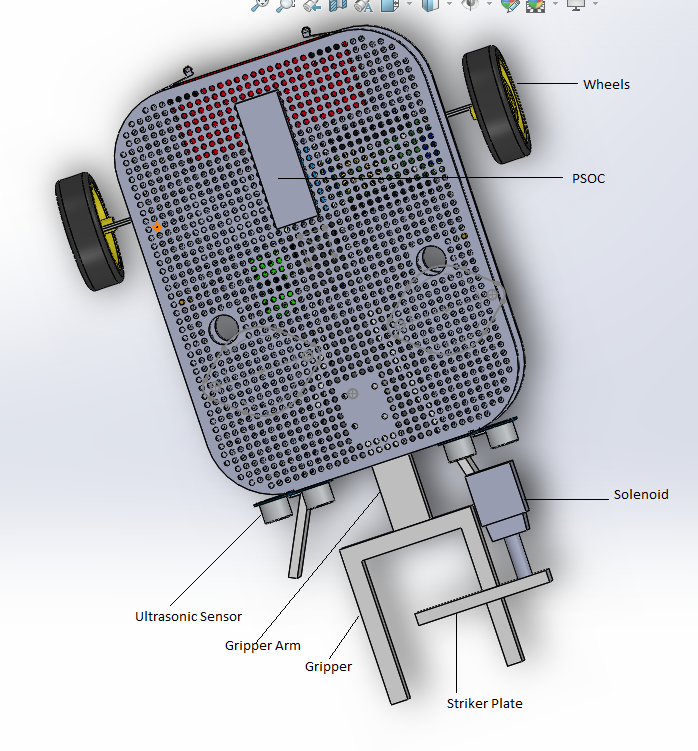
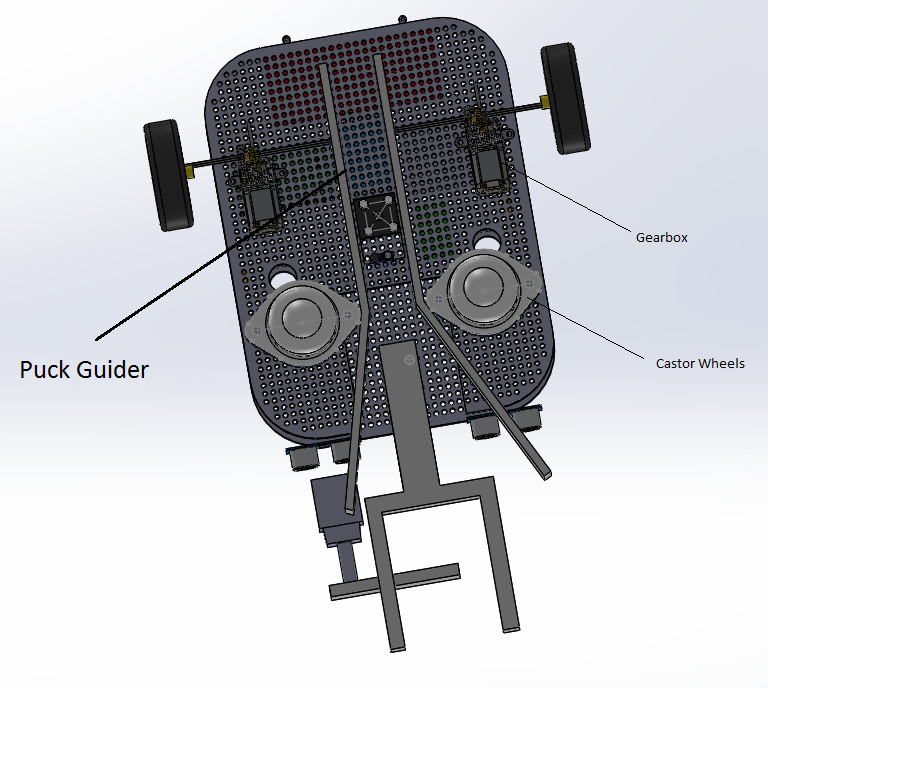


Figure 2: Top View of the Robot



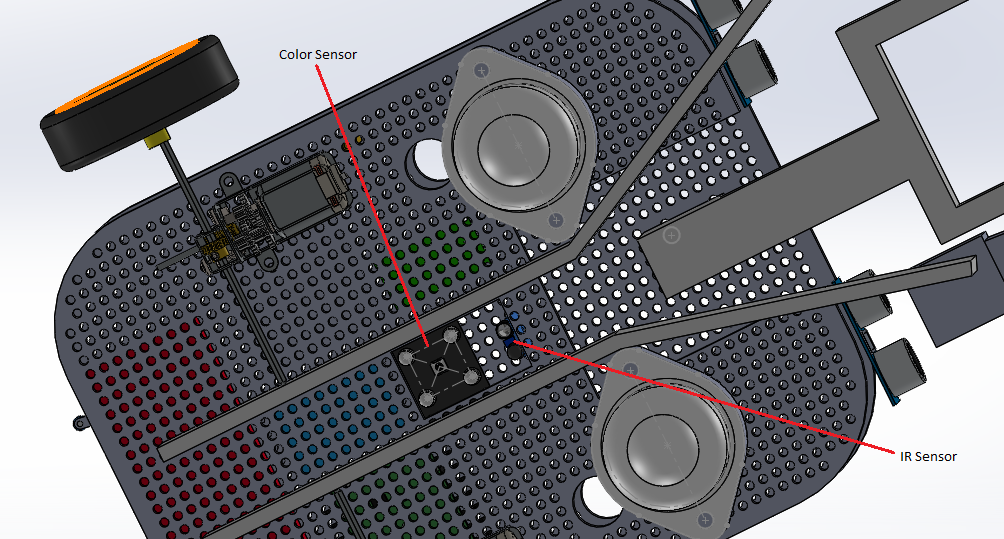


Figure 3: Bottom View of the Robot

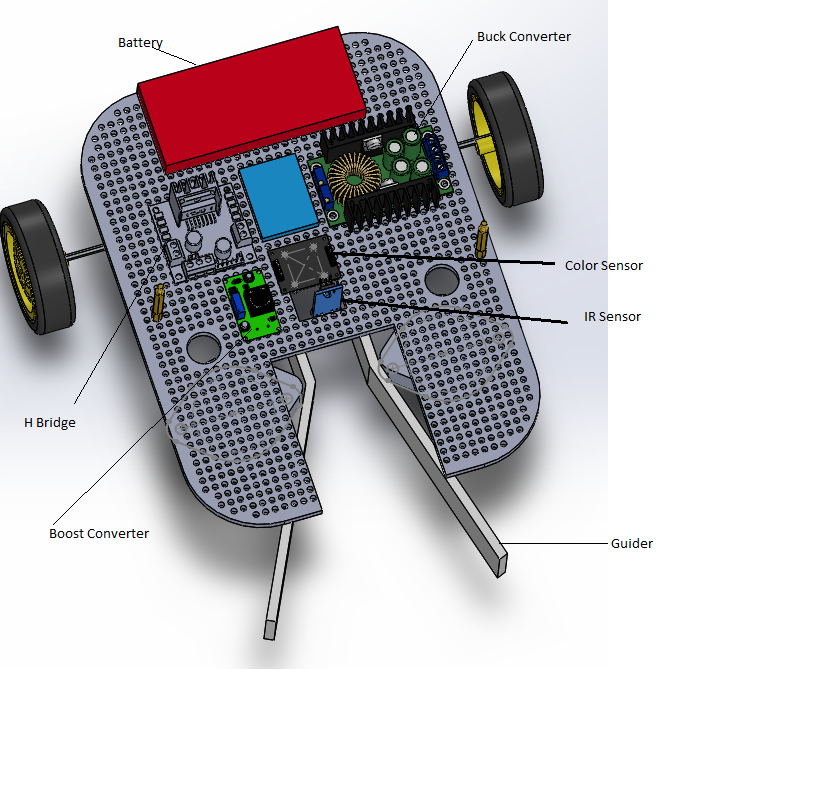


Figure 4: Bottom Base

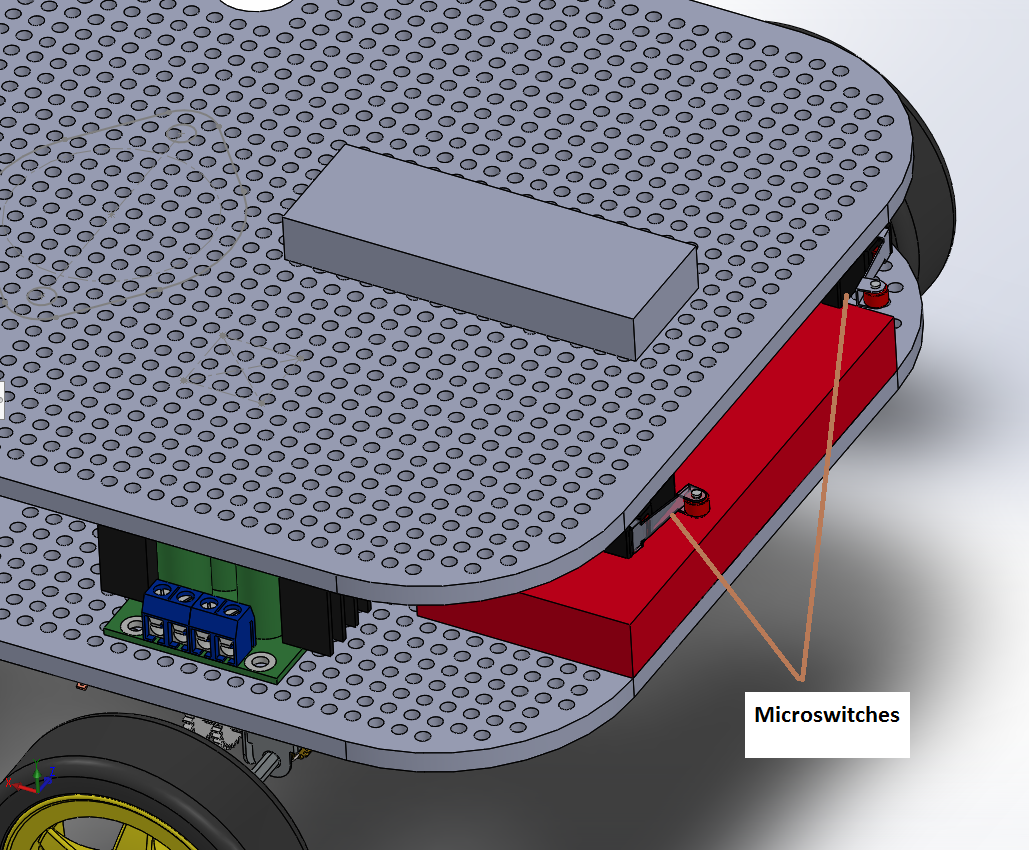


Figure 5: Back View

**2.0 MECHANICAL DESIGN**

The mechanical design of the robot is very important for stability and mobility.

