e-Yantra Robotics Competition - 2017
Theme and Implementation Analysis – Planter Bot

931

Team leader name	Abhaysheel Anand
College	Bharati Vidyapeeth College of Engineering, New Delhi
Email	abhaysheelanand@gmail.com
Date	9th January 2018

Scope and Preparing the Arena

Q1. a. State the scope of the theme assigned to you. (5)

The theme *Planter Bot (PB)* is focussed on the implementation of practical problems related to planting of seeds. The wider scope of this theme, if analysed, could reach the higher limits of solving the huge grade problems in our primary sector i.e. Farming. If implemented efficiently on a larger scale it could reduce the workload of the farmers as the need for labour would be minimised. This might have a negative impact as reduced labour means reduced employment but expanding the horizon of our thoughts, the introduction of these autonomous robots would eventually increase the pace of development of our country.

b. Upload the Final Arena Images.

(20)

< Prepare the arena according to the steps given in Section 3: Arena, of the Rulebook. Please follow the arena configuration shown in "Figure 1: Arena Design" and "Figure 4: Arena Design with Dimensions" of the rulebook.
Configuration for Zone Indicators and Color Markers associated with them are as per following Table:

Zone Indicator Number	Cell number for Zone Indicator	Color Marker Type	Number of Color Markers
1	N3	Red Circle	3
2	F7	Green Triangle	4
3	O11	Red Square	1
4	E16	Blue Square	2

In addition to this, place a Zone Indicator at Cell number J16. This has no Color Markers associated. Refer to Section 2: Theme Description and Section 6: Theme Rules of Rulebook for more information about this.

Take 4 photos of the completed arena from different angles such that the entire arena along with its components such as Terrains, Zone Indicators, Cells, etc., is clearly visible in the photos.

Answer Format: The four image files should be uploaded as **.jpg** along with this document as per instructions in Read Me for Task 3. >

Building Modules

Q2. Identify the major components required for designing the robotic system for the theme assigned to you.

(5)

