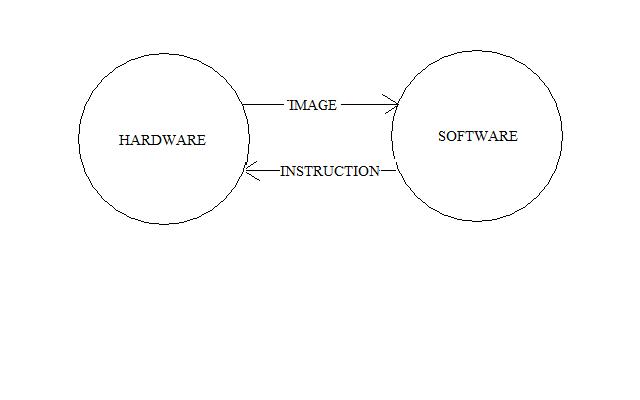
|  |
| --- |
| Birla institute of technology |
| **SYSTEM DESIGN DOCUMENT** |
| An autonomous robot with pseudo-vision system capable of obstacle detection and path recognition |
|  |
| **Team:** |
| **Ayushi Garg 3BE\4048\06**  **Isha Gautam 3BE\4054\06**  **Mayank Kumar Dadheech 3BE\4057\06**  **Sumit Singh 3BE\4074\06** |
| **Project Guide:**  **Mr. Shripal Vijayvargiya** |

|  |
| --- |
|  |

# Data Flow Diagrams

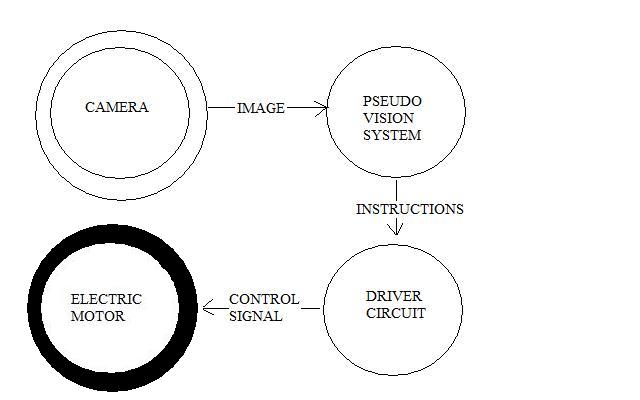
## Level 0

****

**Fig. 1 Level 0 DFD**

## Level 1

### Hardware



**Fig 2. Hardware Components**

#### Electric Motors

Two Electric motors will be used to drive the vehicle wheels. One is for left wheel and the other is for right wheel.



Fig 3: Electric Motor

#### Driver Circuit

The Driver circuit is used to control the operation of motors. It consists of an H Bridge. L293D is an H Bridge IC. The Motors are connected to it as shown in the circuit diagram. Input is given at Pin 4,5,6,7. Pin 4 and 5 control motor A. Pin 6 and 7 control motor B.

#### Camera

The Camera is mounted in front of the robot. It is connected to the PC through USB 2.0 port.

The Camera is set to take pictures automatically at regular intervals or when triggered through the pc. All the pictures are stored in a folder named ‘Images’ and the images are named by the time stamp at which they are taken. Having the images named by time stamp makes it easier to get the last taken image (and images taken just before that) for processing the result.

### Software

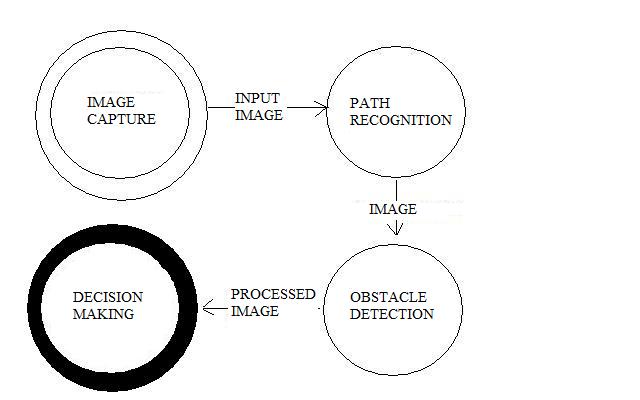


Fig 4: Software Modules

# Circuit Diagram

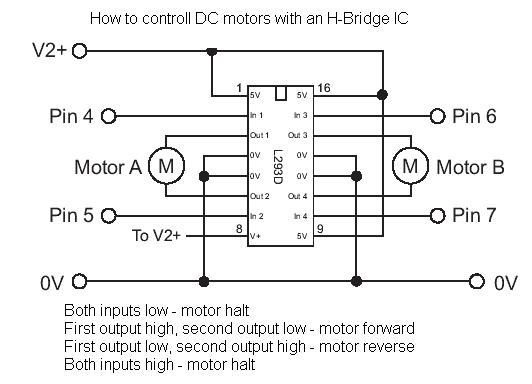


Fig 5: Circuit Diagram of Driver Circuit

# Algorithms

## Obstacle Detection

Fig 6: Process Flow of Obstacle Detection Algorithm.

fig 7: Detailed Process In Object Detection Algorithm

##### Image Capture

The image captured by camera will be transmitted to computer and will be stored in image folder. The images in this folder would serve as input to further processing.

##### Floor Identification

The floor on which the robot is running is identified in this module. The sub steps involved in this are as follows:

1. sample space in any user specified or default shape eg triangle or square.
2. Select the Threshold level: If the immediate floor space includes obstacles which is setting the threshold high, then it will help remove those obstacle pixels from consideration. This, however, can also remove floor pixels that are slightly “discolored” from the rest.
3. Handling highlights (reflections etc).
4. Histogram spread.