**Wheels**

A four wheel configuration(Figure 6) was chosen for the final design as it provided the correct balance and support. The wheel system consists of two wheels at the back connected to the encoders and motors and two ball casters at the front for support. The two ball casters helped reduce the front friction thus increasing the speed of the robot to achieve the targeted performance.

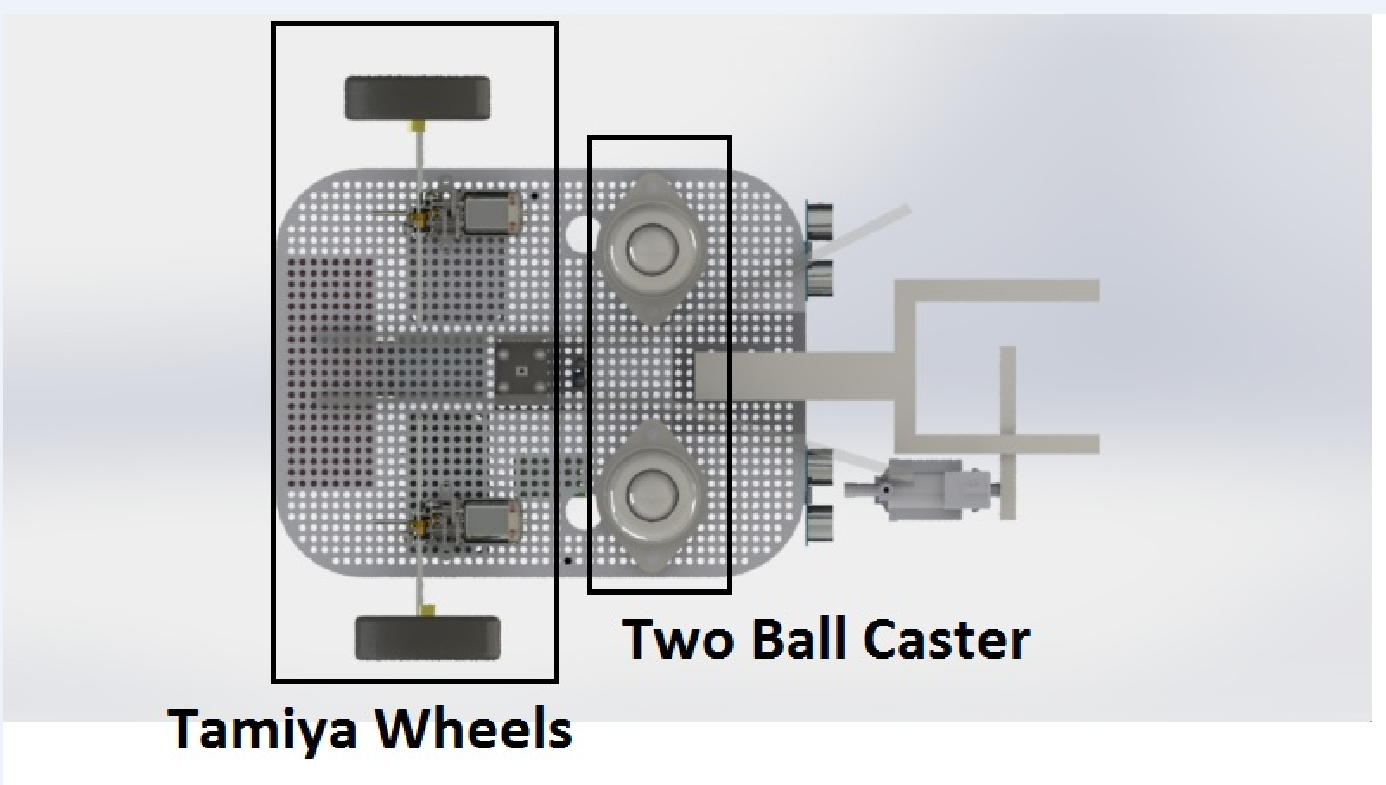
****

Figure 6: Final wheel arrangement

Alternative design:Initially a three wheel configuration(Figure 7) with two wheels at the back and one ball caster at the front was implemented but it was rejected as it would hit the pucks in its way and a complex puck guider will be needed to direct the pucks towards the infrared and color sensor which are placed at the center of the bottom base.

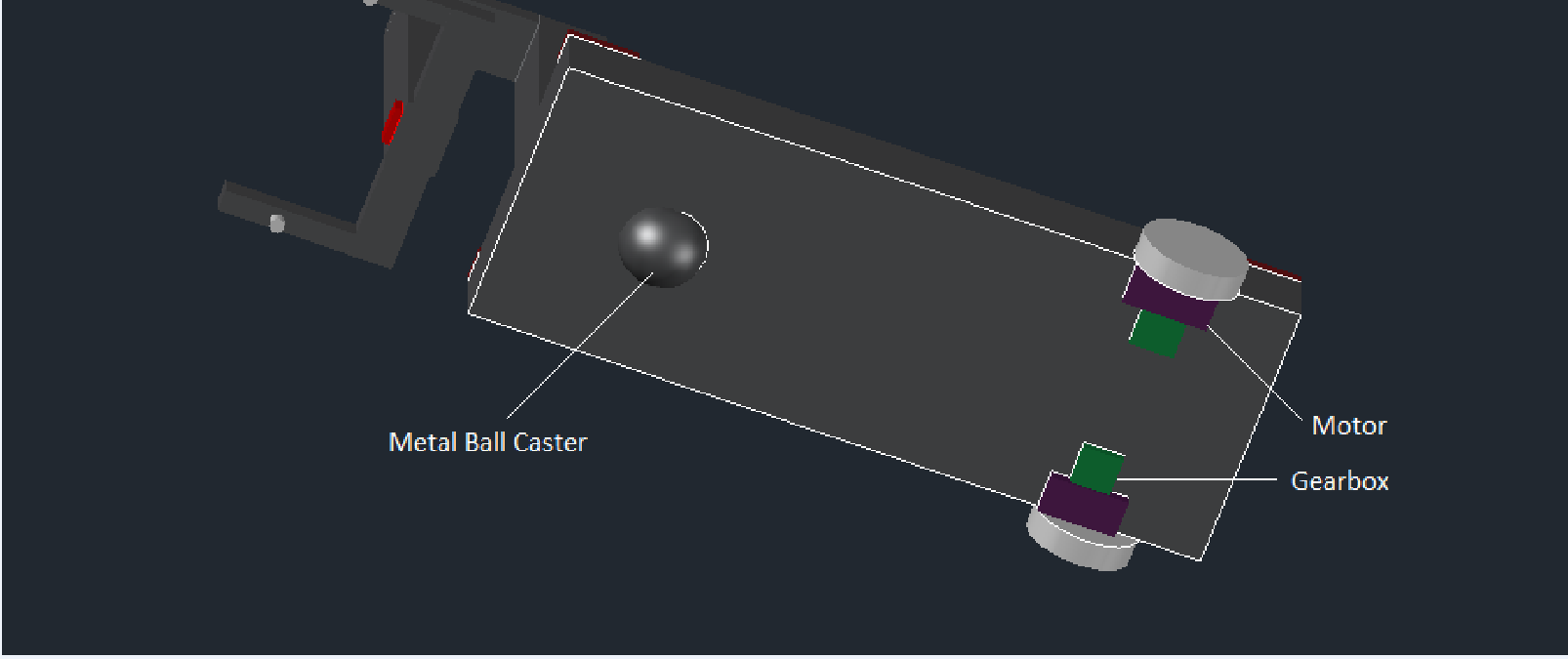
****

Figure 7: Initial wheel arrangement

**Puck Guider**

In our first design we did not have any puck guider attached at the bottom base of the robot. After the color sensor was placed it was found that it could not take readings correctly and displayed wrong color and sometimes the gripper was not able to pick up the pucks consistently. Another problem was the pucks were hitting the caster balls as well as the rear wheels which disrupted the movement of the robot. Therefore a puck guider was cut out of a acrylic sheet and placed at the bottom to make a pathway for the pucks to enter and prevent them from hitting the wheels and caster balls.

