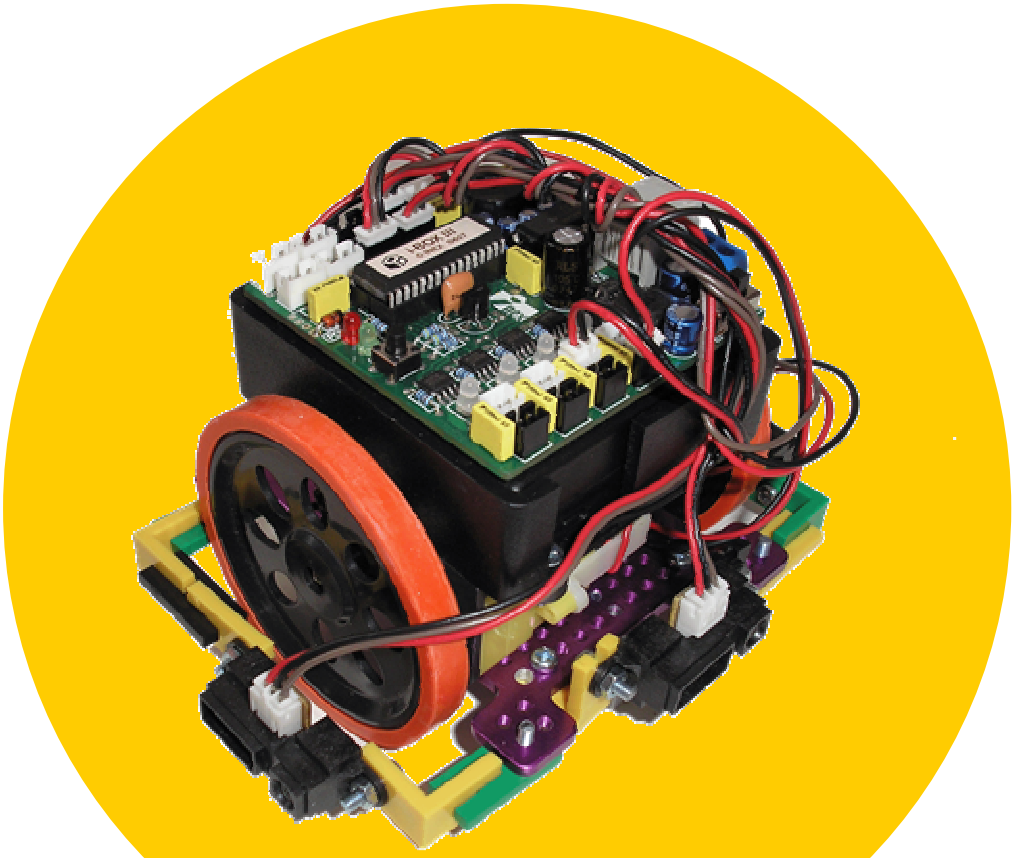


MAZE ROBOT

MAZE ROBOT COMPETITION

This is a competition that is about a robot that is able to navigate out of the maze in the quickest time possible from the designated starting point to the ending point.

The Maze size can be dependent on the difficulty level, types of robotic platforms used or based on the organizers discretion.

SAMPLE RULES

TASK
The task is to build and program an INEX robot that navigates through a maze as fast as possible. The robot will start at the entrance (already inside the maze) and has to find its way to the exit autonomously once started there can be no more input to the robot). The figure below shows an example scenario.

The Robot
The robots participating in the competition have to be built exclusively with the use of INEX microcontrollers and sensors. Additional 3rd party parts are allowed in the competition. Some suggested sensors that should be used are as follows : distance sensors, light sensors, touch sensors.

Each team is allowed only one robot for the competition which has to have a maximum Width, Breath and Height of 20cmx20cmx20cm. Any parts that is larger than this measurements have to be removed during the competition or the team will be disqualified. Flags are allowed outside this size limitations for team identification.

All parts that drop off the robot during the competition will not be allowed to be placed back onto the robot during the competition.

Robot Programming
All code used in the robots has to be written by the team members. Only. Codes can be changed at anytime during the competition but not during the running of the robot. Teams are reminded that competition lighting conditions are

The Maze

The maze will be constructed on a rectangular *30.5 cm* × *30.5 cm* (*1'* × *1'*) grid. The size of the maze will be *2.14 m* × *2.14 m* (*7* × *7* cells). All passages and crossings will be aligned with the cells of the grid.