##### Dilate

The image is dilated in this module. Sub-steps involved are:

1. Specify the direction from which object should be dilated.
2. Specification of kernal.
3. Border’s role.
4. No. Times dilation will take place.

##### Negate

The image will simply be negated.

##### Side-fill

The empty space at the bottom of the image is filled as long as an edge is not encountered. It’s sub-steps are as follows:

1. Specify the direction from which object should be filled from eg top to bottom.
2. Isolate mask.
3. Skip border.

##### Erode

Just eroding or shrinking the current image horizontally by an amount large enough such that the resulting white areas would be large enough for the robot to pass without hitting any obstacle. It is useful for removing noise from an image.

Sub-steps involved are:

* 1. Specify the direction from which object should be eroded.
  2. Specification of kernal.
  3. Border’s role.
  4. No. Times erosion will take place.

##### Smoothening

It involves following two steps:

1. Specify the Window Size that the outline of the blob will be averaged across.
2. Specify the weight that the final outline is biased towards. 0 % is unchanged, 100% means the outline is replaced by the averaged.

#### Path Recognition

Fig 8: Process Flow Of Path Recognition Algorithm

#### Decision Making

**Fig 9. Process Flow Of Decision Making**

# Use Case Diagram



**IMAGE CAPTURE**

**Function:** Capture the image

**Input:** Pixel intensity matrix

**Source**: Front area of camera

**Output**: Image

**Destination**: PC memory

**Action**: Camera captures the image and then stores in PC memory

**Requires**: camera ,connecting wires, PC

**Pre condition**: Good quality camera , ample space and processing speed

**Post condition**: Image should be saved and processed.

Electric Motor

Wheel Rotation Control

**WHEEL ROTATION CONTROL**

**Function**: Rotation of wheel

**Description**: wheels are rotated according to the instructions given by the PC through Motor driver

**Input**: Instructions given by the PC

**Source**: PC

**Output**: voltage instructions to the wheels

**Destination**: electric motors

**Requires** : Instructions , driver circuit , electric motor

**Pre condition**: Input image has been processed and instructions are send to the motors

**Post condition**: wheels are rotated and follows the path.



**MOTOR CONTROL**

**Function**: Control right and left motors for rotation.

**Description**: wheels are rotated according to the instructions given by the PC through Motor driver.

**Input**: Instructions given by the PC.

**Source**: PC.

**Output**: voltage instructions to the wheels.

**Destination**: electric motors.

**Requires** : Instructions , driver circuit , electric motor.

**Pre condition**: Input image has been processed and instructions are send to the motors.

**Post condition**: wheels are rotated and follows the path.

****

**IMAGE STORAGE**

**Function**: Image to be stored at appropriate location.

**Description**: Path and name based on timestamp is assigned to each image.

**Input**: Image Captured.

**Source** : Camera.

**Output**: Image saved.

**Destination**: The saved image will be used for further processing.

**Requires:** Captured image.

**Pre condition:** Path should be specified.

**Post condition**: Image will be retrived for image processing algorithms.

**GRAPHICAL USER INTERFACE**

**Function**: It will provide basic control and help to user.

**Description**: User friendly interface would be generated.

**Input**: Control instructions to user through input devices such as keyboard or mouse.

**Source** : Input devices such as keyboard or mouse..

**Output**: Image display and instruction to driver circuit.

**Destination**: Output devices and driver circuit.

**Requires:** Captured image.

**Pre condition:** Nil.

**Post condition**: Nil.

**PATH DETECTION**

**Function**: Path to be traversed.

**Description**: Path has been detected after the processing of the images.

**Input**: Image captured.

**Source** : Camera.

**Output**: Processed Image for path detection.

**Destination**: Processed image will go into the Decision making.

**Requires:** Captured image.

**Pre condition:** Quality Image should be captured and passed into the PC memory.

**Post condition**: Processed Image will be send to the PC.

**OBJECT DETECTION**

**Function:** Obstacle has been detected and avoided.

**Description :** It finds the floor and detects all the rest things around as obstacles.

**Input :** Image captured.

**Source :** Camera.

**Output**: processed image for object detection.

**Destination** : processed image has been send to the decision making.

**Requires**: Captured image.

**Pre condition**: Quality Image should be captured and passed into the PC memory.

**Post condition**: Processed Image will be send to the PC.

**DECISION MAKING**

**Function:** Final path , avoiding objects ,has been detected.

**Description:** Further functioning for the robot has been detected by processing the outputs from path detection and object detection algorithms.

**Input:** Processed images from path detection and object detection algorithms.

**Source:** Algorithms from PC.

**Output:** Instructions to the driver circuit.

**Destination:** Driver circuit.

**Requires**: Processed images, PC , Driver circuit.

**Pre condition**: Processed Images from path detection and object detection algorithm should be send .

**Post condition:** Instructions will be send to the motor and the robot is moved accordingly.

**Error Handling**

**Function:** To handle any run time errors.

**Description:** Error modules will handle any exception to avoid any abnormal termination.

**Input:** Nil.

**Source:** Other software modules.

**Output:** To print the exception.

**Destination:** Program control would be tranferred to image capture routine.

**Requires**: Nil.

**Pre condition**: Nil.

**Post condition:** Nil.

# Class Diagram

# Graphical User Interface