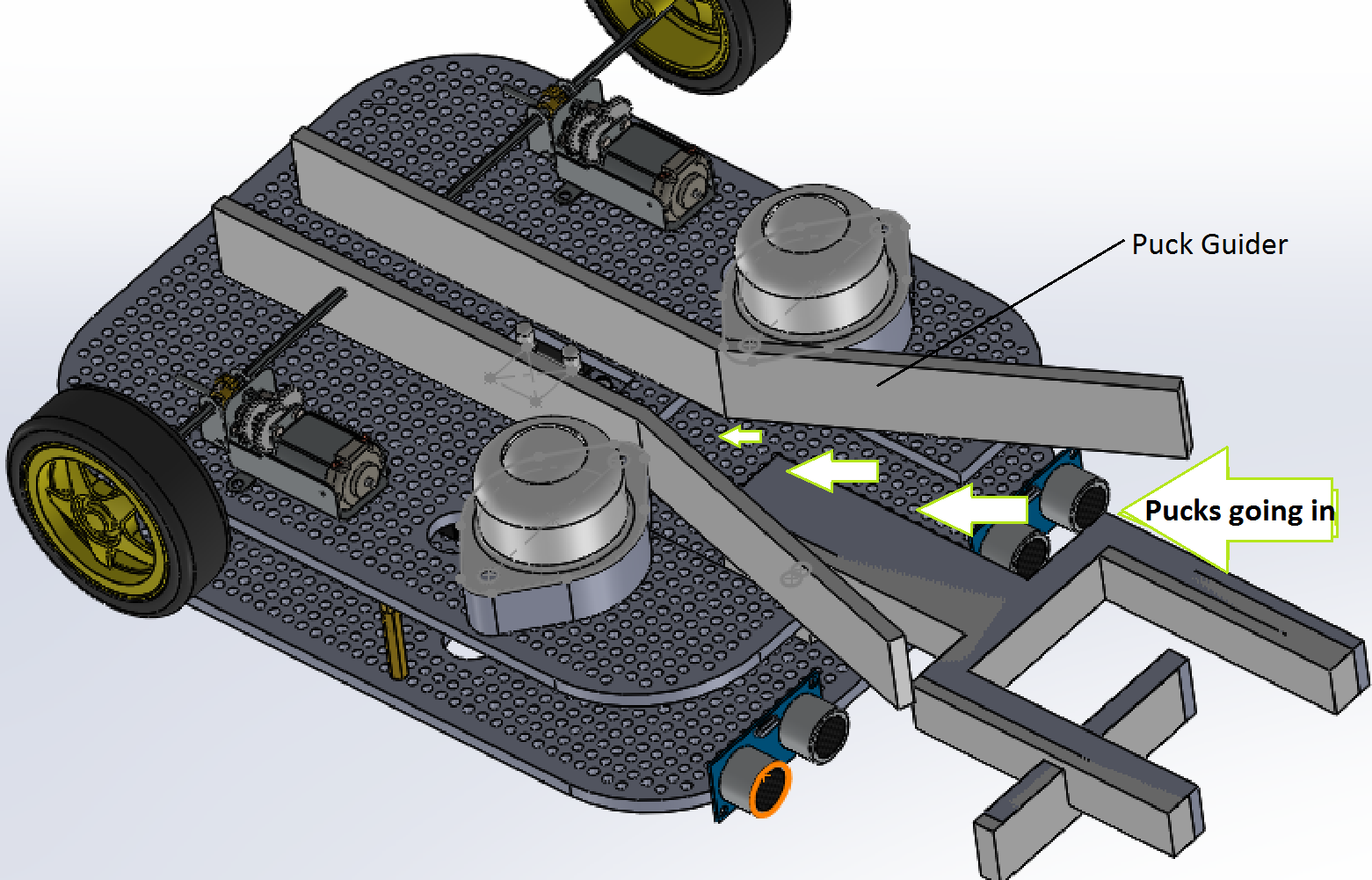


Figure 8: Puck Guider

**Gear Configuration**

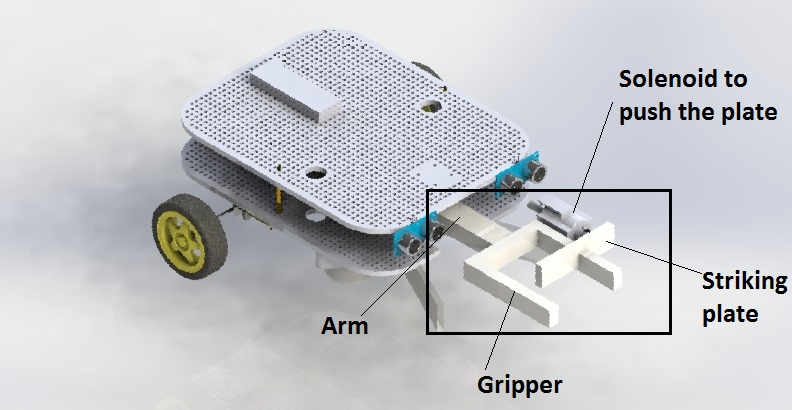
The Tamiya double gearbox given to us is a compact unit with two independent motors and gear trains. The kit includes two motors and all gears and parts to build any of the four possible gear ratio configurations shown below.

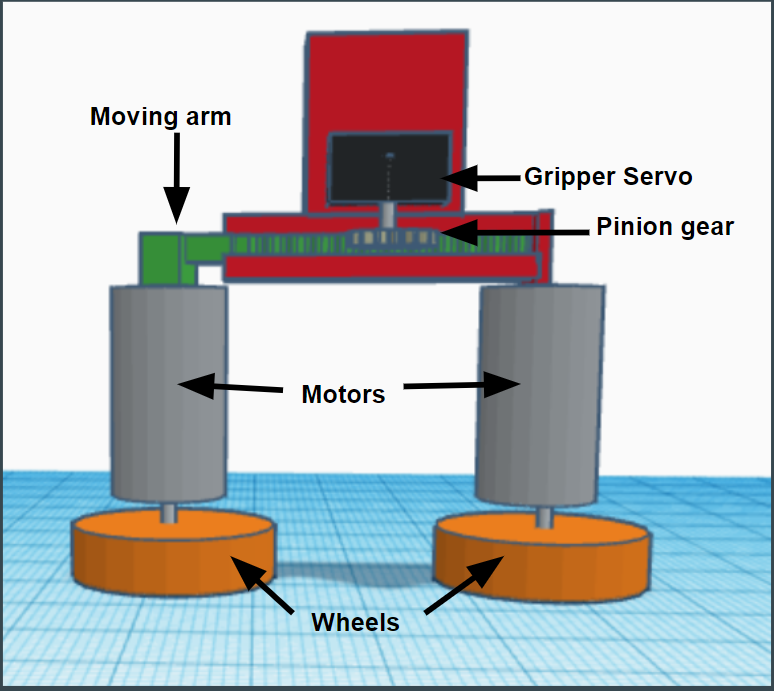


Type B has both moderate speed and torque whereas type C has moderate torque but speed is lower than type B. Type B and C both met our requirement but it was observed that type B was more suitable for our design since the torque was enough for the movement and the speed was neither too slow nor too fast.

Alternative design**:**Type A has very high speed but very low torque.This configuration was chosen at first but it did not provide enough torque for our robot to move. On the other hand Type D has very high torque,more than our requirement but a very low speed.

**Gripper and Firing Mechanism**



Figure 9: Alternative Gripper Design Figure 10: Current Gripper.

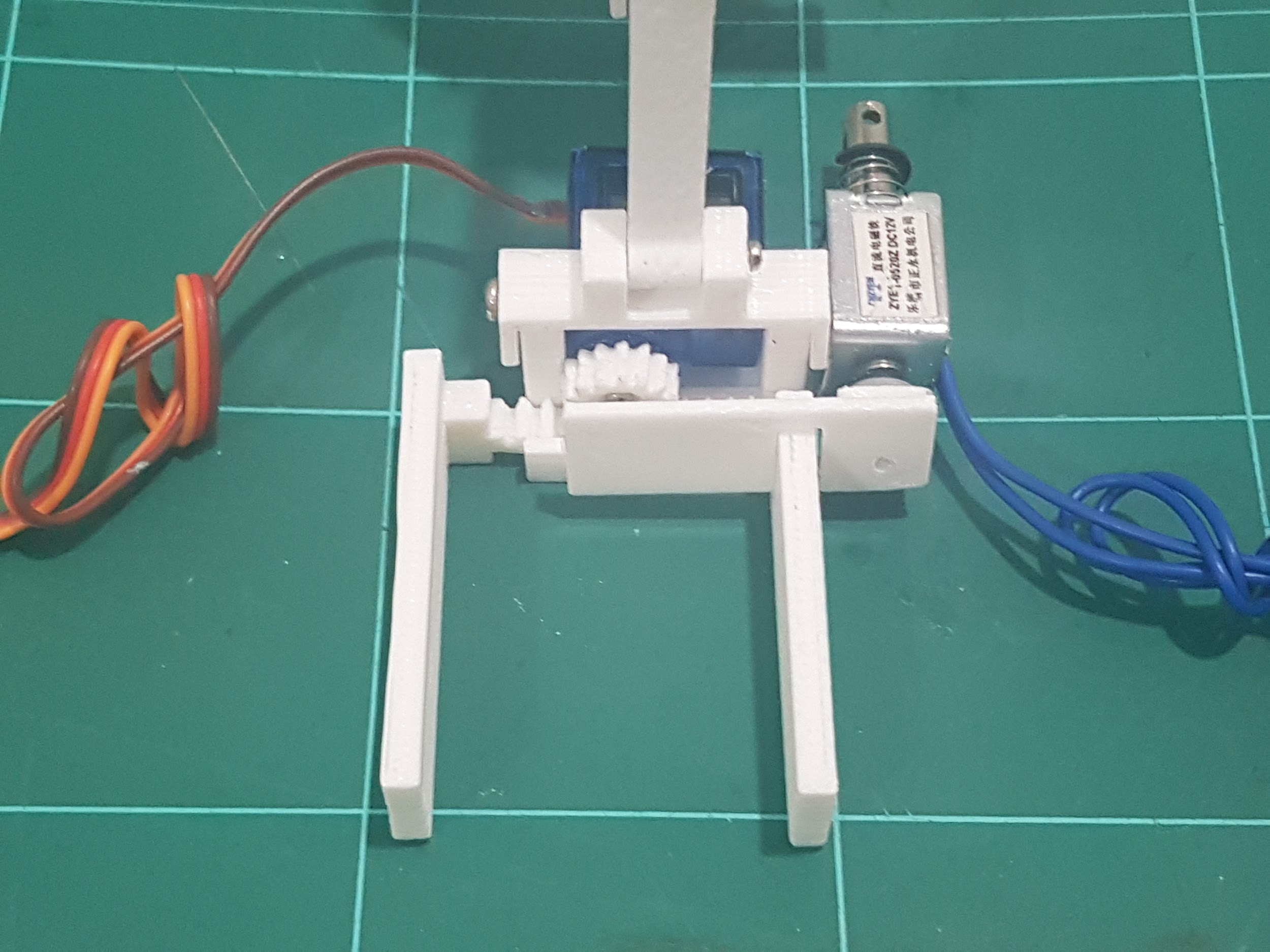


Figure 11: Picture of the current gripper

The current gripper comprises of a stationary arm and a an arm that is connected to a rack. The gripping motion is performed using a pinion gear that moves the rack inwards and outwards. The firing mechanism is integrated into the gripper with the use of a striking plate that is connected to the linear solenoid actuator and attached at the side of the gripper. The firing mechanism launches the puck accurately and at a sufficient distance and the gripper is able to hold onto the puck tightly. However, it was seen that if the puck was not touching the striker plate even in the slightest bit, it would not launch very far. However, this could be rectified if the robot moved forward slightly before firing.

Alternative design: The previous design that was considered had the same gripping mechanism but the arms were replaced with motors attached to wheels. This design good potentially have launch the puck at a great distance but had a number of disadvantages. One of them being that the wheels had to turn at the same speed in order to be accurate. Another disadvantage is that the wheels would not be able to grip the puck securely because there is not enough surface area in contact between them. It is because of these reasons we did not chose this design.

**3.0 ELECTRONIC DESIGN**

The electronics of this robot form one of the basis on how the robot would perform its tasks. It is important that all components under this subsystem work cohesively.

