School of Engineering, Malaysia Campus

ENG3091: Engineering design

The Monash Puck Shifting Robots Project Report

Team ID: 15
System Integrator: Chia Kok Bond (30893062)
Programmer: Mah Yuen Yee (30213673)

Submission Date: 24/10/2021

Contents

1.0 Executive summary

1.1 General project description

In a team of 2, we are tasked with building an autonomous robot to shift pucks, whose colour matches the colour of adjacent capture zones, from the respective collection zone into the respective capture zone through dragging or flicking action on the arena as shown in Fig. 1 during the design competition held in Week 12. The competition will be divided into 2 rounds, with each round being 5 minutes and consisting of 2 half-rounds. The robot would restart every half round with random puck allocation, i.e., puck colour in each collection zone is not known beforehand.

During each half round, the robot would be starting on the starting base and move towards the puck collection zones. Once the robot reaches any of the collection zones, it should detect the colour of the puck within and check if the puck colour matches the colour of the respective puck capture zone. If the colour matches, the puck is deemed valid, and the robot should shift the valid puck into the adjacent respective puck capture zone using a flicking or dragging action. Successful shifting of a valid puck into the respective capture zone signifies a successful capture which awards 1 point. The robot should try to get as many points as possible to win the competition while following all the rules set.

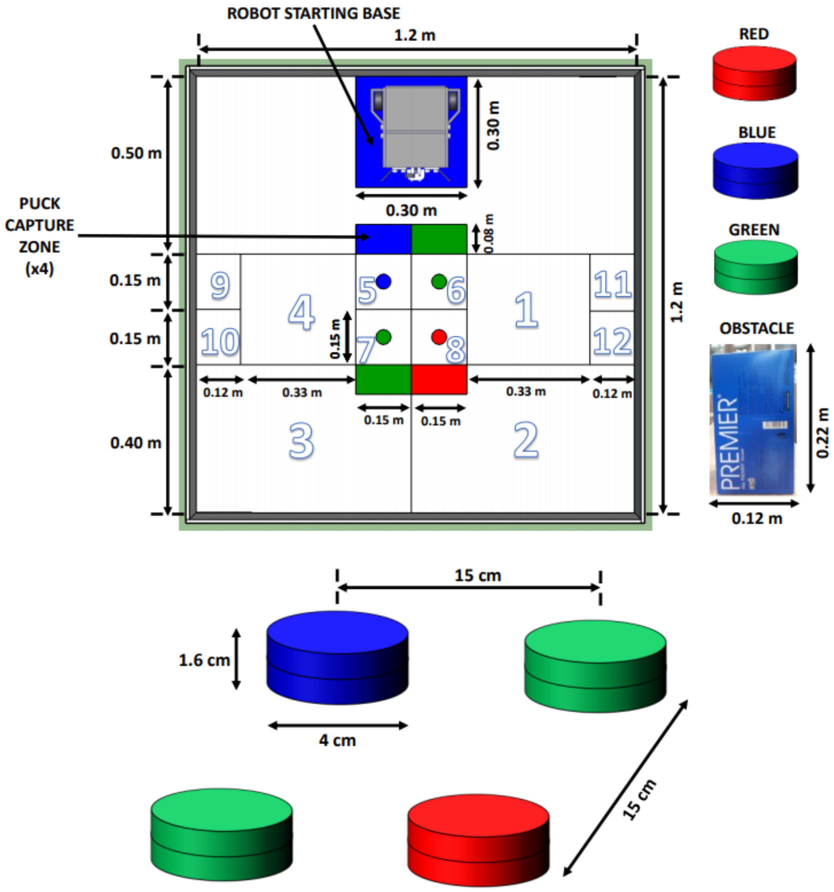


Fig. 1. Competition arena overview and puck overview.

1.2 Problem statement

Given the rules and project description above, there will be several design problems that need to be tackled to achieve the project objective. The design problems will be classified into a few categories: Robot construction, Movement, Colour sensing, Puck handling, and Mixed which involves several other categories in the problems.

1. Robot construction
    a. The robot must be totally autonomous once it enters the arena.
    b. The robot must only use the PSoC5LP Prototyping Kit (CY8CKIT-059) provided in the project kit as the microcontroller.
    c. A budget of only RM380 is allocated for purchasing extra commercially available items atop the already provided project kit.
    d. The robot must only be powered by alkaline or Ni-MH rechargeable batteries. For rules compliance inspection, the batteries should be visible on the robot.
    e. The robot must be in one piece, i.e., parts are structurally connected and not attached by wires or strings, throughout the competition.
    f. The robot must not have hanging wires or a poor casing.
    g. Both the length and width of the robot must not exceed 30cm.
    h. The robot should have enough power to run for 2 rounds, each lasting 5 minutes.
2. Movement
    a. The robot should never touch any of the obstacles placed in zones 9 to 12.
    b. The robot should never travel outside the edge of the arena.
    c. The robot must use a differential drive mechanism.
    d. The robot must not use any line or indicator on the arena for activity guidance.
3. Colour sensing
    a. The robot should be able to sense red, green, and blue correctly.
    b. The robot should know the colour of each capture zone.
4. Puck handling
    a. The robot must not manipulate pucks without touching them.
    b. The robot must not manipulate more than 1 puck at a time.
5. Mixed
    a. The robot should only shift valid pucks (whose colour matches the respective capture zone) to be completely inside the correct capture zone for the capture to be awarded 1 point.
    b. The robot should never manipulate invalid pucks (whose colour does not match the respective capture zone).
    c. The robot should visit, i.e., detect the puck, sense the puck colour, and handle the puck based on its colour, at least 3 out of the 4 puck collection zones in each half-round.
    d. The robot must not manipulate pucks when not colour sensing nor capturing.
    e. The robot should have no idle time in the competition.
    f. A captured puck should stay in the capture zone for at least 2 seconds.

g. The robot must not touch the starting base when visiting and manipulate pucks in zones 7 and 8.
h. The position of the before-capture puck in each collection zone is fixed throughout the competition.
i. After capturing a valid puck, the robot should not cover the black circle, where the valid puck stayed in the collection zone before capturing, to allow the system integrator to replace the puck.
j. The same camera must be used to capture both the robot activity and puck capture activity.
k. The replaced puck must be exactly inside its black circle with no error margin.

In short, the robot must be able to solve all the problems mentioned above, consistently and reliably while trying to accumulate as many points as possible during the competition.

1.3 Crucial aspects of the design

1.3.1 Shortest route strategy

According to the project rules, the robot must visit at least 3 collection zones. To follow the rule while attempting to achieve as many points as possible, i.e., capture valid puck as many times as possible, we designed a movement route such that the robot would first visit all collection zones, then focus on the last valid puck (according to the puck collection zone numbering) for continuous repetitive capturing.

Free zones, which are zones 1, 2, 3, and 4, are empty as shown in Fig. 1. This may prompt the robot to visit all the collection zones by moving through these zones since there is no obstacle nor pucks there that obscure robot movement. However, the shortest possible route, i.e., the route where the robot directly moves in and through the puck collection zones and puck capture zones, is preferred as it results in a shorter travel time which allows more time to capture pucks. Hence, the movement route is designed as such:

1. A half-round starts with the robot on the starting base.

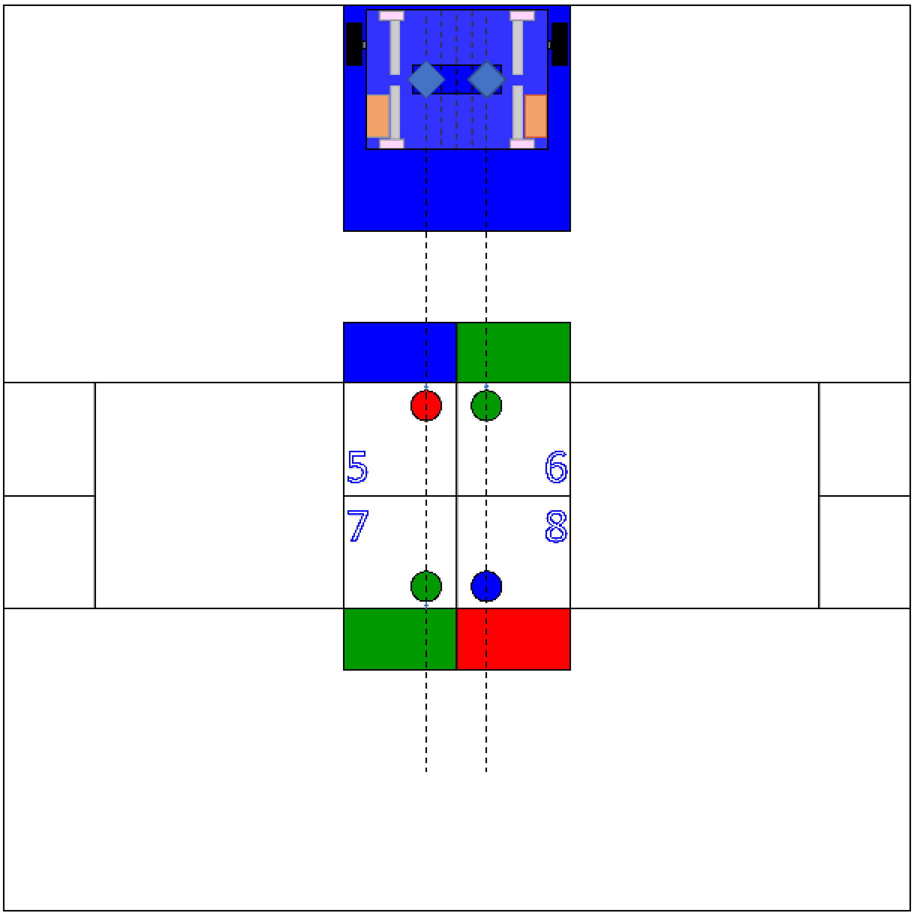


Fig. 2. Step 1 of movement route.

2. The robot moves forwards into puck collection zones 5 and 6 to sense the pucks simultaneously.

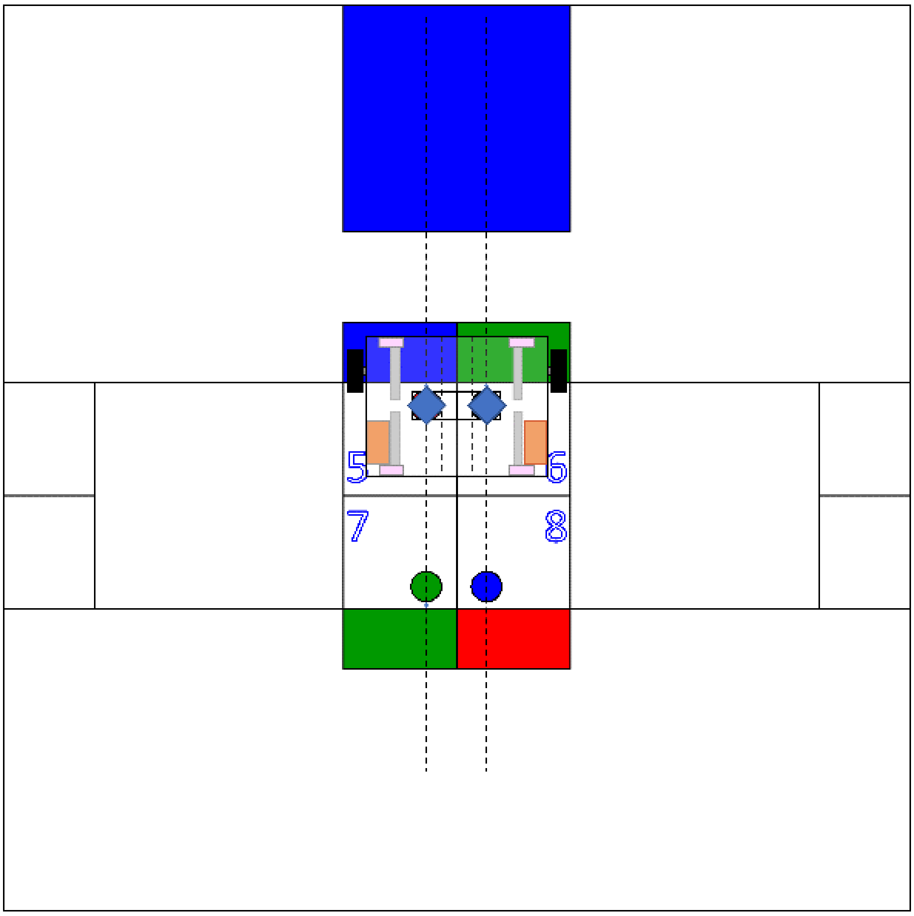


Fig. 3. Step 2 of movement route.

3. The robot moves forwards to have a 1cm gap between the back chassis edge and the nearest puck edge as shown in Fig. 4(a). This is to ensure the robot does not cover the black circle in the collection zone where the to-be-captured valid puck is in. The robot thus captures the valid puck(s) in zone 5 and/or zone 6 via the 4-servo motor puck dragging mechanism as explained in Section 1.3.2. As shown in Fig. 4 which shows an example of the puck in zone 6 being valid, the robot only moves forwards and backwards during puck capturing and involves no point turning.

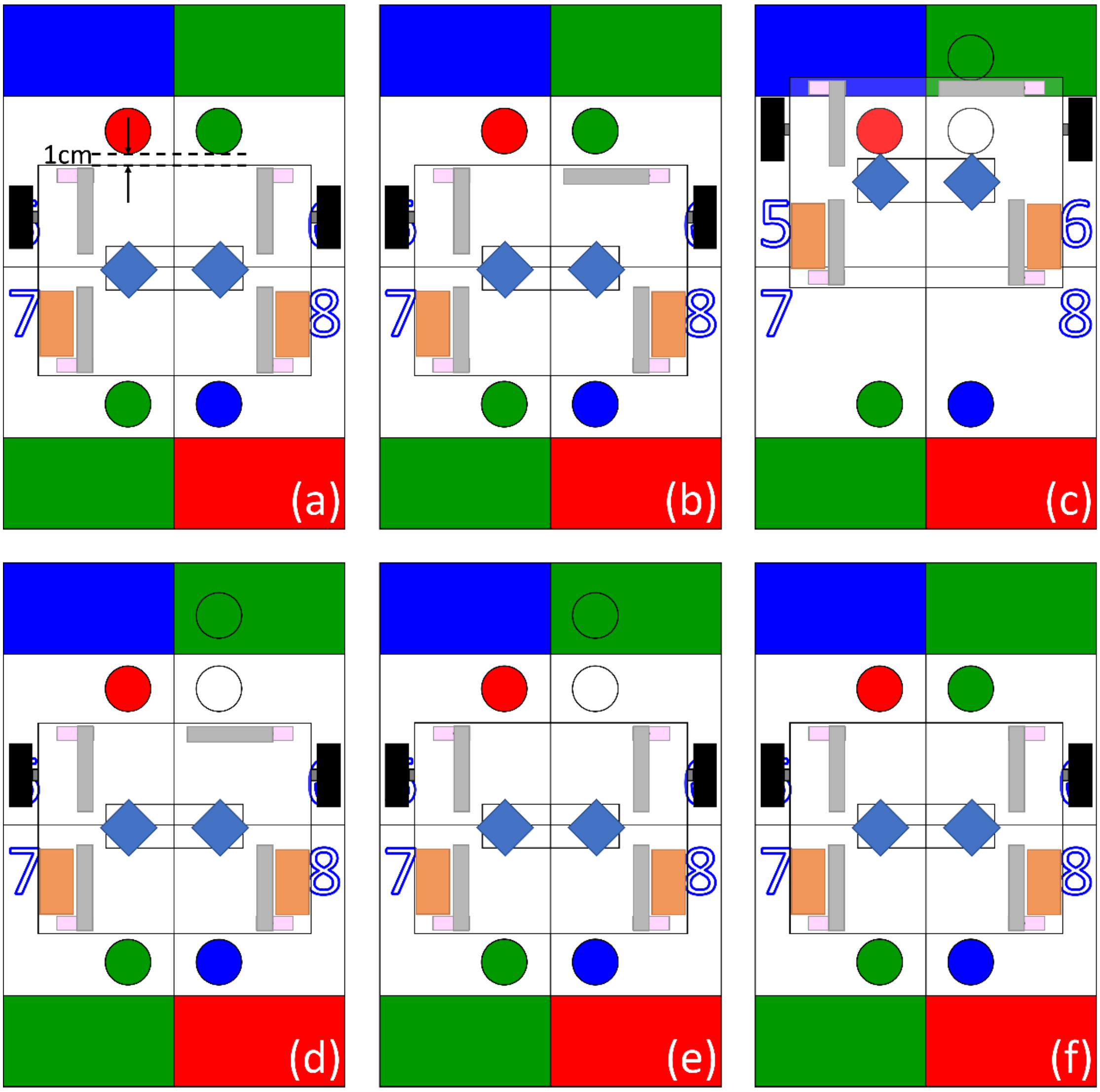


Fig. 4. Step 3 of movement route. (a) 1cm gap between the back chassis edge and the nearest puck edge. (b) The robot closes the servo rod 6. (c) The robot moves backwards to push the valid puck into the capture zone via its closed servo rod 6. (d) The robot retracts to its initial position. (e) The robot opens the servo rod 6. (f) The robot waits 8 seconds for manual puck replacement to be done.

4. The robot moves forwards into puck collection zones 7 and 8 to sense the pucks simultaneously.

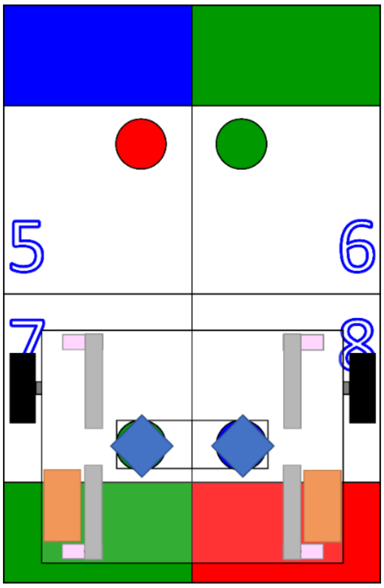


Fig. 5. Step 4 of movement route.

5. The robot moves backwards to have a 1cm gap between the front chassis edge and the nearest puck edge as shown in Fig. 6(a) and captures the valid puck/s in zone 7 or/and zone 8 if there is any. As shown in Fig. 6 which presents an example of the puck in zone 7 being valid, the capturing process is similar to step 3. By the same token, the robot only moves forwards and backwards when capturing.

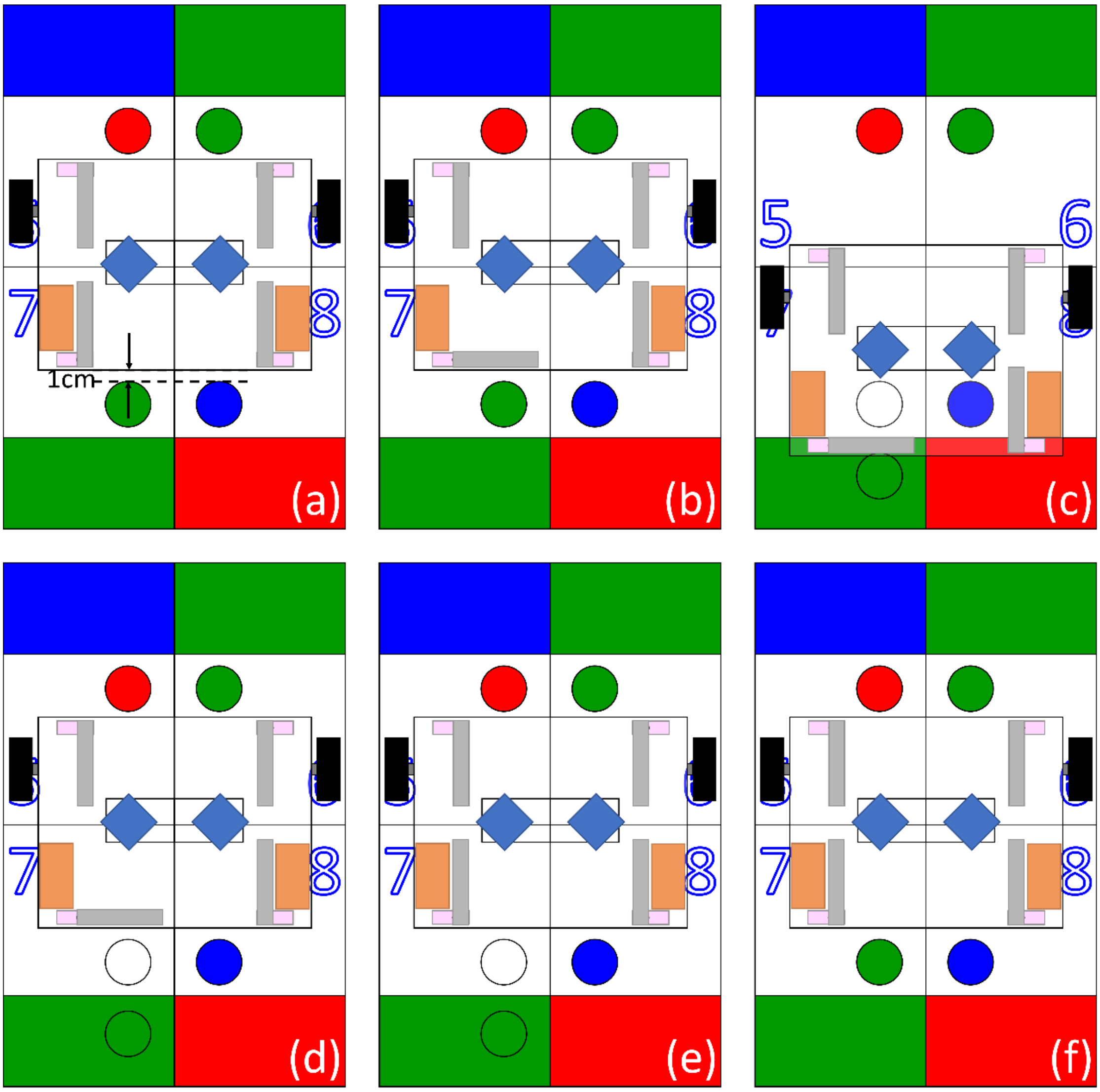


Fig. 6. Step 5 of movement route. (a) 1cm gap between the front chassis edge and the nearest puck edge. (b) The robot closes the servo rod 7. (c) The robot moves forwards to push the valid puck into the capture zone via its closed servo rod 7. (d) The robot retracts to its initial position. (e) The robot opens the servo rod 7. (f) The robot waits 8 seconds for manual puck replacement to be done.

6. As the robot is designed the focus on the last valid puck (the puck in zone 7 in this case) for continuous capturing, it stays 1cm away from the nearest edge of the puck in zone 7 as shown in Fig. 7(a) and keeps capturing the puck, i.e., repeating Fig. 7(b) to Fig. 7(d) until the end of the half-round. The puck handling mechanism is explained in Section 1.3.2.

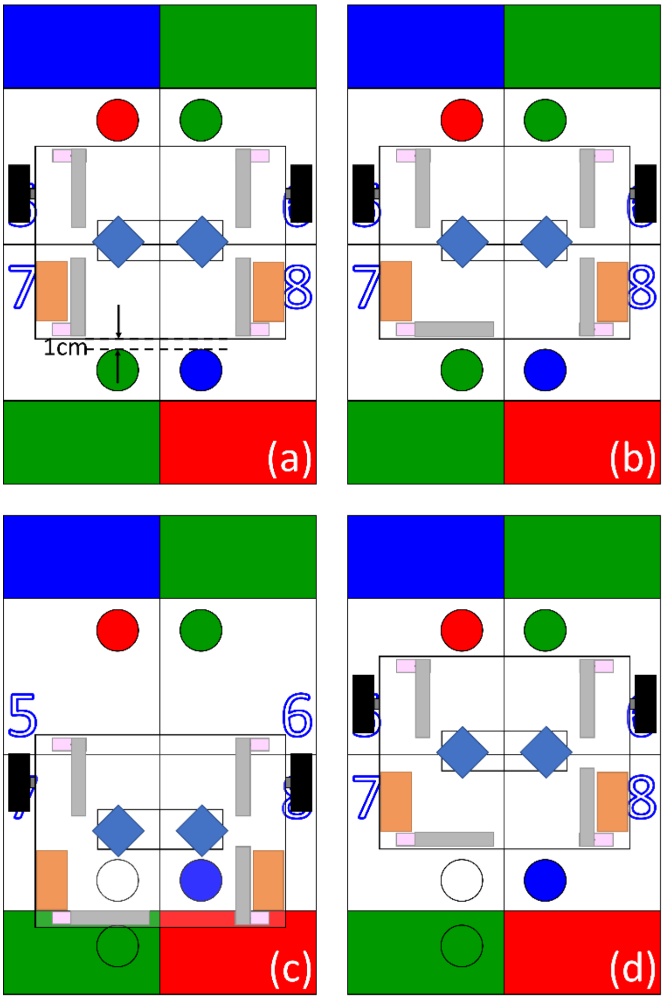


Fig. 7. Step 6 of movement route. (a) 1cm gap between the front chassis edge and the nearest puck edge. (b) The robot closes the servo rod 7. (c) The robot moves forwards to push the valid puck into the capture zone via its closed servo rod 7. (d) The robot retracts to its initial position.

This is the robot movement route for each half-round. The robot visits all collection zones in the shortest time and distance possible, leading to longer puck collection time. In addition, the robot stays around the centre, between the pucks after visiting all the collection zones. This allows the robot to continuously capture the valid puck immediately after a slight adjustment of the position depending on where the last valid puck was. There will be no need for further movement by the robot to initiate the repeated puck capturing process. Besides, the shorter the travel distance, the fewer dead reckoning errors due to physical interactions between the robot and the arena surface. In essence, this shortest route strategy shortens the travel time to allow for more puck collection time all while reducing error.

1.3.2 4-servo motor puck dragging mechanism

To make the shortest route strategy possible, it is obvious that the puck handling subsystem requires more than 1 servo motor. Therefore, a 4-servo motor puck dragging mechanism is used as the puck handling subsystem. Each servo motor is attached to each chassis corner as shown in Fig. 8 to be responsible for shifting the puck in each collection zone.

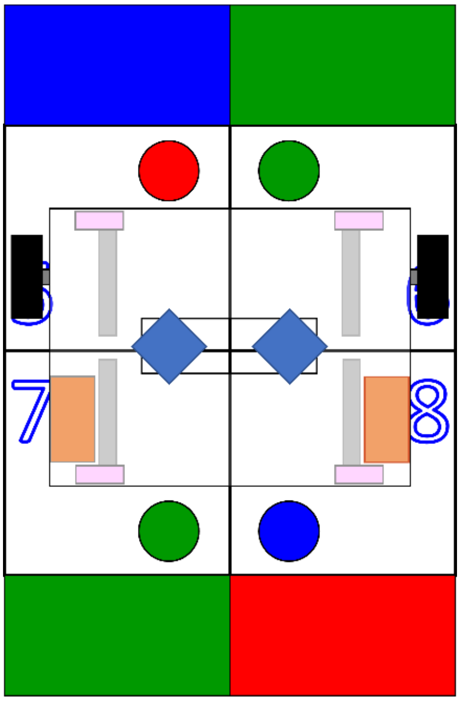


Fig. 8. Each chassis corner has a servo motor to be responsible for puck shifting in each collection zone.

The process to capture a valid puck is as follows:

1. The robot adjusts its position to have a 1cm gap between its chassis edge and the nearest edge of the puck.
2. The robot closes the corresponding servo rod i.e., rotates the corresponding servo rod clockwise or anticlockwise towards the puck by 90 degrees, so that the servo rod is parallel to the back or front chassis edge.
3. The robot moves forwards or backwards to push the valid puck with its closed servo rod until the valid puck is completely inside the corresponding capture zone.
4. The robot retracts to where it was in step 1.
5. There are 2 cases that can happen in this step:
    a. If the valid puck is not for continuous capturing, the robot opens the corresponding closed servo rod, i.e., rotates the servo rod by 90 degrees, so that the servo rod is parallel to the side chassis edge as shown in Fig. 8. This allows the robot to continue its motion sequence without any unintentional puck manipulating.
    b. If the valid puck is for continuous capturing, the robot keeps its corresponding servo rod unrotated.

6. The robot waits 8 seconds for the system integrator to place the puck back to the corresponding black circle.
7. If the valid puck is not for continuous capturing, the robot continues its motion sequence as described in section 5.1. Else if the valid puck is for continuous capturing, the robot repeats steps 1 to 6 until the end of the half-round.

An example of the process is provided in Fig. 9 which assumes the case where the robot is capturing the valid puck in zone 6 which is not for continuous capturing.