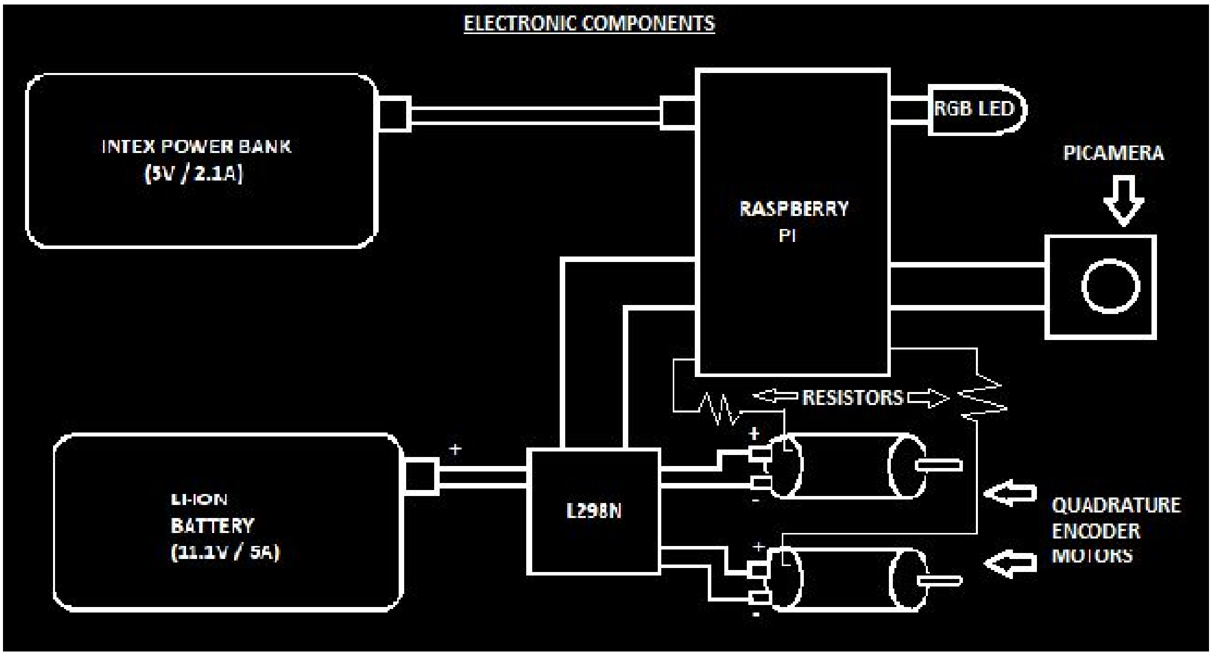


Diagram illustrating Electronic Components used in designing

The major components required for designing the robotic system can be listed under the following systems:

A. **Electronic system** :- It consists of the following components:
   1. Raspberry Pi
   2. PiCamera
   3. Motor Driver(L298N)
   4. Lithium Ion Battery
   5. Power Bank (INTEX)

6. Quadrature Encoder DC Motors
7. RGB LEDs
8. Resistors

B. **Mechanical System** :- It consists of various mechanical components such as
   1. Wheels (including Castor Wheel)
   2. PiCamera Stand
   3. Chassis
   4. Studs
   5. Screws
   6. Clamps

Power Management

Q3. a. Explain the power management system required for a robot in general and for the theme assigned to you in particular.

(5)

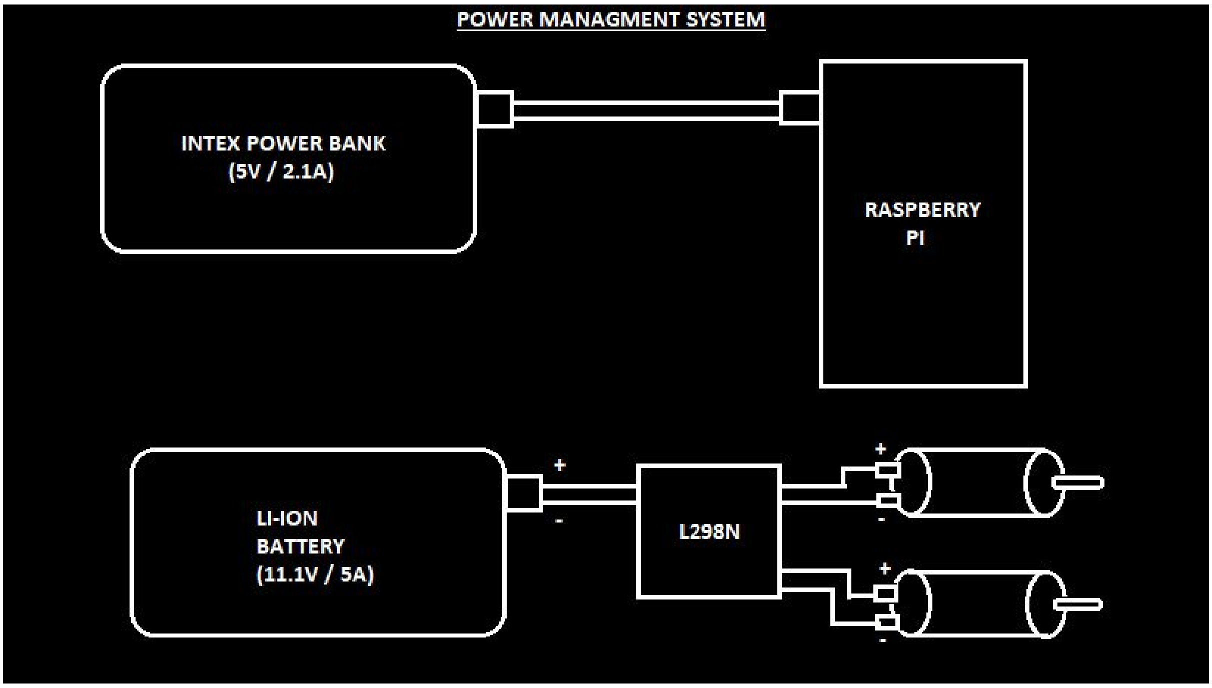


Diagram illustrating Power Management system of the robot

In our *Planter Bot (PB)*, we are using two different power sources. Those are :