This section will cover:

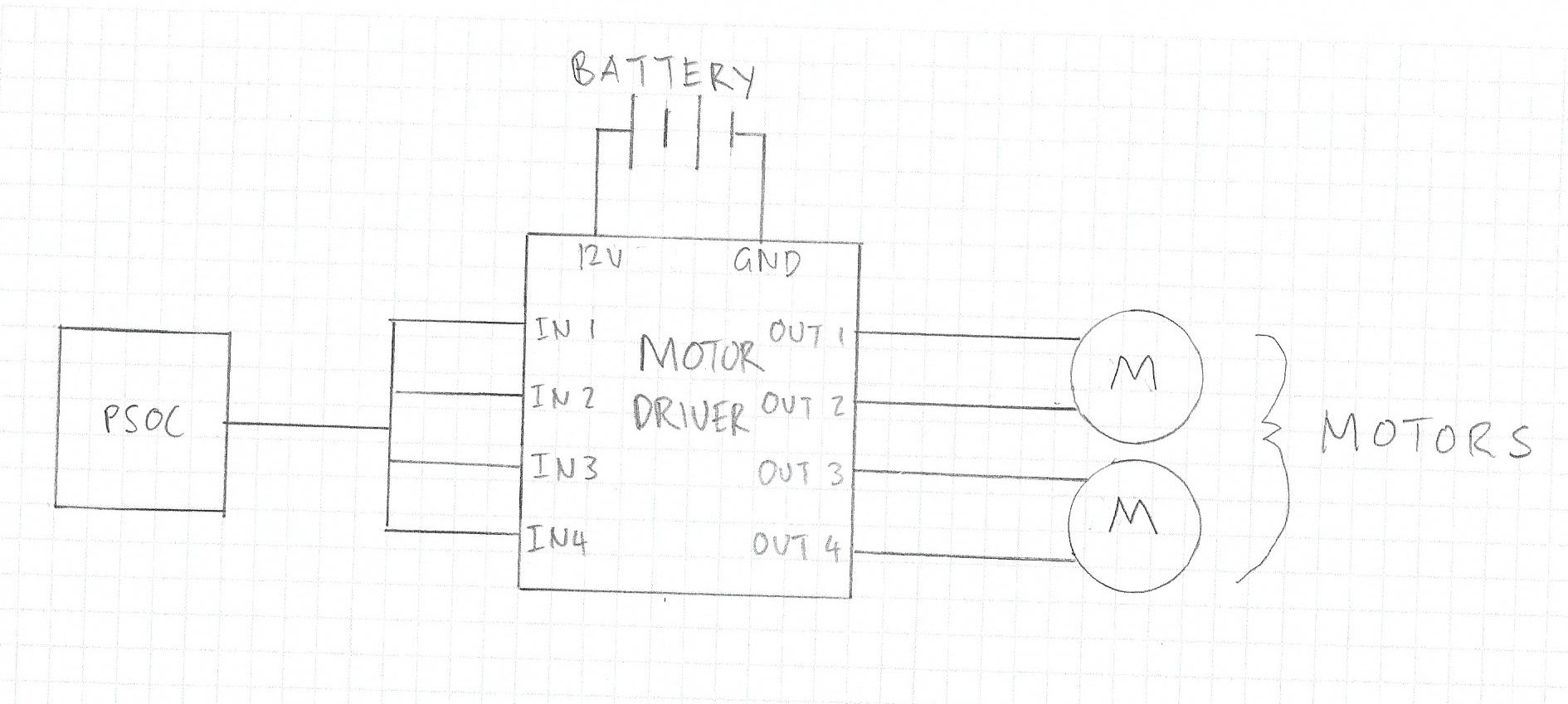
* The working principles of the actuators and sensors.
* How the power subsystem optimized the energy consumption of the robot.
* The electronic system integration.
* Alternative components that were considered.

**3.1 Overview of the actuators and sensors**

**3.1.1 Actuators**

Motors and motor driver

The motors work with 9.6V but normally us 12V. They have two terminals and the direction in which they spin depends on the polarity. There are two motors that the robot uses to move. Each motor has both of their terminals connected to 2 output pins on the motor driver. The input pins of the motor driver are connected to the PSoC which control what kind of output the out pins deliver. For example, if the input pins 1 and 2 receive digital inputs 1 and 0 respectively, output pins 1 and 2 will deliver power to the motors causing it to spin in one direction. If the inputs are reversed, the motor will spin in the opposite direction. Digital inputs of 1 and 1 or 0 and 0 will cause the motors to stop.



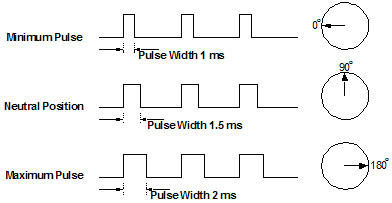
Linear solenoid actuator

The linear solenoid actuator consists of a wire wrapped around an actuator which is inside a casing. When there is a voltage sent through the wire, a magnetic field is established causing the actuator to move in one direction. On the casing itself is spring attached to it; once the voltage supply is cut off, the actuator is returned to its initial position by the kinetic force provided from the elastic energy of the spring.



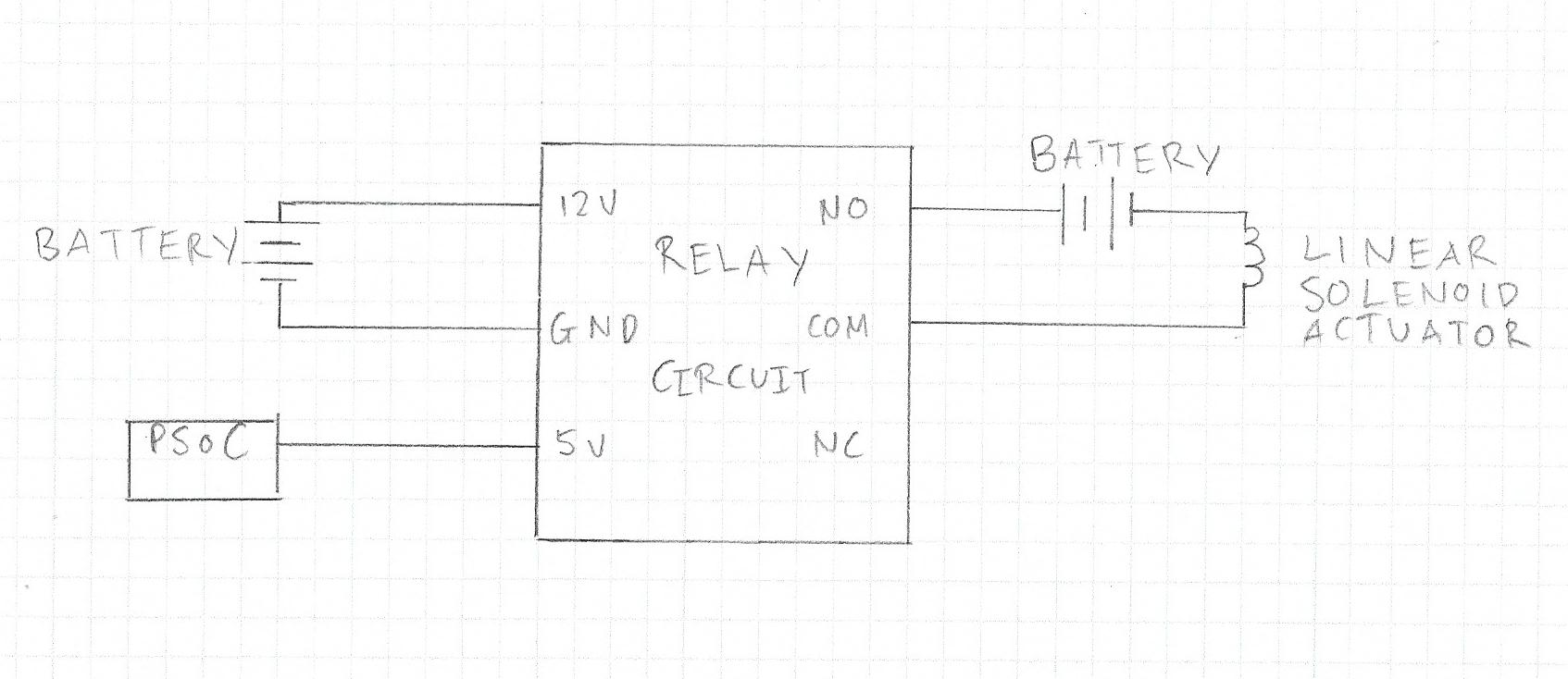
Servos

The servos have three terminals. Two is for the 5 volts supply and 1 is for a PWM (Pulse Width Modulated) signal from the PSoC. The servos can only turn from 0 to 180 degrees. The number of degrees it can turn is dependant on the length of time the PWM signal is high in a single cycle. The minimum time the PWM signal can be high is 1 ms; this will turn servo toward the maximum degree of rotation to one side. The maximum time the PWM can be high is 2 ms; this will turn the servo toward the maximum degree of rotation to the opposite.



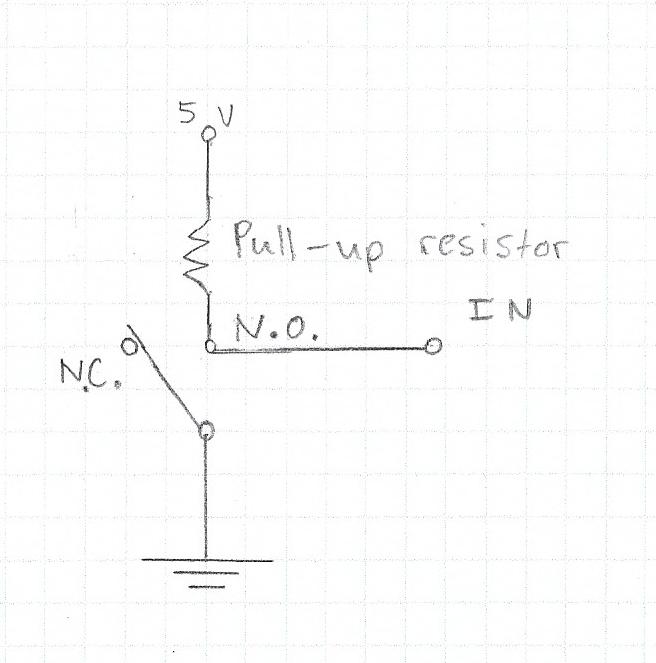
Relay circuit

The relay circuit is an electromechanical switch that has 6 terminals: A 5V control input that activates the switch, a 12V supply input ( can work with 9.6 volts) that magnetizes the electromagnet, a ground terminal, a normally open contact, normally closed contact, and a common junction. The normally open contact is connected to the battery which is connected in series to the solenoid. The solenoid in turn is connected to the common junction. When the control input is high, the electromagnet is active and flips the the switch inside the relay, completing the connection between the solenoid actuator and battery. Once the control input goes low, the electromagnet is off and the connection is broken.



Microswitches

The microswitch has 3 terminals: a normally closed contact, a normally open contact and a common junction. The input is connected t a pull-up resistor which is connected to the 5 volt supply. This configuration is all internal in the PSoC. The PSoC input is connected to the normally open contact (N.O.) on the microswitch. With the switch open, the input is high because it connected to the 5 volt supply but once the switch is closed the input is connected to ground and becomes low.



**3.1.2 Sensors**

Ultrasonic sensor

The sensor comprises of 2 transducers at the front, 2 pins for the supply voltage, and uses 2 data pins: trigger and echo. The trigger gets the input from the microcontroller to send a ultrasonic sound wave from the transmitter transducer. The sound wave bounces off the object and reenters the sensor through the reading transducer. The echo sends a signal back to the microcontroller to indicate when the sound wave has returned. From that, the time is obtained and the distance can be calculated by multiplying the speed of sound with half the time it took for the soundwave to be transmitted and reflected back.