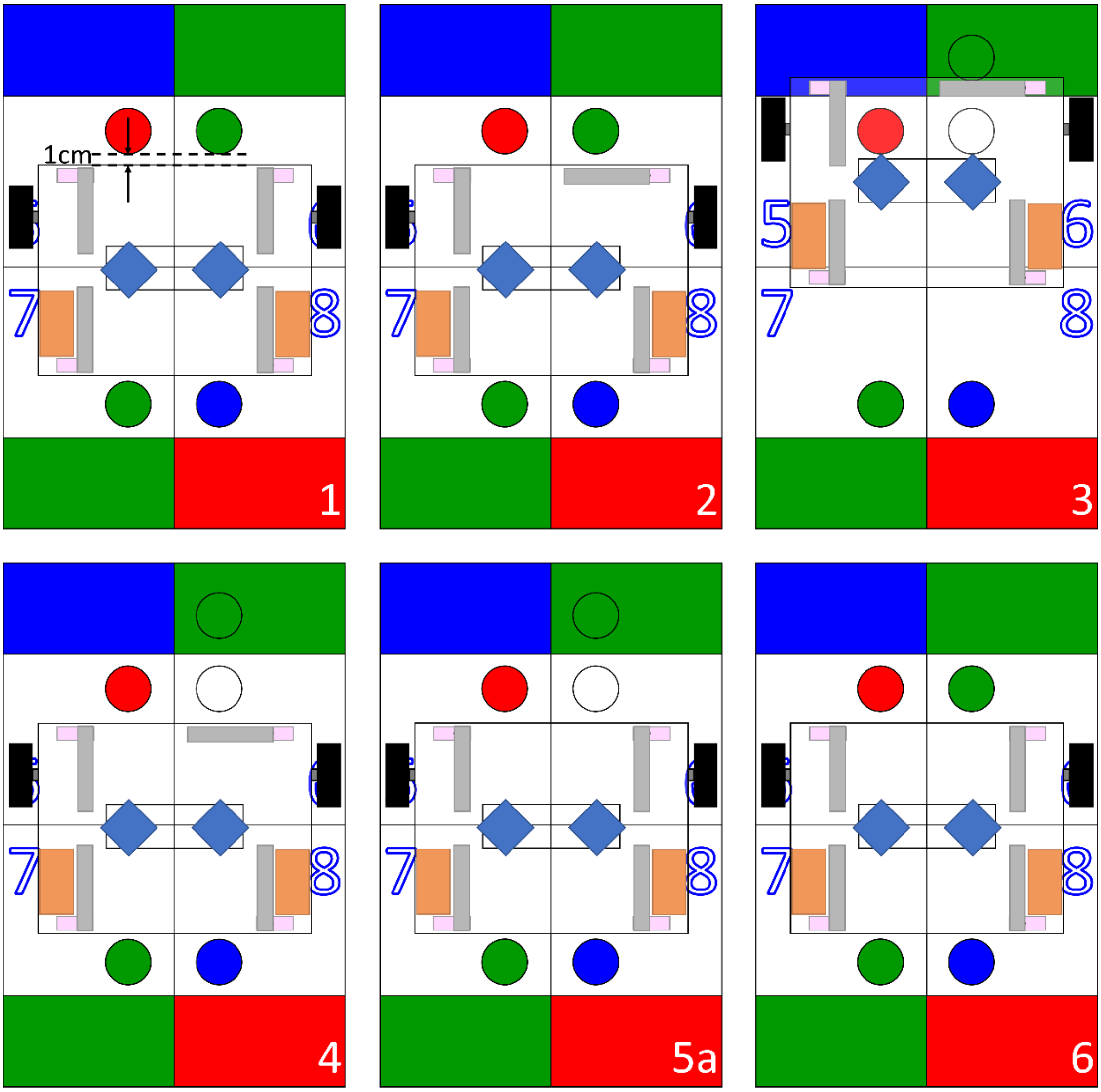


Fig. 9. An example of the process using the 4-servo motor puck dragging mechanism the capture a valid puck assuming the valid puck is in zone 6 and not for continuous capturing.

As described above, the 4-servo motor dragging mechanism requires robot movement to complete a capture. Nevertheless, this dragging mechanism complements the shortest route strategy well to allow a short amount of travel distance and travel time as well as the immediate capturing of valid pucks after the mandatory 3 puck zone visits.

1.4 Overview of the robot design

1.4.1 Overview of the robot structure

The top-down view, underside view, front view, side view, back view, and 3D view of the robot are provided to show an overview of the robot structure. Wiring and minor details in each component are omitted for clarity.

1.4.1.1 Top-down view of the robot

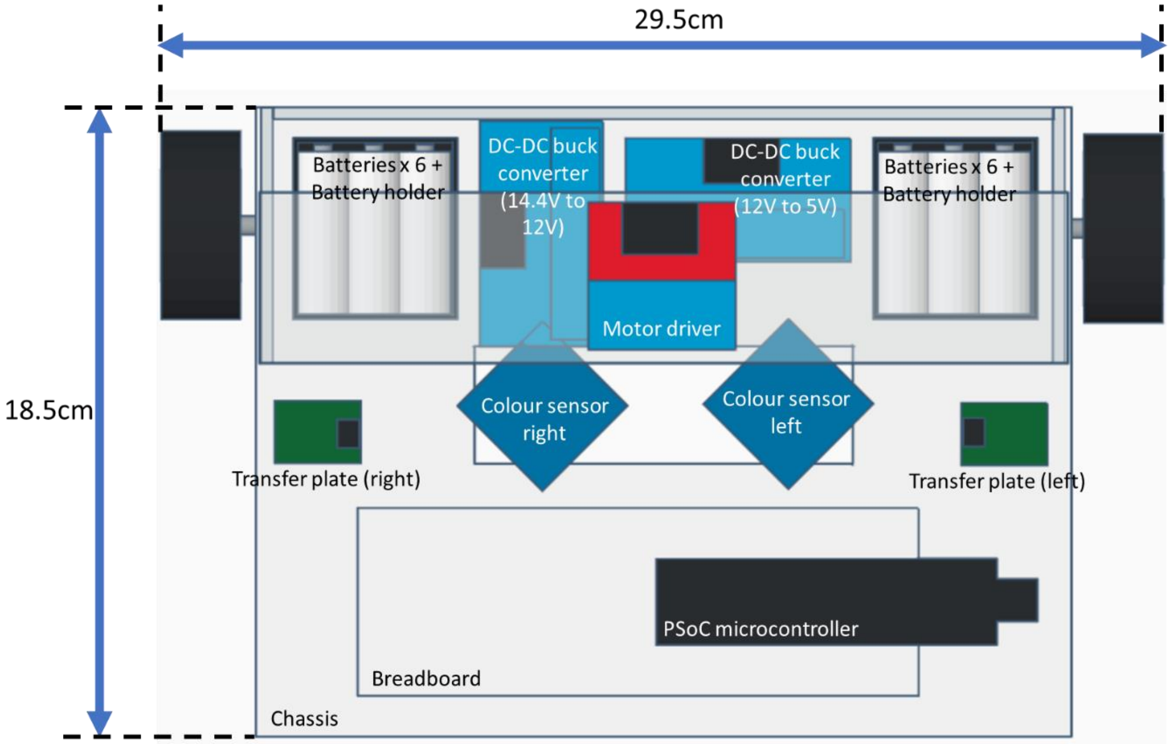


Fig. 10. Top-down view of the robot.

The robot uses only 1 level of robot base, i.e., chassis, for system integration as it is large enough to accommodate all components with the use of some corrugated boards to lift the motor driver up for neater component arrangement. The PSoC microcontroller, i.e., PSoC5LP Prototyping Kit (CY8CKIT-059), with pins soldered, is inserted into a breadboard to allow all the connections between the components and the microcontroller via jumper wires.

1.4.1.2 Underside view of the robot

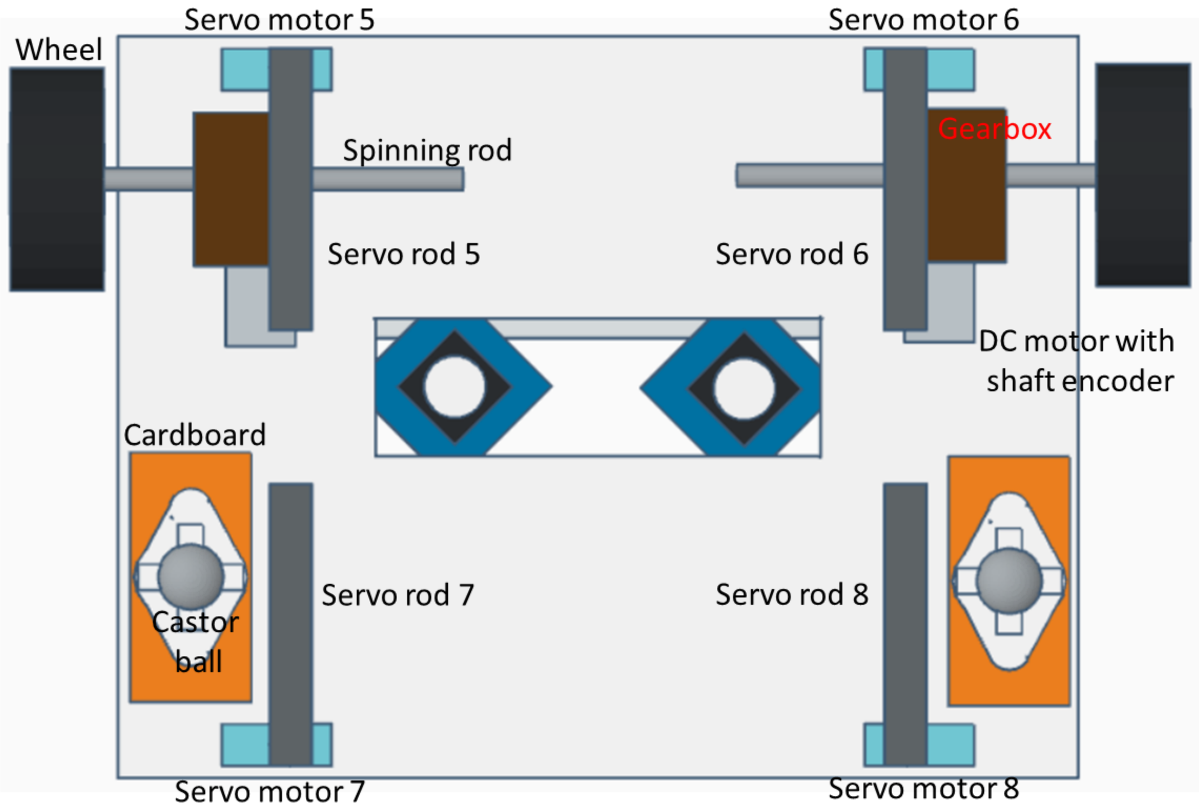


Fig. 11. Underside view of the robot.

1.4.1.3 Front view of the robot

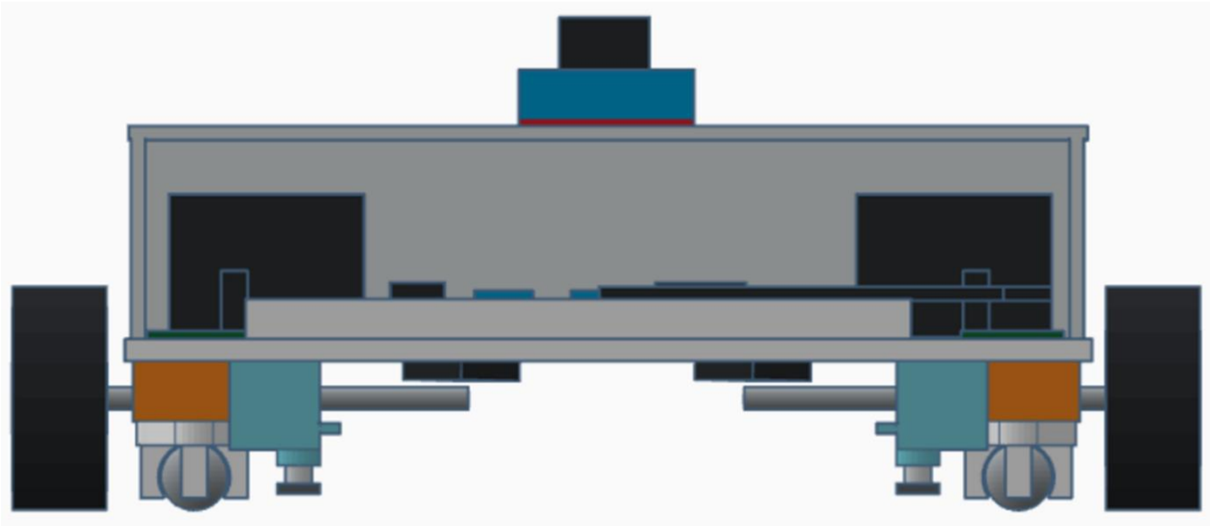


Fig. 12. Front view of the robot.

1.4.1.4 Side view of the robot

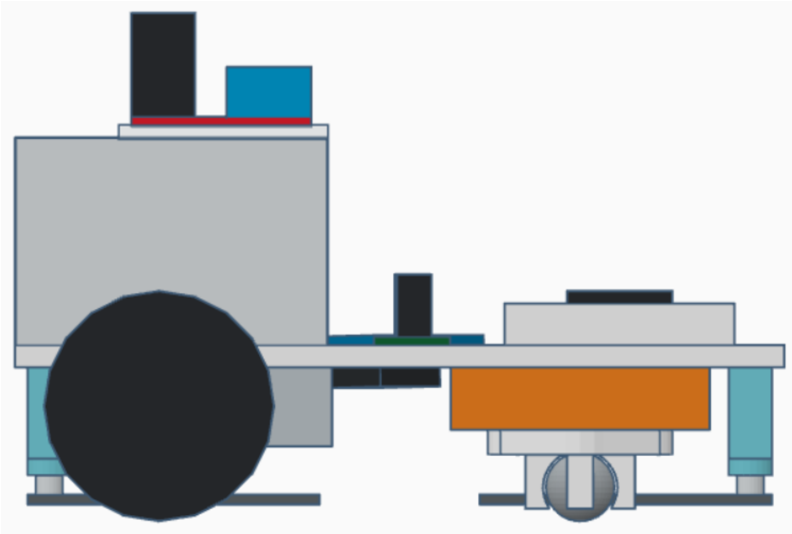


Fig. 13. Right side view of the robot.

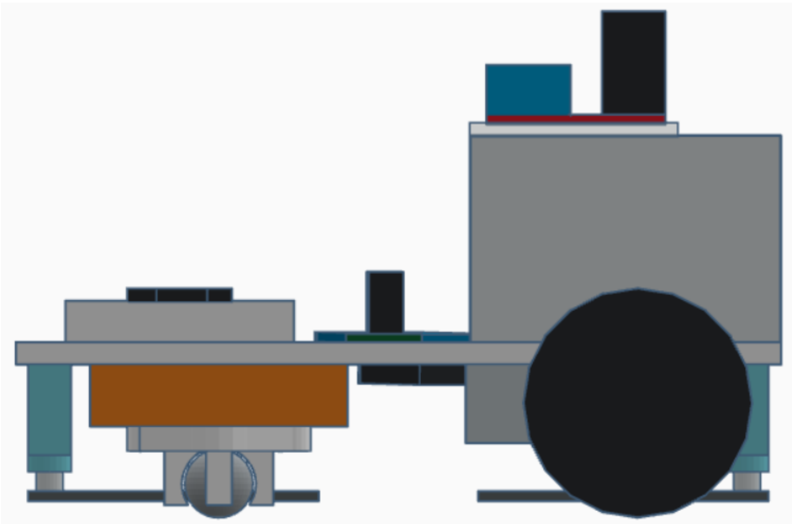


Fig. 14. Left side view of the robot.

1.4.1.5 Back view of the robot

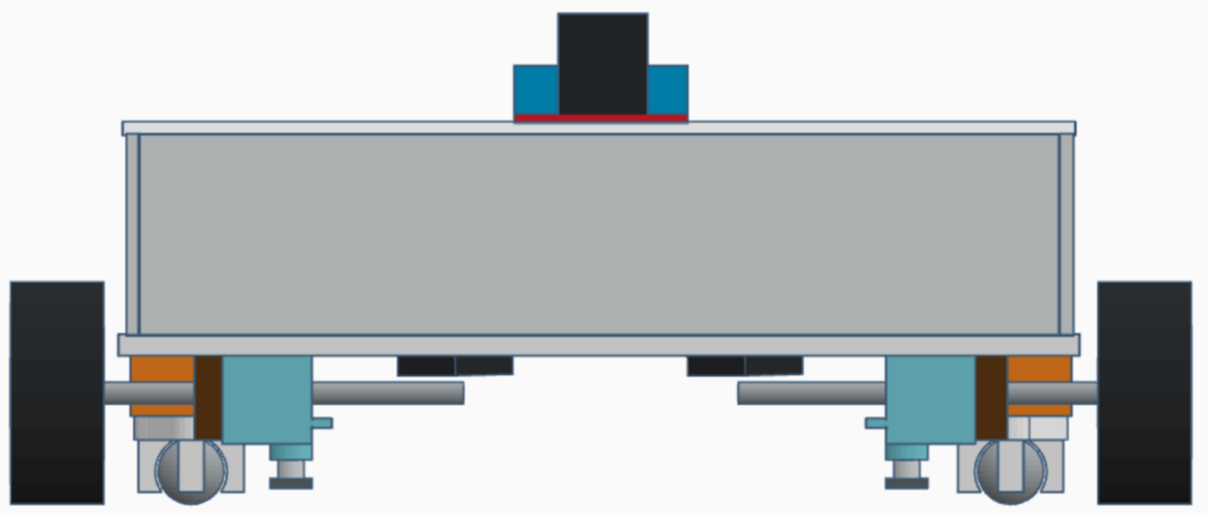


Fig. 15. Back view of the robot.

1.4.1.6 3D view of the robot

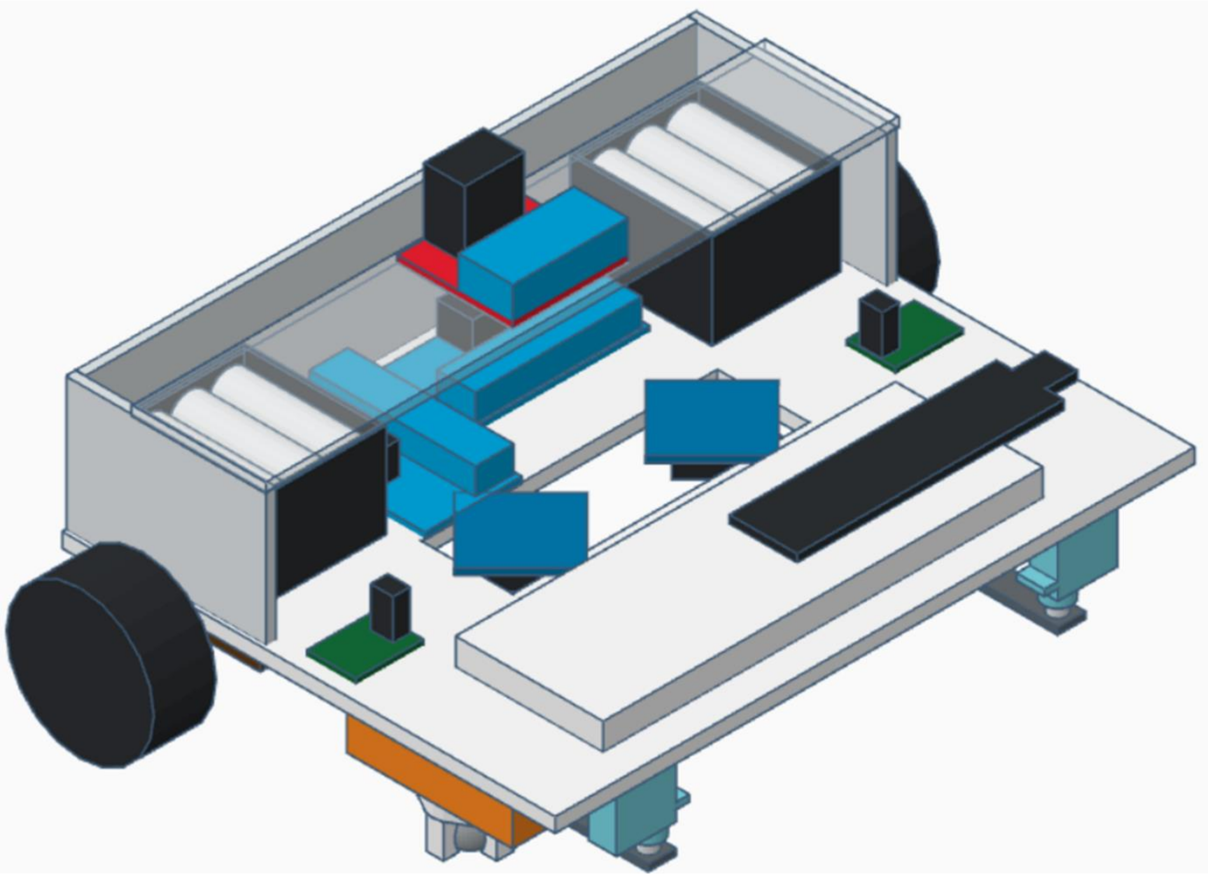


Fig. 16. 3D view of the robot.

1.4.2 Robot design by subsystem

1.4.2.1 Robot design for movement subsystem

The movement subsystem involves 2 major components. Firstly, the wheel sets are controlled by the PSoC microcontroller via motor driver and transfer plates. Secondly, the castor balls, which are attached to the bottom side of the chassis using several layers of cardboard, act as the multidirectional wheels for movement.

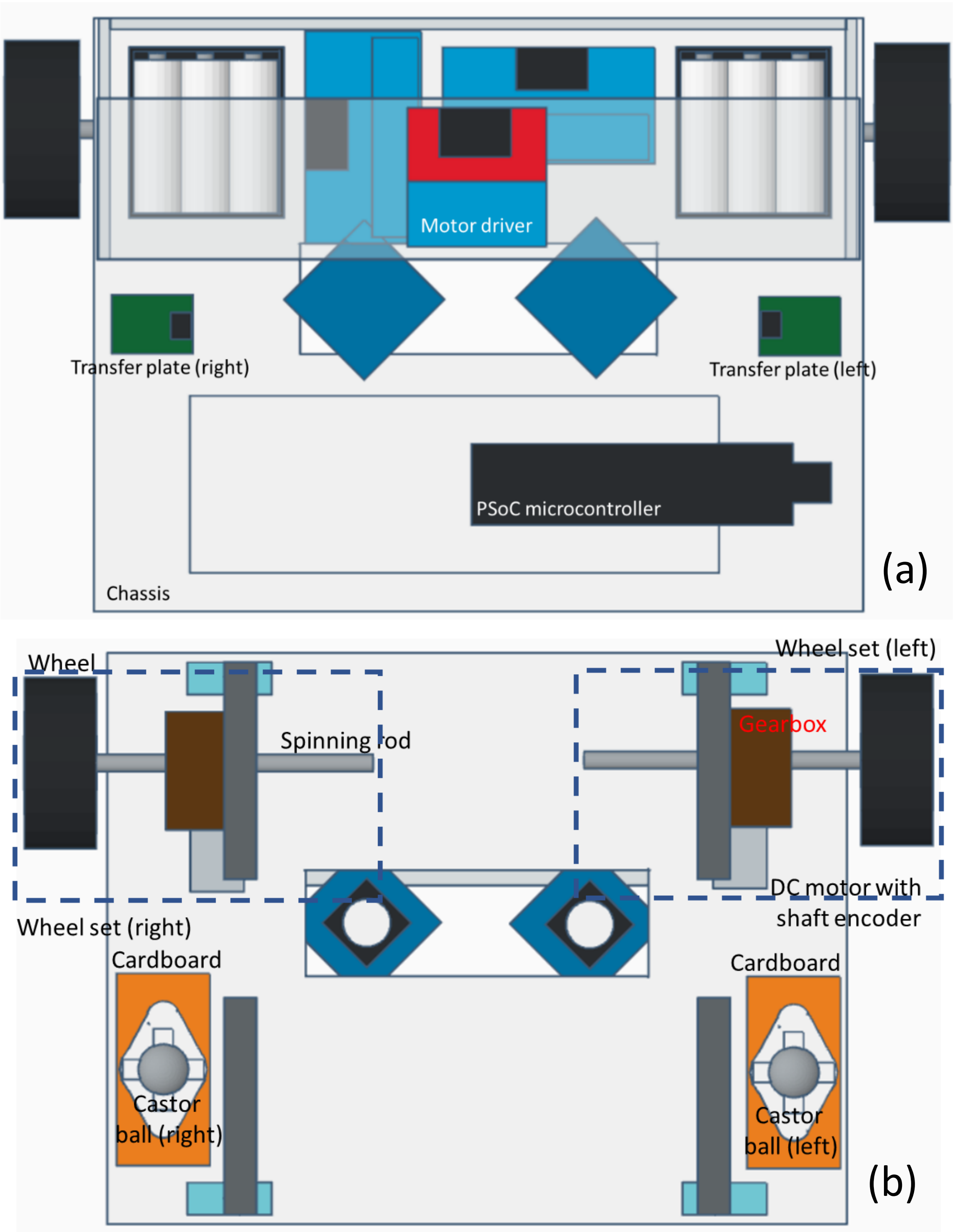


Fig. 17. (a) Top-down view and (b) underside view of robot design for movement subsystem.

1.4.2.2 Robot design for colour sensing subsystem

The colour sensing subsystem only consists of 2 colour sensors, both with the photodiode array facing downwards, placing around the centre of the chassis in a rectangular hole as shown in Fig. 18.

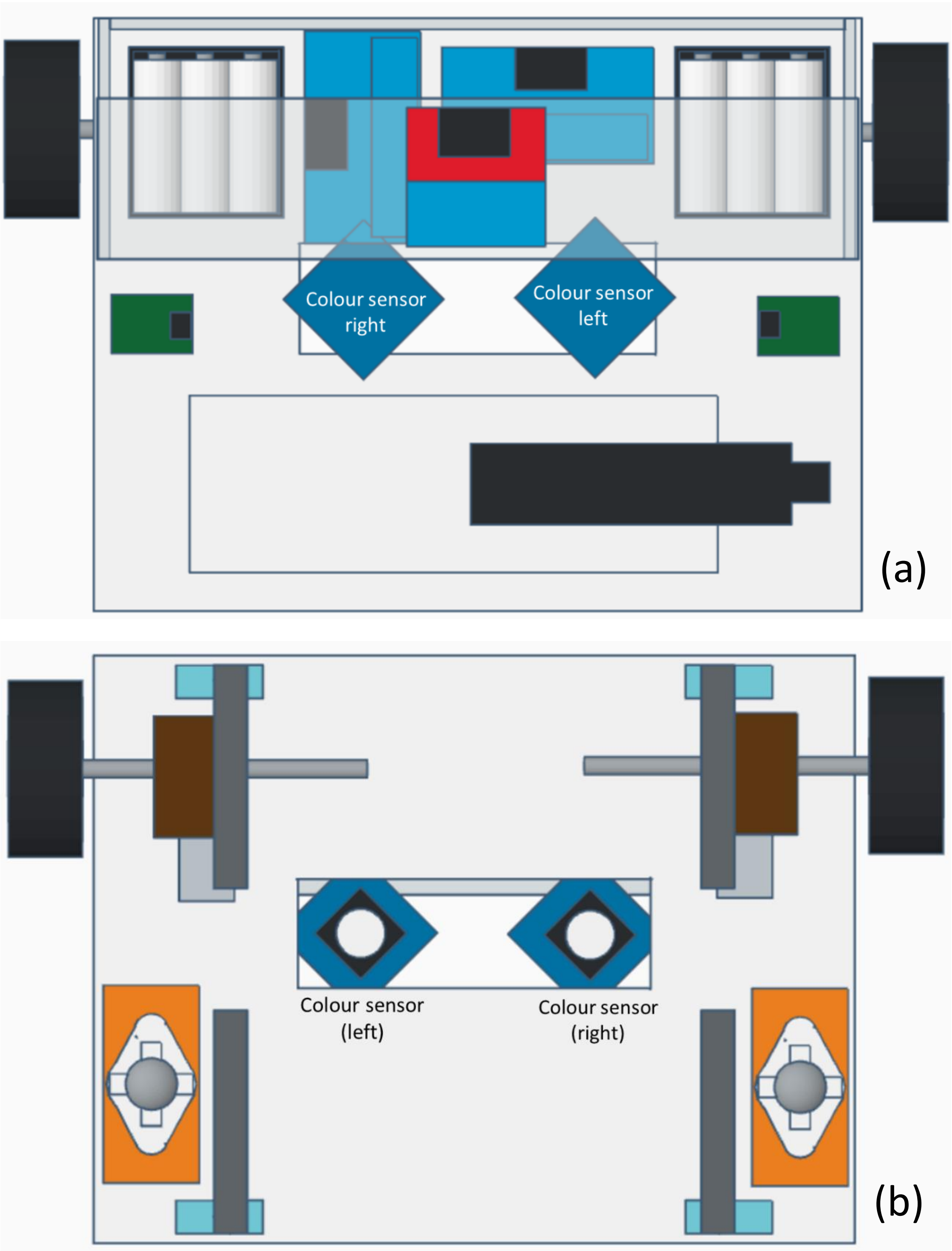


Fig. 18. (a) Top-down view and (b) underside view of robot design for colour sensing subsystem.

1.4.2.3 Robot design for puck handling subsystem

The puck handling subsystem consists of 4 servo motors, each attached to a rod. Each servo motor and its rod are named accordingly to the collection zone they are each responsible for. The names will be used in later sections for succinctness.

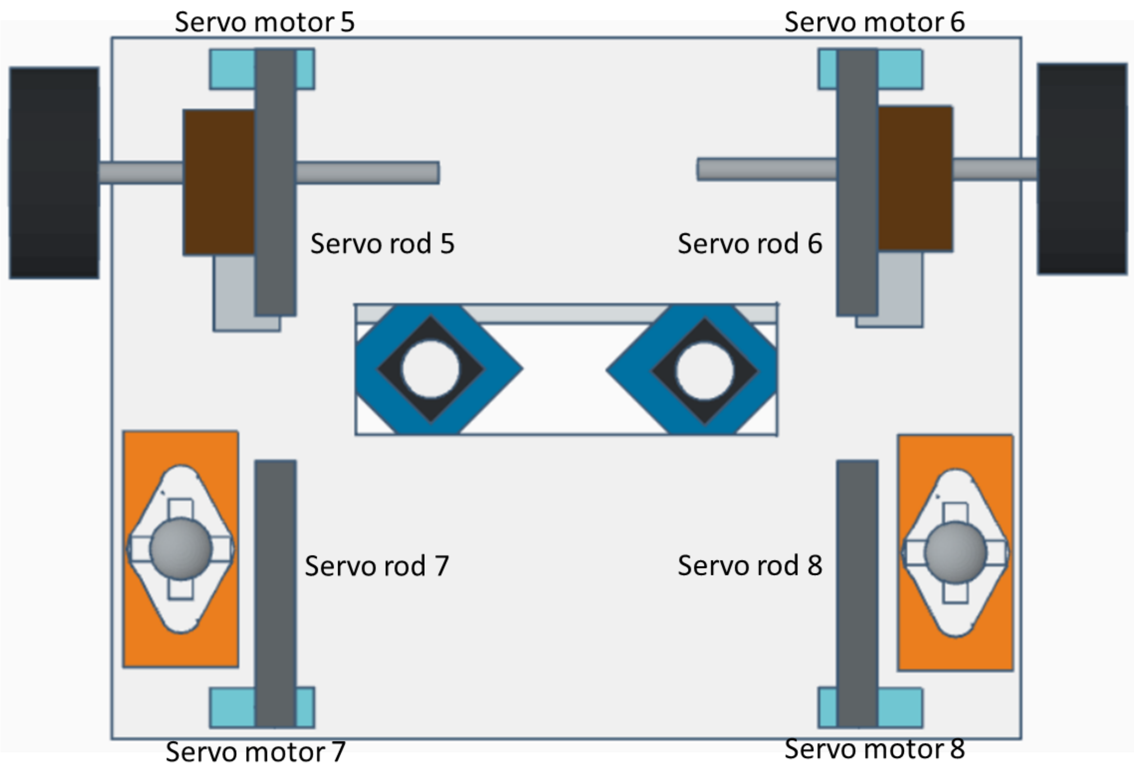


Fig. 19. Robot design for puck handling subsystem.