- First a *Lithium ion* battery,
  - **Voltage rating = 11.1 V**
  - **Current rating = 5 A (approx.)**
- Second one is an *INTEX power bank* with two options for different current output,
  - **Voltage rating = 5 V**
  - **Current rating (I) = 2.1 A**
  - **Current rating (II) = 1 A**

We will use the *li-ion battery* for powering the dc motors and the *power bank* with **5V, 2.1 A** configuration for powering *Raspberry Pi*.

The motors provided are *geared DC motors with encoders* so they will each use a minimum of **1 A**. and the *Raspberry pi* needs approximately **2 A** for its proper functioning. It is prefered to use battery instead of auxiliary power supply so as to rectify the mobility issues.
.

b. Can there be a single power supply for your robot? - Yes/No/Don't know. Please elaborate/justify your answer choice.

(5)

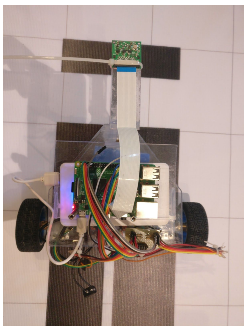
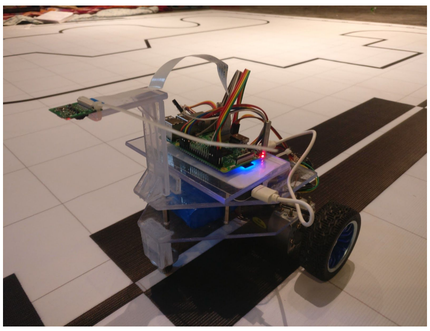
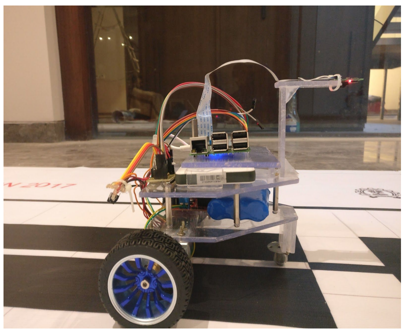
No.
The reason behind this is that if we try to power both the *motors* and the *Raspberry Pi* with a single *Lithium Ion battery* (single power supply), the motors will draw a large amount of current from the battery and as both of them would be connected to the *L298N* there might exist an instant at which the *Pi* might not be able to draw the required current (minimum 1.5A for basic operations) for its continuous operation. Due to this the *Pi* would restart and that is obviously not desirable.
So it would preferable to use two power source for the robot in this case.

Design Analysis

Q4. Teams have to design a robot which traverses a arena following a given path and simulate planting by overlaying image in GUI.

a. How will your robot traverse a field represented by the Arena given in the rulebook?

(5)

Design of the Planter Bot (PB)

Components used for designing the bot (for traversal) :

1) ***Raspberry Pi 3***
2) ***PiCamera v1.3***
3) ***Quadrature Encoder Motors***
4) ***L298N Motor Driver***
5) ***Power Sources (11.1V/5A Li-Ion Battery, 5V/2.1A INTEX Power Bank)***
6) ***Jumper Wires***
7) ***Resistors***

- The *Planter Bot(PB)* is equipped with a *PiCamera* which is controlled using *Raspberry Pi, a L298N Motor Driver* and a set of *DC motors* with position encoders. The *PiCamera* is adjusted on a stand which is custom made along with the chassis, the angle is set such that the frame captured by the camera covers the nearest required region of the arena for traversal. It then sends the data to the *Raspberry Pi* for further processing which then direct the motors accordingly.