Color Sensor

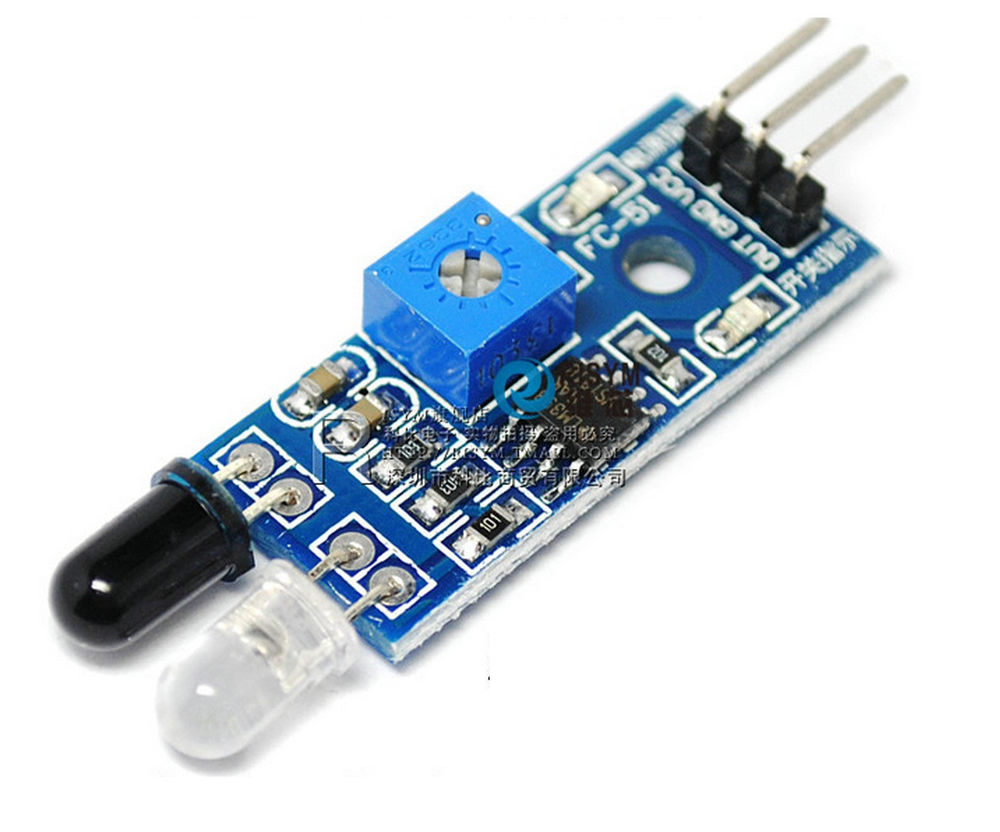


The sensor mainly consists of 4 white LEDs and are made of photodiodes with 4 types of color filters; red , green, blue and no color filter. There are 16 photodiodes with with the filter for each corresponding color.

The color sensor has 8 pins. 2 are for the voltage supply, 2 pins (S2 and S3) are used to change the photodiodes for a different color filter by driving digital inputs into them, and another 2 pins ( S0 and S1) are use to scale the output frequency by factors of 100%, 20%, 2% or completely powered down.

IR Sensor

The IR (Infrared) sensor consists of an infrared transmitter (IR LED), an infrared receiver (photodiode) , a potentiometer and 3 pins ( 2 for the voltage supply and one for the output). When the sensor is active, the transmitter emits an infrared signal which bounces of the object and reenters through the receiver. The output pin will read high depending on what the distance it is set for using the potentiometer.



**3.2 Power Management.**

The robot has access two sources of power: a 9.6V battery and a 4.5 volt supply from a DC-DC buck converter. The buck converter supplies power to all actuators and sensors with the exception of the motors, ultrasonic sensors and shaft encoders. The converter also powers the PSoC.

The buck converter is a type of switching voltage regulator. Therefore, it is very efficient in stepping down the voltage as compared to a linear voltage regulator. This means that less energy is lost to heat. The energy consumed by the various components was further reduced by decreasing the initial 5 volt output of the buck converter to 4.5 volts. This value is the minimum voltage required to power all the components connected to the buck converter.

**3.3 Integration of electronics**

A switch is turned on which connects the 9.6 volt battery to the motors, the relay circuit, the motor driver, and the DC-DC buck converter.

The buck converter supplies 4.5 volts to the IR sensor, color sensor, the crane and gripper servos, and the PSoC ( which in turn powers the microswitches, the shaft encoders and the ultrasonic sensors).

The relay circuit is connected to the boost converter which is connected to the linear solenoid actuator. This layout is similar to the layout discussed in 3.1.1 with exception of the boost converter. The relay circuit receives a 5 volt digital input from the PSoC in order to activate the linear solenoid actuator.

The motor driver receives 4 digital inputs from the PSoC to control the motors. The PSoC receives the counts from the phase A and B points on the shaft encoder.

The IR sensor outputs a digital signal to the PSoC when a puck is detected. The color sensor receives 2 PSoC input to set the color filter of the phototransistor and another 2 PSoC output to set the frequency scaling. The color sensor outputs a frequency into a PSoC pin.

Both the crane servo and the gripper servo each receive a PSoC input that controls the degree of rotation. There are 2 ultrasonic sensors with 2 input terminals each ( echo and trigger ) that receive inputs from the PSoC.

Lastly, the 2 microswitches are connected to 2 PSoC pins in the layout previously described in 3.1.1.

**3.4 Alternative components**

Puck detection: Ultrasonic sensor Vs. IR sensor

During the preliminary task, the ultrasonic sensor was tested to see whether it was suitable for puck detection. The results showed that it could detect the puck but was very inaccurate and sometimes gave false positives. The form factor of this sensor was also a problem because it needed to point towards the ground and there was limited space on the base of the robot. The IR sensor, when calibrated correctly, can detect the puck much more effectively and was much smaller than the ultrasonic sensor making it the obvious choice for puck detection.

Color sensing: Handmade color-sensor Vs. Pre-made Sensor

A handmade color sensor would consist of a photoresistor, red, green, and blue LEDs and a some resistors. The components themselves are very cheap and it is easy to build. However, when it comes to programming the sensor, it is hard in that it needs to be calibrated every so often if there is a change in its position. An uncalibrated sensor could lead to inaccurate results.

The pre-made sensor is effective in detecting the colors accurately. However, it was relatively expensive to obtain. Nevertheless, there was enough money in the budget to acquire it and it was worth it given how easy it was to work with.

Firing the puck: Motors Vs. Linear solenoid actuator

When attached with a wheel, the motor can launch a puck at large a distance. The motor was also provided and so it would be inexpensive to implement. However, given that firing action needed consists of a rotational force, it could be inaccurate when firing.

The other option was the linear solenoid actuator. The actuator just need to push forward in order to launch the puck making it very easy to control. However, this component was the single most expensive part that was obtained apart from the batteries. Despite that, since the actuator is very easy to control and was able to launch the puck at a sufficient distance, it was chosen because it was integral in making our strategy during competition simple.

Activation of firing mechanism: Motor drive controller Vs. Relay switch

The motor driver is a solid state device. This can be used to send a PWM signal to the solenoid actuator to make it launch the puck with a greater force at a higher voltage. However this is unnecessary as it was shown that the actuator launch the puck at a sufficient distance anyways which is why the much simpler and cheaper relay circuit was used instead.

Navigation: Digital compass Vs. Microswitches

The digital compass is very effective in aligning the robot to any degree. It is also effective in helping to robot get back on course if it gets lost. However, the compass is very hard to program and implement.

The microswitches are only used align the robot’s back to the wall making which makes it perpendicular. Therefore, it is limited to only to being 90 degrees to the wall. However it is easy to implement and is very easy to program to the PSoC and integrate into our strategy for the competition.

**4.0 SOFTWARE DESIGN**

The software of the robot governed the core decision making process of the robot during the competition. It dictated the fundamental behavior of the robot and guided the robot in the rules that were determined by the team. The software itself was programmed onto the robot using PSOC Creator 4.1 by Cypress and was programmed into the PSoC Microcontroller. In the perspective of the entire system, the PSoC Microcontroller was the central processing unit of the robot, in which it would take in the inputs of various sensors and would output to the correct actuators by following the software that was programmed into the microcontroller.

In this section, we will cover the following aspects of the software:

* The overall design aspects of the software.
* The main algorithms used by the robot during the robot.
* Other algorithms that were considered but was not used in the final competition.

**4.1 Overall Software Design**

The software of the PSoC microcontroller was programmed using the C language that was specific to the PSoC brand of processors. The differences between the actual language and the microcontrollers version of the language was very minimal. This allowed for the program to be interpreted as a normal C program, with a few additional libraries and API added. The program itself also used certain API’s that resembled real world devices that was programmable and configurable by the C code and this allowed the C Code to easily handle tasks and actions that would be extremely time consuming.

The main code of the program can be divided into two major parts, the initialization section and the program loop section. The initialization section consisted of a set of code that would only run once as soon as the robot was switched on. This section was mainly used to initialize, startup and configure functions and API’s that was used throughout the rest of the code. It was also used to initialize variables and states that were referred to in the program. The other set of code that was used can be found in the Program Loop. The program loop contained a set of code that was confined inside of an infinite for loop. This code would continue to run as for as long as the robot was powered. This was where most of the code for the functionality of the robot was present. The code for certain algorithms and decisions was coded into this section. Also, any functions that needed to be called was located in this section of code. Using both of these section helped to create structure and order to the code, so that the code can be easily referred to during the debugging stage and new code could easily be implemented into the code.

The extensive use of the program loop meant that the robot performed most of its functions using the polling method. The robot would run the code within the infinite for loop as fast as the internal clock of the microprocessor would allow it to. This meant that our code would read and write to the PSoC microcontroller extremely quickly and the response time of the code would be drastically improved. The polling methods major drawback was that it was imprecise due to it’s quick and reactive nature. Tasks that require accuracy had trouble with the polling method. In these cases, interrupts were used to perform the task instead of polling. But majority of the code was based around polling. In order to work around the disadvantages of the polling method, we used delays within the code, so that the operations of the robot isn’t completely instantaneously completed. Another work around was to use slower clocks in certain API’s. This allowed the API’s to function slower and more precisely. But this also caused certain API’s to not function correctly. In the case of the robot's functions though, the API’s didn’t react to the slower clock rates in any undesirable way.

The robot’s code made extensive use of predefined parameters in the form of Preprocessor Directives. These parameters were used to define certain traits and values for the instruction set of the robot. The parameters ranged from arena measurements, to measured values of the robot’s construction, to values that were constant throughout the program or to debug and calibrate the robot to the final competition conditions.

It is worth mentioning that despite the original plan of implementing the subsystems into the code of the robot was through the method of dedicating a function for each subsystems, in the final design of the software, this wasn’t the case. Some of the subsystems was implemented as functions and interrupts into the code, such as the locomotion system and the color detecting system. But in the case of other subsystems, it wasn’t feasible to write them into functions. Instead, they were implemented into the code as short sections of code into the main program loop and was accessed using if and switch statements.