Therefore, all passages will be straight and crossings will be at *90°* angles. The width of each passage can vary between *30 cm* and *30.5 cm* and the height of the walls will be between *11.5 cm* and *12.75 cm*. The entrance to the maze will always be in the left upper corner of the maze and each team can place their robot at an arbitrary position within the first grid square at the entrance (the contour of the robot can not fall within any other square of the maze).

The exit to the maze can be at any point on the boundary of the *7* × *7* grid (except that entrance and exit can not coincide).

Maze Materials

The competition maze will be constructed from plywood and will not be painted. The floor will be smooth, slightly textured, and without grid markings. The floor color will be different in each rounds. These specifications are of course flexible and up to the organizer's discretion.

Preparation

In order to give all teams better preparation, a practice day will be held to allow the teams and robots to adjust for the maze materials and to simulate competition conditions for testing purposes.

This practice day will set by the organizers which are around 2 weeks before the actual competition. Each team will be scheduled for approximately an hour and a half of time to work with one stationary maze set-up.

Teams will not be visible to one another. All teams will practice with identical maze set-ups and no changes will be allowed to the maze during the practice time.

This practice maze IS NOT the actual competition maze. Teams are to practice their programming logic as well as to take note of lighting conditions during this practice round.

The Competition
The competition will contain two elements: a design and knowledge part and the actual running of the mazes.

Design and Team Knowledge
Every team has to provide a printout of the code that is running on the robot and will be asked by the judges a series of questions about their robot design and the code they are running. In this, the team is expected to explain all aspects of their design and each team ember is expected to be able to describe at least the parts that she/he has been involved in.

The important element here is not so much the best and most ingenious design (there will be a separate engineering design competition for this) but rather the degree of knowledge of the team with respect to their design and implementation choices.

Since it is expected that every team will be able to explain their code well, the scoring for design and team knowledge will only indicate a team that did not have good knowledge of its code.

Poor results in this assessment would mean that the Team might note have written the code or design themselves.

Maze Running
The competition will be conducted as an elimination tournament. In every round teams will be paired up and compete against each other on identical mazes. In each of these contests, the teams will have two attempts on the maze and the winner will be the team that reaches the exit of the maze in the shortest amount of time or, in case none of the robots can find the exit within the time limit (approximately *10* minutes - actual time limits will be announced prior to the competition), the winner will be the robot that came closest to the exit of the maze.

If this leads to a tie, the result of the second run will be used to determine the winner.

All mazes used within one round will be identical (except for fabrication tolerances). However, different mazes will be used for the different rounds of the tournament.

To adjust for fabrication tolerances, competing teams will swap mazes between their two runs. In the preliminary round, teams are going to be given approximately *10* minutes between their runs to fix potential errors in their code or to recalibrate their sensors. However, no different program can be downloaded at that time and no information about the maze can be added (only information that the robot acquires autonomously can transfer from one run to the next).

The Tournament

In the preliminary round, the eight teams with the fastest time (including time penalties) will qualify for the second round of the competition. At most two teams from any given high school can qualify for the second round.

The two fastest teams from the second round (including time penalties) will qualify for the final while the third and fourth fastest teams will compete for third place.

The precise format of the competition is subject to change depending on the number of teams participating in the competition.

Judging and Rules

For the design and knowledge judging, a group of judges will interview teams individually prior and during the first round of the tournament. The resulting time penalties will move with the team through the different rounds of the tournament.

During the maze running part, a judge will be assigned to each of the mazes to keep time. The judge will be the only person permitted to touch the robot during a run. Each contact of a team member with the robot will result in an automatic re-start of the maze task without a reset of the clock (i.e. if a team member touches the robot, it has to be moved back to the entrance of the maze).

The exception to this rule is in case of a mechanical breakdown (i.e. if a part breaks off the robot during the run). In these situations, teams have to inform the judge that they want to perform repairs.