1.4.2.4 Robot design for power source

12 batteries, 6 in each battery holder, of 1.2V are placed in series to create a supply voltage of 14.4V. The 14.4V is stepped down to 12V via a DC-DC buck converter to power DC motors. The 12V is stepped down again to 5V by another DC-DC buck converter to power all other electronics and PSoC microcontroller.

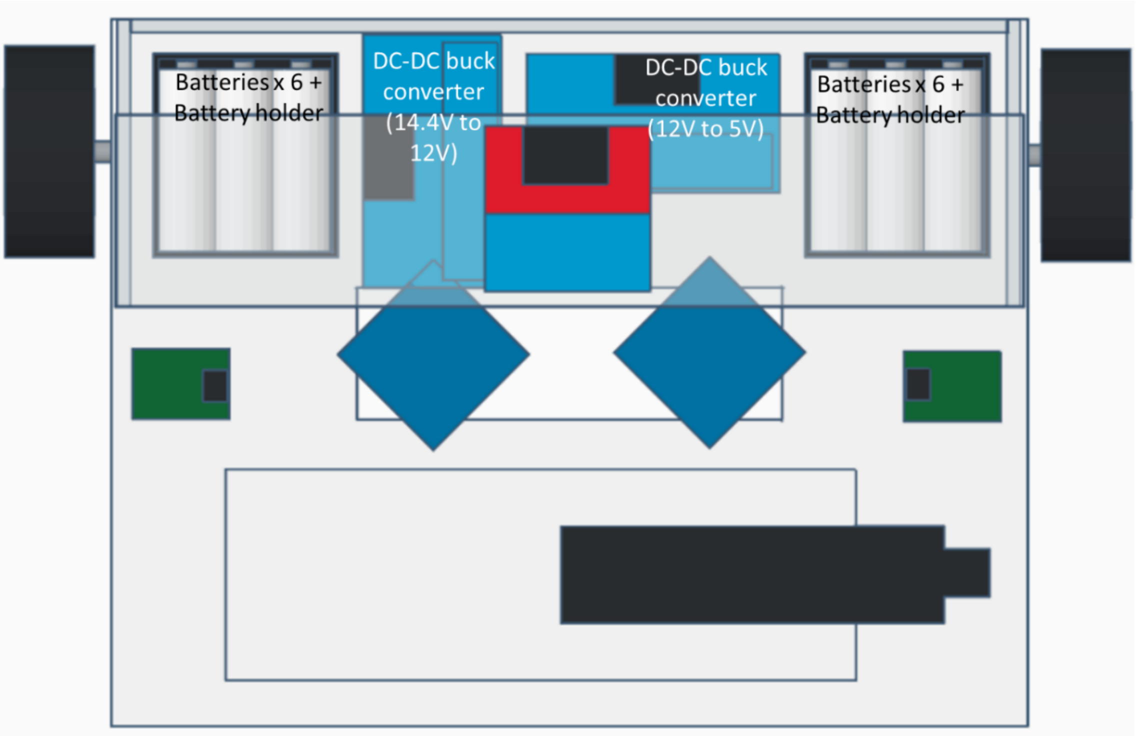


Fig. 20. Robot design for power source.

2.0 Mechanical design

2.1 Mechanical aspects of the design

2.1.1 Movement mechanism

The wheel configuration used in this robot is the popular two-wheeled turtle geometry as shown in Fig. 21. The robot is driven by 2 side-by-side wheels which are placed near the hind side of the chassis. Each wheel is connected to the DC motor with shaft encoder via a spinning rod and a gearbox for driving control. 2 castor balls are used as the undriven wheels to provide the third and fourth point of contact for robot stability and smooth robot movement.

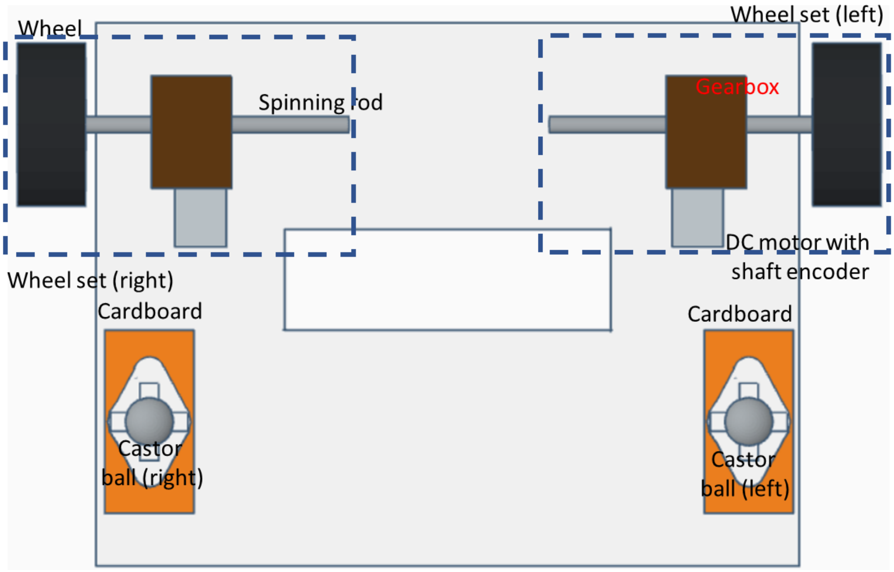


Fig. 21. Underside view of movement subsystem.

As shown in Fig. 22, each castor ball has a height of 2.1cm while the height between the bottom side of the chassis and the floor is 4.1 cm. The fill the gap, each castor ball is attached to a stack of several layers of cardboard. Cardboard is used for its recyclability and ability to support the mass of the robot.

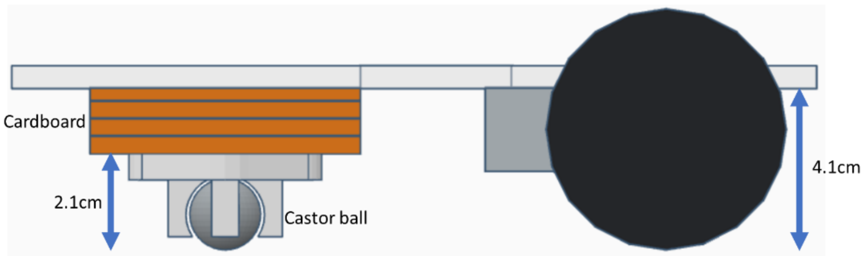


Fig. 22. Side view of movement subsystem.

Since 2 wheels, each connected to its own DC motor with shaft encoder, are used, the robot makes use of the differential drive mechanism. Nevertheless, the robot involves no point turning for motion sequence, and thus the differential drive mechanism is only used to move the robot forwards and backwards.

2.1.2 Robot balance and stability

As shown in Fig. 10, the robot uses only 1 level of robot base, i.e., chassis, for system integration as it is large enough to accommodate all components, only with the use of some light-weighted corrugated boards to lift the motor driver up for neater component arrangement. By lowering the center of mass with only 1 rather than 2 levels of robot bases, the robot has higher stability.

As shown in Fig. 10 which shows the top-down view and Fig. 11 which shows the underside view of the robot, the robot has almost a symmetrical component arrangement both above and below the chassis except the buck converters, breadboard, and the PSoC microcontroller. Symmetrical component arrangement is preferred over its even mass distribution at the left and the right part of the robot, providing better robot balance. Besides, the heavy components, mainly the batteries, are placed atop the load-bearing wheel. This reduces the chances of chassis bulging and robot imbalance.

2.1.3 Components arrangement

2.1.3.1 Underneath the chassis

There are several components underneath the chassis, which are 2 wheel sets, each consisting of a wheel, a spinning rod, a gearbox and a DC motor with shaft encoder, 4 servo motors with rod, and 2 castor balls as shown in Fig. 11.

As described in sections 1.3 and 5.1, the robot has to move over 2 pucks from time to time due to the shortest route strategy and for colour sensing. Therefore, the arrangement of the components underneath the chassis is designed carefully to ensure the components would not unintentionally manipulate any puck which is against the project rule. The puck paths are shown in Fig. 23 and Fig. 24. Each puck path has a width of 6cm, although the puck only has a width of 4cm, to allow a 1cm error margin at both sides of the puck.

The wheel sets have all their components, i.e., gearbox, spinning rod, and DC motor with shaft encoder, above the height of a puck. Therefore, the wheel set has no issue of possible interference. The servo motor with rod is placed near to the side edges of the chassis to allow a 1cm gap between the servo rod and the puck path as shown in Fig. 23 and Fig. 24. The gap is not designed to be larger since the robot has to accommodate 2 castor balls at its two sides for movement mechanism. The components underneath the chassis have symmetrical positioning for better robot balance as described in section 2.1.2.

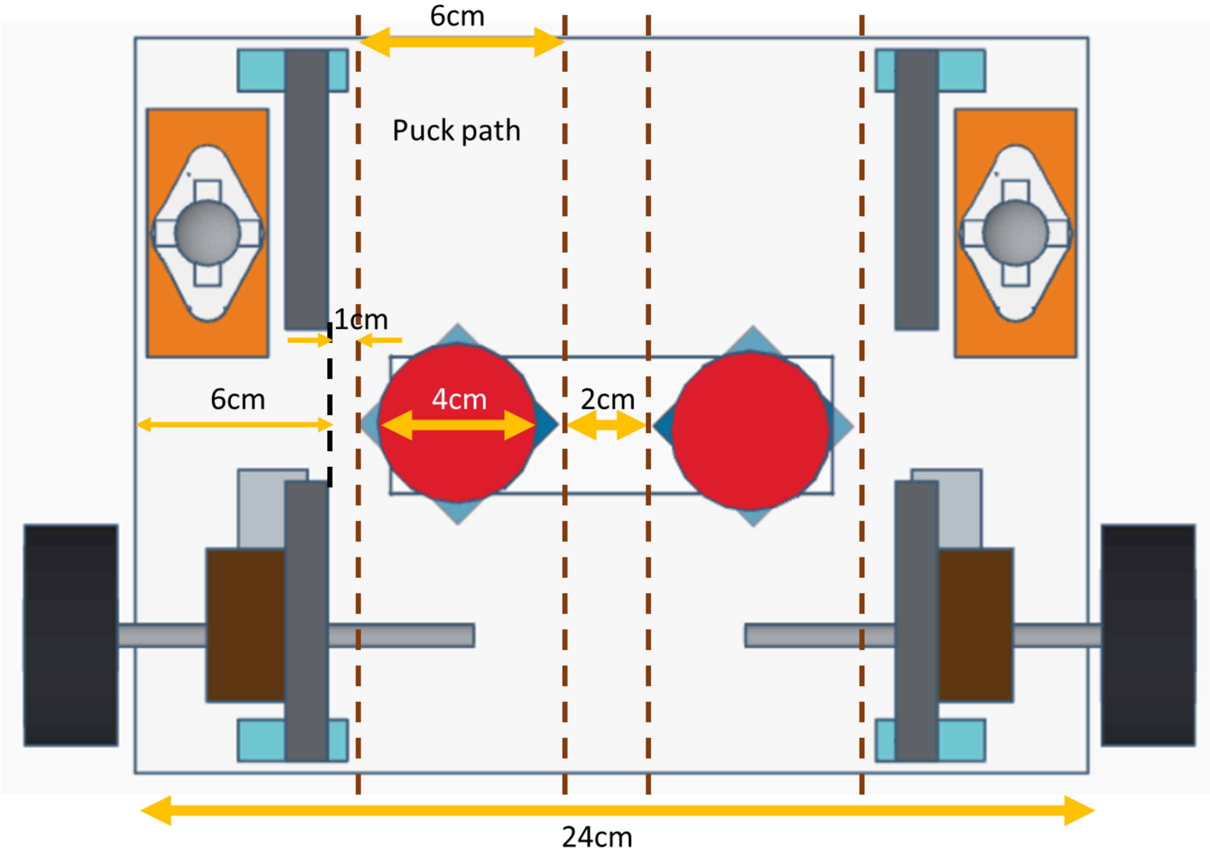


Fig. 23. Component arrangement underneath the chassis.

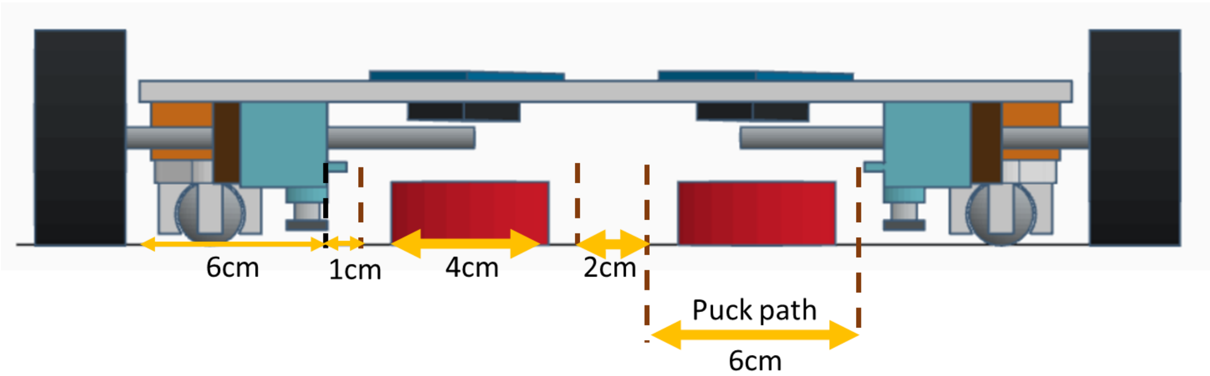


Fig. 24. Front view of the component arrangement underneath the chassis.

2.1.3.2 Above the chassis

There are several components above the chassis, i.e., batteries in holders, 2 DC-DC buck converters, 2 colour sensors, 2 transfer plates, a motor driver, a breadboard, and a PSoC microcontroller.

The reasons for almost symmetrical component arrangement and batteries atop the wheel sets are explained in section 2.1.2. The transfer plates connect the shaft encoders to the PSoC microcontroller. Therefore, they are placed between the PSoC microcontroller and the wheel sets for neater wiring. The motor driver is lifted by some corrugated boards for the same reason.

Besides, 2 colour sensors are used for simultaneous colour sensing. A colour sensor is sensitive to ambient noise as the ambient light can interfere with the light reflected from a puck to the photodiode array on the colour sensor. Therefore, the colour sensors are inserted into a rectangular hole around the center of the chassis to reduce ambient noise. The chassis comes with a smaller rectangular hole which the system integrator enlarges to accommodate the sensors. Each colour sensor is placed at the middle of each puck path as shown in Fig. 23 so that each colour sensor can accurately detect the puck colour but not accidentally detect colour from the floor. Although not shown in the overview of robot design, the robot has a casing to follow the project rule. The casing covers over the PSoC microcontroller, transfer plates, and the colour sensors as shown in Fig. 25. This further reduces the ambient light noise to the colour sensors.

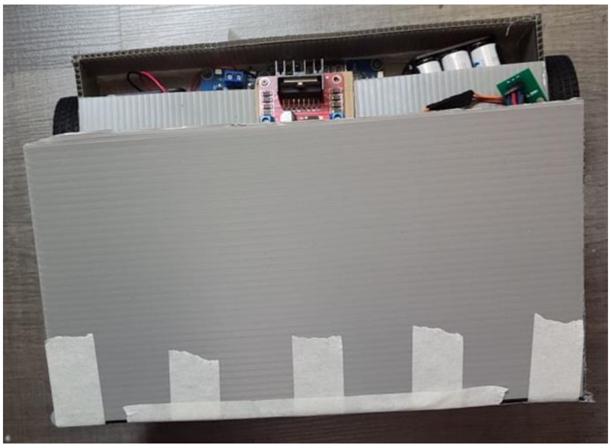


Fig. 25. Casing of the robot.

2.1.4 Puck handling mechanism

As shown in Fig. 26, each servo rod is designed to only have 2 possible angular positions during the motion sequence. The servo rods are normally opened rods. Only when the robot wishes to drag a valid puck, the corresponding servo rod would close, i.e., rotate 90 degrees anticlockwise or clockwise to be parallel to the back or front chassis edge. After closing, the robot would move towards the valid puck and push it towards the capture zone until it is completely inside. After capturing, the robot would retract to its before capture position. If the valid puck is only to capture once but not for continuous capturing, the robot would open the servo rod as described in section 1.3.2 and section 5.1. This allows the robot to continue its remaining activities before starting continuous capturing.

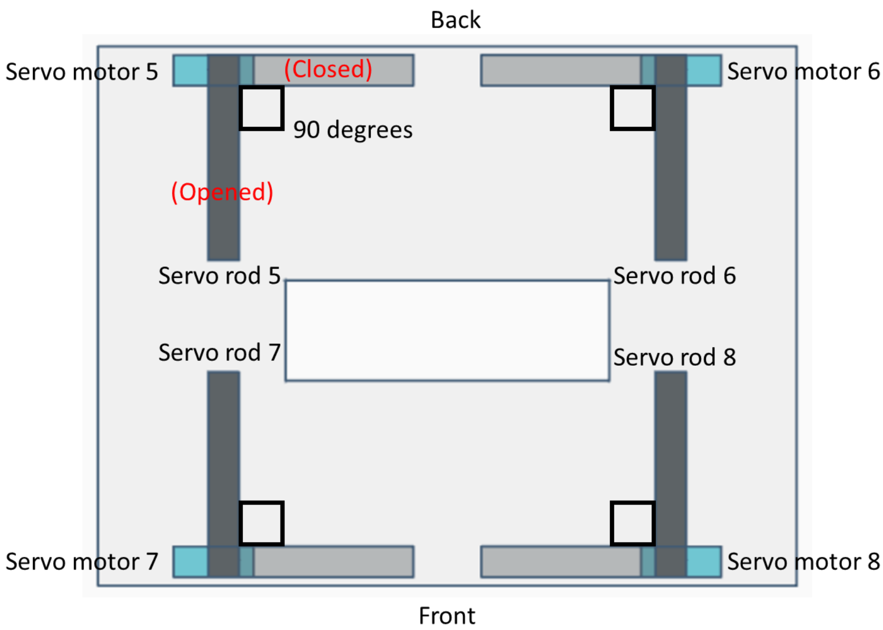


Fig. 26. Under-side view of puck handling subsystem.

Each servo rod is around 7cm long. This ensures the rod is long enough to push the valid puck, assuming the valid puck is within the corresponding puck path as shown in Fig. 27. It is also short enough to have no interference with other servo rods when it is closed and opened as shown in Fig. 26. Each servo rod is made of a corrugated board which is around 3mm thick and is thin enough to have no contact with the arena. It is attached to the servo motor by simply inserting the shaft screwed to the servo motor into the hole within the board as shown in Fig. 28.

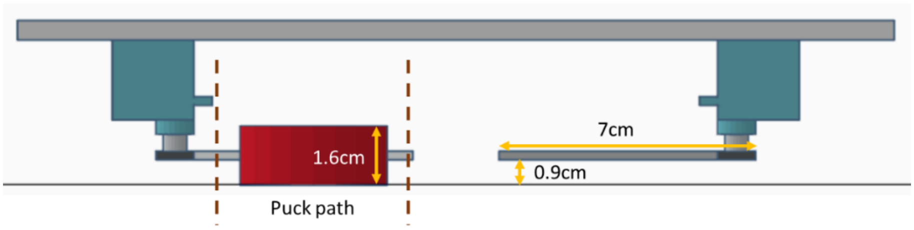


Fig. 27. Front view of puck handling subsystem.

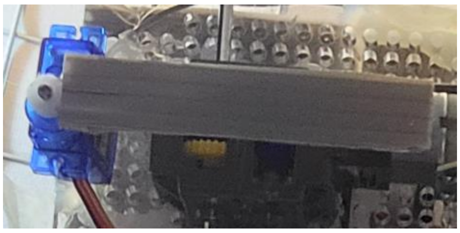


Fig. 28. The servo rod is attached to the servo motor by inserting the shaft into the board.

2.2 Recyclable materials

As shown in Fig. 22, each castor ball has a height of 2.1cm while the height between the bottom side of the chassis and the floor is 4.1 cm. To fill the gap in between, each castor ball is attached to a stack of several layers of recyclable cardboard. Cardboard has a strong capability to support the mass of the robot firmly and shows no deterioration in supporting performance until the end of the project.

Besides, 4 680mm x 575mm x 3mm corrugated plastic boards are purchased to construct the casing. After casing construction, there are some leftover pieces of corrugated plastic boards. These leftovers are used to make the servo rods to reduce wastage.

2.3 Alternative mechanical design

2.3.1 Flicking vs dragging

Current mechanical design: capture a puck by dragging via robot movement

The current mechanical design uses the dragging mechanism where the corresponding servo rod would close before the robot moves towards the to-be-captured valid puck to push the puck to the capture zone with the closed servo rod as described in section 1.3.2 and section 5.1 and shown in Fig. 29(a).

Alternative robot design: capture a puck by flicking via a servo rod

An alternative mechanical design uses the flicking mechanism to capture a valid puck. The robot first stops 1cm away from the valid puck to ensure the robot does not cover the black circle. After that, the robot rotates its servo rod as shown in Fig. 29(b) to flick the to-be-captured valid puck into the corresponding capture zone.

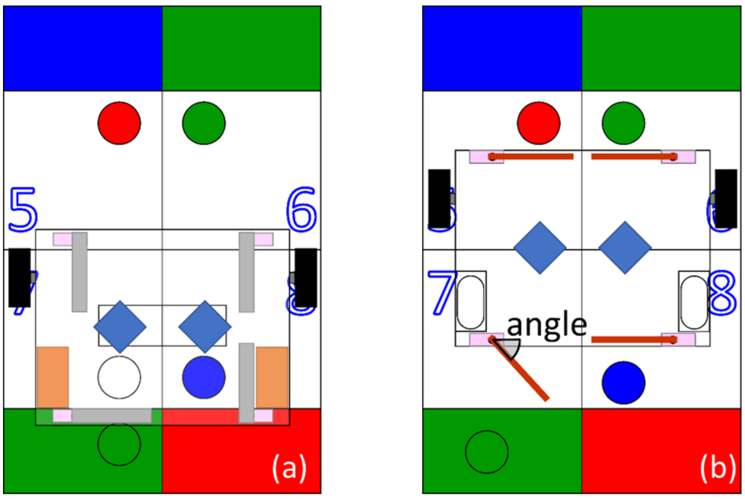


Fig. 29. (a) The current design of capturing a puck by dragging via robot movement. (b) The alternative design of capturing a puck by flicking via a servo rod.

Discussion:

We experimented with both methods in a test environment of continuously capturing the valid puck in zone 7 for 20 times. Two different flicking angles are tried. The result of the experiment is shown in TABLE I. We found that the flicking method provides unreliable results where it is unlikely to have the error rate, i.e., puck shifted too far or too near, to be 0% since increasing rotation angle reduces "too near" error rate but increases "too far" error rate while decreasing rotation angle reduces "too far" error rate but increases "too near" error rate. Therefore, the alternative mechanical design of capturing a puck by flicking via a servo rod is not used.

TABLE I
PERFORMANCE OF DRAGGING AND FLICKING MECHANISMS

	Puck shifted too near	Puck is rightly captured	Puck shifted too far
Dragging	0%	100%	0%
Flicking (angle = 36 degrees)	4%	81%	15%
Flicking (angle = 41.4 degrees)	2%	78%	20%

2.4.2 2-servo motor dragging mechanism vs 4-servo motor dragging mechanism

Current mechanical design: 4-servo motor dragging mechanism

The current mechanical design uses the 4-servo motor dragging mechanism as explained in section 1.3.2 and shown in Fig. 30(a). Each servo motor with rod is responsible for the puck in each collection zone. This makes the shortest route strategy possible.

Alternative mechanical design: 2-servo motor dragging mechanism

An alternative mechanical design uses the 2-servo motor dragging mechanism as shown in Fig. 30(b). An example of how the dragging mechanism captures a puck is shown in Fig. 31. The mechanism requires the robot to visit all the collection zones one by one which lengthens the movement route and travel time before starting continuous capturing.

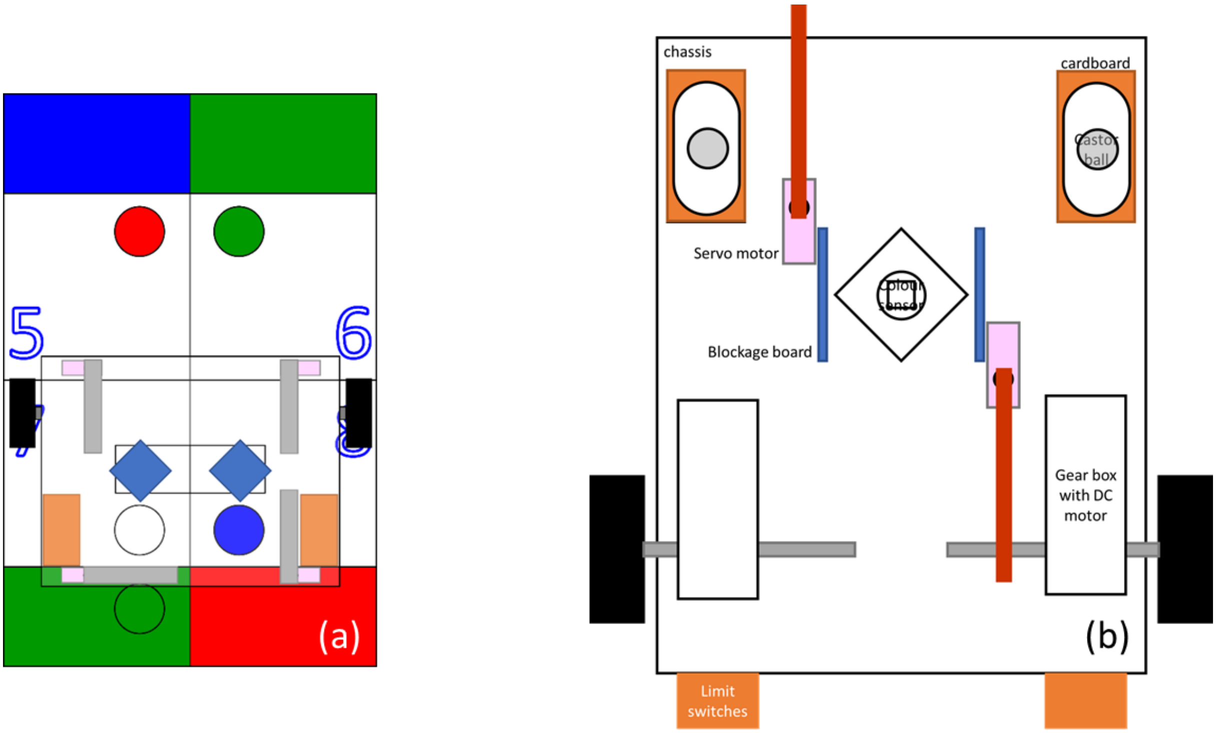


Fig. 30. (a) The current design of 4-servo motor dragging mechanism. (b) The alternative design of 2-servo motor dragging mechanism.

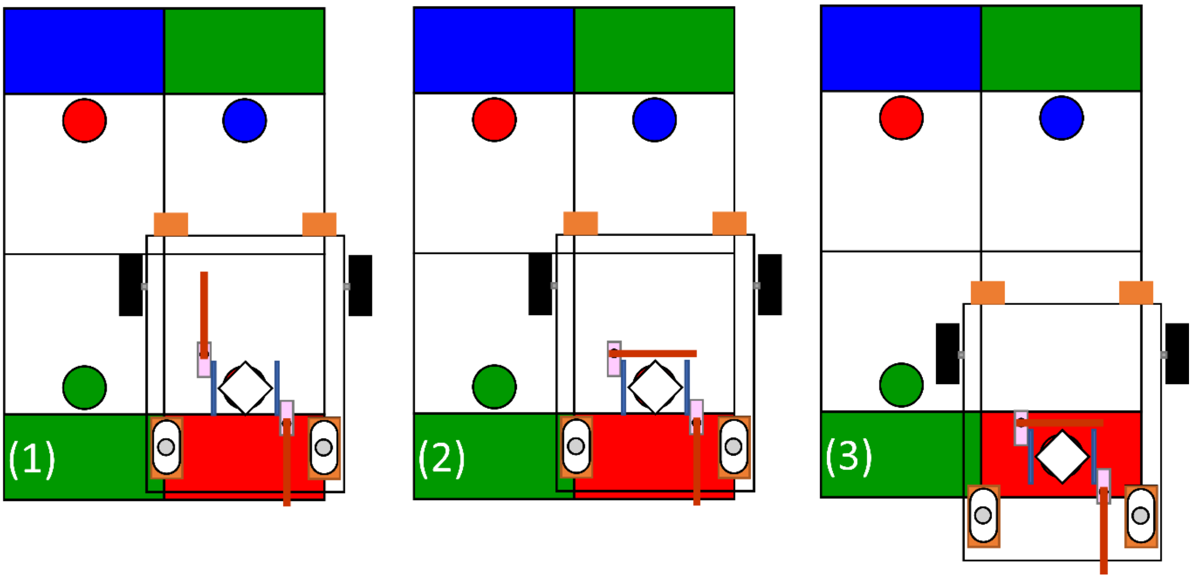


Fig. 31. The capturing process of the alternative design.

Discussion:

We experimented with both methods by timing how long each takes to visit all the zones. The results are shown in TABLE II. The 2-servo motor dragging mechanism requires roughly 1.5 times the travel time compared to the 4-servo motor dragging mechanism to visit all the collection zones. The movement route to visit all the zones in the case of 2-servo motor dragging mechanism is greatly lengthened. It is because the robot has to visit all the zones one by one and enters enter free zones for point turning. In addition, point turning increases the chances of robot misalignment as it is almost impossible for the robot to perform an exact 90-degree turn and the errors can accumulate. This requires a reorientation mechanism by pushing limit switches against the wall. The reorientation further increases the travel time to visit all the zones. Due to the obvious comparative disadvantage, the alternative mechanical design of 2-servo motor dragging mechanism is not used.