**4.2 Software Algorithms**

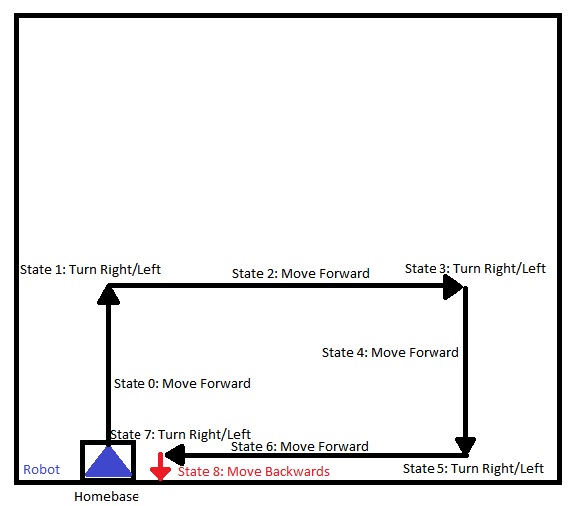
The algorithms used in the robot dictated the instruction the robot would do according to the situation. While the code takes in the input and stores it, the algorithms would then tell the robot what is happening in the arena and what needs to be done. There are a multitude of different algorithms that were implemented into the final design of the software of the robot, each with it’s own set of inputs, conditionals and outputs.

4.2.1 Puck Searching Algorithm

The puck searching algorithm was the algorithm used to move the robot into the puck area and to search of pucks in that area, and possibly in the area around that. The primary focus of this algorithm was to search for pucks mostly within that specific area, only branching out other areas occasionally. This allowed for the robot to focus all it’s resources into searching for pucks in the area of the arena where pucks are mostly likely going to be found, thereby saving time and resources on the other aspects. Despite the efficiency of this method, the major drawback to this is the possibility that as the competition progresses, the pucks will have been completely dispersed throughout the arena and the likelihood of finding pucks in the puck area will be less.

The algorithm was coded as if the robot was a finite state machine, with each part of the instructions coded into ‘states’. The state of the algorithm determined which part of the algorithm was currently running and which part to execute once the current state has finished it’s instruction. This allowed a linear sequential approach to the design of this algorithm, which allowed the robot to easily follow the instructions one after another. This also allowed more instructions to be coded into the algorithm without having to change the code too much or in an undesirable way. This however meant that the robot would have a very specific behavior and would only try to follow these instruction, making the robot unaware of any changes that have happened in the arena while the round is undergoing. The states of the algorithm was coded using a switch statement and the current state of the algorithm was stored in a variable.

The states and the instructions are illustrated below:



N.B. The diagram above is not to scale. The distance travelled may vary greatly, depending on the conditions of the final competition.

The robot, in searching for a puck, would go through all these states, in an attempt to search for pucks. To make the robot more robust, the states 1, 3, 5 and 7 would turn according to the starting base of the robot. If the robot started in the red or green base, then the robot would turn right. If the robot started in the blue or yellow base, then the robot would turn left instead. This allowed the implementation of the algorithm is different cases. The puck searching was only done in states 2 and 4, so that the focus can only be near the puck area. It is also worthwhile to note that in state 5, the state change was made when the robot detected the edge of the arena and in state 8, the robot turned and used the back switches to orient itself again with the arena and start the algorithm again. The algorithm would immediately be suspended when the correct colored puck was detected.

4.2.2 Color Sensing Algorithm

The color sensing algorithm executed whenever the IR sensor, which was placed near to the pre built color sensor, detected a puck underneath the robot. The sensor would then execute the algorithm which is used to determine the color of the puck in question. This particular algorithm followed a set of instructions that would involved mostly mathematical computations and logical decisions, rather than following a set of instructions.

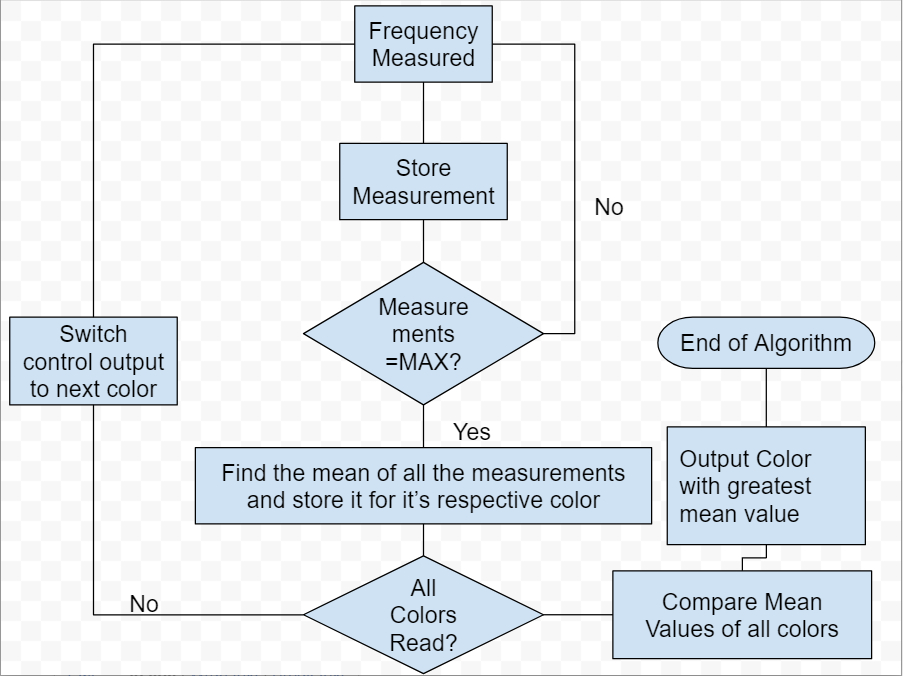
The color sensor contains a 8x8 array of photodiodes. The photodiodes are divided into 16 sets. Each set of photodiodes have a color filter on them. These colors are Red, Green, Blue and Clear (Transparent). To use only one of these set of photodiodes, we need to set a combination of pins on the color sensor to high or low. The truth table of this is found below.

|  |  |  |
| --- | --- | --- |
| **S2** | **S3** | **Photodiode Filter Colors** |
| 0 | 0 | Red |
| 0 | 1 | Blue |
| 1 | 0 | Clear |
| 1 | 1 | Green |

Using the Red, Blue and Green filters, we can determine which color the puck is by measuring the frequency output of the color sensor using three of these different filters and the one with the greatest frequency is the color of the puck in question.

The frequency is measured using a combination of a PWM and a counter. The PWM is connected to the enable of the counter and the reset as well. The counter then counts the number of times the signal goes high. If we set the PWM to have a certain clock speed, the number counts is equivalent to the frequency that is measured from the signal of the color sensor.

The sensor itself doesn’t quite output an accurate enough frequency, so the algorithm measures the frequency and then stores the value in an array and takes the measurement again. After a set of frequencies have been measured, the algorithm then takes the mean of all of these frequencies and stores the mean value for later use. Then the algorithm changes the filter and measures the frequencies again. After doing this for all the filters, the algorithm now has a mean frequency value for all the filter colors. The algorithm then compares them and outputs the one with the greatest value, which is the color of the puck underneath. The algorithm however only detects the color. Whether the color detected is the team color of the queen color is irrelevant to the algorithm.



4.2.3 Direction Searching Algorithm

This fairly simple algorithm is used to determine the direction the robot is currently facing. It is only used briefly in the puck handling algorithm to correctly orient the robot so that it can return back to home base.

The algorithm works by using two crucial pieces of information, the color of the starting base and the state the puck searching algorithm is in. Using this, the algorithm compares this to a lookup table that is programmed into it and then determines the direction of the robot. This information is then used by the puck handling algorithm to return back to home base and fire the puck in.



4.2.4 Puck Handling and Firing Algorithm

The puck handling algorithm is the algorithm that is used to pick up the puck that was detected, go back to the home base, aim at the hole, fire and return back to the starting position to search for pucks again. The algorithm itself resembles the structure and methodology of the puck searching algorithm, in which it follows a finite state machine like order to complete the task at hand. This algorithm however is much more complex than the puck searching algorithm simply because there are more instructions that needs to be done.

The puck itself is picked up by writing different duty cycles to the PWM’s that are controlling the crane and gripper servos. The robot itself briefly moves backwards and forwards to completely enclose the puck in the grasp of the robot. After the set of instructions to grip and pick up the puck is done, the direction searching algorithm is executed.

After the direction has been determined, the robot then orients itself according to the direction of the home base. Once it is in the correct orientation, the robot will move backwards until the back switches of the robot hit the wall of the arena, which corrects any error in motion caused by the puck searching algorithm. Once the robot has moved to the end of the arena, then it turns left or right (depending on the home base) and move backwards again until it hits the wall of the arena, like previously. Once it has done both of these task, the robot should be near the corner of the arena where the hole of the home base is. Afterwards, the robot turns towards the hole, lowers the crane and releases the puck, move a few centimeters forward to have the puck as close to the firing plate as possible and then fires the puck into the hole. Then it proceeds to turn back and tries to return to the starting position of the homebase.



4.2.5 Collision Detection Algorithm

Despite this algorithm proving to be difficult to implement and not being good enough to use in the final competition, it is still part of the robot’s original design and it was the chosen collision algorithm of the robot. The collision detection algorithm worked with the collision detecting strategy that was discussed and chosen. The robot first waited to see whether the object in front of the robot moved away within the next few seconds. If the robot doesn’t detect anything in front of it after that set time period, then it continues the previous task at hand. If the object still persists, then the robot would assume the object is a wall and would proceed onto the next state of it’s current algorithm. The algorithm follows those instructions as long as the robot was in the puck searching algorithm, which was robust enough to work along side with this algorithm, not with the puck handling algorithm. This is because the puck handling algorithm cannot skip states, as this would lead to the robot firing the puck inaccurately. Also, it is unlikely that the robot would undergo collisions during the execution of that particular algorithm because the robot would be near the home base, where the probability of finding another robot would be very low.



**4.3 Alternative Algorithms**

The following are descriptions of algorithms that are alternatives to the algorithms that selected and implemented into the system. The algorithms that were considered were analysed from a critical point of view in terms of design, feasibility, efficiency, the condition present and the limitations of the robot. Some of the algorithm designs listed below were implemented but were changed or modified to make the final algorithms of the robot.

4.3.1 Block Search and Spiral Search Algorithms.

Block search and spiral search are two algorithms that were briefly considered and implemented into the robot in order to search for the pucks in the puck area in the final competition, but were instead, chosen for a more simpler design. Block search was when the robot took on a fixed rectangular path along the puck area in order to search for pucks while spiral search spiraled downwards into the puck area. Both of these algorithms were much more efficient and effective in terms of their search time and detection accuracy.

The two algorithms weren’t considered for the final competition due to the limitations of the robot. The robot was unable to make fully accurate ninety degree turns. It the first couple of turns can be done accurately, but the error produced by the turns would carry forward and affect the other turns, making the algorithm uneffective as it should be. So both ideas were discarded as a result of them being unable to be implemented properly and so a more simpler design was applied, one that would require less turns than both of these algorithms.