- The direction and the intensity of the turn is calculated by the *PiCamera* through the image processing algorithm. The *PiCamera* captures the frames at the specified frame rate. Each frame is then divided into 4 parts. The cropped parts are stored in an array. The array is traversed and the contour with maximum area is found in each contour. The coordinates of the centroid of all the four parts are determined and their shift from the centre of the X-axis is calculated.
- Thus, the shift for the contour whose centroid lies at the left of the line is negative and similarly positive for the right one. The desired turn is obtained by the quadrature encoder motors depending upon the shift.
- The channel A and channel B of the *quadrature encoder motor* is given to the interrupt pins of the *Raspberry Pi*. Thus whenever a change in edge is encountered at channel A or channel B, an interrupt is called which updates the encoder count variable according to the lookup table and thus the rotation of the shaft of the motors is controlled.
- Thus by these operations the *Planter Bot(PB)* traverses the whole arena.

b. If you were to implement this theme in the real-world scenario, what would be the actuators you will employee? Explain their purpose.

(5)

The only actuators used at present in the *Planter Bot (PB)* are only responsible for the movement of the robot that are the **DC quadrature encoder motors** itself. If this theme is to be implemented in the real-world scenario these actuators will not be enough to work in those situations.

For the processes to work properly we will need some more actuators, most importantly **standard servo motors**.

Such type of actuators will be used in the implementation of this theme. According to the theme, we will need two **servo motors** to accomplish this task. If ploughing is taken into consideration then one **servo motor** will be used to plough a small amount of area to put the seedlings. Then comes the task where we have to put the the seedling into the ploughed field area. For this task we will need one extra servo motor that will be integrated in an mechanism which will be used to plant the seedling.

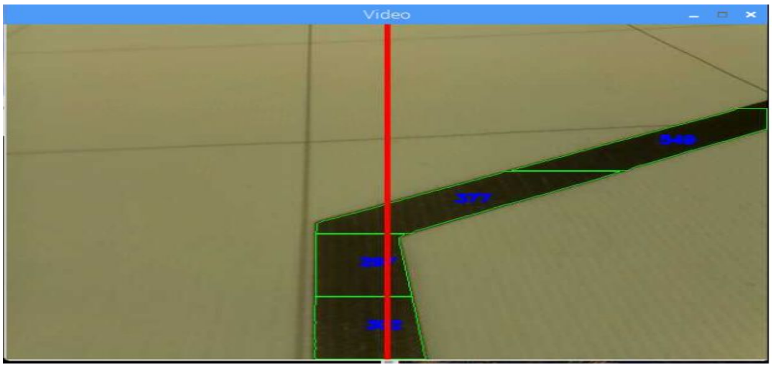
Advantages of servo motors :

- These are a type of dc motors which have the capabilities to turn to a specific degree of rotation or to turn degree by degree.
- High intermittent torque.
- High torque to inertia ratio.
- These motors are available in all sizes.

Environment Sensing

Q5. a. Explain how you will use the PiCam to decide the course of traversal.

(5)



Example of a frame depicting use of PiCamera for traversal

The robot will traverse the field represented by the arena using image processing. *Raspberry Pi* and *PiCamera* are used to serve the purpose. The frame captured by the *PiCam* is divided into four equal parts horizontally. The four parts are stored in the array. Then each element of the array is processed by the image processing algorithm.

- First of all, contour with maximum area is identified in the element and its centroid is detected.
- Then the shift between the midpoint of element and the centroid of the contour with maximum area helps us to identify the direction and the intensity of the turn. For this a vertical line in the middle of the frame is drawn. The distances between the centroid of all the 4 contours of the image-parts and the middle line is calculated. If the centroid is in the left, then its distance is taken to be negative and if the centroid lies to the right of the middle line, then its distance is taken to be positive. The sum of all the four distances is calculated. The sign of the sum gives us the direction and the magnitude of the sum gives the intensity of the turn.

b. Would the webcam be a better choice of camera over the PiCam? Explain.