TABLE II
PERFORMANCE OF 4-SERVO MOTOR AND 2-SERVO MOTOR DRAGGING MECHANISMS

Dragging mechanism	Travel time to visit all the zones provided all pucks are valid (s)
4-servo motor	52
2-servo motor	77

2.4.3 Automated vs manual puck replacement mechanism

Current robot design: manual puck replacement mechanism

The current robot is designed to use the manual puck replacement strategy between each capture. This signifies that, after a valid puck is captured, i.e., pushed into the capture zone completely, for 2 seconds, the system integrator would then place the captured puck back to its black circle. Since there is no replaced puck detection mechanism in this robot, the robot would wait for 8 seconds to allow enough time for the system integrator to replace the puck. This ensures that the replaced puck does not against the problem statement that it must be exactly inside the black circle after replacement since the system integrator can put it back to the exact position.

Alternative mechanical design: automated puck replacement mechanism

Before refining the robot to focus on manual puck replacement, we proposed an automated puck replacement mechanism that makes use of 2 servo motors as shown in Fig. 32. The automated puck replacement mechanism is expected to take lesser time to replace a puck since it is faster to respond and to shift the puck back into the black circle.

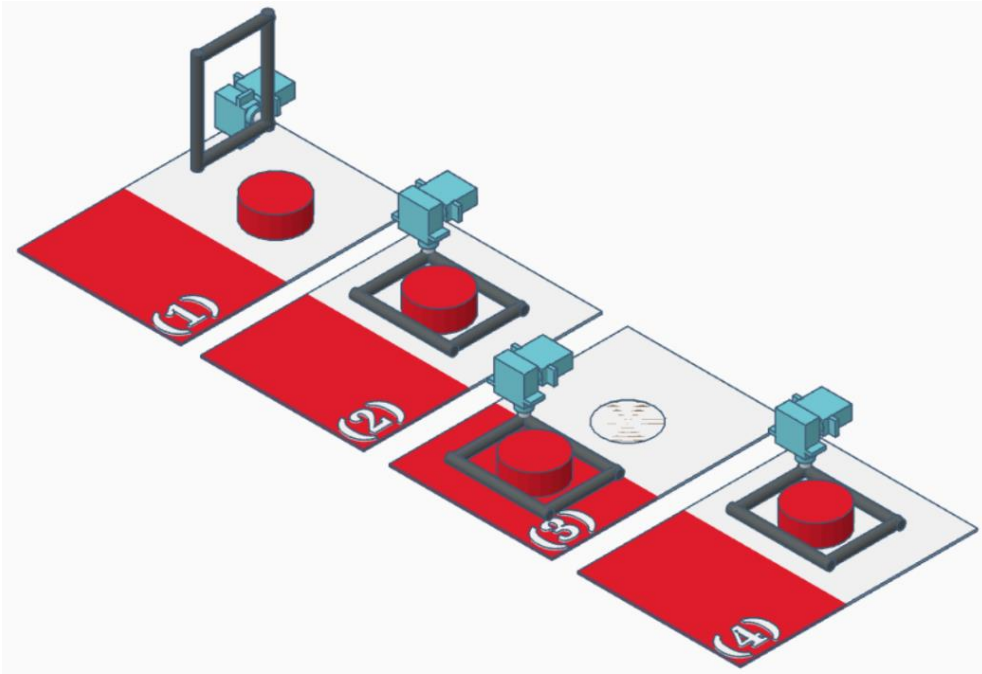


Fig. 32. Automated puck replacement mechanism.

Discussion:

We experimented with both methods by timing how long each takes to replace a puck. The results are shown in TABLE III. The alternative design takes around only half the time the current design needs. Nevertheless, the alternative design uses 2 servo motors with a loop for automated puck replacement. It is almost impossible to use this method to replace the puck exactly back to the black circle since the loop always has a hole of larger size than the puck size to ensure the loop can accommodate the puck. According to project rules, if the captured puck is not replaced into the circle exactly, the puck can no longer be captured. This project rule makes it unfavourable to use the automated puck replacement mechanism since the mechanism proposed is not capable to provide an accurate replacement. Due to the limitation, the alternative design of automated puck replacement mechanism is not used.

TABLE III
PERFORMANCE OF MANUAL AND AUTOMATED PUCK REPLACEMENT MECHANISMS

Puck replacement mechanism	Time to replace a puck (s)
Manual	3.2
Automated	1.7

3.0 Electronic design
3.1 Working principle
3.1.1 Circuit diagram

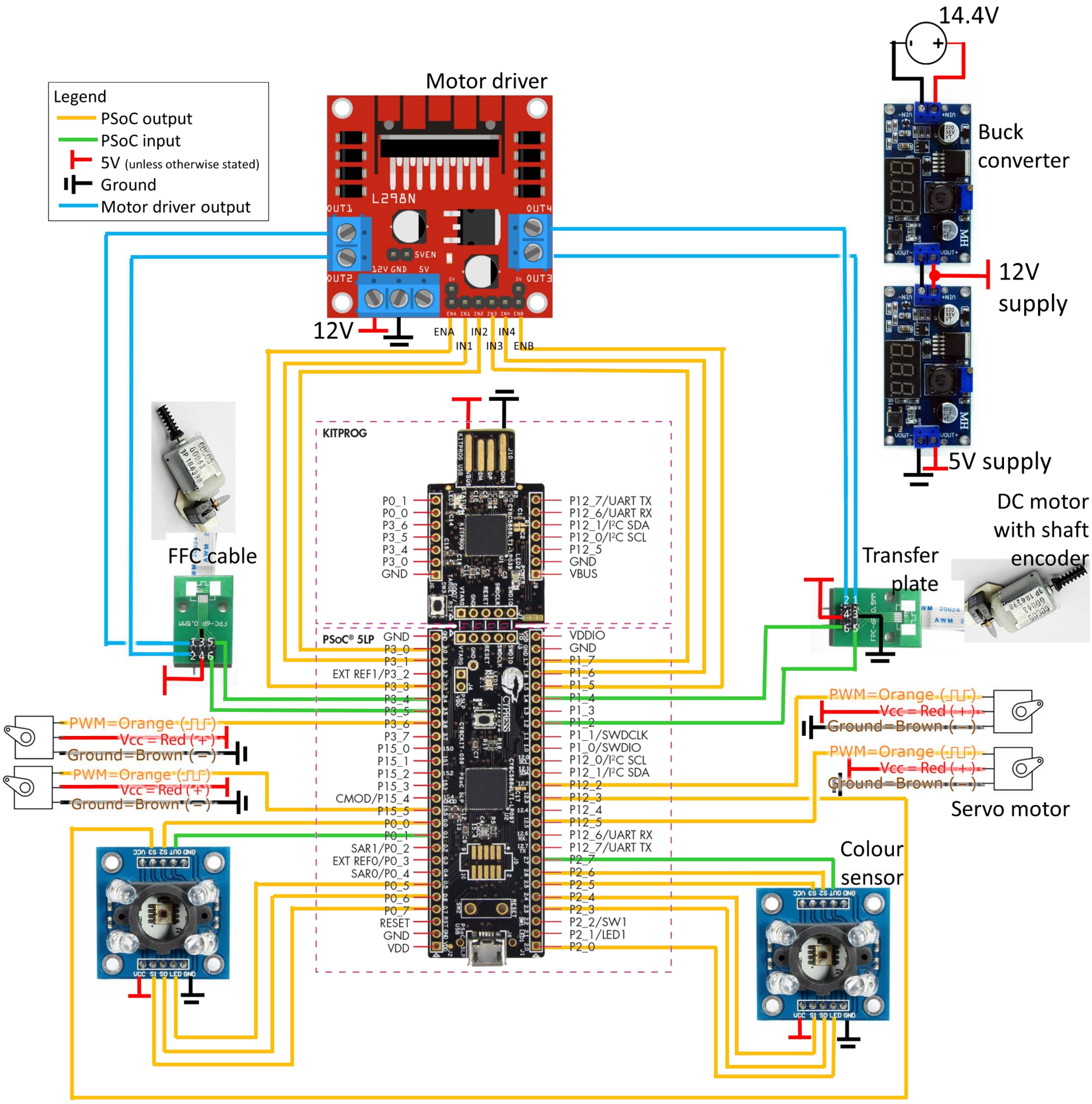


Fig. 33. Circuit diagram of electronics [1], [2] [3], [4], *[5]*, *[6]*.

The PSoC microcontroller has extremely flexible input and output ports. Every general-purpose input and output port (GPIO) has analog and digital I/O capability. Therefore, GPIOs of the PSoC microcontroller are used for communication between the microcontroller and every electronic component. All pins on the PSoC board with special connections, such as pins connected to ground through a bypass capacitor, and special functions, such as UART-TX and UART-RX, are not used to avoid the burden of very careful usage. The GPIOs are powered by 4 separate VDDIOs. To avoid overloading of any VDDIO segment, the connections between the electronic components and the microcontroller are distributed more or less evenly among the 4 power segments. Pins connecting to the same electronic are connected to the GPIOs on the same power segment to allow modulation for easier debugging. All pins, power supply and ground are connected via male-to-male or female-to-male jumper wires besides the voltage supply and ground connection of the PSoC board which are connected via a USB cable.

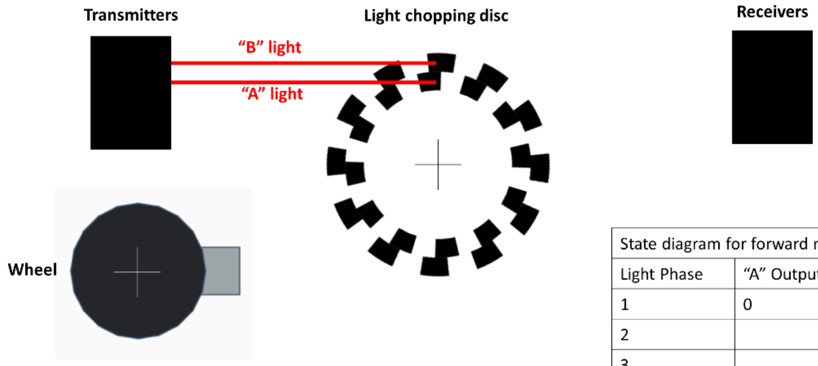
The wiring between the components and the PSoC microcontroller is carefully designed to enhance wiring neatness. Each colour sensor has all its pins connected to one side of the PSoC board beside the S3 pin of the right colour sensor. The S3 pin is deliberately connected to P12_3 on the opposite side of the PSoC instead of any pin on the same side. This to keep all the pins of the colour sensor connected to the same power segment as P12_3 is powered by the same VDDIO as the other pins connected to the right colour sensor. Besides, the right DC motor has all its connections on one side of the PSoC. So does the left DC motor. As the PSoC board is placed on the chassis with its USB port facing the left side as shown in Fig. 10, each servo motor has its PWM connection as shown in Fig. 33 such that the servo motor wires would not crossover the PSoC board.

12 batteries of 1.2V in series supply a total of 14.4V. The 14.4V is stepped down to 12V via a DC-DC buck converter to power the 2 DC motors. The 12V is stepped down again to 5V by another DC-DC buck converter to power all other electronics and PSoC microcontroller.

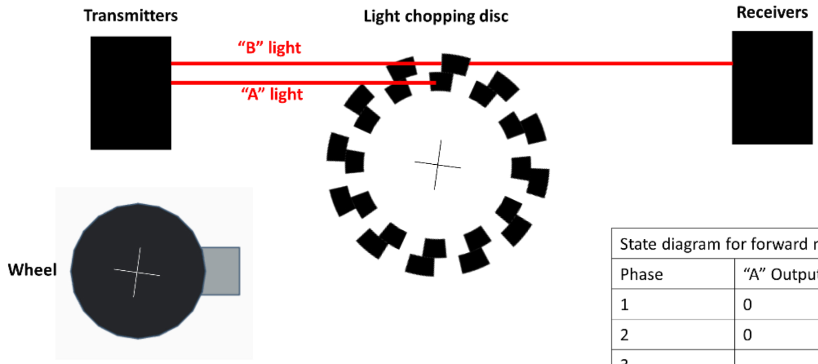
3.1.1 DC motor with shaft encoder for movement mechanism

As the robot makes use of the differential drive mechanism, 2 DC motors are used, each to drive a driven wheel. Each DC motor is attached to a shaft encoder which makes dead reckoning possible.

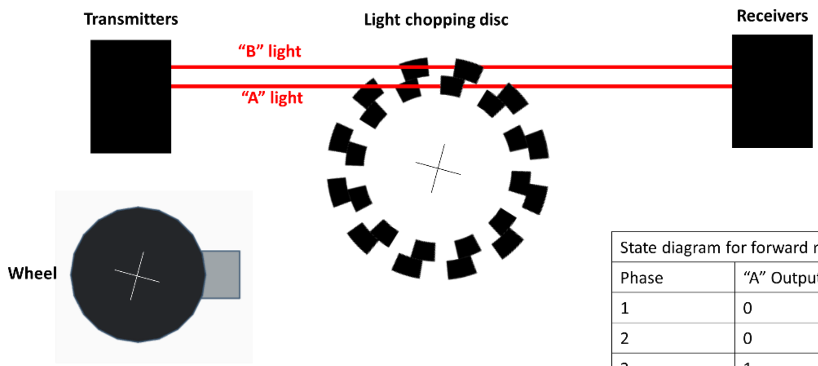
A shaft encoder has 2 light transmitters, 2 receivers, and a light chopping disc as shown in Fig. 34. The transmitters emit fixed light beams A and B to the disc. The corresponding receiver receives the light beam only when it is not blocked by any black spot on the disc. When the wheel keeps rotating, so does the encoder disc as shown in Fig. 34. This is reflected by the changing light phases output which is fed into the software algorithm constantly to compute the wheel's travelled distance to acknowledge immediate robot position.



State diagram for forward motion		
Light Phase	"A" Output	"B" Output
1	0	0
2		
3		
4		



State diagram for forward motion		
Phase	"A" Output	"B" Output
1	0	0
2	0	1
3		
4		



State diagram for forward motion		
Phase	"A" Output	"B" Output
1	0	0
2	0	1
3	1	1
4		

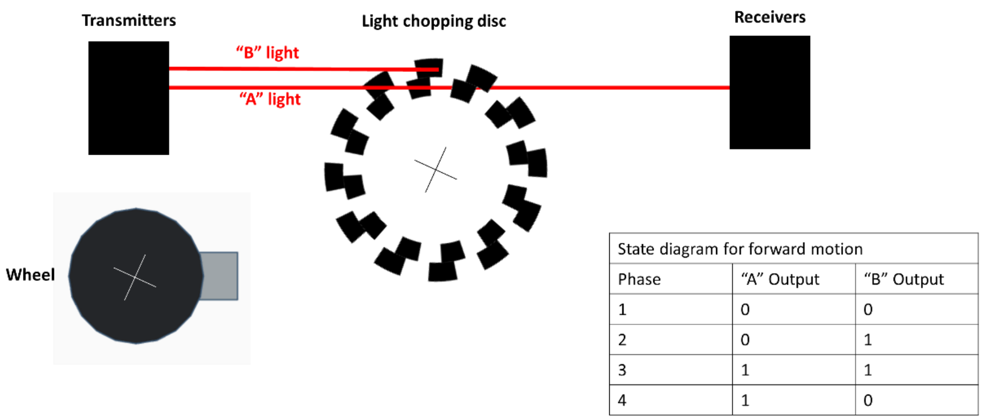


Fig. 34. Light phases change when the wheel is rotating *[7]*.

Fig. 34 shows a sequence of light phases output for forward motion of the wheel. Backward motion of the wheel has a different state diagram as shown in TABLE IV.

TABLE IV
STATE DIAGRAM FOR BACKWARD MOTION

State diagram for backward motion		
Phase	"A" Output	"B" Output
1	1	0
2	1	1
3	0	1
4	0	0

The changes of light phases output of the shaft encoder signal the microcontroller the transitions of quadrature input that are count in the software to represent the travelled distance. A shaft encoder count of 100 with the counter resolution of 1x, i.e., count increments every 4 transitions of quadrature input (or light phases), roughly represents 0.2cm of travelled distance.

3.1.2 Motor driver for movement mechanism

L298N dual H-bridge motor driver is used to control the speeds and directions of the 2 DC motors. For each DC motor, the motor driver has a pair of direction pins (IN1&IN2 or IN3&IN4) and a PWM motor, i.e., speed pin (ENA or ENB), as shown in Fig. 35. These pins receive signals from the microcontroller for direction control and speed control of the DC motor and its wheel. The motor driver changes the direction and speed of the DC motor by varying the voltage supplied to the DC motor via the 2 output ports (OUT1&OUT2 or OUT3&OUT4).

For speed control, the software outputs a PWM signal to the enable pin. The varying duty cycle in the software PWM is reflected by the varying duty cycle in voltage the motor driver outputs to the DC motor. This changes the average voltage supplied to the DC motor and thus its speed.

For direction control, the motor driver switches voltage supply port and power ground port between the pair of output ports according to the signals received via the direction pins. The relationship between the direction pin inputs and the resultant motion of the robot is shown in TABLE V.

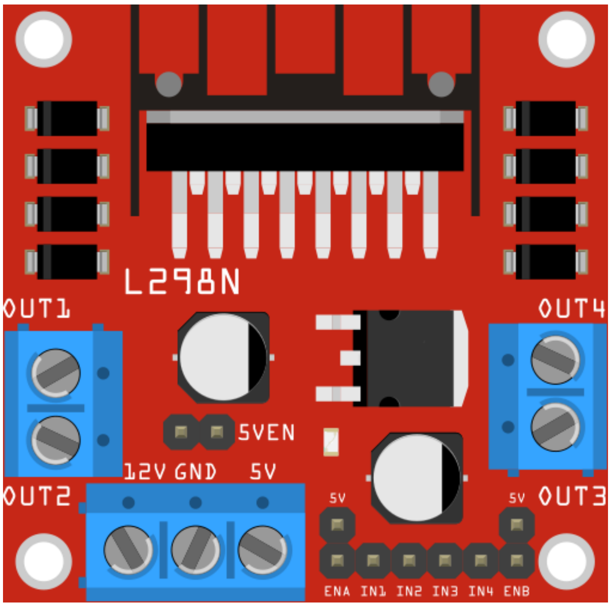


Fig. 35. L298N dual H-bridge motor driver [1].

TABLE V
DIRECTION PINS AND ROBOT MOTION

IN1	IN2	IN3	IN4	Robot motion
0	1	0	1	Move forwards
1	0	1	0	Move backwards
0	0	0	0	Stop movement
1	1	1	1	Stop movement

3.1.3 Colour sensor for colour sensing mechanism

2 TCS3200 colour sensors are used for the simultaneous colour sensing strategy. A TCS3200 colour sensor has an 8 x 8 array of photodiodes in the middle. 16 photodiodes have red filters, 16 have green filters, 16 have blue filters and the remaining 16 have or no filters. To avoid location bias among the colours, the filters of each colour are evenly distributed throughout the array. Internal the sensor has a current-to-frequency converter, as shown in Fig. 37, that inputs current and outputs a square wave of 50% duty cycle which has its frequency directly proportional to light intensity. Outside the sensor has 4 LEDs around the photodiode array to ensure there is large enough light intensity to allow colour sensing.

Fig. 36. TCS3200 colour sensor *[4]*.

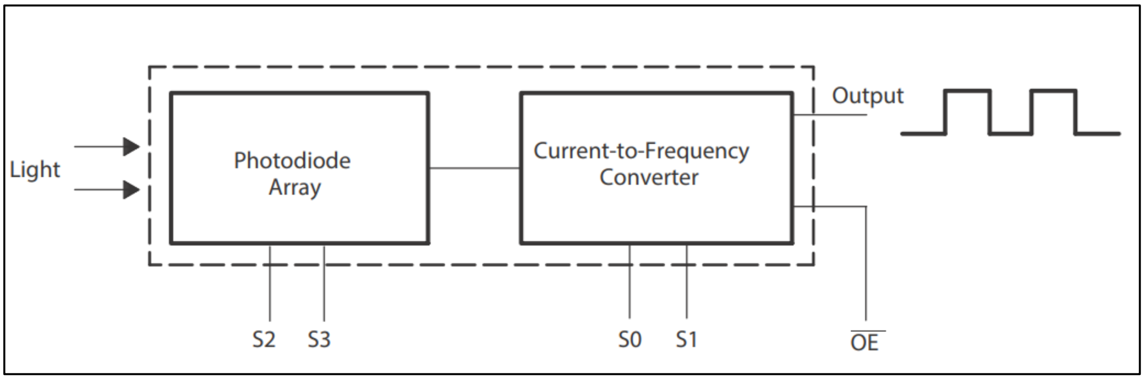


Fig. 37. TCS3200 colour sensor block diagram *[8]*.

Assuming a red puck is under, i.e., facing the photodiode array of, a colour sensor. When the LEDs illuminate the red puck, the red puck reflects a higher irradiance of the red component of light to the photodiode array. Since photodiodes with filters of a colour can only receive the incident light of that colour, the photodiodes with red filters would thus receive a higher light intensity compared to the photodiodes with filters of different colours. Therefore, when only red photodiodes are active, the photodiode array (photodiodes with blue, green, and clean filters are off) receives light of higher intensity compared to when photodiodes with filters of a different colour are active.

Higher irradiance of reflected light received by the red photodiodes induces a higher current passing through the photodiodes. The larger current is thus converted to an output square wave of 50% duty cycle with a larger frequency. In short, the process to determine puck colour is:

1. Activate photodiodes with red filters.
2. Count the number of square waves received for a fixed time window to compute the frequency.
3. Repeat steps 1 and 2 for blue photodiodes and green photodiodes.
4. Compare the frequencies obtained, the highest frequency determines the puck colour.

As shown in Fig. 33, the colour sensor has its S0, S1, S2, S3, LED and OUT pin connected to the PSoC microcontroller. The functions of each pin are described in the tables below.

TABLE VI
TERMINAL FUNCTIONS

Terminal	Input / Output (I/O)	Description
VDD		Supply voltage
GND		Power supply ground
S0, S1	I	Output frequency scaling selection inputs
S2, S3	I	Photodiode type selection inputs
OUT	O	Output frequency
LED	I	LEDs activation input

TABLE VII
SELECTION INPUTS FOR OUTPUT FREQUENCY SCALING

Selection Inputs		Output Frequency Scaling
S0	S1	
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

TABLE VIII
SELECTION INPUTS FOR PHOTODIODE TYPE

Selection Inputs		Photodiode Type
S2	S3	
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

3.1.4 Servo motor for puck handling mechanism

4 SG90 180-degree plastic gear servo motors are used for the 4-servo motor puck dragging mechanism. Each servo motor has 3 pins, i.e., pin for 5V voltage supply, ground connection and pulse width modulator (PWM) input. The PWM pin is to connect to the PSoC microcontroller for software PWM to control the angular displacement or position of the servo shaft and thus the servo rod. This allows the servo to close and open its rod for puck handling as described in section 1.3.2 and section 5.1.

Fig. 38. SG90 180-degree plastic gear servo motor *[9]*.

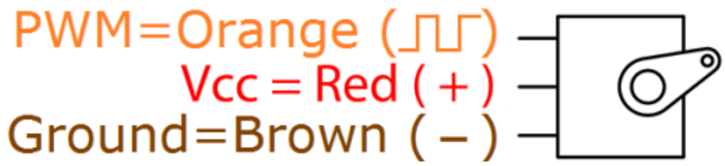


Fig. 39. Pins of SG90 servo motor *[5]*.

The servo motor can rotate its rod clockwise by 90 degrees or anti-clockwise by 90 degrees. The servo uses PWM period of 20ms with a clock of 50kHz in software. A 1.5ms pulse, i.e., 7.5% duty cycle rotates the rod to be at 0° position; a 2ms pulse, i.e., 10% duty cycle, rotates the rod to be at 90° position; a 1ms pulse, i.e., 5% duty cycle, rotates the rod to be at -90°.

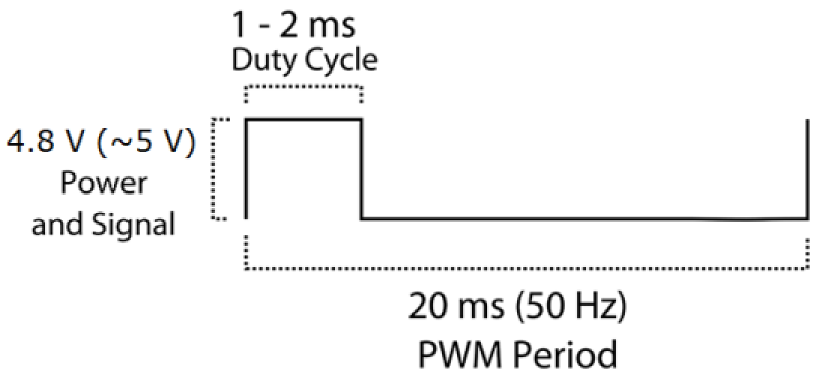


Fig. 40. PWM duty cycle and positioning of the servo rod *[5]*.

In other words, by initiating the rod angular position to be 90° or -90°, the rod is capable to displace by 180°. Nevertheless, the puck handling mechanism used in the robot only requires a servo rod rotation of 90°.

3.2 Power management

3.2.1 Robot weight

Theoretically, power is equal to the energy converted per unit time. The electrical energy is effectively converted into kinetic energy to move the robot. Kinetic energy has an equation of $KE = \frac{1}{2}mv^2$, where m is mass and v is velocity. In other words, to reduce the kinetic energy to move the robot and thus the power consumption of the robot, the mass of the robot should be reduced.

As shown in Fig. 10, the robot uses only 1 layer of robot base, i.e., chassis, with some corrugated boards to provide extra space to accommodate all components. We chose not to use 2 levels of robot base provided since an extra level of robot base is significantly heavier than the corrugated boards to lift the motor driver up.

Besides, cardboards are used to fill the gap between the castor balls and the chassis while the casing is made of corrugated boards. Even the servo rods are made of plastic corrugated boards. All these materials used have a very light weight. These materials are deliberately chosen for their lightness. With these measures, the weight of the robot is kept to a minimum without affecting the performance of the robot.

The cardboards used may seem less reliable compared to their wooden or metal counterparts. Nevertheless, the cardboards remain undeformed since installing until the end of the final competition. The robot did not show any deterioration in the performance of its movement mechanism throughout that period.

3.2.2 Off the idle PWMs

Even though 4 servo motors are used for puck handling, they are idle most of the time in each half-round Therefore, each of them is switched off by using PWM_Stop() function in the software when they are idle. By switching off the PWM, there is no current supplied to the servo motor and thus less power is consumed. Since the puck handling mechanism relies on a motionless servo rod to push the valid puck towards the capture zone rather than a rotating servo rod, the problem of sudden current, due to switching on and off of the PWM, causing inconsistent rotation speed of servo rod is not an issue to the robot.

3.3 Integration of the electronics

3.3.1 Movement subsystem

3.3.1.1 Overview

The movement subsystem involves the communication between 3 electronics, which are PSoC microcontroller, motor driver and DC motors with shaft encoders as shown in Fig. 41. The integration of these electronics allows the movement algorithm to implement travelled distance feedback, motion control, speed control and proportional controller for differential drive mechanism as described in section 6.1.

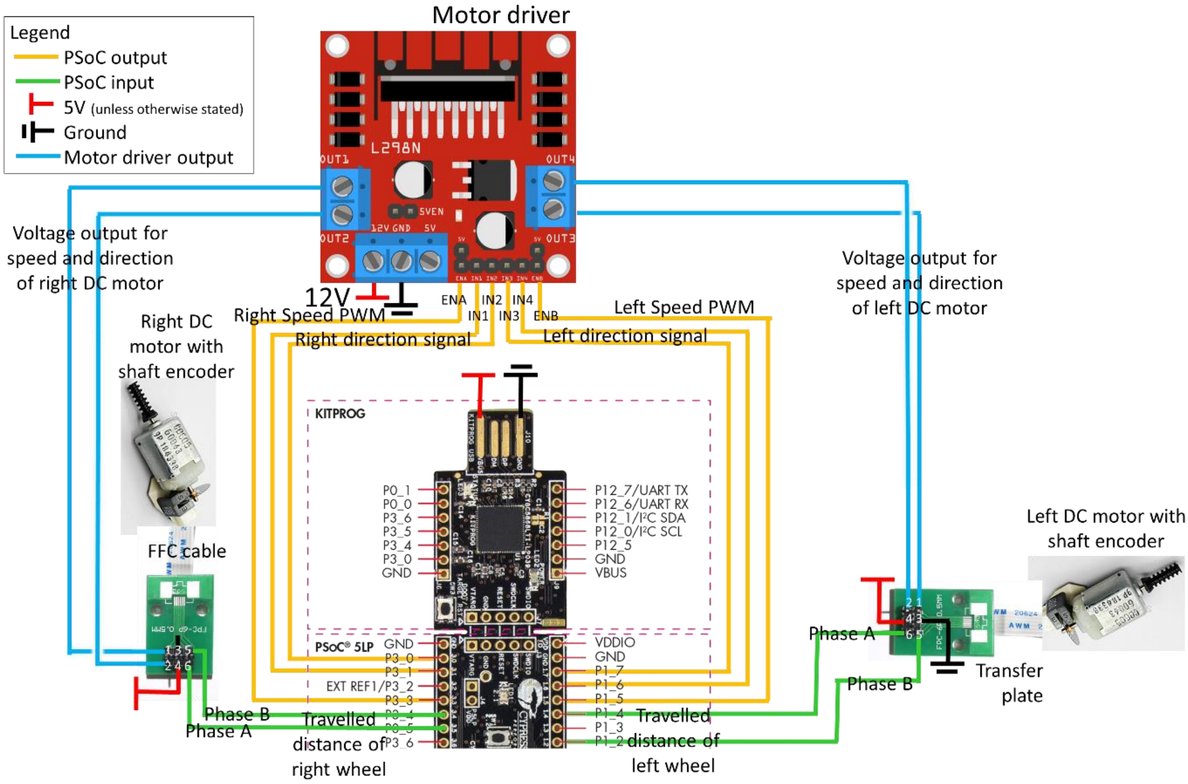


Fig. 41. Electronics for movement subsystem *[1], [3], [6]*.