4.3.2 Color Sensing Algorithm (Median)

A similar concept to the color detection algorithm was briefly considered. The algorithm would almost resemble the chosen idea entirely, except that instead of calculating the mean of the measured frequencies, the median is found instead. This leads to the algorithm being less prone to errors and noisy produced by the square wave produced by the sensor. But the idea itself was discarded when an attempt to modify the code was made. In order to calculate the median, the data had to be sorted, which meant that a sorting algorithm had to be implemented as well. When considered in terms of the number of values the sorting algorithm had to sort, the robot would take up a lot of time just to sort the values in order and then find the mean. This had to be done to a maximum of three times. So in conclusion, the implementing this algorithm wouldn’t be feasible enough for the final competition.

4.3.3 Preliminary Competition Collision Detection Algorithm

The collision detecting algorithm that was used in the preliminary competition was also considered to be used in the final competition. The algorithm consisted of it’s own set of instruction similar to the finite state machine like structure of the puck searching and handling algorithm. If the robot faced an object, it would then proceed to execute a set of instructions that would move the robot through a path that would avoid the obstacle and continue on its previous algorithm.

The reason why it wasn’t considered in the final competition was because the object in the preliminary competition was a stationary object in the middle of the puck area. This is an unlikely scenario that would occur in the final competition. So giving the robot another set of instruction to avoid the object by setting the robot on a predefined path would prove to be disadvantageous, especially since the robot would then once again have the risk of detecting the object again during the path. Also, the team weighed out both the options and deemed that this collision detection strategy is too obsolete for the purpose of the final competition and the better solution was implemented into the robot.

4.3.4 More Advanced Puck Handling Algorithm

This algorithm was considered to solve the problem of returning back to base. The picking up part of the algorithm was the same as the one that was used in the final competition, but the returning to home base part was largely different. The difference being that the solution to the problem was entirely accurate and much more innovative that the solution used in the final, but was never actually even attempted or tested at all.

The solution to going back to home back was being able to keep track of the number of counts that shaft encoders recorded and then using that to pinpoint the exact location of the robot. This means that the robot knows how far it has travelled in certain directions and can then know how to get back to base by travelling the same distance it travelled to reach where it is now. This method can also be used to orient the robot for the firing, using trigonometric calculations. In theory, this method would be far greater than the method that was used for the final competition.

However, it was discovered that the shaft encoders were not accurate enough for this solution. The wheels themselves gave off inconsistent errors, which lead to the number of counts recorded being only accurate up to a certain distance. In order to use this method, the counts had to be much more accurate and consistent. Therefore, it was not feasible enough to be coded into the robot.

**4.4 Software Design Conclusion**

From the algorithms stated above, the trend that can be seen is that the software techniques and algorithm designs favored simplicity the most, rather than efficiency. Despite the limitation of the robot, the program that was used to guide the robot throughout the competition was fairly simple and easy, which in the end, proved to be the crucial aspect in the success of the completion of the robot.

**5.0 STRATEGIES AND INNOVATION**

**5.1 Innovation**

Mechanical Innovation

The integration of the gripper and firing mechanism was one of our key innovations because of how it simplified how the robot picked up and launched pucks. With unintegrated mechanisms, the robot would have to realign itself in some way after dropping the puck in order to launch it. For example a robot might drop a puck right in front of itself then turn around 180 degrees to align the puck launching mechanism at the back of the robot in order to fire. Such realignments may introduce errors and create complications in firing the pucks.

Electronic Innovation

The main energy source for the robot is a 9.6V battery. The linear solenoid actuator used is normally supplied with 12V. If the actuator is only supplied with 9.6V, the force is insufficient to launch the puck into the slot. To solve this problem, we used a DC-DC boost converter to increase the 9.6V input voltage into the actuator to 12.5V. This lead to the actuator in outputting a higher force which was able to launch the puck into the slot consistently.

Software Innovation

A key innovation in terms of the software of the robot was the idea of separating each of the algorithms that had a fixed set of instructions into a finite state machine like structure, which allowed the algorithms to be debugged more easily, expanded on more efficiently and it also allowed for the exclusion of instructions that we’re similar to each other, for the sake of simplicity.

**5.2 Strategies**

The idea behind the underlying concept of the robot was to be able to detect the correct color of the pucks in the arena and pick, target and shoot in the pucks that was of the team’s chosen color, or if a point has already been scored, the queen colored puck. The strategies that were adopted by the team revolved around this basic idea, and most of the time, this is what lead to the design choices that were made throughout the course of the competition.

One major strategy that was thoroughly discussed and implemented, was the collision detection and obstacle avoidance strategy. The basic assumption that was made for this particular strategy was that there are two types of obstacles that the robot would face. The robot would have to overcome static objects like walls and disabled robots, or dynamic obstacles like robots that are actively moving around. For the static objects, the robot would try to actively avoid the obstacle in it’s path by redirecting it’s path and continuing on with it’s currently running algorithm. But in the case of a dynamic object, it was assumed that the object would eventually move out of the way of the robot. This meant that the robot had to merely wait until it’s path was clear, so that it could continue its instruction set.

Another strategy was the order of pucks that was to be shot. Originally, the robot would detect its designated color and then shoot it. Then it would search for the queen pucks only after that. But after thinking about the number of pucks available, and using the fact that the other team will also plan to shoot in queen pucks, the robot might end up searching for pucks that are no longer available in the arena anymore. Instead, it was decided that the robot would loosely follow that order, but the robot would not only search for queen pucks, but would also search for team colored pucks as well. This allowed for the robot to achieve the maximum number of points available.

An often neglected aspect of the robot that is ignored, but proved to be a vital strategic decision that was made, was the increase in the speed of the robot. The speed of the robot was controlled using pulse width modulated signals. This allowed for the robot to undergo different speed. The speed itself before the competition date was around 35% of the duty cycle of the PWM, but on the day itself, it was increased to around 45%. This means that the robot will be doing it’s tasks much faster than the opponent's robot, meaning that it will be able to score more points quicker. The drawback to this is that the robot might be more prone to wear and tear and the possibility of a malfunction happening was greater, but in the short run, the risks were almost negligible for the final competition.

**6.0 SYSTEM INTEGRATION AND TESTING**

**6.1 System Integration**

The system as a whole was integrated over the course of the semester as small subsystems that put together, make up the whole robot. Little by little the subsystems were incorporated into the robot to slowly get the robot to the finished product. This proved to be time consuming, since the software and hardware had to implemented together only in small parts. It also caused compatibility issues within the software when two of the subsystems were implemented. But using this method of implementation allowed the robot to have a meaningful order in the design, which allowed the robot to effectively perform for the final competition. It also allowed testing to be done much easier by isolating the subsystems and locating which of the subsystems is faulty. The subsystems also allowed the robot to be almost fully independent from each other in terms of hardware. This meant that if one subsystem malfunctioned, it didn’t cause the entire robot to be affected by fault. The other subsystems that were present in the robot was isolated from the damaged subsystem, which also protected the subsystem from any damages from the defective subsystem.

The system overview is shown below. This illustrates how each component comes together to make up the system as a whole:



The main components of the system described above are the PSoC Microcontroller and the 9.6 V Battery. The battery provides the power to all the components through the DC-DC Buck Converter, DC-DC Boost Converter or the Motor driver. The PSoC Microcontroller controls the input and output of all the components that are attached to it. In a sense, the PSoC microcontroller acts as the brain of the system and combines all the subsystems together to make up the robot.

**6.2 Testing, Debugging and Calibration**

The testing or the robot under final competition conditions were done near the end of the project life cycle, close to the competition date. This was because then the robot could be tested as a whole system. At first, the black box testing method was used to test if all the algorithms and components worked in accordance with the guidelines. This allowed for testing whether or not the system worked as a whole and that it was up to regulation for the final competition. Afterwards, any debugging that needed to be done was done using white box testing, where the code of the robot was analysed carefully and any bugs present was eliminated as the testing phase went.

A useful tool in the debugging phase was the PSoC Creator 4.1 debugging component. This allowed for close line by line analysis of the code present on the robot. The debugging component displayed all the variables that were present in the code and how they changed as the robot went through the program. It also allowed for close inspection of algorithms and functions to see if they executed according to the design. One shortcoming of the debugging option in PSoC Creator 4.1 was that it couldn’t be used to test any system that was not powered using the PSoC Microcontroller. This is because the PSoC microcontroller and the battery of the robot could not both be activated at the same time, which would cause high amounts of current passing into the microcontroller, causing the PSoC to malfunction. This meant that the locomotion system couldn’t be tested using this method. Also, testing with this system doesn’t work for real time systems, for example, the color detection systems. The debugging system only worked line by line. The debugging method is not able to update the present variables real time, while the robot is operational. Nevertheless, the real time systems were still tested under a different code monitoring method called UART (Universal Asynchronous Receiver Transmitter).

UART is a device that is present on the PSoC Microcontroller that is able to communicate with the computer that is programming the microcontroller. The device is able to send and receive information to the computer for display, in real time while the robot is running under final competition conditions. This allowed certain variables to be monitored closely and how they behave when the robot was subjected to different situations and different condition.

Using the two tools above, the software of the robot was tested and debugged extensively. This allowed the robot’s software design to be up to standard and ready for the final competition. The hardware aspects of the robot was also tested and debugged in accordance to the conditions that was outlined. Consistent measurements of the output voltage of the battery and buck converter made sure that the robot continued to function correctly and in compliance with the rules of the competition. Also, the individual components of the robot was tested constantly to ensure they are operational up until the final competition.

Calibration was another aspect that the robot underwent extensively. The robot used components that was prone to give off significant errors if the robot, as a whole, was not calibrated properly. Calibration was the final step of the testing phase because at this stage is when the robot functioned completely. In order to have the robot execute the tasks accurately, the robot needed to be calibrated according to the arena and the task at hand. Certain algorithms and instructions needed to be tweaked in order to effectively accomplish the objective of the robot design. The Calibration was done by reprogramming the robot with different parameters and rules sets. This method itself was time consuming but proved to be very effective in the final competition.

**7.0 PROJECT MANAGEMENT**

Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.It involves planning and monitoring project activities to ensure goals and objectives are accomplished on time.The development of an autonomous robot involves intense research, testing, and monitoring. An initial planning was carried out to allocate tasks and identify the main subsystems.Every week group meetings were held to plan out the following week’s work and to monitor the progress of the project.

**7.1 Team Roles**

Due to the scale of the project, the work has been divided accordingly amongst the group members based on their ability and experience. Though some parts of the project will be worked on collaboratively such as feasibility study, project report and robot testing to prevent absolute separation of the whole project.

To help our project move forward in an orderly manner our group member Farhana Hossain was appointed as the team leader. She was responsible for overseeing the project and interfacing with the manager. Calling meetings, finalizing and submitting reports, and making sure that team members are accomplishing their goals. Due to her skills in hardware and design and little knowledge in AutoCAD, she was in charge of robot design and construction, assembling chassis of Robot, 3D designing and modelling.She has experience in programming therefore helped Nayanaka occasionally and as she is more confident with electronics she helped Nico with design of the robot and other tasks like development of the motors,soldering etc.