The judge will then permit the team to pick up the robot, make repairs, and re-place the robot in the identical position and orientation in which it was prior to the mechanical breakdown. The clock will continue to run during the repairs. In situations where a robot gets stuck (i.e. it does not make any progress for a duration of at least *5 s* - no progress includes cyclic behavior that the robot can not recover from) teams can ask the judge to ``nudge'' the robot to move it out of the position.

Nudging here will be a displacement and reorientation of maximally *2 cm* and *30°*. If teams have special requirements for this (e.g. doing so without picking the robot off the ground) they have to inform the judge of this prior to the run. Each ``nudging'' of the robot will incur an automatic *10 s* time penalty.

Scoring
The scores of each team will be according to the following :
Given the maximum of 10 minutes, each team will start with 10 x 60s = 600 points

Time to exit (TE)
Time taken for the robot to travel from start to the exit successfully. The seconds will be the number of points for this.

Stuck (STK)
In the event the robot is stuck in an endless loop in the maze.
Every of such event, a score of 10 points will be deducted.

Breakdown (BKD)
In the event the robot stops moving / parts fall off / immobilize / or any other situation whereby the robot cannot function anymore.
Every of such event, a score of 15 will be deducted.

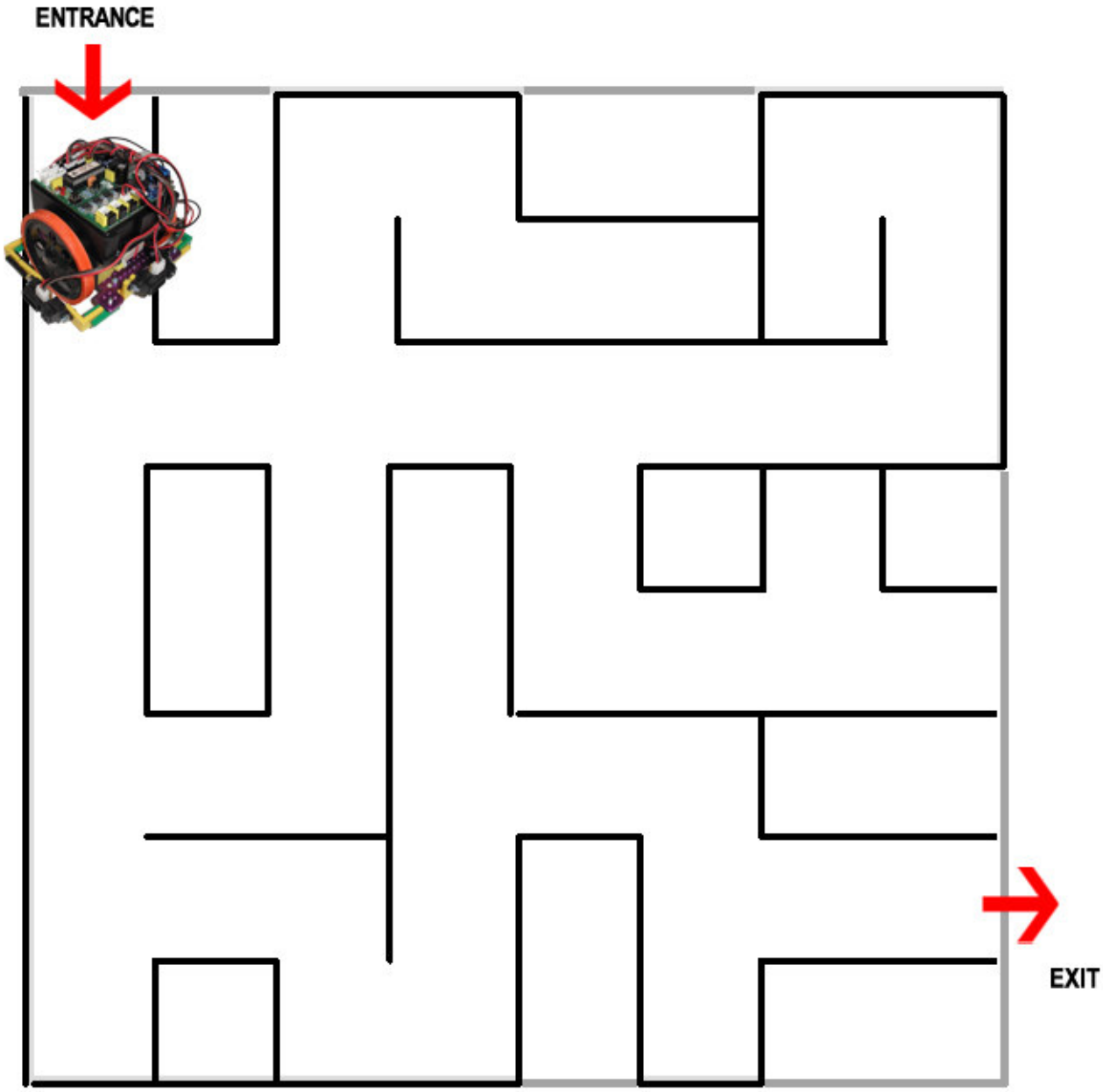
Total Team Score = ***600 – (TE + STK + BKD)***