(5)

The *PiCam* would be the better option. The following reasons justify this statement:

- It is easy to interface *PiCam* with *Raspberry Pi*.
- There is no need to install any drivers for *PiCam* whereas there is a need to install drivers for the webcam. It is highly unlikely that one can start experimenting with a *Webcam* before doing a whole lot of work on its drivers.
- *PiCam* is compact as compared to a *Webcam*.
- *PiCam* has its own library in *OpenCV*.
- *PiCam* can click low light images and the feed could be enhanced using *OpenCV* for better results.
- *PiCam* has better resolution.

- It also has faster frame rate in comparison.

c. What other sensors will the robot require to complete its task successfully?

(5)

Here are some sensors which could be useful.

- Colour Sensor :-

  USE :- The colour sensor can be used in detecting the colour of the different types of colour markers. It will detect the actual more accurately because it uses its own light to detect the colour and independent of the surrounding light conditions.

  WORKING :- Colour sensors consist of RGB and clear light sensing elements. This sensor works by shining a white light at an object and then recording the reflected colour. It contains red, green, blue filters. These filters convert the light energy to current and then further the current signal is converted into the voltage signal, which our Raspberry pi can read.

- Infrared Sensors :-

  USE :- Bunch of infrared sensors will be used to detect the black path line present in the arena. Path can be followed more precisely by using merely 3 IR sensors.

  WORKING :- Infrared sensor works on the principle of detecting the amount of reflected infrared light using photodiodes. They consist of an IR LED and a Photodiode, the IR LED transmits the light on the surface and the reflected light is catched by photodiode. The output of the sensor is passed through an op-amp circuit from where we can get our binary signal having only 0 or 1. When the sensor will be on black line, it will output 0 and otherwise 1. So by combination of the output of these we can follow the black line path.

- 3 Channel Line Sensor :-

  USE :- This sensor can be used instead of Infrared sensors. They can also be used to follow the path present on the arena even more precisely than IR sensors.

  WORKING :- Line sensor consists of high intensity red LED for illumination and directional phototransistor for line sensing. Phototransistor consists of a photo transistor and convex lense. This line sensor is highly immune to ambient light. The output of the sensor is analog in nature.

Testing your knowledge (Theme Analysis and Rulebook-related)

Q6. a. If a team has an overlay similar to one shown in the Figure 1, how many points will you score for the overlay in total. Specify score for accuracy, penalty if any and total. Elaborate on penalty if any - why it will be applicable?

(5)

Note: The team has selected the correct seedling image upon detection of Color Marker and there are three such Color Markers at the Zone Indicator.

Figure 1: Overlay Example

The total formula for scoring is stated as :

*Total Score = (600 - T) + (ZD*100) + (CMD*75) + (TT*100) + (IPP*200) + (O*25) + (B) - (P*50)*

(In this **O** defines the accuracy of the overlays and **P** defines the number of penalties.)

Since in this figure, the overlay is partially overlapping, therefore there will be 2 penalties for the partial overlay and the team will be awarded points only for the middle Seedling image which is in full view and the 2 partially hidden Seedling image overlays will be penalized.

Therefore considering the above stated formula only for the scenario of the overlay, the following data would be produced,

Points Awarded (+) : O = 1*25 = 25 points
Penalties (-) : P = 2*50 = 100 points
Total : 25-100 = -75

Therefore **75 points** would be deducted for this overlay.

b. Name the different Terrains in the Arena.

(3)

1: Hill Side Road (HR)
2. Berms (BE)
3. Cliff Roads (CR)
4. Inverted Plains (IP)

c. How many possible unique Color Markers can be made in this theme?

(3)

9

d. If there are 3 Blue Triangle Color Markers placed in front of a Zone Indicator, how will you indicate this via hardware only?

(3)

The presence of 3 blue triangle color markers in front of a zone indicator can be indicated by using an *RGB LED*. The presence of blue can be indicated by lighting up the blue color of the *RGB LED* and the number of blinks i.e. 3 can specify the number of color markers.