Nico Sorreta was mainly in charge of hardware implementation and testing. These tasks involved building, troubleshooting, and integrating of the electronic subsystems to form the robot. Nico also provided programming support to Nayanaka occasionally. Nico is detailed-oriented and adept at electronic design; combined with his affinity for building and tinkering with electrical devices, he was motivated to build a robot that not only completed its tasks efficiently but also managed its power consumption wisely.

Nayanaka De Silva was in charge of handling the implementation of the algorithms, programs and the artificial intelligence of the robot. This also included understanding the computer architecture of the PSoC microprocessor, implementing the logic the robot will use extensively and the combining of the code used to program the decision making ability of the robot with the mechanical aspects the robot uses to function. This role was given to Nayanaka because of his immaculate coding capability and his knowledge on designing and implementing algorithms. He did not have any experience in coding for hardware and microprocessors but he understands the basic requirements of it and was willing to improve his existing knowledge of the subject to further the construction of the robot.

**Responsibility Matrix**

It is a method to display, in tabular form, the individuals responsible for accomplishing the work items in the Work Breakdown Structure: P – Primary responsibility, S – Support responsibility.The Responsibility Matrix is given below.



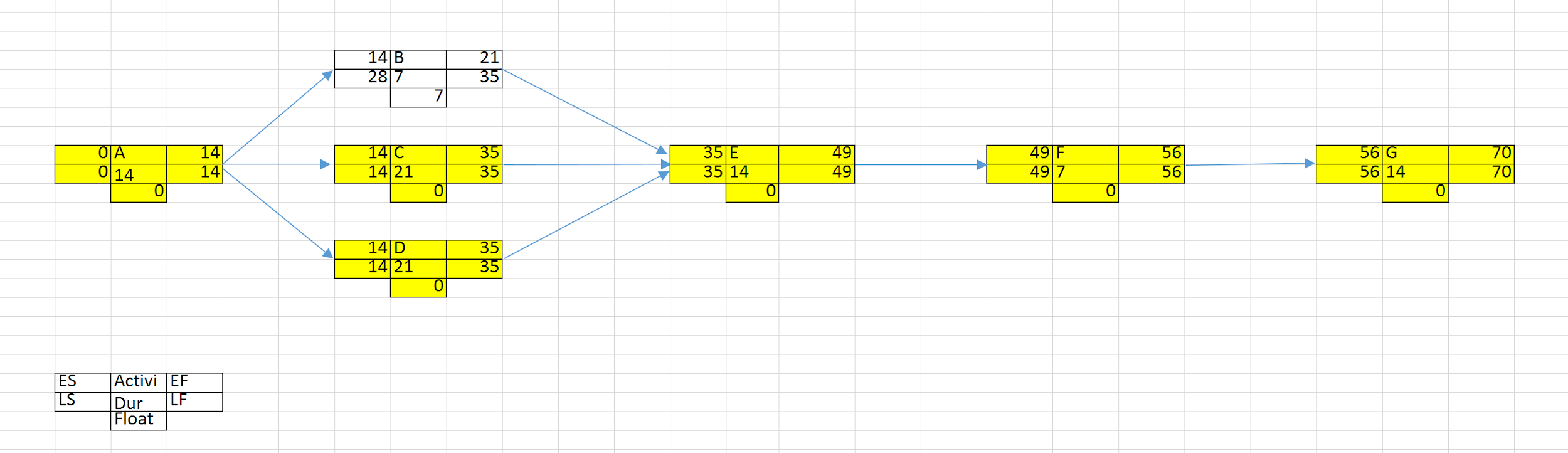
**7.2 Gantt Chart**

The Gantt Chart below shows the timeline of the project throughout the entire semester



**7.3 Critical Path Identification**

Critical path analysis is an important element of project planning.The critical path for the project is shown below.

****

**CPI: A,C,D,E,F,G**

|  |  |
| --- | --- |
| Activity |  |
|  |  |
| A | Locomotion |
| B | Collision Detection |
| C | Puck Detection |
| D | Color Detection |
| E | Gripping Mechanism |
| F | Firing Mechanism |
| G | Integration |

**7.4 Project Budget**

The following table shows all our expenses incurred during the course of this project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Supplier** | **Item name** | **Quantity** | **Price per item** | **Cost + shipping** |
| **MSC Supply & Service** | Color Sensor (TCS 3200D) | 1 | 26 | 26 |
|  | DC-DC buck converter | 1 | 10 | 10 |
|  | T plug | 2 | 2 | 4 |
|  | G4-271ADXL345 compass | 1 | 18 | 18 |
|  | 40 pcs ribbon jumper | 1 | 7 | 7 |
|  | Single channel relay module | 1 | 15 | 15 |
|  | IR sensor | 1 | 12 | 12 |
|  | M33 Stand offs | 10 | 0.7 | 7 |
|  | 12 DC Solenoid actuator (Push) | 1 | 39 | 39 |
|  | 40 pcs connector sockets | 2 | 2 | 4 |
|  | 40 pcs connector pins (Straight) | 1 | 1.3 | 1.3 |
|  | 40 pcs connector pins (Angled) | 1 | 2 | 2 |
|  | DC-DC boost converter | 1 | 15 | 15 |
|  |  |  |  |  |
| **KAG Systems SDN BHD** | M3 nuts and bolts x10 | 3 | 4 | 12 |
|  |  |  |  |  |
| **Lelong** | Ball caster | 2 | 3.9 | 7.8 |
|  |  |  |  |  |
| **C&C HOBBY MARKETING** | NiMh 6V 1.8Ah Battery | 3 | 29 | 87 |
|  |  |  |  |  |
| **Hi-Scan Wholesale SB** | Pvc model board | 1 | 5 | 5 |
|  | Acrylic Sheet | 1 | 5.2 | 5.2 |
|  |  |  |  |  |
| **Nasamas Electronics SDN BHD** | 40 pcs jumper wire | 1 | 7 | 7 |
|  | 2 pin rocker switch | 1 | 1.2 | 1.2 |
|  |  |  |  |  |
| **Mobicon-Remote Electronic SDN BHD** | LM7805 Voltage Regulator | 1 | 2 | 2 |
|  | 0.1 uF Electrolytic Capacitors | 5 | 0.5 | 2.5 |
|  | 0.47 uF Electrolytic Capacitors | 5 | 0.3 | 1.5 |
|  | Ribbon Cables | 1 | 14.5 | 14.5 |
|  |  |  | **TOTAL** | **306** |

Table : Budget for the Project

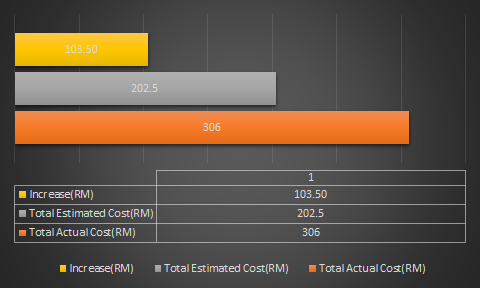


Figure 12 : Actual,Estimated and Increase in Cost

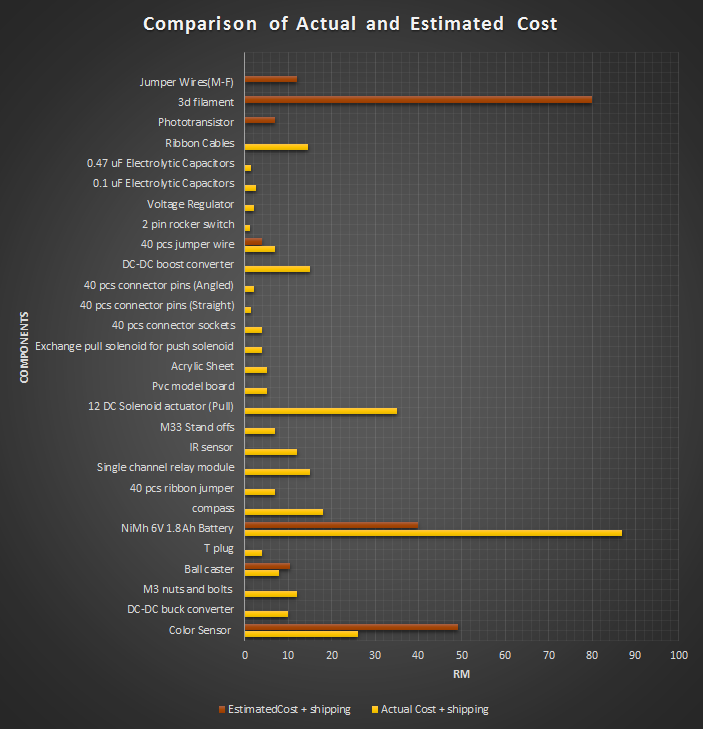


Figure 13 : Comparison of Actual and Estimated Cost

As we can see from the comparison tables above,initially when we planned our budget for the project some important components were included but a lot of components were missing.As we progressed, working with each subsystem we bought components according to our needs but keeping it within the budget.

**8.0 RECOMMENDATIONS**

We tried to achieve the simplest and finest design but there still remain a large possibility for improvement. Some of the aspects of the project that can be further improved are listed below that would improve the functionality of the robot.

* In our robot design we have used two ultrasonic sensors at the front, to further improve the functionality ultrasonic sensors can be placed on the sides and back to avoid any robot hitting from the sides and back.

* Navigation was one of the most daunting subsystem. We have used the encoders in our design for movement, but this can be improved with a very useful component: digital compass. We have come up with a correction code to make the robot move in a straight line but instead a digital compass can be used which is way more accurate with an error of just 1 or 2 degrees.

* We have also used the encoders to make 90 degree turns, which again is not always perfect, either it overshoots or undershoots so we had to carry out trial and error methods and come up with a code to make almost perfect turns. To improve this criteria a digital compass can be used to precisely make the robot turn 90 degrees.
* Another option is to use PID control system to control the robot. It provides robust performance for correction mechanism and speed adjustment.
* Implementation of an RTOS (real-time operating system) as a foundation of the PSoC’s software would have a positive impact of how are robot runs. An RTOS would be able to switch between tasks based on its priority, which is important in the competition as the variables (eg. the amount pucks are available), during which, can change at any given moment.

**9.0 CONCLUSION**

In the ECE3091 project, we had to apply our knowledge and skills in electronics, mechanical construction, and programming. Intense research and analysis, and skills like budgeting, communication, teamwork, time management and work planning helped us to design and construct an efficient autonomous robot with minimum power consumption while keeping the design simple. With over 10 weeks of work, all the 7 subsystems: Locomotion, Power, Gripper, Firing, Color Sensor, Collision Detection and PSOC microcontroller, were integrated to the robot which was able to function fully by navigating in the arena, avoiding obstacles and walls, detecting and read different puck colors, and shoot the correct pucks successfully.

We believe we have achieved our goal to a great extent.