3.3.1.2 Travelled distance feedback

As explained in section 3.1.1, the shaft encoder outputs phase A and B to the microcontroller signalling the transitions of quadrature input that are count in the software to represent the travelled distance. The shaft encoder outputs these 2 signals via a transfer plate. These 2 components are connected by using a FFC cable. The pin 6 and 5 on the transfer plate are connected to two GPIOs, i.e., P3_5 and P3_6 for the right DC motor; P1_4 and P1_2 for the left DC motor, on the PSoC microcontroller to output the phase A and B signals from the shaft encoder to the microcontroller. The connections are shown in Fig. 41.

3.3.1.3 Speed control

As explained in section 3.1.2, the motor driver has a PWM motor, i.e., speed pin (ENA or ENB) receiving signals from the microcontroller for speed control of each DC motor and its wheel. ENA, responsible for speed PWM signal of the right DC motor, is connected to the P3_3 pin on the PSoC board while ENB, responsible for speed PWM signal of the left DC motor, is connected to the P1_5 pin.

The motor driver changes the speed of the DC motor by varying the voltage supplied to the DC motor via the 2 output ports (OUT1&OUT2 or OUT3&OUT4). OUT1 and OUT2, responsible for the speed and direction of the right DC motor, are connected to pin 1 and 2 of the transfer plate respectively. OUT3 and OUT4, responsible for the speed and direction of the left DC motor, are connected to pin 1 and 2 of the transfer plate respectively. The connections are shown in Fig. 41.

3.3.1.4 Motion control

As explained in section 3.1.2, the motor driver has a pair of direction pins (IN1&IN2 or IN3&IN4) receiving signals from the microcontroller for motion (direction) control of each DC motor and its wheel. IN1 and IN2, responsible for the direction signal of the right DC motor, are connected to P3_1 and P3_0 pins on the PSoC board respectively. IN3 and IN4, responsible for the direction signal of the left DC motor, are connected to P1_7 and P1_6 pins on the PSoC board respectively.

The motor driver changes the rotation direction of the DC motor by switching voltage supply port and power ground port between the pair of output ports (OUT1&OUT2 or OUT3&OUT4) according to the signals received via the direction pins. OUT1 and OUT2, responsible for the speed and direction of the right DC motor, are connected to pin 1 and 2 of the transfer plate respectively. OUT3 and OUT4, responsible for the speed and direction of the left DC motor, are connected to pin 1 and 2 of the transfer plate respectively. The connections are shown in Fig. 41.

3.3.2 Colour sensing subsystem

As explained in section 3.1.3, each colour sensor has 5 input pins (S0, S1, S2, S3, LED) and 1 output pin (OUT). S0 and S1 are output frequency scaling selection inputs while S2 and S3 are photodiode type selection inputs. LED pin is self-explanatory, i.e., LEDs activation input. OUT pin outputs frequency which is directly proportional to the light intensity received. The function of each pin is explained in section 3.1.3. Connections between the pins on the colour sensors and GPIOs on the PSoC microcontroller are shown in Fig. 42 and TABLE IX.

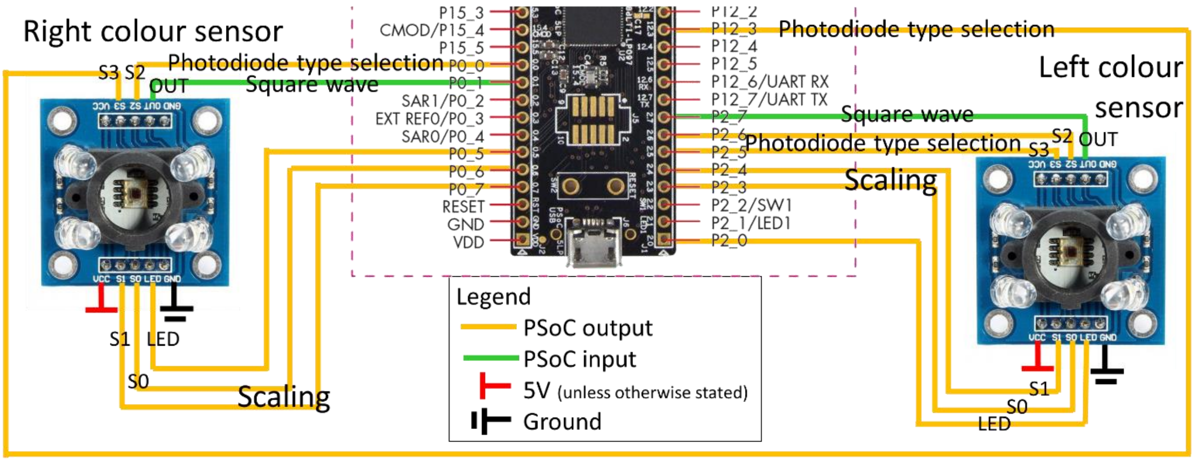


Fig. 42. Electronics for colour sensing subsystem *[4], [6]*.

TABLE IX
CONNECTIONS BETWEEN COLOUR SENSORS AND THE PSOC MICROCONTROLLER

Left / right colour sensor	Pin on colour sensor	Pin on PSoC board
Left colour sensor	S0	P0_6
	S1	P0_7
	S2	P0_0
	S3	P12_3
	LED	P0_5
	OUT	P0_1
Right colour sensor	S0	P2_3
	S1	P2_4
	S2	P2_6
	S3	P2_5
	LED	P2_0
	OUT	P2_7

3.3.3 Puck handling subsystem

As explained in section 3.1.4, each servo motor has a PWM pin to connect the PSoC microcontroller for software PWM to control the angular displacement or position of the servo shaft and thus the servo rod. Connections between the PWM pins of the servo motors and GPIOs on the PSoC microcontroller are shown in Fig. 43 and TABLE X.

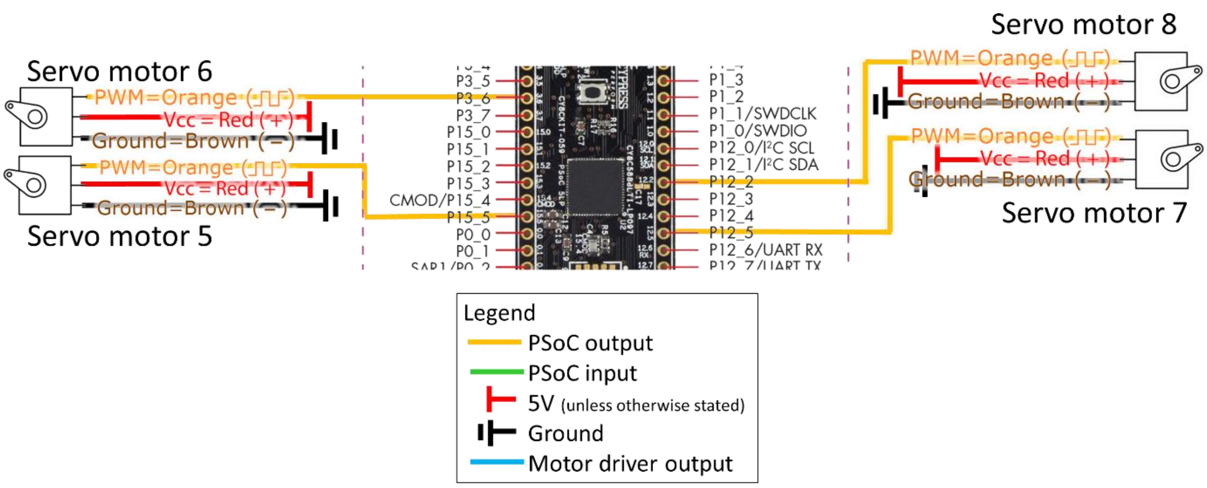


Fig. 43. Electronics for puck handling subsystem *[5], [6]*.

TABLE X
CONNECTIONS BETWEEN SERVO MOTORS AND THE PSOC MICROCONTROLLER

Servo motor	Pin on servo motor	Pin on PSoC board
Servo motor 5	PWM	P15_5
Servo motor 6	PWM	P3_6
Servo motor 7	PWM	P12_5
Servo motor 8	PWM	P12_2

3.4 Alternative sensors and actuators

3.4.1 1 colour sensor vs 2 colour sensors

Current electronic design: 2 colour sensors

The current electronic design uses 2 colour sensors for simultaneous colour sensing strategy as described in section 5.1.

Alternative mechanical design: 1 colour sensor

An alternative mechanical design uses 1 colour sensor to detect the puck colour one by one.

Discussion:

It is obvious without experiment that the simultaneous colour sensing strategy, made possible by 2 colour sensors, only takes half the time needed for colour sensing compared to only 1 colour sensor being used. Since it is preferred to have more time for continuous capturing to achieve as many points as possible, 2 colour sensors are used instead of 1.

3.4.2 IR sensor vs hardcoding for puck detection

Current electronic design: Hardcoding puck positions in the collection zones

Current electronic design hard codes the puck positions in the collection zones for the robot to stop over them to conduct colour sensing. The hardcoding process is as follows:

1. Calculate the distance the robot has to move to reach the puck positions. There are 3 distance parameters to be set, i.e.,
    a. the distance to travel from the starting base to puck collection zones 5 and 6,
    b. the distance to travel from puck collection zones 5 and 6 to puck collection zones 7 and 8 (as the robot would move directly from puck collection zones 5 and 6 to zones 7 and 8 if there is no valid puck in zones 5 and 6),
    c. and the distance to travel from being 1cm away from pucks in zones 5 and 6 to puck collection zones 7 and 8 (if there is/are valid puck/s in zone/s 5 or/and 6).
2. Compute the target shaft encoder count by the relationship between the shaft encoder count and the distance in cm mentioned in section 3.1.2.
3. Test and fine-tune the target shaft encoder count.

To complete the hardcoding process for all 3 distance parameters, it takes the system integrator around 3 hours.

Alternative electronic design: IR sensor to detect pucks in the collection zones

An alternative electronic design uses an IR sensor to detect pucks in the collection zones. This strips the burden of tuning the distance parameters off.

Discussion:

The main reason to make no use of any IR sensor for puck detection is this project is short. There are only 12 weeks for it and there is only 1 final competition. The puck positions in the collection zones are fixed throughout the competition and hardcoding can certainly provide an error-free “puck detection” mechanism due to the reliability of travel distance feedback by the shaft encoder. The reliability requires no experiment since the travel distance feedback just did not show any unexpected error since the team started using the shaft encoders until the end of the competition.

Nevertheless, the distance parameter tuning takes 3 hours which may seem long. However, the team has never used an IR sensor before. It would definitely take way longer than 3 hours to learn and tune the right voltage threshold for puck detection. If the project is longer or if the puck positions in the collection zones are not fixed throughout the competition, an IR sensor for puck detection is definitely needed. However, given the actual situation, the high development cost (effort and time) of using an IR sensor stops us from using it for the project. Therefore, the alternative electronic design of using an IR sensor to detect pucks in collection zones is not used.

3.4.3 Solenoid vs robot movement for dragging

Current electronic design: Robot movement for dragging

The current electronic design uses DC motors as actuators for puck dragging. This accumulates the errors in the difference of travel distance between the 2 wheels as the robot keeps moving back and forth when continuous capturing. This can cause robot misalignment and poorer dead reckoning result.

Alternative electronic design: Solenoid for dragging

An alternative electronic design uses solenoid for puck dragging. Solenoids can effectively drag a valid puck with faster speed yet no error accumulation in the difference of travel distance between the 2 wheels.

Discussion:

There are three main reasons to make no use of solenoid for puck dragging:

1. High cost for each unit of solenoid which challenges the limited budget the team has [10].
2. Require soldering which the team can hardly practice due to lack of experience and tools [11].
3. Short extended length (much smaller than even the puck width of 4cm which makes it impossible to push the puck into the capture zone) [10], [11].

Due to technical and cost constraints, the alternative design of using solenoid for puck dragging is not used.

4.0 Software design

4.1 Overview

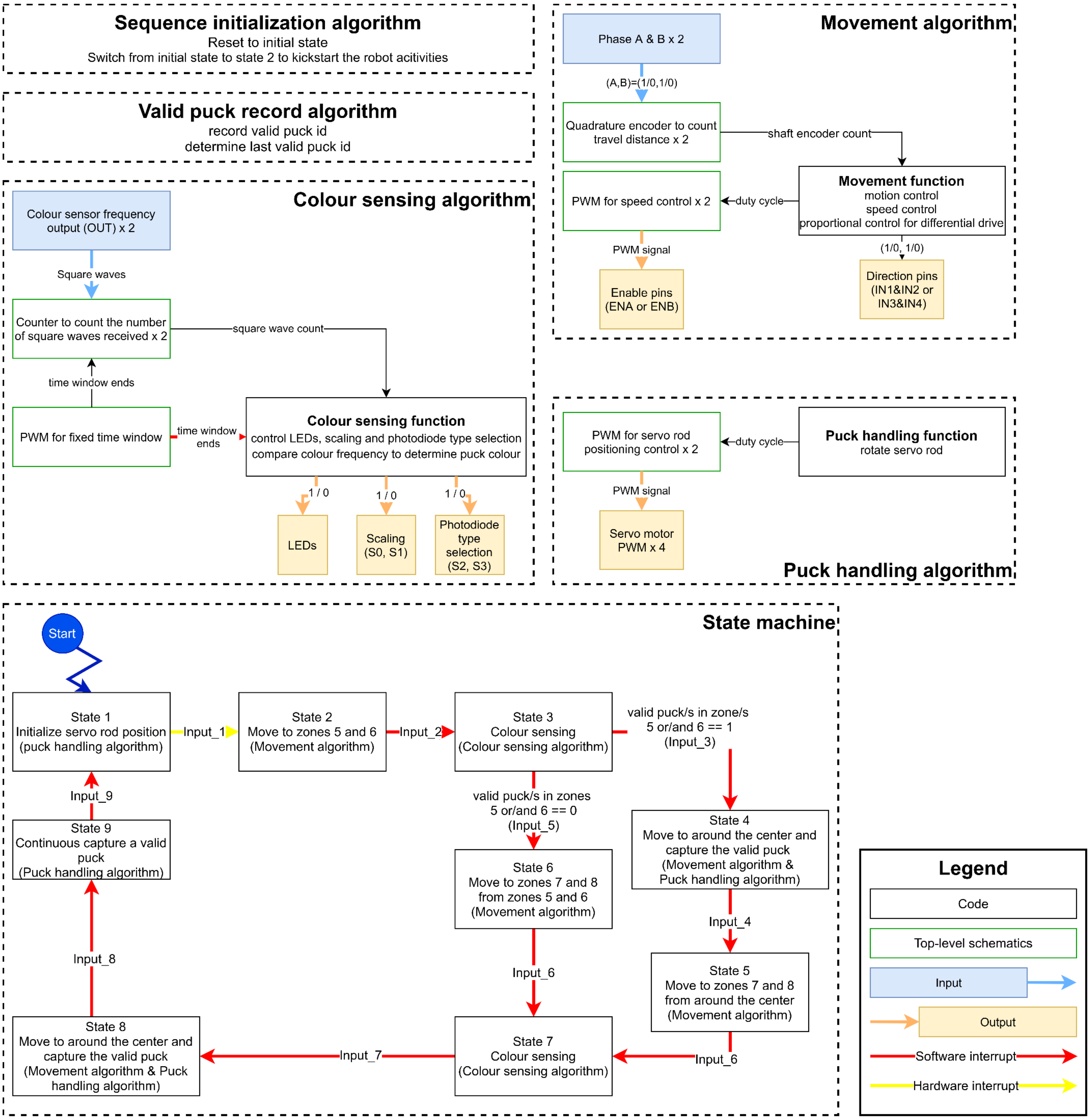


Fig. 44. Software block diagram.

As shown in Fig. 44, software mainly consists of 5 algorithms and 1 state machine. The state machine is used to implement the motion sequence as modulation of code eases code debugging and refinement. Every state uses 1 or 2 algorithms to complete the activities assigned. After resetting the robot, Input_1, a hardware interrupt, kickstarts the robot activities by switching the software from state 1 to state 2.

4.2 Software algorithms

4.2.1 Sequence initialization algorithm

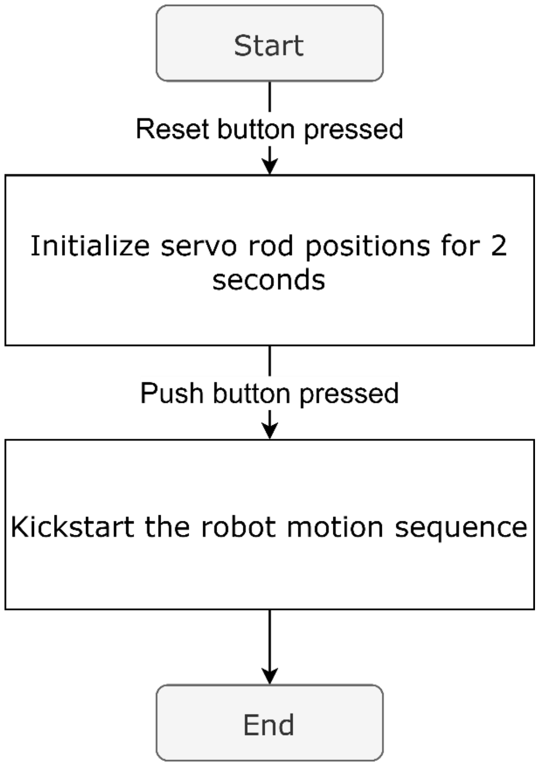


Fig. 45. Flowchart of sequence initialization algorithm.

Once the reset button on the PSoC microcontroller is pressed, the software is reset to initialize the servo rod positions by opening all of them. This takes around 2 seconds. After that, the push button is expected to be pressed to signal the algorithm to kickstart the robot activities by switching from state 1 (more or less idle except the servo rod position initialization) to state 2 in the state machine.

4.2.2 Movement algorithm

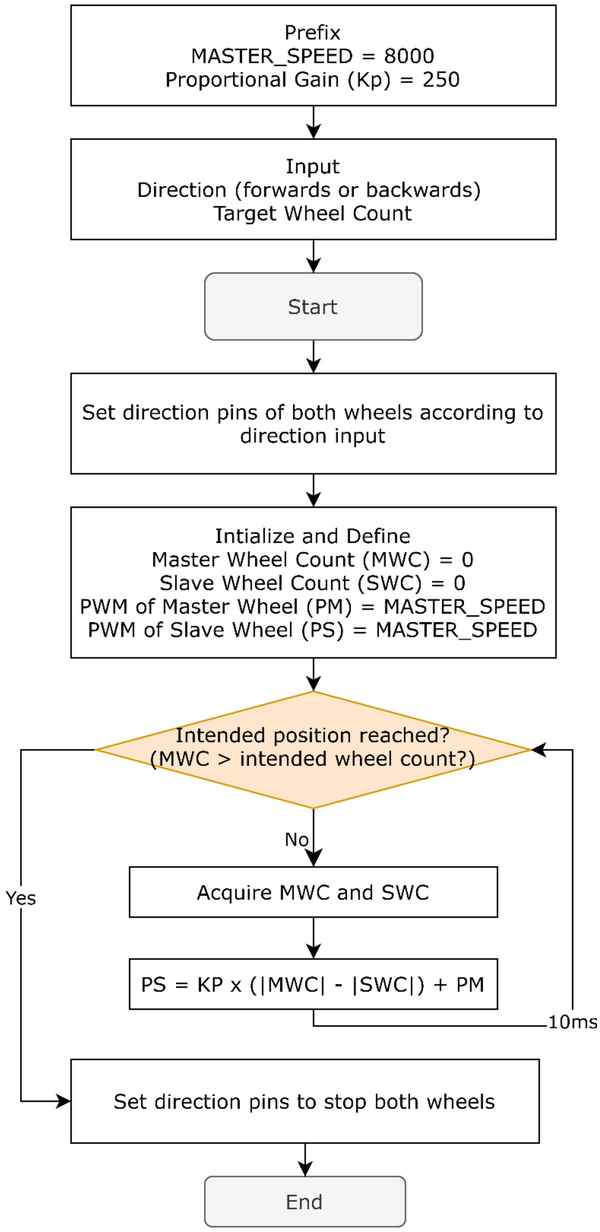


Fig. 46. Flowchart of movement algorithm.

The 2 wheels tend to have different physical speeds, despite the same coded speed due to physical limitations. Therefore, the algorithm uses a proportional controller for the differential drive mechanism for speed alignment and smooth robot movement. The right wheel works as the master wheel whose speed never changes, while the left is the slave wheel whose speed is tuned constantly.

First, the master wheel speed is fixed to be 8000µs which represents 32% duty cycle as a 25ms period is used for the PWM. The proportional gain is set as 250 which shows a small error margin as described in section 6.2. The algorithm thus takes movement direction input and target wheel (shaft encoder) count input to move the robot to an intended position in the intended direction. After that, the algorithm sets the direction pins of both wheels according to the direction input.

The algorithm thus moves the robot continuously until the target wheel count is reached where it stops the robot. During the process, the algorithm keeps adjusting the speed of the slave wheel according to the equation shown in Fig. 46. When the travelled distances between the wheels are different, the speed of the slave wheel would be varied accordingly to compensate for the difference.

4.2.3 Colour detection algorithm

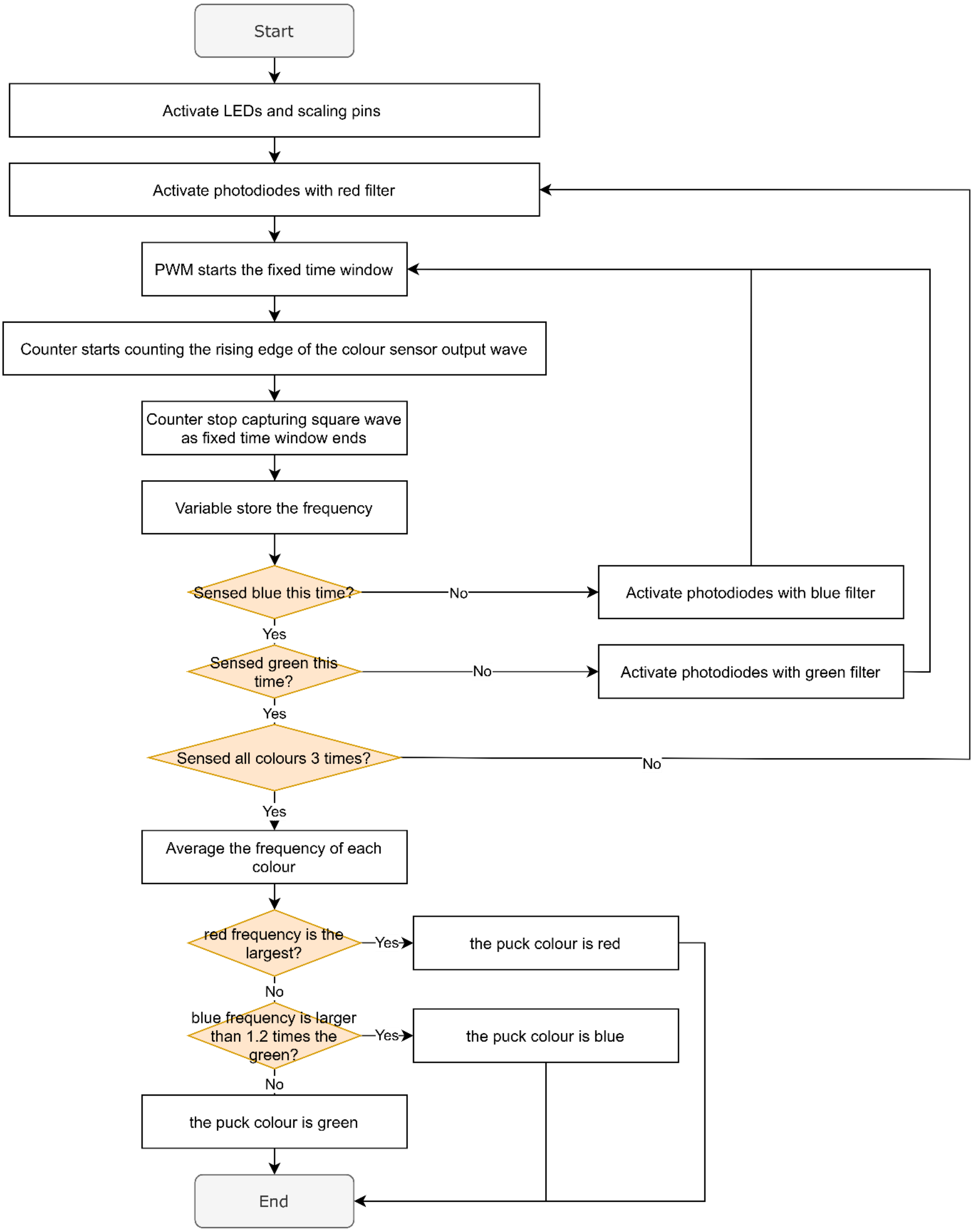


Fig. 47. Flowchart of colour detection algorithm.

The algorithm first activates the LEDs, and the frequency scaling pins as a larger output makes computation easier. It then activates red photodiodes and starts detecting the colour. The number of square waves received for a fixed time window is counted to compute the frequency. The same process repeats for blue and green photodiodes. This whole flow is then repeated 3 times for better accuracy. The highest average frequency would determine the puck colour excepts for the comparison between the blue and green since the colour sensor tends to have a high frequency when activating blue photodiodes to detect green puck. Therefore, when comparing blue and green frequencies, the blue frequency has to be 1.2 times the green frequency to determine that the puck colour is blue.

Since simultaneous colour sensing is used, the algorithm executes the same process for both sensors simultaneously. In essence, the robot calculates each sensor frequency output at the same time, by using 1 PWM and 2 different counters, and determines each puck colour at almost the same time.

4.2.4 Valid puck record algorithm

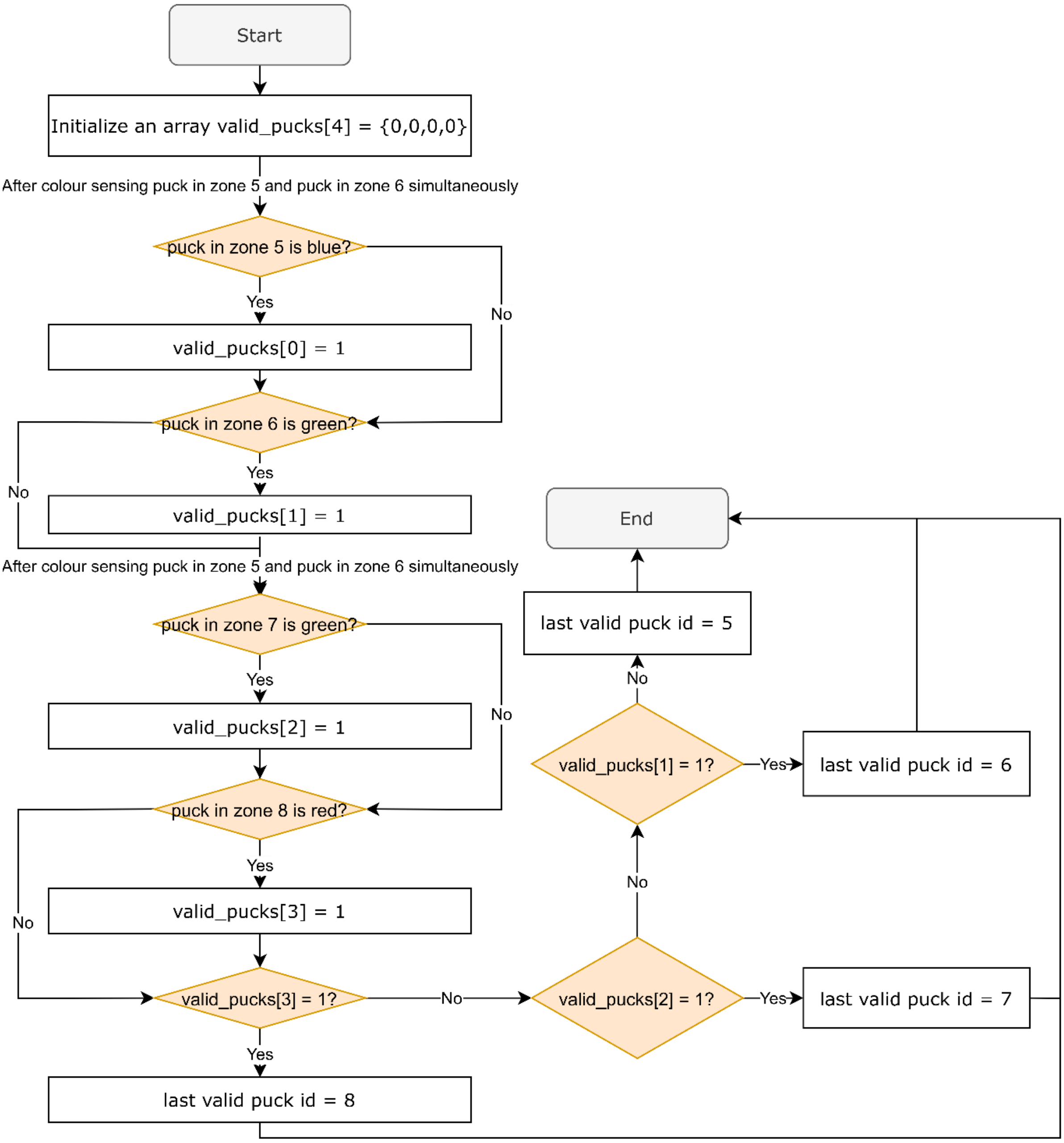


Fig. 48. Flowchart of valid puck record algorithm.

The algorithm starts by initializing an empty array with 4 elements, i.e., valid_pucks[4]. After the robot visits zones 5 and 6 via movement algorithm and colour sensing algorithm, this valid puck record algorithm determines if the puck in each of these zones is valid by comparing its colour with the pre-coded colour of the corresponding capture zone. The valid puck is recorded into the valid_pucks array. The robot would then attempt to capture the valid puck/s in zone/s 5 or/and 6, if there is any via puck handling algorithm. The algorithm repeats the process for pucks in zones 7 and 8 after the robot visits zones 7 and 8.

After visiting all the zones, the algorithm browses through the valid_pucks array to find out the last valid puck according to corresponding collection zone numbering. The algorithm thus sets the last valid puck id accordingly and completes its job. This last valid puck id would then be fed into the puck handling algorithm for continuous capturing but this is out of topic for the valid puck record algorithm.