For indicating the shape of the color marker, we can use another *RGB LED* (provided in the kit) and it would blink the number of times equal to the number of sides of the shape (with the same color of the Color Marker) , i.e. 4 times for depicting a Square, 3 times for a Triangle and 1 time for a Circle.

e. What are the different conditions that indicate end of a run?

(3)

Condition 1 *:* The PB completes the task and blinks RGB-LED at the Shed. PB must stop and blink the RGB-LED in the same sequence of Colors of CMs as they appeared on the Arena at the PZ from first to last. The RGB-LED must blinks with one second interval.

Condition 2 *:* The maximum time limit (600 seconds) for completing the task is reached.

Condition 3 *:* The team needs repositioning but has used both repositioning options of that Run

Algorithm Analysis

Q7. Draw a flowchart illustrating the algorithm you propose to use for theme implementation.

(10)

The **algorithm** is as follows (text format, succeeded by the **flow chart**) :

- ★ START
- ❖ Initialize all the variables and define all the functions.
- ❖ While True.
    - Capture frame
    - Crop the frame to desired sizes
    - Find contour with maximum area in the cropped frame
    - If contour area is greater than shed area,
        - ❑ Call **shed blink** function and stop traversal.
    - Else,
        - If contour area is greater than area of zone indicator threshold area.
            - ❑ Stop the motors until further processing is done.
            - ❑ Find all contours in the frame.
            - ❑ Eliminate the black contour.
            - ❑ Identify the shape, color and count of remaining contours.

- Find the corresponding image to be overlayed from the .csv file by calling **file_handling** function.
- Call **overlay** function and overlay the desired number of images on the desired planting zone
- Blink the led to indicate the identification of color and number of color markers by calling **LED_Blink** and **TURNOFF_LED** function.
- Store the color in a list.

- Else,
  - If contour area in lowest cropped part is greater than the threshold set for 90° turn,
    - If centroid is on left side of the middle line,
      - Turn 90° Left.
    - Else,
      - Turn 90° Right
  - Else,
    - Traverse the image array containing all the cropped parts.
    - If fifth zone indicator is encountered and area is less than a threshold already set.
      - Find contour with maximum area in all the four cropped parts of the frame after inverting the frame.
      - Find centroid of all the contours.
      - Append the distances of all the centroid from the vertical middle line in a list.
    - Else,
      - Find contour with maximum area in all the four cropped parts of the frame.
      - Find centroid of all the contours.
      - Append the distances of all the centroid from the vertical middle line in a list.
    - Find the sum of all the distances in the list according to their weightage.
    - The sum calculated indicates the direction of the turn and its magnitude indicates the intensity of the turn.

★ END

Functions Defined :

❖ **shed blink**

- Traverse the list in which the colors encountered are stored.

- Blink the RGB led with the color of each element of the list for 1 second using **LED_Blink** and **TURNOFF_LED.**