Awards
There will be a Championship award followed by a 1st and 2nd runner's up awards. These which gets the most score out of this will be deemed the winners.

There are also side awards for the following categories :

- BEST CHEER GROUP
- BEST ROBOT DESIGN
- BEST INNOVATION
- BEST FIRST TIMER
- BEST BOOTH DESIGN
- BEST PROGRAMMING
- BEST TEAM WORK

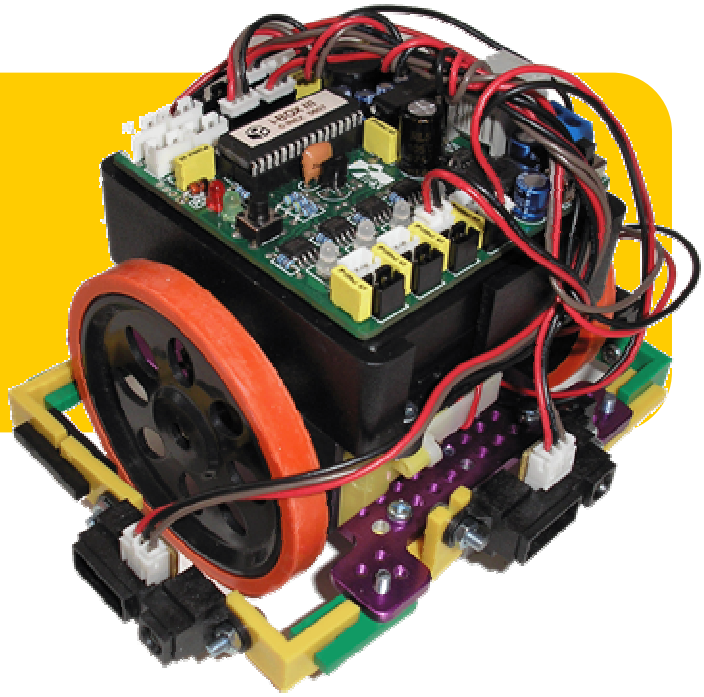
MAZE Layout

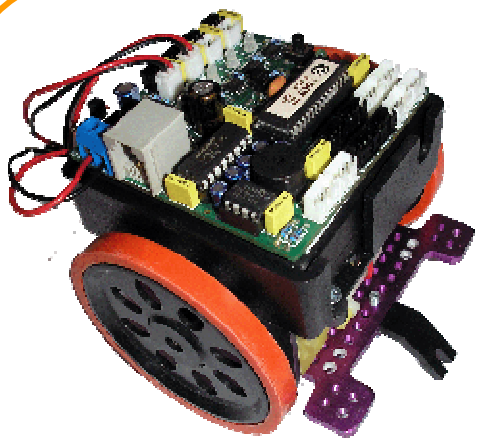


BUILDING a MAZEBOT

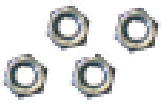
MAZEBOT

This is how your MAZEBOT will look like. The MAZEBOT main Sensors include the GP2D120 distance Sensors.