4.2.5 Puck handling algorithm

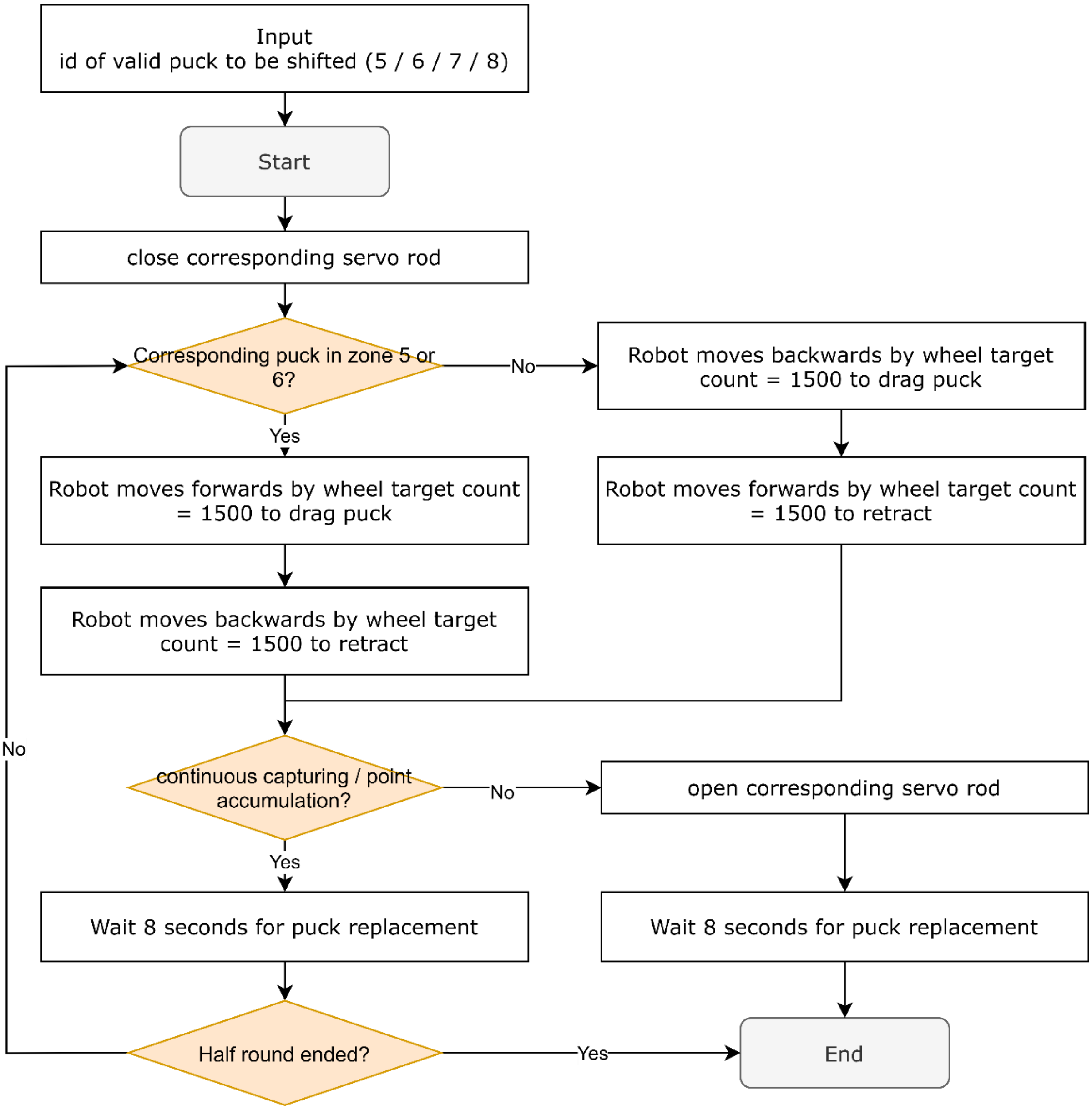


Fig. 49. Flowchart of puck handling algorithm.

The algorithm starts with an input to tell which puck to shift. The corresponding servo rod is thus closed, and the robot would then move towards the puck to push it towards the corresponding capture zone for a fixed wheel count of 1500 for the puck to be completely inside the capture zone. After that, the robot would retract to its initial position and wait 8 seconds for manual puck replacement. If the puck is not for continuous capturing, the closed servo rod would open to allow the robot to continue its activities. Else if the puck is for continuous capturing, the robot would repeat the process of moving towards the puck and retracting every 8 seconds until the half round is ended.

4.3 Alternative software algorithms

4.3.1 Navigation stack vs functions for movement

Current software design: Simple functions for movement

The current software design uses a function to move the robot for a target shaft encoder count. This relies on the distance input between the initial position and the intended position rather than the x and y coordinates input of the intended position.

Alternative software design: Navigation stack for movement

An alternative software design uses navigation stack to move the robot from its initial position to the intended position by inputting the x and y coordinate of the intended position.

Discussion:

Navigation stack provides a very easy method for positioning of the robot by using x and y coordinates which avoids the trouble of computing the distance between the initial and intended position that is needed when using functions for movement. Nevertheless, functions are very easy to develop while navigation stack bears a high development cost (time and effort). In addition, the positioning of the robot is extremely easy according to the motion sequence. The robot only has 4 possible positions after each half-round starts, which are,

1. robot in zones 5 and 6,
2. robot in zones 7 and 8,
3. robot is 1cm away from pucks in zones 5 and 6, and
4. robot is 1cm away from pucks in zones 7 and 8.

Due to the simple positioning requirement for the robot and the high development cost of navigation stack, the alternative design of using navigation stack for the movement mechanism is not used.

4.3.2 Global variable vs pointer to variable as an input to the function for colour sensing

Current software design: Pointer to variable as an input to the colour sensing function

For neater software organization, most self-defined functions are stored in other .c files rather than all in the main.c file. The colour sensing function is stored in colour_sensing.c file. The function has to change 2 local variables in main.c, which are the total frequency from the left colour sensor and total frequency from the right colour sensor after colour sensing 3 times, when calling in the main.c. There are different methods to do so, one of them is to use a void function that changes the variables in main.c by inputting the pointers to the variables to the function. This is the current software design.

Alternative software design: Global variable for the colour sensing function

An alternative software design changes the local variables to extern global variables to allow modifying of variables across multiple .c files.

Discussion:

It is always preferred to avoid the use of global variables, especially extern global variables across multiple .c files. It is because the global variables can be changed easily by any function in any .c file. This conflicts with the good programming practice of code modulation and makes it harder to debug. Even though extern global variables are extremely easy to implement compared to pointers to variables, they are avoided for code robustness and clarity. As a result, the alternative software design of using global variables for access and modification across multiple .c files is not used.

4.3.3 4 1-output PWMs vs demultiplexers with two 1-output PWMs for puck handling

Current software design: Demultiplexers with two 1-output PWMs for puck handling

The current software design uses two 1-output PWMs, each connected to a demultiplexer which outputs to 2 servo motors, to control the PWM duty cycle of each servo motor as shown in Fig. 50.

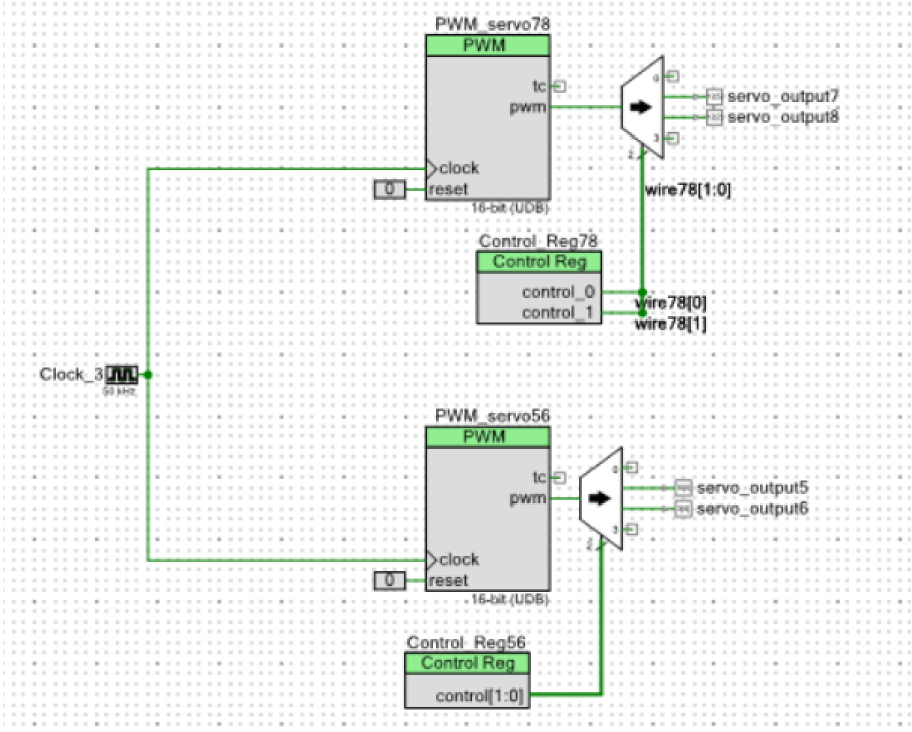


Fig. 50. Top-level schematics of puck handling subsystem.

Alternative software design: 4 1-output PWMs for puck handling

An alternative software design uses 4 1-output PWMs, each connected to a servo motor, to control the PWM duty cycle of each servo motor.

Discussion:

We experimented with both methods by changing the top schematics and building the design. The results of the UDB resource used are shown in TABLE XI. It is impossible to implement the 4 1-output PWMs since the UDB resource used has exceeded 100% where the design is failed to build. Even if the alternative design does not use up all UDB resources, the use of 2 1-output PWMs with demultiplexers saves UDB resources with low development costs. By saving more resources, it allows the software to be refined and modified with less concern regarding not enough software resources. As a result, the alternative software design of using 4 1-output PWMs for puck handling is not used.

TABLE XI
UDB RESOURCE USED IN DIFFERENT SOFTWARE DESIGNS

	2 PWMs with demultiplexers	4 PWMs
UDB resource used up (%)	87.5	104.2

5.0 Strategies and Innovation

5.1 Motion sequence

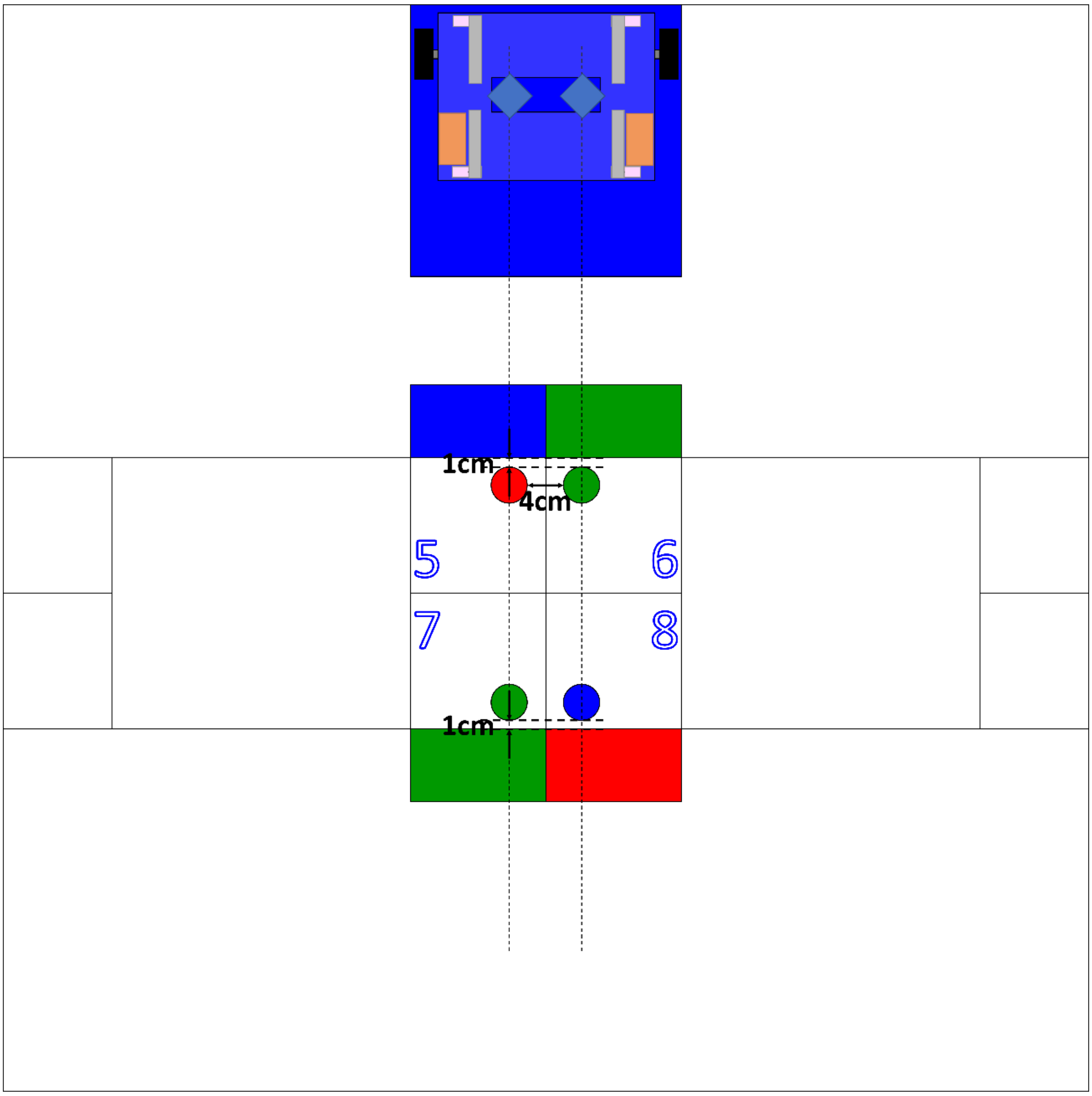


Fig. 51. Step 1 of motion sequence.

1. A half-round starts with the robot on the starting base. The distance between the two side-by-side pucks is 4cm. There is a 1cm gap between each puck and its capture zone to act as an obvious visual cue to tell that the puck circle is not touching any part of the boundary of its capture zone.

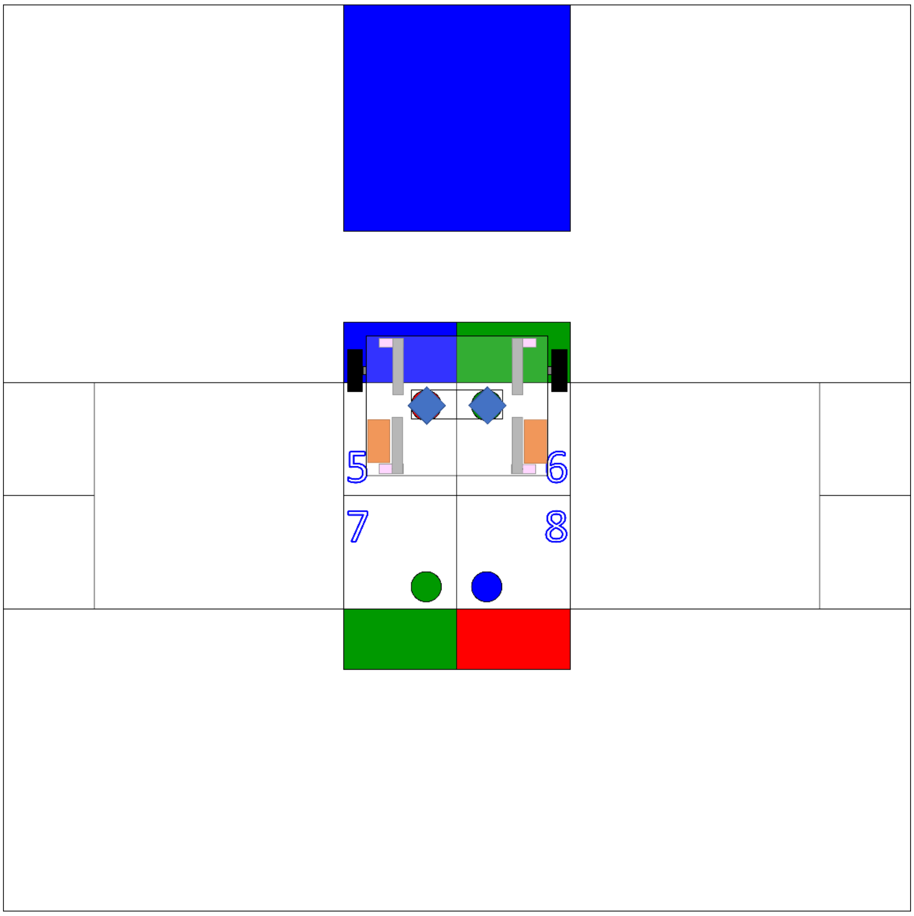


Fig. 52. Step 2 of motion sequence.

2. The robot switches from state 1 to state 2 and moves forwards to enter puck collection zones 5 and 6. The robot moves by using movement mechanism. The direction pins of both wheels are set to (0,1) for the robot to move forwards. The robot uses proportional controller for differential drive mechanism as explained in section 4.2.2. The speed of the master wheel is fixed by fixing the PWM duty cycle output to the enable pin while the speed of the slave wheel is adjusted constantly according to prefixed proportional gain and the difference in travel distance between the 2 wheels as described in section 4.2.2.

   The robot stops when the pucks are exactly under the colour sensors. The robot stops by hardcoding the target travel distance between its position in starting base and the intended position. The travel distance can be measured by shaft encoder as explained in section 3.1.1. The relationship between the travel distance and the shaft encoder count is described in the same section. Once the shaft encoder count reaches the target count, the robot stops movement.

   The robot switches from state 2 to state 3 to conduct simultaneous colour sensing strategy to sense the puck colour in zone 5 and the puck colour in zone 6 at the same time. The colour sensing process repeats 3 times and uses the highest average frequency obtained to determine puck colour to reduce the effect of random noise. After colour detection, each puck colour is compared against the pre-coded corresponding capture zone colour to determine if the puck is valid. Valid puck id, which follows the corresponding collection zone numbering, is recorded for later use of continuous capturing.

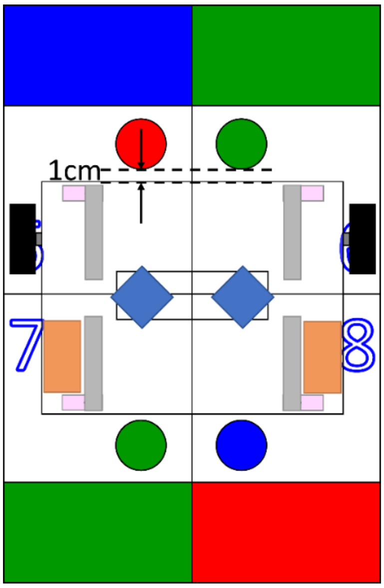


Fig. 53. Step 3 of motion sequence.

3. The robot switches from state 3 to state 4 if there is/are valid puck/s in zone/s 5 or/and 6. The robot moves forwards until its back chassis edge is 1cm away from the nearest edge of the valid puck to ensure the robot does not cover the black circle. If there is no valid puck in zone 5 or 6, the robot switches from state 3 to state 6 where the robot skips steps 4 to 8 in this motion sequence and moves directly from zones 5 and 6 to zones 7 and 8. Nevertheless, this example has a valid puck in zone 6, therefore, the steps are not skipped.

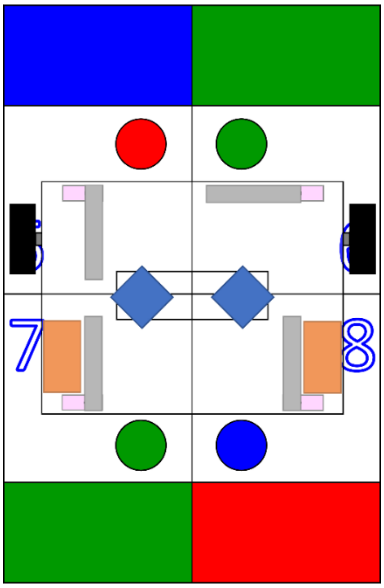


Fig. 54. Step 4 of motion sequence.

4. The robot switches on the PWM of the corresponding servo and closes the corresponding servo rod by using puck handling mechanism. 4 servo motor dragging mechanism is used, each servo motor with rod attached to each chassis corner is responsible for the puck in each collection

zone. The servo rod positioning is determined by the PWM duty cycle output from the microcontroller to the servo motor.

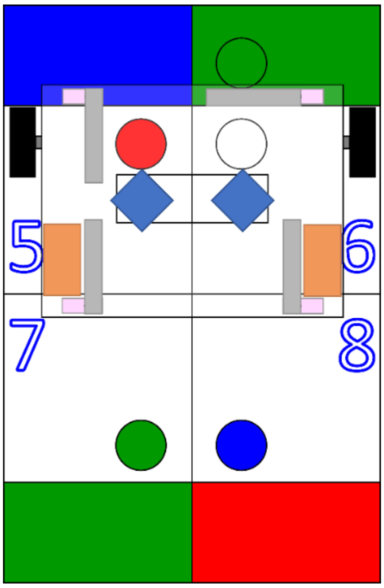


Fig. 55. Step 5 of motion sequence.

5. The robot moves backwards to push the valid puck into the capture zone until it is completely inside. Again, since the zones and the initial puck position are fixed, the target travel distance to capture a puck is hardcoded.

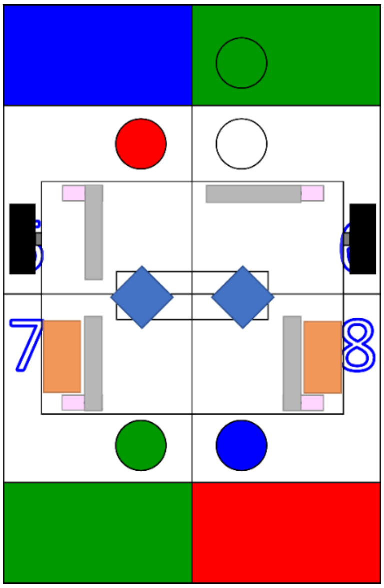


Fig. 56. Step 6 of motion sequence.

6. The robot retracts, i.e., moves forwards to be 1cm away from the nearest edge of the black circle of the valid puck.

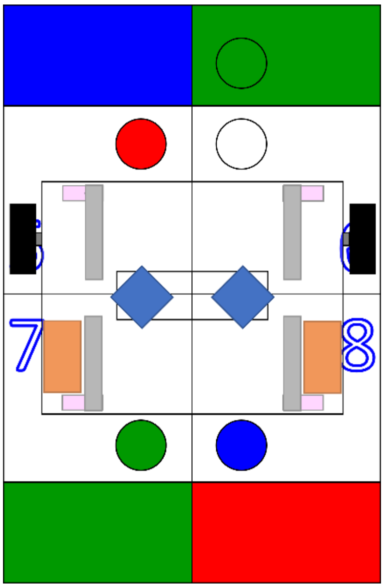


Fig. 57. Step 7 of motion sequence.

7. The robot opens the closed servo rod to allow the robot to continue its remaining activities.

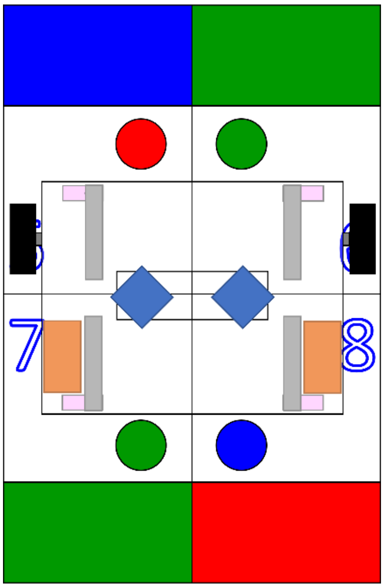


Fig. 58. Step 8 of motion sequence.

8. The robot waits 8 seconds for the judges to show the judges the captured puck and for the system integrator to manually replace the puck back into its black circle. No replaced puck detection mechanism is used since it would require the use of 4 proximity sensors which can incur interference with each other, consumes high power in total, and bear a high development cost of parameter tuning and sensor installing. Therefore, the robot relies purely on the 8-second delay to allow enough time for the manual puck replacement to be executed by the system integrator. If both pucks in zones 5 and 6 are valid, the robot repeats the (1) close corresponding servo rod, (2) move towards the valid puck to capture the puck and retract, (3) open servo rod and (4) wait 8 seconds strategy for the not-yet-captured valid puck. The robot is designed to capture puck 5 and then puck 6 one by one if both are valid.

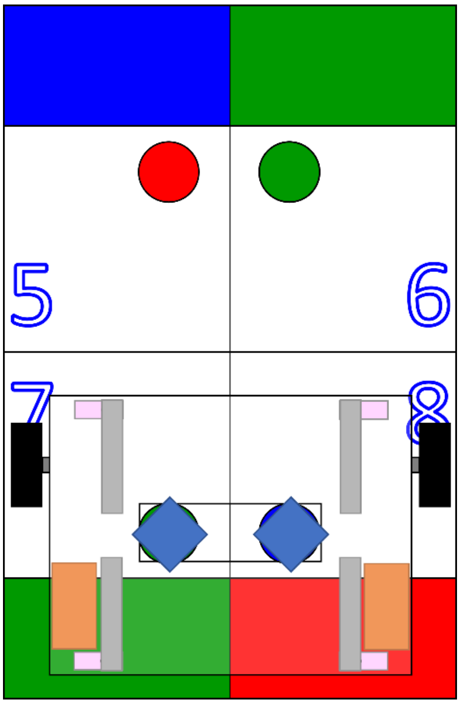


Fig. 59. Step 9 of motion sequence.

9. The robot continues its activities by switching from state 4 to state 5. The robot moves forwards to enter puck collection zones 7 and 8. The robot stops when the colour sensors are exactly above the pucks by using the same mechanism as in step 2 where the robot did so for the pucks in collection zones 5 and 6. The robot switches from state 5 to state 7 to execute simultaneous colour sensing of pucks in zones 7 and 8.

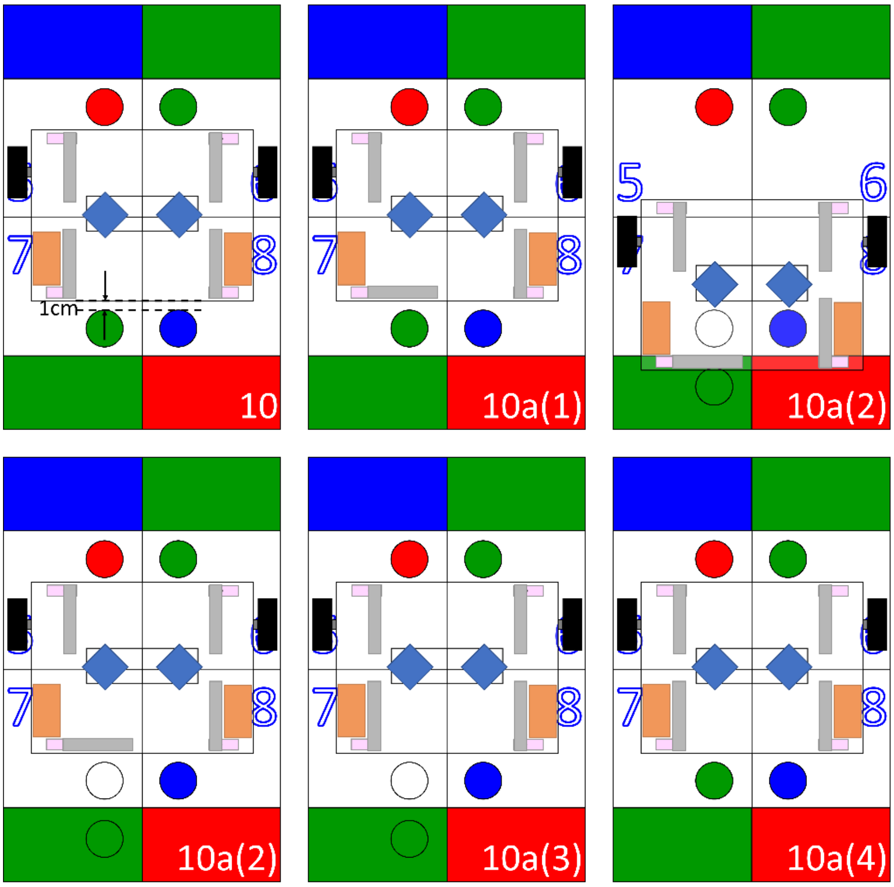


Fig. 60. Step 10 of motion sequence.

10. The robot switches from state 7 to state 8 to move backwards towards to centre to have a 1cm gap between the front chassis edge and the nearest puck edge. There are 2 cases that can then happen:
    a. If there is a valid puck in zone 7 or 8, the robot executes the
        (1) close corresponding servo rod,
        (2) move towards the valid puck to capture the puck and retract,
        (3) open corresponding servo rod and
        (4) waits 8 seconds strategy.

       If both pucks in zones 7 and 8 are valid, the robot executes the same strategy first on the puck in zone 7 and then on the puck in zone 8.

    b. If there is no valid puck in zone 7 or 8, the robot moves backwards even further to have a 1cm gap between the back chassis edge and the nearest puck edge.

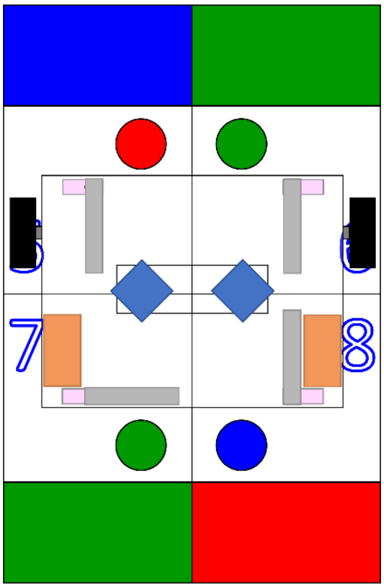


Fig. 61. Step 11 of motion sequence.

11. The robot switches from state 8 to the last state, i.e., state 9, for continuous capturing. Since the system integrator holds the mobile camera to capture the robot activity, and any valid puck before, during, and after capture, the system integrator would be physically nearest to the last valid puck, according to the collection zone number the puck is in, when continuous capturing can start. Therefore, to reduce the burden of moving around to shoot all the puck capturing processes, the robot focuses on only the last valid puck for continuous capture. The robot switches on the PWM of the corresponding servo to close the servo rod.