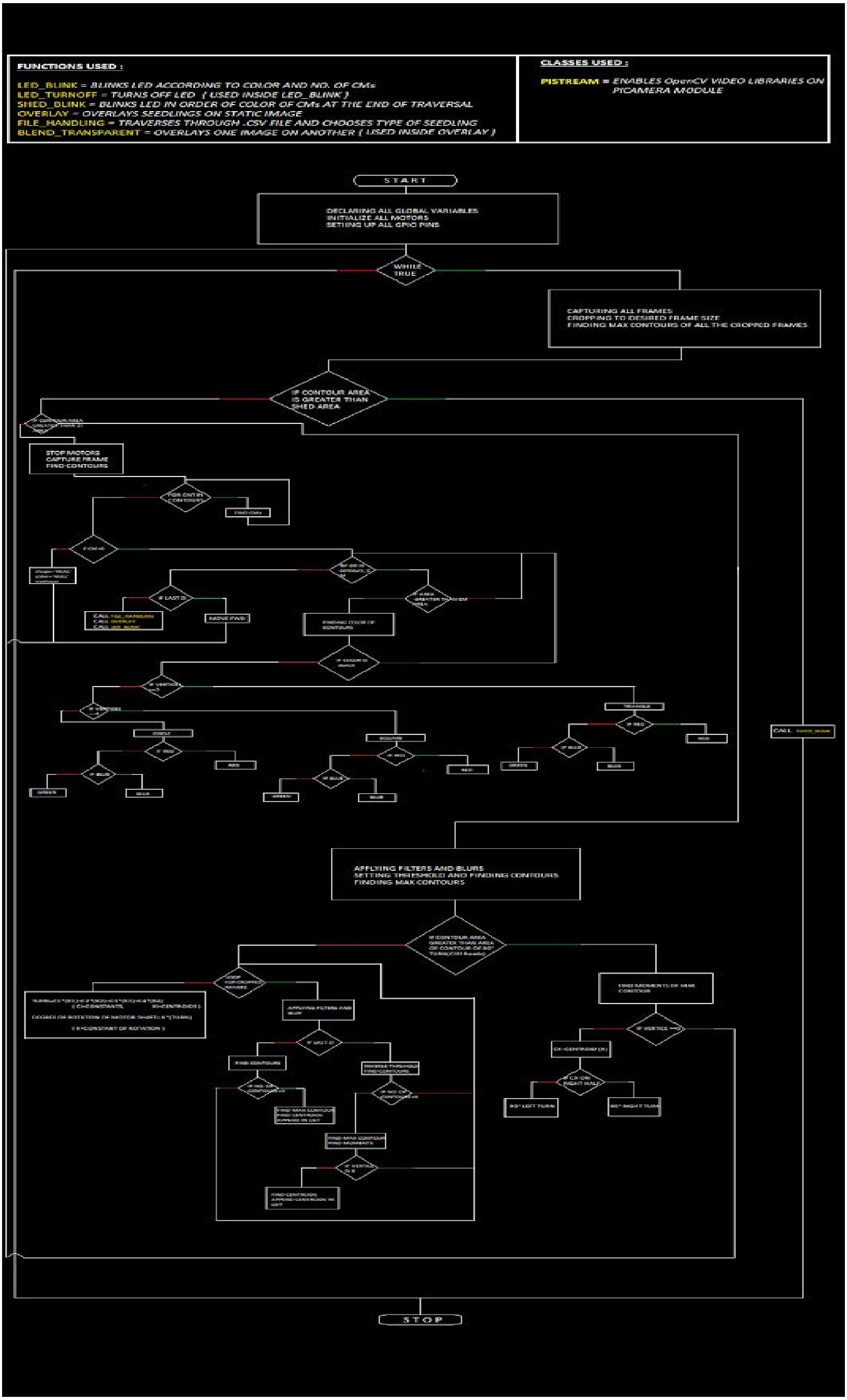
- **file_handling**
  - If color and shape are null,
    - Return null.
  - Else,
    - Open and read the csv file line by line.
    - Find the color and shape in each line.
    - Eliminate the color and shape from the line, remaining string is the name of the image to be overlayed.
    - Return the name of image to be overlayed
- **overlay**
  - If number of CMs are greater than zero,
    - Find the type of zone using ZI variable passed.
    - Find and read the overlay image using the string name passed by **file_handling** function.
    - Overlay the image on the particular zone in the static plantation background using the predefined (by *eyantra*) function **blend_transparent.**

- **LED_Blink**
  - Initiate loop that runs the number of times the color markers are detected.
  - Blinks the led of the desired color.
  - Turns off the led for a small instant before blinking the second color by calling **TURNOFF_LED** function.

- **TURNOFF_LED**

  - Turns the led OFF.