Nuts

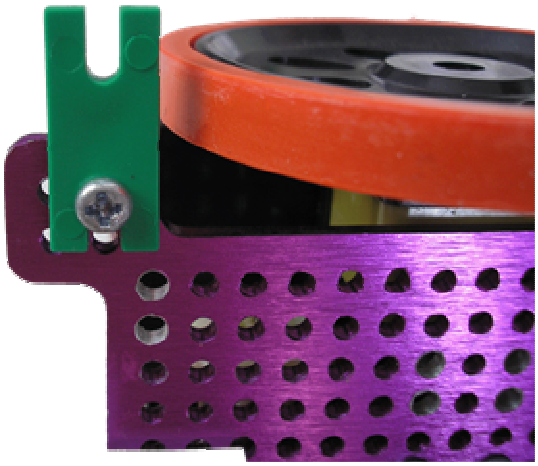


3x10 Screws

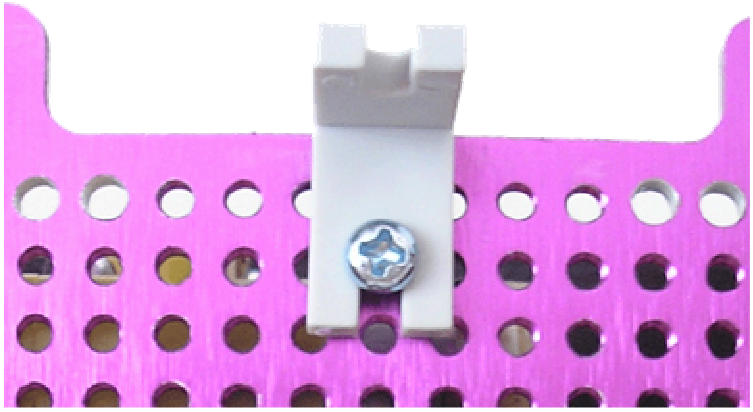
GP2D120 Sensors

Plastic Spacers

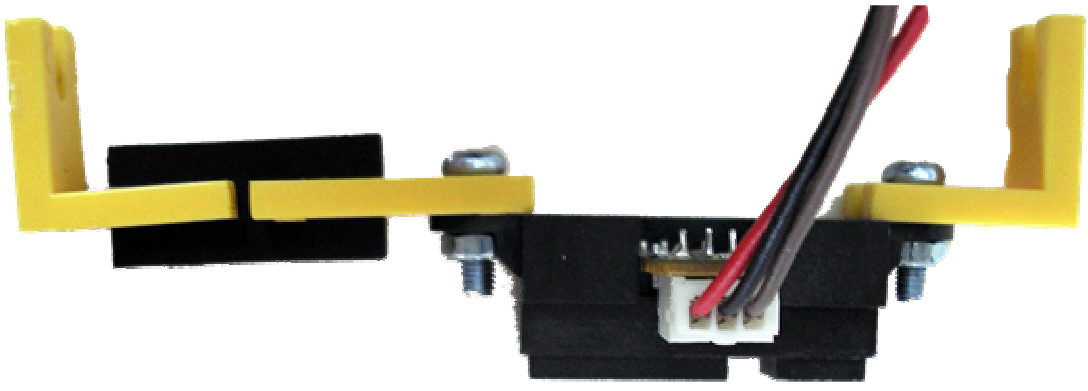
1. The first step is to add 2 straight joiners to the side of your basic robot.



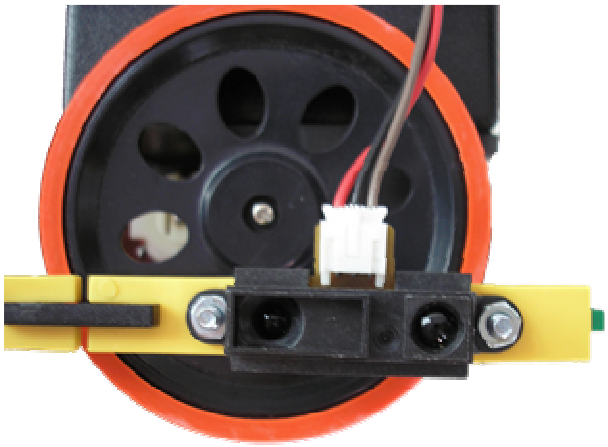
2. Add to angle joiners to the front and back of the robot to give it stability in movement.