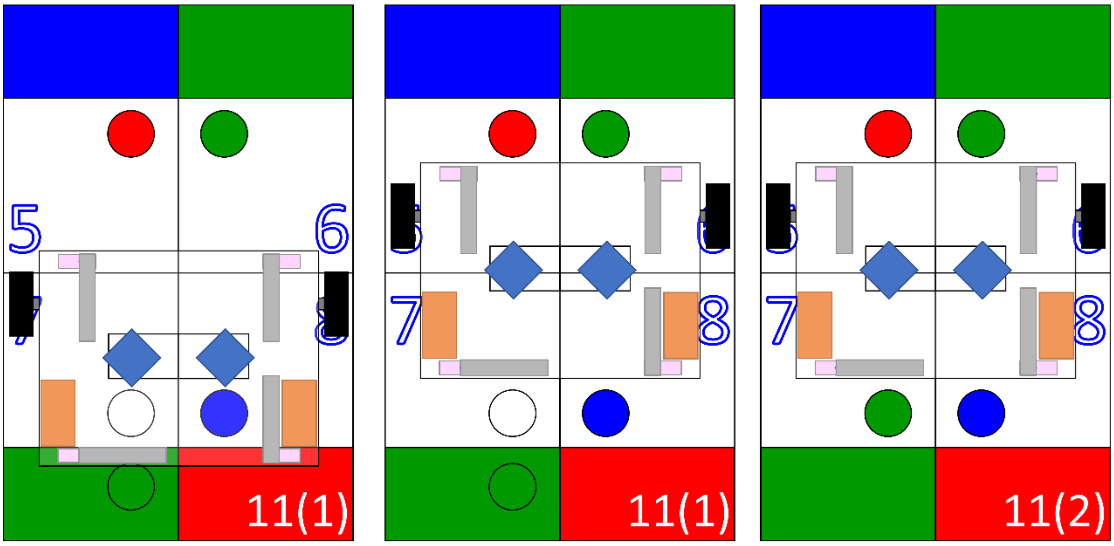


Fig. 62. Step 12 of motion sequence.

12. Once the servo rod is closed, the PWM is switched off and no PWM of any servo motor would be switched on until the end of the half round to lessen unneeded power consumption. After that, the robot keeps executing the

    (1) move towards the valid puck to capture the puck and retract, and

    (2) wait 8 second strategy,

    until the end of the half-round for point accumulation.

5.2 Innovation

5.2.1 Mechanical aspect: Lightweight 2^{nd} layer

As shown in Fig. 10, the robot makes use of 1 layer of robot base provided and corrugated boards for the 2^{nd} layer. As 2 18.5cm x 24cm robot bases are provided, teams are prompted to use either the 2 robot bases to create a 2-layer robot or only 1 robot base to create a 1-layer robot.

The size of 1 robot base is large enough to accommodate all components, but not neatly unless very careful arrangement. Very careful arrangement hardens and lengthens the robot hardware debugging process as it takes more time and effort to install the components back to their places after they are uninstalled from the robot chassis. To enhance neatness, it is possible to use 2 robot bases to create a 2-layer robot. Nevertheless, this method greatly increases the weight of the robot as each robot base bears a considerably large weight. The robot bases can be cut into smaller pieces to reduce the weight which consumes time and effort to do so. In addition, cutting of robot bases is irreversible which would incur troubles if the robot structure had to be changed during robot refinement phase later. Therefore, to avoid the issue of greatly increased weight or irreversible size reduction of robot bases, we used a different method.

We found that 1 layer of robot base is enough to accommodate most components yet provide a neat component configuration, provided the motor driver is placed on a 2^{nd} layer. Instead of using the 2^{nd} robot base provided, the system integrator makes use of corrugate boards purchased for casing to create a 2^{nd} layer that is firm enough to withhold the weight of the motor driver. As the vertical corrugated boards can also be seen effectively as a part of casing as shown in Fig. 25, the 2^{nd} layer added essentially only increases the weight of the robot by the weight of the single horizontal corrugated board on which the motor driver is placed as shown in Fig. 10.

In short, we managed to use a very lightweight 2^{nd} layer for neat component arrangement yet keep the weight added extremely low to reduce power consumption as described in section 3.2.

5.2.2 Electronics: 2 colour sensors for simultaneous colour sensing

The use of 2 colour sensors instead of 1 makes the simultaneous colour sensing mechanism possible. The colour sensors are arranged to have each placed around the middle of each puck path so that the photodiode array faces around the center of the puck as shown in Fig. 23. The use of 2 colour sensors may seem intimidating at first since an additional colour sensor requires connections of 6 more pins to the PSoC microcontroller. This can easily lead to not enough pins from the PSoC board for component connection if the robot design is complicated. Nevertheless, with the use of 2 colour sensors, the robot design can be simplified since it allows the robot to only move forward and backwards in the collection zones for zone visiting. This strips the burden of robot alignment off as only forwards and backwards movement tend not to incur significant misalignment that requires the use of reorientation mechanism. In addition, as described in section 1.3.1, the 2 colour sensors are a must to enable the shortest route strategy which greatly reduces the travel time to visit all the zones. In short, the use of 2 colour sensors for parallel colour sensing improves the efficiency and simplicity of the robot and its motion sequence.

5.2.3 Software: Full control of servo rod positioning

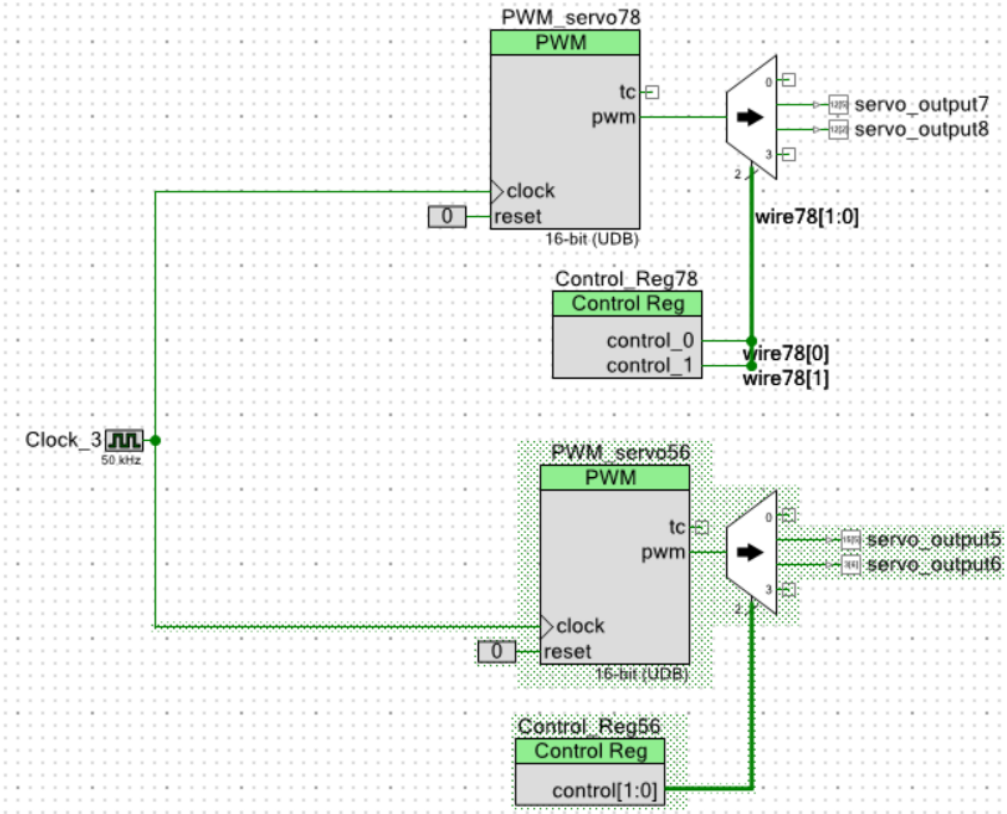


Fig. 63. Top-level schematics of puck handling subsystem.

Two PWMs are used for angular positioning of 4 servo rods to reduce software resources used. As explained in section 4.3.3, 4 1-output PWMs to control angular position of the 4 servo rods do not work since the maximum UDB resource limit is reached. There is another alternative which is two 2-output PWMs that use no demultiplexers. The reason not to use this is that, since each PWM connects to two servo motors, when the PWM is switched on the modify the angular position of one of them, the PWM nevertheless supplies current to both servo motors. Supplying current to idle servo motor not only incurs unnecessary power consumption, but it also may cause the idle servo motor to have unexpected actions.

When testing the two 2-output PWMs for servo rod positioning control, the team found that the idle servo motor which receives current when the PWM is switched on to change the angular position of the other connected servo motor, the idle servo motor tends to produce considerably large noise which shows that there are some unknown events happening within the servo motor. This may be due to the servo motors being of comparatively low quality since they are purchased from sellers with no quality assurance online. Nevertheless, this signifies that idle servo motors receiving current can cause unexpected events to happen when we cannot assure the quality of the servo motors. Therefore, the team decided to try to implement full control of servo rod positioning mechanism by switching from 2 2-output PWMs to 2 1-output PWMs, each connected to a 1:4 demultiplexer.

Assuming now the robot wants to close servo rod 5, the software executes the process as shown in Fig. 64.

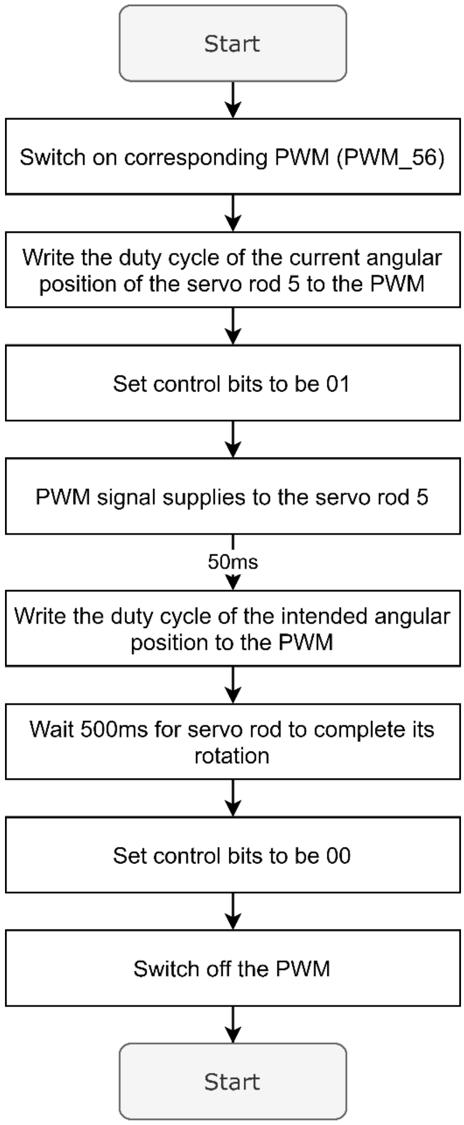


Fig. 64. Flowchart of changing angular position of a servo rod in the software.

First, the corresponding PWM is switched on. It is then written the duty cycle of the current angular position of the servo rod 5. The control bits, by default 00, are set to 01 to supply the PWM signal to the servo rod 5. Since the duty cycle represents the current angular position, the servo rod 5 has no change in angular position. This measure is implemented since the servo motor tends to rotate at an unstable speed if a sudden current (as the PWM is switched off beforehand) with a different PWM voltage duty cycle is fed. After 50ms to let the current supply be stable, the duty cycle of the intended angular position is written to the PWM to rotate the servo rod 5 to its intended position. The robot thus waits 500ms for the rod to complete its rotation. The control bits are then set to 00, their default state, and the PWM is switched off.

This strategy has the following benefits:

1. No unnecessary power consumption due to useless PWM signal outputs to idle servo motors.
2. The target servo motor would not be fed the dummy PWM duty cycle set in PWM components in the top-level schematics when the PWM is switched on. Even though

most of the time, the effect of the dummy PWM duty cycle is negligible as long as the software executes the next line of code that writes the right PWM duty cycle fast enough. But from time to time, the dummy PWM duty cycle shows its effect on the servo rod by rotating it in an unwanted direction before the servo rod follows the right PWM duty cycle in the next line of code. Therefore, to reduce the instability of servo rod performance, it is preferable to not feed the dummy PWM duty cycle into the servo motor.

3. Consistent rotation speed despite switching on and off PWM.

In short, the strategy used reduces unnecessary power consumption and improves the servo rod performance in terms of stability and consistency. Incidentally, this strategy did solve the issue of noisy idle servo motors which proves its functionality.

6.0 System integration and testing

6.1 System integration overview

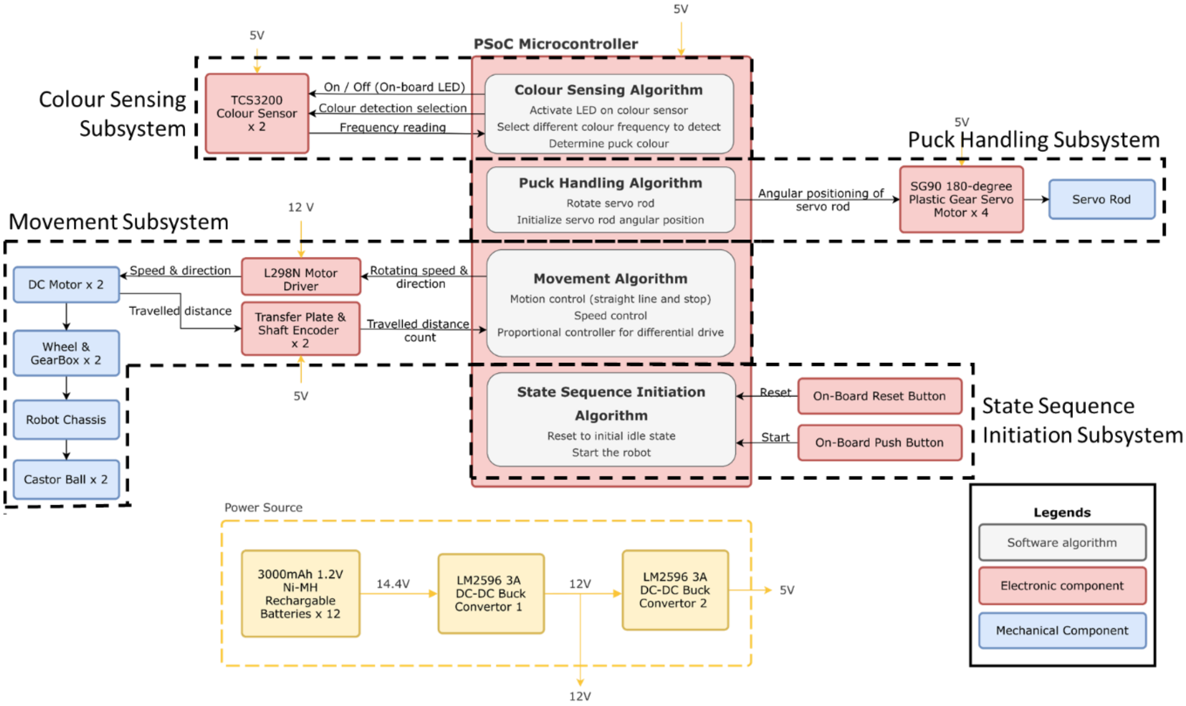


Fig. 65. System block diagram.

As shown in Fig. 65, the robot consists of 4 main subsystems to handle all the tasks provided in the final competition.

6.1.1 Colour sensing subsystem

The purpose of this subsystem is to handle the colour sensing of pucks and allow the software to interpret the colours of the pucks for further manipulation. In order to trigger the colour sensing sequence, the colour sensor’s LEDs will be switched on to act as a light source. The light emitted will then illuminate the puck, which reflects the respective colour rays back towards the photodiodes on the colour sensor. The photodiodes were selectively activated via photodiode type selection inputs from the microcontroller beforehand as described in section 4.2.3. The colour sensor interprets these colour rays and transmit them towards the PSoC board in the form of frequency. As the frequency value received varies with each colour, and that the colour with the highest frequency would be the colour of the puck being detected, the software within the PSoC board would then use the information received to decide on what step to take next.

6.1.2 Puck Handling Subsystem

The purpose of this subsystem is to handle the manipulation of pucks. Based on what data the sensors provide and the pre-planned movement coded within the software, instructions on what to do with the pucks will be given out. The given instructions would normally be in terms of what angle do we want the plastic gear servo motors to be in and how the robot should move correspondingly. Once a puck colour is detected and that the written software has decided to manipulate the puck, values on what angle the servo rod should rotate would be transmitted to the said component in terms of PWM. The frequency of the PWM would then be interpreted as the angular position of the servo rod. Once the servo rod is in place, the PSoC board would then fire out instructions towards the DC motor and the wheels to move either forward or backward. This would in the end lead to the pushing of pucks into the puck capture zone.

6.1.3 Movement Subsystem

The purpose of this subsystem is to be in charge of navigating the robot along the competition arena. As our robot does not use real-time sensors, all the values for how the robot should move have been pre-planned and coded within the software. Whenever there is a need for moving, stopping, or reversing, the software written in the PSoC board will send out respective signals towards the motor driver to manipulate the direction and speed of both wheels.

Whilst moving, the shaft encoder connected to the dc motor would continuously count the number of revolutions made by the wheels. This would then be interpreted into distance travelled by the software written. We have also implemented a proportional differential drive controller for our movement system. The one wheel that is designated as the master wheel will be in charge of how the slave wheel moves. The revolution count of both the wheels will be monitored and that slave wheel’s rotation speed will be increased or decreased to fit the master wheel’s revolution count.

6.1.4 State Sequence Initiation Subsystem

The purpose of this subsystem is to serve as a trigger for initiating the robot and to push the robot through all the allocated states. With the press of the onboard button, the initial idle state would move towards the allocated state 2 as described in section 4.2.1. At the end of each state, there will be a software state traverser that increments the state count of the robot mostly by 1, until all the states have been reached then the robot falls back to the idle state, state 1.

6.1.5 Power Source Subsystem

The purpose of this subsystem is to act as the power source for the whole robot. 12 Ni-MH rechargeable batteries are connected in series to create a total voltage of 14.4V. Then through the use of two separate DC-DC buck convertors, the 14.4V are stepped down to 12V and 5V respectively. The 12V is used to power the motor driver while the 5V is used to power the rest of the electrical components.

6.2 Test environments

6.2.1 Proportional gain (Kp) & error margin of shaft encoder count

The Kp value is the proportional gain of the of the proportional differential controller. This value is in charge of making the robot move in a straight line. To test the optimal value for this value, we had the robot move on the arena with guidelines. We then fixed the movement speed at the target travel distance of the robot. In this case, the movement speed will be 8000 and the travel distance as well. With the travel distance at a large value, although no necessary for the actual competition, we will be able to see the errors collecting from the long travel, further verifying what Kp values would be optimal. Then we proceeded to change the Kp value until the robot no longer deviate from the guidelines with a relatively small error margin. Each Kp value is tested 2 times to obtain the average error margin for evaluation. There was a general guide for changing the Kp value as a higher Kp might cause the robot to oscillate while a low value might not be enough gain to allow the robot to move straight.

TABLE XII
PROPORTIONAL GAIN AND ITS RESPECTIVE ERROR MARGIN

Proportional Gain, Kp	Error Margin (quadrature encoder count)
120	±22
250	±10
480	±18

6.2.2 Distance (Target wheel count)

As mentioned before, our robot relies on the hardcoded distance values to achieve its tasks. As for how the values are obtained, we first start with a general value of 1000 for the robot to move straight into the puck capturing zone. This value would then be used to estimate the rest of the movement distance values. The estimated values would then be adjusted slightly to fit what movement is required. In the final testing and fine-tuning, we had made sure that all the hard-coded values are within ±2 of the actual distance values, that is we have reached a point of accuracy that it would be hard to discern any physical difference if we were to add/minus 2 from the distance wheel count. To conclude, the method of testing here is mostly trial and error with some well-educated guess on the staring values to shorten the testing phase.

TABLE XIII
MOVEMENT INSTRUCTION AND ITS RESPECTIVE DISTANCE WHEEL COUNT

Movement	Distance Wheel Count
Straight-Line movement from starting base to centre of arena	9100
Forward movement for puck capturing	1800
Backward movement for puck capturing	1900
Slight Shift forward for better puck top-down viewing	275
Slight Shift backward for better puck top-down viewing	300

6.2.3 Colour sensing

To ensure that the colour sensing mechanism worked, we had reused the codes from a previous Viva Voce Task to ensure that the codes would not be at fault. Of course, we had also double checked all the assignment pins to be correct as a new pin assignment is made for our new robot. Before attaching the colour sensors to the robot, we had individually tested the functionality of the colour sensor by putting them through a test run of puck colour sensing. Each colour sensor is tested 3 times on each puck colour. The results are satisfactory as the puck colours are detected correctly in all the tests. After that, we then move on to attach them onto the robot and have a final quality check on their functionality. The two sensors are tested the simultaneous colour sensing function with different pairs of puck colour to ensure the colour sensing algorithm has no error. Each combination is tested 3 times. With this, we have removed any concerns of coding error, wiring error and any assembly mistakes that might have happened while attaching the colour sensors onto the robot.

TABLE XIV
COLOUR SENSOR TESTING RESULTS

Colour sensor	Puck colour(s)	Error rate (%)
Individual left colour sensor	Red	0
	Blue	0
	Right	0
Individual right colour sensor	Red	0
	Blue	0
	Right	0
Colour sensor pair	Left: Red; Right: Blue	0
	Left: Red; Right: Green	0
	Left: Red; Right: Red	0
	Left: Blue; Right: Red	0
	Left: Blue; Right: Green	0
	Left: Blue; Right: Blue	0
	Left: Green; Right: Blue	0
	Left: Green; Right: Green	0
	Left: Green; Right: Green	0

6.2.4 Puck handling

The puck handling mechanism had a lot of testing required with our initial flicking design. We had to test that angular speed and movement of the puck flicking would hit the puck into the puck capturing zone consistently. We had also tried doing so in a low powered environment to check in the off chance of the batteries running out of power midway through the competition, would the flicking mechanism still work as intended. This had eventually led us to switch to puck dragging as puck dragging does not involve constant servo rod angular movement. Now, we would just need the PWM values for the servo to maintain an open gate position and closed gate position. To find those values, again trial and error have been conducted. However, some initial values can be calculated as a 50kHz PWM period is used for the servo motors, this would mean that only a change to the pulse width of 1ms is needed to rotate the servo rods by 180 degrees. With this in mind, the servo rod open and closed gates positions are as in table XV. Although the PWM values for servo 5, 8 and servo 6,7 should be the same in theory, the slight errors in assembling the servo motor onto the robot has caused deviations in the orientation of the servo motors, thus, micro adjustments on the PWM values of the servo open/close positions are made.

TABLE XV
SERVO ROD POSITIONING AND ITS RESPECTIVE PWM VALUES

Servo Rod	Closed Gate PWM Value	Open Gate PWM Value
5	940	890
6	915	965
7	925	975
8	930	885

6.2.5 Puck replacement

For the design of our robot, we have decided to go with a delay-based puck recognition design. This means that the robot will acknowledge that the puck will be replaced by the time that the delay counter has finished. To determine this value, we have decided to go with a slow and steady approach to avoid potential touch penalties. Taking into account the amount of streaming delay on Zoom, and that the spectators might not be paying full attention, we have generously increased the delay count to 8 seconds. This value is not at all random as we have deemed 3 seconds enough for a distracted spectator to observe the puck capturing process, a generous 2 second stream delay is allocated for fear of the worse while another 3 seconds is given to the system integrator to replace the puck fully within the black circle. The 3 seconds for puck replacement is an average value of the amount of time the system integrator uses to place the puck back within the black circle firmly for a total of 10 times. The way we monitor this replacement is strict because we did not know how accurate this placement is monitored, and again with the safety is key policy, we decided to be very generous with the time we allocated for puck replacement. However, after having experienced the competition first-hand, we can say that we have been too generous with how much delay we built into our robot. A hind-sight analysis would say that about 3 second delay would be adequate for the competition.

7.0 Project management

7.1 Team roles

Belbin Team Roles is developed by Dr Meredith Belbin in 1981 and is, nowadays, one of the most effective and common tools for team construction. Despite infinite types of possible behaviour of human engagement, the range of useful behaviours contributing high team performance is finite. These behaviours are grouped by Dr Meredith Belbin into 8 related clusters, i.e., the 8 team roles. The 8 team roles consist of 3 people-oriented roles, i.e., co-ordinator, resource investigator and team worker; 3 action-oriented roles, i.e., implementer, shaper, and completer finisher; and 2 cerebral roles, i.e., plant and monitor evaluator.

TABLE XVI
BELBIN TEAM ROLES SCORE RESULT

Team roles	Chia Kok Bond (System Integrator)	Mah Yuen Yee (Programmer)
Implementer	9	9
Co-ordinator	7	11
Shaper	11	15
Plant	12	6
Resource investigator	7	6
Monitor evaluator	7	9
Team worker	8	8
Completer finisher	9	6

Note: blue-shaded cells signify the top 4 team roles of the highest score of each / both members and the respective score of that member

The 4 team roles of the highest score of Chia Kok Bond are implementer, shaper, plant, and completer finisher; that of Mah Yuen Yee are implementer, co-ordinator, shaper, and monitor evaluator. Both members are implementers and shapers.

Implementers are known to be disciplined, practical, reliable, and efficient. As such, implementers are very good at actions. The work of turning ideas into practical actions is what an implementer is good at. It helps the team a lot that both the system integrator and the programmer are implementers. Kok Bond, the system integrator, is capable to turn ideas into actions by experimenting different methods of robot construction. Since robot construction requires many trials and errors during refinement and improvement phase, Kok Bond being an implementer can certainly improve the robot construction with reliable and practical ideas and actions. On the other hand, Yuen Yee, the programmer, is capable to turn ideas into implementation by experimenting different software designs when inspiration comes. This can certainly improve the software design which has numerous approaches. Only implementation of ideas and thus experiments can keep refining the software design. However, as implementers, we may be slow to recognise and respond to new possibilities. We may be reluctant to change plans for positive changes. With this in mind, we will consciously keep ourselves more open to possible positive changes, such as suggestions from each other, peers, or the lecturer, since more heads are better than one.

Besides, both of us are shapers. Shapers are dynamic, assertive and may even thrive on pressure. Both of us are highly motivated for achievement and have the courage and confidence to confront challenges. This is a good sign for the project since this project has a final competition. The system integrator must be increasingly stressful when the final competition is approaching. Being thrive on pressure, the system integrator is capable to debug some last-minute unexpected errors of the robot when the final competition is imminent. On the other hand, the programmer may need to make last-minute changes to the software algorithm when things go wrong right before the competition. Being thrive on pressure, the programmer is capable to support the team by making sensible software design decision even when feeling stressful. In addition, motivation from being sharpers would counteract the reluctance to change as implementers. However, shapers are prone to provocation. Double shapers in a 2-member team may lead to conflicts with no one in the team can act as the moderator. With this in mind, we will be mindful to be reasonable in discussions. Anyhow, double shapers mean that both of us understand that effective discussions can cause occasional aggressiveness. Therefore, it should not be a problem for us to deal with negative emotions during discussions since both of us know the positive intention of each other.

Yuen Yee is a co-ordinator who is described as confident, calm, and mature. Being able to focus on the team's main goal, listen to others and identify talents, she is good at delegating tasks and organising the team for the best collective performance. In case the robot construction has any unexpected error and requires more time and effort from the system integrator than expected, Yuen Yee, the programmer, can reanalyse the situation and redistribute tasks among the members to lessen the burden of the system integrator. In this pandemic situation, each team members are limited in terms of what they can do for the project due to geographical and distance constraints. Therefore, the skill of being able to adjust task distributions constantly is very much helpful for the team to work out the project efficiently. In addition, the drawback of coordinator being able to over delegating tasks is weakened since it is almost impossible to do so in a group of 2, where tasks over-delegating is obvious and ineffective for the project, especially when both have their own defined tasks as a system integrator or a programmer.

Kok Bond is a plant who is often described as creative, unorthodox, and individualistic. As such, he is good at generating original and innovative ideas for the team to tackle difficult problems. It is extremely constructive to have the system integrator to be a plant. It is because robot construction always requires creative and innovative approach for refinement. Being innovative increases the chances of creating a robot of high quality and with silver bullet to win over other teams in terms of robot design and competition strategy. However, he may tend to be forgetful or absent-minded to carry out the tasks assigned. Nevertheless, there are numerous digital tasks organization applications such as to-do-list apps for reminders of daily tasks. Therefore, by making use of alike tools, it should not be a major issue.

Yuen Yee is a monitor evaluator who is often characterized as strategic, serious, and discerning. Being apt to analyse problems and evaluate options, she is good at assessing feasibility and values of suggestions. A strategic programmer definitely helps the team as the software organization would be designed to be clearer and cleaner that allows easier debugging. Besides, since she is apt to evaluate different possibilities and provide the pros and cons for judgement, she is capable to make sensible software design decision among numerous choices. This saves time and effort as she tends not to design and troubleshoot the software blindly by trials and errors. However, monitor evaluator tends to lack ideas or inspirations for solutions. This can be greatly counteracted by the plant in the team who can provide novel ideas.

Kok Bond is a completer finisher who is labelled as conscientious and painstaking. Possessing a great capability of polishing and refining the robot, he is very good at ensuring the robot is as near perfect as possible and error-free before the project delivery. It's self-evident that this skill is extremely important to the project since the project has a heavy weightage based on the final competition. Being able to ensure nothing in the robot goes wrong before the competition is a must to reduce the risk of robot malfunction during competition. However, a completer finisher may be prone to worrying disproportionately and reluctant to delegate tasks to others. Therefore, he should try to strike a balance between the project practical outcomes and perfection.

There are 2 team roles that are not in the top 4 of either members, which are resource investigator and team worker. Resource investigators are known to be enthusiastic, outgoing, and curious. They are capable to explore available resources and build external contacts that may be useful for the team. Being poor of diplomacy in this team should not be a huge issue since it is limited to gain physical help from external contacts under current pandemic situation. However, we have to be mindful that we should consult the lecturer when we need help and advice. Besides, the team lacks team worker. Team worker is often characterized as perceptive, sociable, and co-operative. Lack of team worker may worsen the negative effect due to double shapers who can easily led to conflicts. It may take some time for the team to become cohesive and efficient as none of us is team worker.