****please refer to the link (if flowchart image seems too pixelated) :**
https://drive.google.com/open?id=1G5VS85pZoAJ2uL5nbv7fNcgNDD2HIYHv


FUNCTIONS USED :
LED_BLINK = BLINKS LED ACCORDING TO COLOR AND NO. OF CMs
LED_TURNOFF = TURNS OFF LED { USED INSIDE LED_BLINK }
SHED_BLINK = BLINKS LED IN ORDER OF COLOR OF CMs AT THE END OF TRAVERSAL
OVERLAY = OVERLAYS SEEDLINGS ON STATIC IMAGE
FILE_HANDLING = TRAVERSES THROUGH .CSV FILE AND CHOOSES TYPE OF SEEDLING
BLEND_TRANSPARENT = OVERLAYS ONE IMAGE ON ANOTHER { USED INSIDE OVERLAY }
CLASSES USED :
PISTREAM = ENABLES OpenCV VIDEO LIBRARIES ON PICAMERA MODULE
START
DECLARING ALL GLOBAL VARIABLES
INITIALIZE ALL MOTORS
SETTING UP ALL GPIO PINS
WHILE TRUE
CAPTURING ALL FRAMES
CROPPING TO DESIRED FRAME SIZE
FINDING MAX CONTOURS OF ALL THE CROPPED FRAMES
IF CONTOUR AREA IS GREATER THAN SHED AREA
STOP MOTORS
CAPTURE FRAME
FIND CONTOURS
FIND CMs
MOVE FWD
FINDING COLOR OF CONTOURS
TRIANGLE
CIRCLE
SQUARE
RED
GREEN
BLUE
CALL SHED_BLINK
APPLYING FILTERS AND BLURS
SETTING THRESHOLD AND FINDING CONTOURS
FINDING MAX CONTOURS
FIND CONTOURS
STOP

Challenges

Q8. What are the major challenges that you can anticipate in addressing this theme and how do you propose to tackle them?

(8)

The following were the challenges that we faced during the implementation of this theme practically using the components provided to us in the kit,

- ➔ The *line following* was the biggest task at hand as it has to be accurate if all the other processes have to be done accurately too.

  Solution : Along with our line following algorithm (which is highly dependent on detection of contours), we used a couple of blurring techniques and filters to increase the accuracy of the contour detection for better line following.

- ➔ To *detect the Zone Indicators,* it was surely a challenge as the usual line traversal algorithm had to be altered so as to detect when a *ZI* is detected and then the bot has to stop there to do further process.

  Solution : We used the fact that the *ZI* is larger in size than a normal block of the line of same length, so we compared the area of the contour in every frame to detect if a *ZI* has been encountered.

- ➔ To determine the seedling from colour markers given in the arena.

  Solution : First we determined the number, color and shape of colour markers, combination of these will result in the type of flower seedling to be planted. The list will be given in the .csv file from where we have extracted the information about what combination of colour and shape will result in which flower seedling using file handling.

- ➔ To overlay the image of the particular flower of which the seedling is being planted after detecting the colour marker.

  Solution : First we determined the coordinates of the different planting zones in the 'plantation.png'. Then according to the detected colour marker we overlaid that specific picture onto the 'plantation.png' on a particular coordinate. And incremented the coordinates according to the counts of flower we have to plant.

- ➔ To turn the bot on the cliff region in the arena as it has 90 degree turns which are not easily accomplished.

  Solution : We put a seperate condition for checking the 90 degree turn in the line and by this detection of the turn a special input to the motors is sent which makes it easy to accomplish the task.

- ➔ Traversing the bot through the Inverted Plains was a great challenge.

  Solution : This problem is resolved by our unique robot traversing algorithm in which it detects four maximum contours. When our robot will reach the Inverted plains, aa new algorithm which enables inverted threshold for detecting contours nis activated so the white line is followed instead of black.

- To determine that the path has ended and the shed has been reached was a challenge of its own.

  Solution : We modified our algorithm to another extent and added a condition which checks the area of the contour with a static threshold value of the shed, basically it will stop the motors when the condition of comparison is satisfied.

- To blink the LED in the exact same order of colors that it had been blinked during the traversal.

  Solution : During the traversal when the function which blinks the led is called, it also stores the corresponding color of the LED in a list and at the shed the list is traversed to get the colors in the definite order and then the LEDs are blinked again in the very same order.