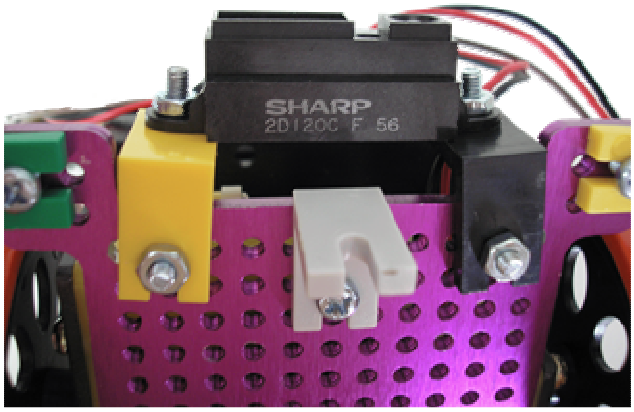
3. Next, place a few joiners together and screw them with a GP2D120 sensor. It should look something like this. Assemble 2 of these.



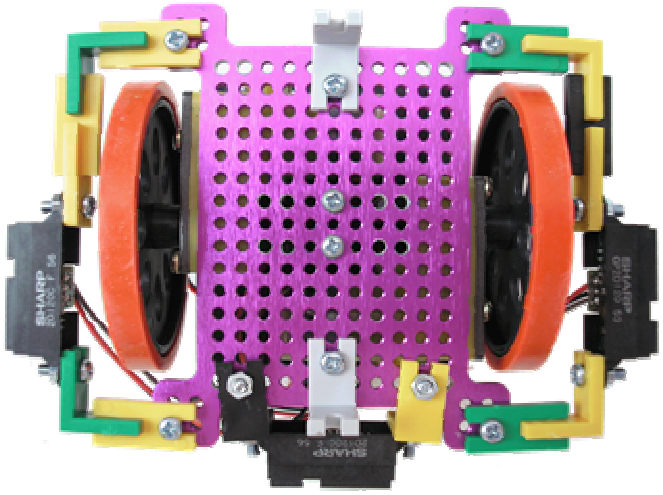
4. Place these 2 sensors at the right and left of your robot.