In conclusion, overall, the team has a diverse team roles constitution, as top 4 of both members account for 6 out of the 8 team roles. For seamless execution and timely completion of the project, both members must make best use of the strengths of their Belbin team roles when executing their own project-oriented team role (system integrator or programmer). For the two team roles that are not included in the top 4 of both members, we should keep in mind to be more co-operative and ask for help when needed to learn to be more like a team worker and a resource investigator.

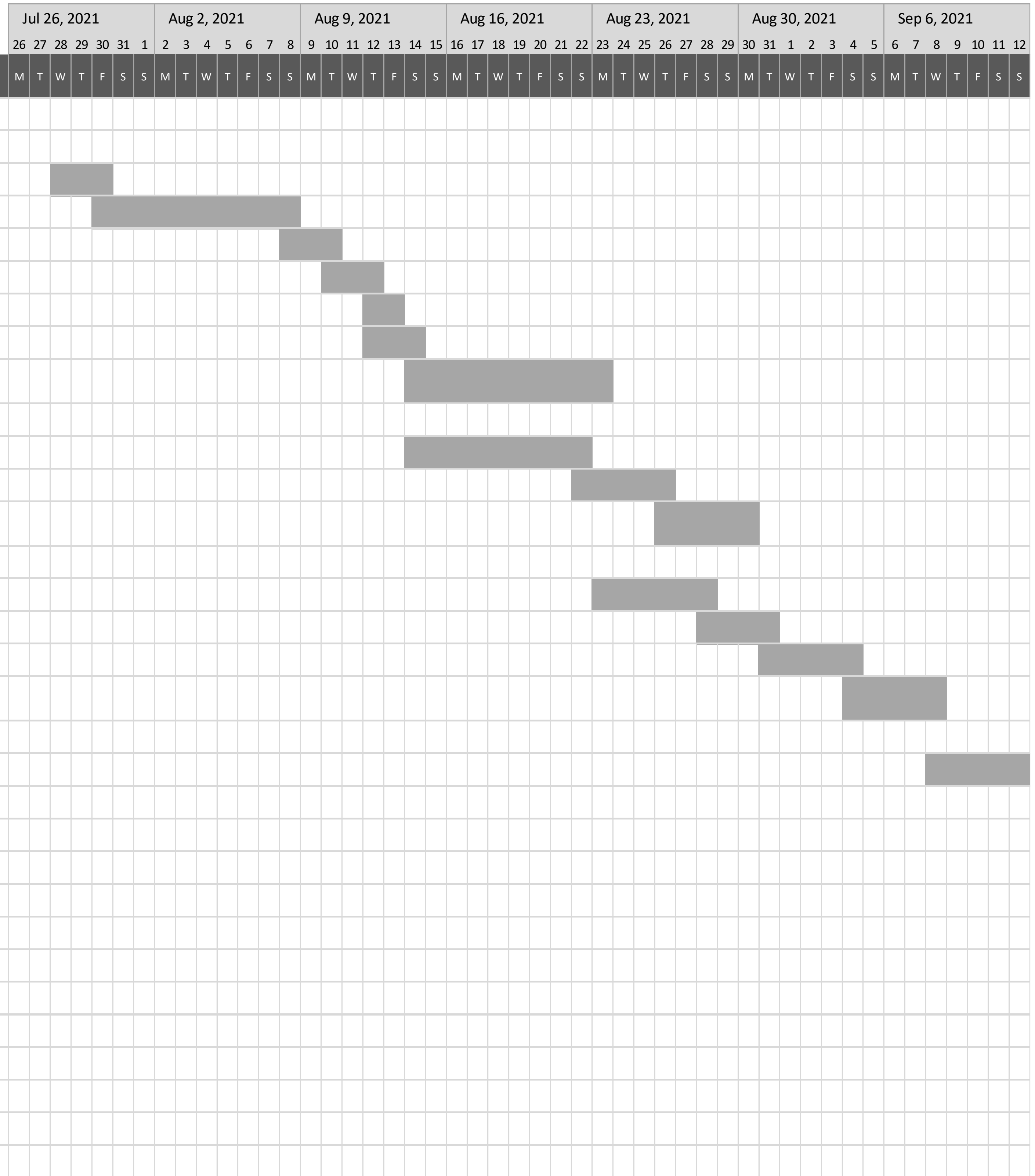
7.2 Gantt chart

ECE3091 The Monash Puck Shifting Robot

Team 15
Chia Kok Bond (Project Lead)
Mah Yuen Yee

Project Start: 2021-07-26
Today: 2021-10-23
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Preparation and design of robot				
Analyse the requirements	Both	100%	2021-07-28	2021-07-30
Think of solutions	Both	100%	2021-07-30	2021-08-08
Discuss and choose an optimal solution	Both	100%	2021-08-08	2021-08-10
Develop detailed design	Both	100%	2021-08-10	2021-08-12
Determine components needed	Both	100%	2021-08-12	2021-08-13
Prepare project proposal	Both	100%	2021-08-12	2021-08-14
Purchase additional components and test arrived components	CKB	100%	2021-08-14	2021-08-23
Programming of robot for prelim tasks				
Program subsystem for movement for prelim tasks	MYY	100%	2021-08-14	2021-08-22
Program subsystem for colour sensing for prelim tasks	MYY	100%	2021-08-22	2021-08-26
Program subsystem for puck handling for prelim tasks, and integrate all the code	MYY	100%	2021-08-26	2021-08-30
Construction of robot for prelim tasks				
Build subsystem for movement for prelim tasks	CKB	100%	2021-08-23	2021-08-28
Build subsystem for colour sensing for prelim tasks	CKB	100%	2021-08-28	2021-08-31
Build subsystem for puck handling for prelim tasks	CKB	100%	2021-08-31	2021-09-04
Integrate subsystems into, test and troubleshoot the system for prelim tasks	CKB	100%	2021-09-04	2021-09-08
Robot for final competition				
Improve and finalize robot design	Both	100%	2021-09-08	2021-09-19
Purchase additional components	CKB	100%	2021-09-19	2021-09-21
Test arrived components	CKB	100%	2021-09-21	2021-09-26
Adjust the software for final competition	MYY	100%	2021-09-19	2021-09-27
Remodel the system for final competition	CKB	100%	2021-09-27	2021-10-04
Test and troubleshoot the system	Both	100%	2021-10-04	2021-10-09
Refining and finalizing				
Prepare presentation	Both	100%	2021-10-04	2021-10-10
Fine tuning and improvement of robot	Both	100%	2021-10-09	2021-10-13
Prepare project report	Both	100%	2021-10-10	2021-10-19
Final test and troubeshoot of the system	Both	100%	2021-10-13	2021-10-19
Final competition	Both	100%	2021-10-19	2021-10-20
Finish project report	Both	100%	2021-10-20	2021-10-23



ECE3091 The Monash Puck Shifting Robot

Team 15
Chia Kok Bond (Project Lead)
Mah Yuen Yee

Project Start: 2021-07-26
Today: 2021-10-23
Display Week: 8

TASK	ASSIGNED TO	PROGRESS	START	END
Preparation and design of robot				
Analyse the requirements	Both	100%	2021-07-28	2021-07-30
Think of solutions	Both	100%	2021-07-30	2021-08-08
Discuss and choose an optimal solution	Both	100%	2021-08-08	2021-08-10
Develop detailed design	Both	100%	2021-08-10	2021-08-12
Determine components needed	Both	100%	2021-08-12	2021-08-13
Prepare project proposal	Both	100%	2021-08-12	2021-08-14
Purchase additional components and test arrived components	CKB	100%	2021-08-14	2021-08-23
Programming of robot for prelim tasks				
Program subsystem for movement for prelim tasks	MYY	100%	2021-08-14	2021-08-22
Program subsystem for colour sensing for prelim tasks	MYY	100%	2021-08-22	2021-08-26
Program subsystem for puck handling for prelim tasks, and integrate all the code	MYY	100%	2021-08-26	2021-08-30
Construction of robot for prelim tasks				
Build subsystem for movement for prelim tasks	CKB	100%	2021-08-23	2021-08-28
Build subsystem for colour sensing for prelim tasks	CKB	100%	2021-08-28	2021-08-31
Build subsystem for puck handling for prelim tasks	CKB	100%	2021-08-31	2021-09-04
Integrate subsystems into, test and troubleshoot the system for prelim tasks	CKB	100%	2021-09-04	2021-09-08
Robot for final competition				
Improve and finalize robot design	Both	100%	2021-09-08	2021-09-19
Purchase additional components	CKB	100%	2021-09-19	2021-09-21
Test arrived components	CKB	100%	2021-09-21	2021-09-26
Adjust the software for final competition	MYY	100%	2021-09-19	2021-09-27
Remodel the system for final competition	CKB	100%	2021-09-27	2021-10-04
Test and troubleshoot the system	Both	100%	2021-10-04	2021-10-09
Refining and finalizing				
Prepare presentation	Both	100%	2021-10-04	2021-10-10
Fine tuning and improvement of robot	Both	100%	2021-10-09	2021-10-13
Prepare project report	Both	100%	2021-10-10	2021-10-19
Final test and troubeshoot of the system	Both	100%	2021-10-13	2021-10-19
Final competition	Both	100%	2021-10-19	2021-10-20
Finish project report	Both	100%	2021-10-20	2021-10-23

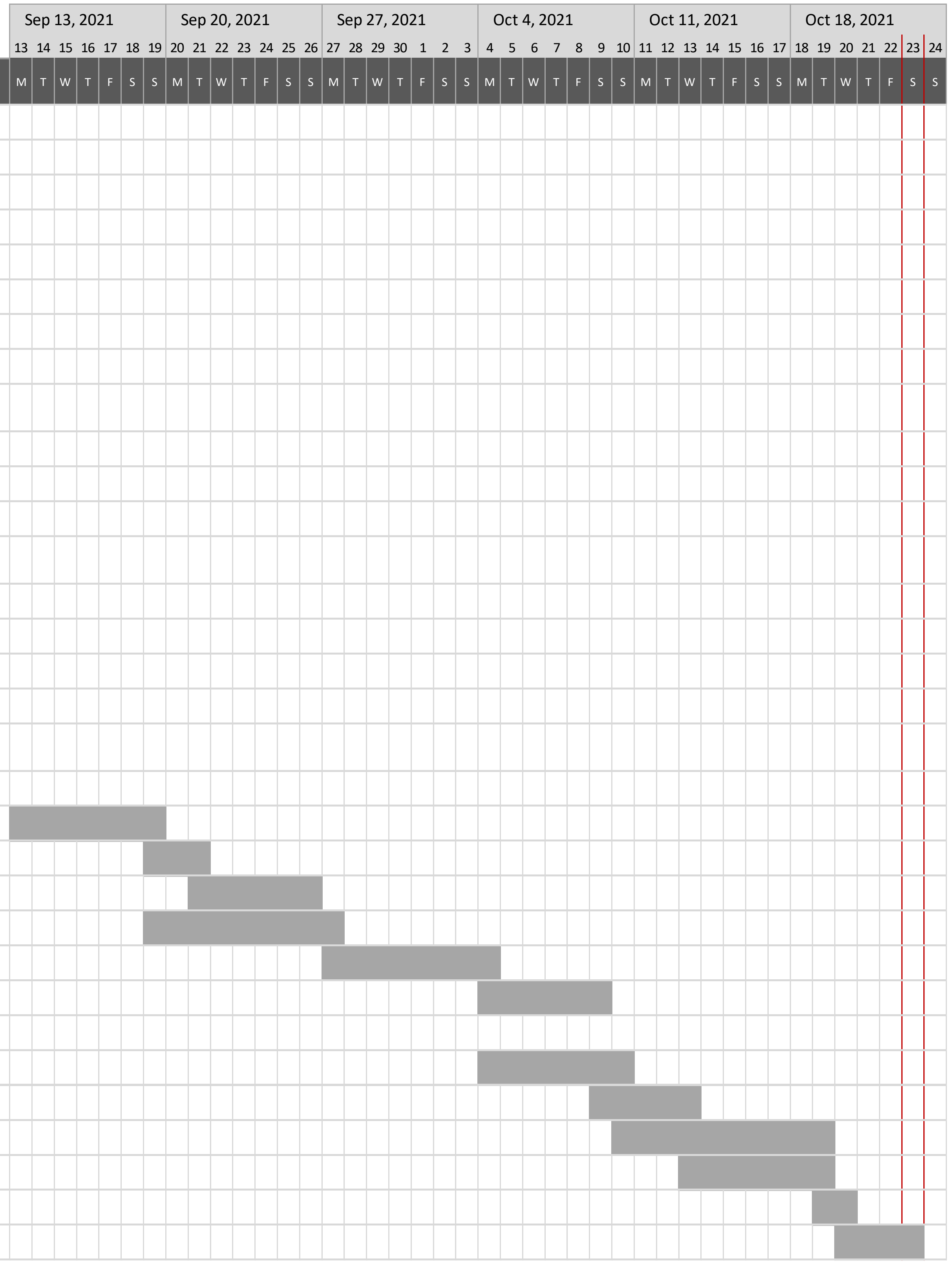


Fig. 66. Gantt chart.

7.3 Critical path identification

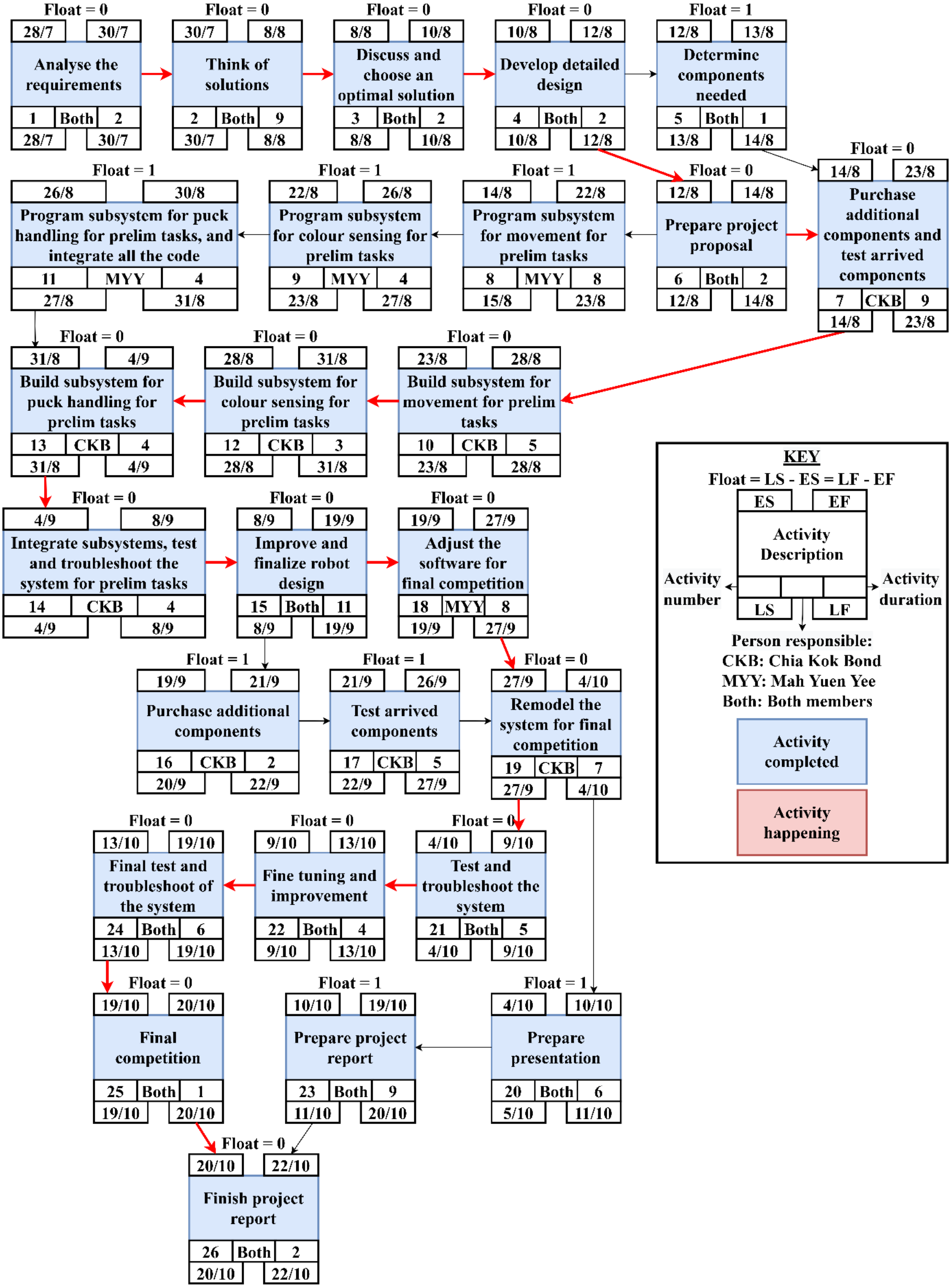


Fig. 67. Critical path network diagram.

7.4 Project budget

TABLE XVII
PROJECT BUDGET LIST

Item	Price	Unit	Total (Including Shipping and Discounts)	Source
Needle File Set	RM7.00	1	RM11.60	https://shopee.com.my/!!%EF%B8%8FREADY-STOCK!!%EF%B8%8F10PCS-NEEDLE-FILE-SET-KIKIR-KAYU-KIKIR-HALUS-PRECISION-NEEDLE-FILE-KIKIR-BESI-i.450195036.9168359436
Felt Pads	RM8.50	1	RM10.50	https://shopee.com.my/3M-Scotch-White-Felt-Pads-SP851-NA-12pcs-2.85cm-diameter-Round-i.264160247.5970051221
Screwdriver Set	RM5.50	1	RM7.50	https://shopee.com.my/66-Happy-Tool-DIY-High-Quality-Convenient-Hand-Tools-Screw-Driver-Set-(2-Pcs）-i.136569371.6901927558
Eye Protection Safety Glasses	RM1.20	1	RM3.20	https://shopee.com.my/(READY-STOCK)-Eye-protection-Safety-Glasses-goggle-(clear-grey)-cermin-mata-i.832674.751200706
Battery Holder	RM3.60	2	RM9.20	https://shopee.com.my/Battery-Holder-only-Casing-with-On-Off-Switch-Single-Double-Slot-AA-AAA-14500-18650-9V-3.7V-1.5V-2xAA-4xAA-2-18650-i.33091591.526016810
Self-Locking Cables	RM1.79	1	RM6.39	https://shopee.com.my/(10pcs)-2-Pin-3-Pin-Self-Locking-Electrical-Cable-Connector-Quick-Splice-Lock-Wire-Terminals-Set-2-3-Way-Fit-Led-Strip-i.298072081.6347403758
LM2596 Voltage Regulator	RM13.20	2	RM31.00	https://shopee.com.my/LM2596-DC-DC-Voltage-Regulator-Adjustable-Step-Down-Module-w-Display-i.33287405.2148334760
Rechargeable Batteries (X 4)	RM16.99	3	RM55.57	https://shopee.com.my/Beston-4pcs-AA-3000mAh-Ni-MH-Rechargeable-Batteries-i.139879612.2111664617
Battery Charger	RM18.50	1	RM23.10	https://shopee.com.my/Beston-AAA-1300mAh-Pack-of-8-pcs-Ni-MH-Rechargeable-Batteries-Beston-Battery-Charger-C9009-Suitable-For-AA-AAA-Battery-i.139879612.9130683661
mg90s servo	RM9.90	3	RM34.57	https://shopee.com.my/genuine-tower-pro-9g-micro-servo-sg90-and-mg90s-180degree-360°-i.46842355.9854308897?position=6
sg90s servo	RM14.00	3	RM42.00	Purchased from Monash
sg90s servo	RM7.90	2	RM15.80	https://shopee.com.my/MG995-MG996r-SG90-9g-MG90s-S3003-Metal-Plastic-Gear-180-360-

				Deg-Degree-Micro-Servo-Motor-Set-for-Arduino-TowerPro-i.33091591.465890496
Corrugated Plastic Sheets	RM6.50	4	RM28.00	https://shopee.com.my/Ready-stock-authentic-Corrugated-Plastic-Sheet-680mm-x-575mm-x-3mm-customize-size-i.112108068.2387239630?position=5
Caster Balls	RM3.00	2	RM10.60	https://shopee.com.my/W420-Steel-Ball-Universal-Wheel-(Castor)-i.6641351.1598235593?position=5
Hot Glue Gun and Hot Glue Set	RM12.80	1	RM17.40	https://shopee.com.my/【40W-100W-】Hot-Glue-Gun-Silicone-Adhesive-Melt-Glue-Gun-Glue-Stick-40W-100W-i.162363634.2456457864?ads_keyword=hot%20glue%20gun%20pink&adsid=721377&campaignid=721706&position=0
Bamboo Skewers	RM1.55	1	RM6.15	https://shopee.com.my/Batang-Lidi-Satay-Stick-Bamboo-Skewer-Kayu-Cucuk-Satay-Kayu-Bunga-Telur-DIY-Batang-Telur-Buah-Bouquet-15cm-25cm-30cm-i.5048558.5561774866?position=5
Female to male Jumper Wires	RM2.50	1	RM2.50	Purchased from Monash
TCS3200 Colour Sensor	RM18.00	1	RM18.00	Purchased from Monash
Sg90s servo	RM5.53	3	RM16.59	https://shopee.com.my/product/33091591/465890496?smtt=0.82604871-1635064174.9
TCS3200 Colour Sensor	RM16.90	1	RM7.12	https://shopee.com.my/product/33287405/464088223?smtt=0.493806379-1635064086.9
Female to male Jumper Wires	RM3.90	1	RM3.90	https://shopee.com.my/product/33287405/567926308?smtt=0.493806379-1635064122.9
Male to male Jumper Wires	RM3.90	1	RM3.90	https://shopee.com.my/product/33287405/567896940?smtt=0.493806379-1635064102.9
Total	RM 364.59			Budget left: RM380 – RM364.59 = RM15.41

7.5 Comparison of project budget

Initial Budget Proposed:RM254.78

Final Budget Summary:RM254.78 + RM109.81 = RM364.59

Table XVIII

ADDITIONAL PURCHASED COMPONENTS COMPARED TO BUDGET LIST IN PROJECT PROPOSAL

Extra Items List	Price	Unit	Total
sg90s servo	RM14.00	3	RM42.00
sg90s servo	RM7.90	2	RM15.80
Sg90s servo	RM5.53	3	RM16.59
TCS3200 Colour Sensor	RM18.00	1	RM18.00
TCS3200 Colour Sensor	RM16.90	1	RM7.12
Female to male Jumper Wires	RM3.90	1	RM3.90
Male to male Jumper Wires	RM3.90	1	RM3.90
Female to male Jumper Wires	RM2.50	1	RM2.50
			Total: RM109.81

As shown in table XVII, these are the additional items purchased that were outside of the proposed budget list. A total of RM109.81 of extra expenses have been made in order to create the competition robot. As shown above, we have made 3 different instances of buying sg90s servos. This is due to the initial mg90s servo purchased being faulty and does not function as what we intended it to do. The reason for a total of 8 servos being purchased is that 3 servos are bought by the programmer to test the servos herself while the other 3 are bought for the actual assembly of the robot. Pay mind that initially including the sg90s servo that was provided to the system integrator there was a total of 4 sg90s servos. However, due to a misconnection, the system integrator has caused 1 servo to short circuit, essentially burning the component. As a precaution for the similar events happening, we decided to buy an additional 2 servos with 1 serving as a replacement and another as a back-up. Similarly, we bought 2 TCS3200 colour sensor. As the final robot would have 2 colour sensors implemented, 1 colour sensor is bought by the programmer for while the other one is bought by the system integrator for robot assembly. We find a need for the programmer to also buy extra components themselves as a coding and programming a component through a shared screen will not be productive and would waste a lot of time. Hence, the price for more productivity and more programming freedom is the purchase of essentially a standalone component for the programmer to use themselves. Lastly, the extra purchases of jumper cables are made as initially the amount of jumper cables provided were deemed enough. This was however far away from the truth as more jumper wire cables were needed. The 3 instances of jumper wire cables are hence justified with 1 used during assembly and 2 for the programmer to use.

8.0 Recommendations and conclusions

8.1 Recommendations

After the completion of the competition, there are a few adjustments and improvements that we can recommend towards the effectiveness of the robot.

8.1.1 Alignment and puck detection mechanism

Firstly, we would like to mention that the robot currently heavily depends on its initial position on the starting base. As a lot of the travelling values set for the robot are fixed, the robot would only function optimally when it starts off where it is meant to be. This brings about a lot of inconsistency to our robot. To help with this problem, alignment and puck detection mechanism are recommended. The alignment mechanism would allow for the robot to be always orientated the right way while the puck detection mechanism ensures that the robot is always hitting or dragging the puck at the same position, all in all increasing consistency.

8.1.2 Adhesive

Another thing worth mentioning is the assembly of the robot. As this competition is a one-off event, the adhesives used were permanent in a way that they are hard to remove. We have used hot glue to connect major pieces of the robot together. This method is effective as hot glue is a fast-acting and sturdy adhesive. The downside of doing so is that in the occasion that there is a need to remove the components, either for replacement or repurposing, the removal process tends to be messy and arduous. Hence, a better solution for the long term would be to make a custom component holder that can be attached to the machine chassis through screws and bolts. This way assembly and disassembly can be done easily without causing any mess.

8.1.3 Momentum

Following it, we have also run into the issue of the robot stopping inconsistently. By inconsistently, we mean that the robot would slightly rotate when stopping. In addition, as the robot repeats moving forwards (or backwards) and retracting during continuous capturing, the robot could not retract back to its initial position accurately. The errors accumulate and thus displace the robot from its intended position. Based on testing and some discussions among peers after the competition, we have deemed the main source of the problem to be the speed of the robot. The momentum of the robot from moving fast would slightly move the robot forward when stopping. As the solution to this problem is however quite easy, one would just need to decrease the robot’s movement speed to decrease the likeliness of such issues appearing.

8.1.4 Wire connection

Due to the design of our robot and the battery holders we have purchased, the way of extracting batteries from the robot is quite tedious and sometimes may cause jumper wires to come loose. Such an instance has appeared during the competition day. To prevent this problem from happening, we could 1) Get single face battery holders and position them in a wire-free environment or 2) secure the jumper wires to the breadboard using adhesive instead of relying on the friction between jumper wire tips and the breadboard insertion points.

8.1.5 Delay for puck replacement

Lastly, a crucial aspect of the robot for point gathering is the delay we set for puck capturing. In the competition, we went for a safe route and allocated a lot of delay time for the robot. An example would be that we allocated an 8-second delay between puck capturing. The reason for this is that we were afraid that our puck machine might start capturing before we receive the confirmation from the competition inspector that we can replace the puck. With knowledge of how the puck capturing inspector acknowledges puck captures, and that our puck capturing is consistent, we would suggest lowering the delay to as low as 3 seconds. We would also recommend manipulating pucks alternatively, if possible (if there are two valid pucks on one side), since this would also give the system integrator more breathing room for puck replacement.

8.2 Conclusion

To conclude the whole event, we are proud to say that we have been able to adapt well to all the sudden changes this competition has thrown at us. We have changed our robot design at least 5 times, with all the key concepts being unique to each other. On the competition day, some mishaps happened as a wire came loose while the system integrator took out the robot's batteries. This would then lead to a bad score for the first half of the second round. Thankfully, we were able to inspect and realise the issue as well as make quick fixes to ensure that the second half of the second round goes smoothly. Our robot did what it was supposed to do and scored us a total of 50 points at the end of the competition. Although the points did not seem to be too outstanding, both of us did agree beforehand to go the safe route and to go for a "Slow and Steady" kind of strategy. With this, we were able to guarantee ourselves to be at least within the top 20 and that no touch penalties will be given. That said, we were also confident that a slightly riskier design of the robot that we had thought of would have also worked for the competition and that it would most likely allow us to go for the top 10 leader boards. All in all, we are proud of all that we have achieved during this semester and that to further improve our robot, the recommendations from above would definitely work.

9.0 Reference

[1] codefls. “H-Bridge with L298N motor driver.” Fritzing Forum. https://forum.fritzing.org/t/h-bridge-with-l298n-motor-driver/7711 (Accessed Oct. 24, 2021).

[2] “LM2596 3A Buck Module with Display.” Cytron. https://my.cytron.io/p-lm2596-3a-buck-module-with-display?r=1&gclid=CjwKCAjwx8iIBhBwEiwA2quaq3v3F6hXLGpGe9ybfT0OdBRwqcSm55jbO2NBUbKHs_kZ3i3WsphPFBoCaL8QAvD_BwE (Accessed Oct. 13, 2021).

[3] “Johnson Standard 130 Motor Green Miniature DC Motor with Encoder Mini motro 25g Shaft Dia 2mm.” Amazon. https://www.amazon.in/Johnson-Standard-130-Miniature-encoder/dp/B07372CWTZ (Accessed Oct. 13, 2021).

[4] “GY-31 TCS3200 COLOR SENSOR MODULE.” JOGJAROBOTIKA. http://www.jogjarobotika.com/sensor-warna/1317-gy-31-tcs3200-color-sensor-module.html (Accessed Oct. 13, 2021).

[5] “SG90 Servo Motor Datasheet,” SG90 datasheet, 2021.

[6] Mouser Electronics, “CY8CKIT-059 PSoC 5LP Prototyping Kit Quick Start guide,” CY8CKIT-059 datasheet.

[7] “Optical Encoder Project.” MIT.edu. https://groups.csail.mit.edu/mac/users/pmitros/encoder/ (Accessed Oct. 13, 2021).

[8] ams AG, “TCS3200, TCS3210 Programmable Color Light-to-Frequency Converter,” TCS3200 datasheet, 2016.

[9] “TOWER PRO SG90 9G – Miniature Servo Motor.” ELECTRONICA EMBAJADORES. https://www.electronicaembajadores.com/en/Productos/Detalle/MMSV002/motors-servo-motors-linear-actuators/servo-motors/tower-pro-sg90-9g-miniature-servo-motor (Accessed Oct. 13, 2021).

[10] Autobotic. “Solenoid – 5v (small).” Shopee, https://shopee.com.my/Solenoid-5v-(small)-i.6641351.1107125929?sp_atk=479e2754-a938-436c-8fa7-9e514a679926 (Accessed Oct. 24, 2021).

[11] Almencla. “Almencla ZYE1-0630Z Linear Mini Push Pull Solenoid Electromagnet Reset.” Lazada. https://www.lazada.com.my/products/almencla-zye1-0630z-linear-mini-push-pull-solenoid-electromagnet-reset-i2478788167-s10840592834.html (Accessed Oct. 24, 2021).