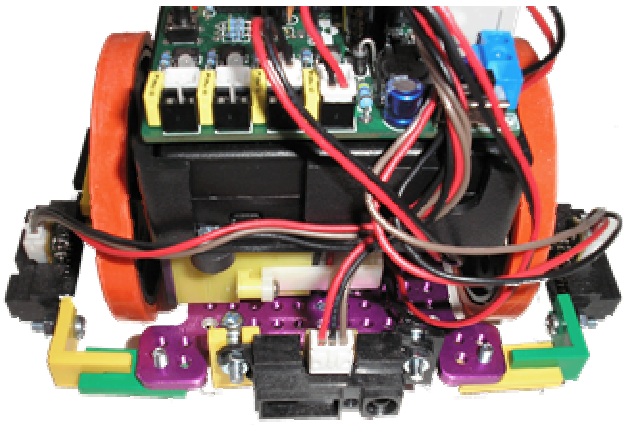


5. Attach the Front GP2D120 sensor with screws and nuts.



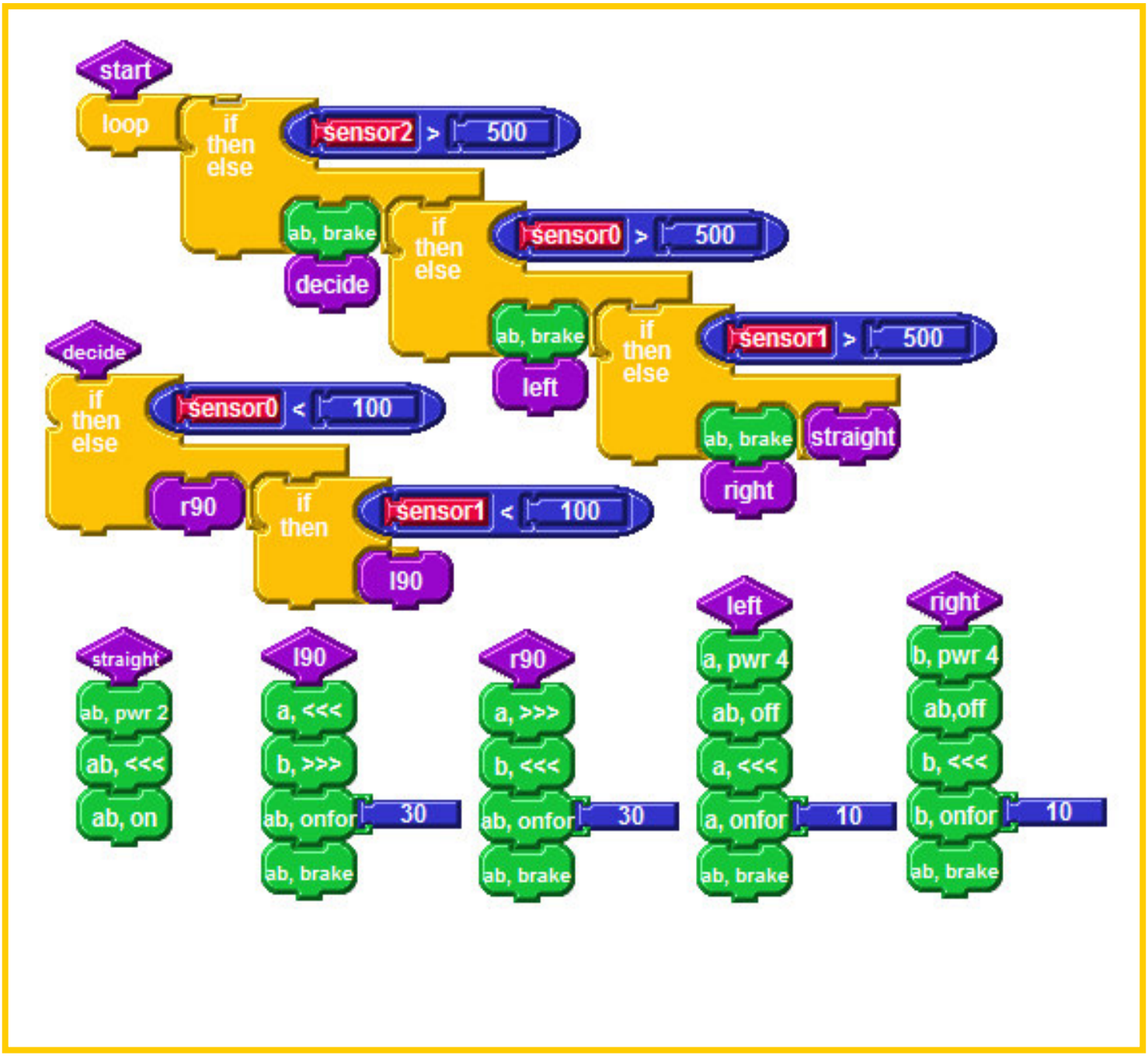
6. Your Overall Robot should look like this.





PROGRAMMING simple Maze Logic

In Programming of Maze Movement, there are many methods that you can use. We will be trying out a simple maze logic program that will allow you maze robot to move propoerly in the maze field. For decision making and higher level of robot analysis, you will need to further explore and develop on this program.



In this simple program, the MAZEBOT first has pre-defined functions, "straight, l90, r90, left, right,decide" .

The MAZEBOT then starts with an continuous loop that does first checking of its Front. If the Front is near to a wall, it will then go into its "decide" sub-function. This function does 2 checks that if the right is empty, turn right, and if the left is empty, turn left.

If the MAZEBOT is not hitting a wall, it will constantly check if its moving near to a wall or not. If it is, it will turn slightly left or right respectively to move back into position.

Cricket Logo Code

```
to start
loop [ ifelse ((sensor 2) > 500)
[ ab, brake
decide ]
[ ifelse ((sensor 0) > 500)
[ ab, brake
left ]
[ ifelse ((sensor 1) > 500)
[ ab, brake
right ]
[ straight ] ] ] ]
end

to decide
ifelse ((sensor 0) < 100)
[ r90 ]
[ if ((sensor 1) < 100)
[ l90 ] ]
end

to straight
ab, setpower 2
ab, thisway
ab, on
end

to l90
a, thisway
b, thatway
ab, onfor 30
ab, brake
end

to r90
a, thatway
b, thisway
ab, onfor 30
ab, brake
end

to right
b, setpower 4
ab, off
b, thisway
b, onfor 10
ab, brake
end

to left
a, setpower 4
ab, off
a, thisway
a, onfor 10
ab, brake
end
```