### AI POWERED HAND GESTURE CONTROL CAR

A

Project Report

Submitted In Partial Fulfillment of the Requirements For The Award Of the

**Bachelor of Technology**

In

**Electronic and Communication Engineering**

***Under Guidance Of***

**Mr. RAHUL SHARMA**

**MICROSOFT & HPE CERTIFIED TECHNICAL TRAINER**

**Project Carried Out At**

****

## Ardent Computech Pvt. Ltd.

**(AnISO 9001:2015 Certified)**

**SDFBuilding, Module No:132, Ground Floor Sector V, GP Block, Kolkata- 700091**

### Mrudu Malay Kundu

### ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

We would like to show our greatest appreciation to **Rahul Sharma, Technical Trainer at Ardent, Durgapur**. We always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized. We also want to thank them for sharing their pearls of wisdom with us during the course of this project.

We are also immensely grateful to **Rahul Sir** for his comments on earlier versions of the manuscripts, although any errors are our own and should not tarnish the reputations of these esteemed professionals.

Words are inadequate in offering our thanks to the other trainees, **project assistants** and other members at **Ardent Computech Pvt. Ltd.** for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

## Table of Contents

|  |  |  |
| --- | --- | --- |
| **SERIAL NO.** | **TOPIC** | **PAGE NO.** |
|  |  |  |
| **1** | **Company Profile** | **01** |
|  |  |  |
| **2** | **Abstract** | **02** |
|  |  |  |
| **3** | **Introduction** | **03** |
|  |  |  |
| 4 | **Objective and Scope** | 04 |
|  |  |  |
| **5** | **Related Works** | **05** |
|  |  |  |
| **6** | **System analysis** | **07** |
|  |  |  |
| **7** | **Work Flow** | **08** |
|  |  |  |
| **8** | **Hardware and**  **Software Requirements** | **09** |
| **9** | **System Architecture** | **10** |
| **9.1** | **Block Diagram** | **11** |
| **9.2** | **Circuit Diagram** | **12** |
|  |  |  |
| **10** | **System design** | **15** |
| 10.1. | Gantt chart | 15 |
| 10.2. | Pert Chart | 16 |
| 10.3. | Sequence diagram | 17 |
| 10.4. | Activity diagram | 18 |
| 10.5. | Snapshot | 19 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **11** | **Implementation and Testing** | **21** |
| 11.1 | Introduction | 21 |
| 11.2 | Objective ofTesting | 21 |
| 11.3 | Process Overview | 21 |
| 11.4 | Test Cases | 22 |
| 11.5 | Testing Step | 22 |
| 11.6 | Validation | 23 |
| 11.7 | White Box Testing | 24 |
| 11.8 | Black box testing | 24 |
| 11.9 | SystemTesting | 24 |
| 11.10 | Output Testing | 25 |
| 11.11 | User Acceptance Testing | 25 |
| 11.12 | Integration Testing | 25 |
| 11.13 | Functional Testing | 25 |
|  |  |  |
| **12** | **System Security** | **26** |
| 12.1 | Database Security | 26 |
| 12.2 | System Security | 26 |
| 12.3 | Limitation | 26 |
|  |  |  |
| **13** | **Cost Estimation** | **27** |
|  |  |  |
| **14** | **Conclusion** | **29** |
|  |  |  |
| **15** | **Future scope and Future**  **Enhancement** | **29** |
| **16** | **References** | **30** |

### ABSTRACT

Robots are now an essential part in the modern society. Nowadays robots are everywhere, their application are far reaching and can encompass a variety of different machines with different purposes. The major goal of the project work is to control the movement of a robot through gestures. The system is made up of two major components: An laptop device with an app capable of reading the gesture from camera embedded in the device, which is controlled by tilting the device. The tilting of the device sends data signals through a Wi-Fi network. This signal is processed with the assistance of the NodeMCU, which instructs the robot to move in the specified direction by sending voltages to the motor driver. The essential operation of the robot is the transmission of data signals from an android device, readings, to the NodeMCU board installed on the robot. The software compiled in the NodeMCU executes according to that value, causing the robot to work properly while connected to the hotspot of the android device.

### INTRODUCTION

* 1. **Background:**

Today, robotics is one of the most sophisticated fields of technology. A robot is often an electromechanical machine that can execute tasks automatically. Robots can be fully autonomous or semi-autonomous. An autonomous robot is not directed by people and acts on its own preprogrammed decisions based on detecting its surroundings. The bulk of industrial robots are self-contained because they must work at fast speeds and with excellent accuracy. However, semiautonomous robots, also known as human-controlled robots, operate under the control of humans through the use of a control system as an interface. Voice recognition, touch-controlled, and gesture-controlled systems are some of the most widely utilized control methods. Historically, significant attempts have been made to create intelligent and natural interfaces between people and computer-based systems that are based on human gestures. Gestures can come from anybody action or state, although they most often come from the face or hand. Gesture recognition is a method for a computer to comprehend human body language. Gestures offer both people and computers with an intuitive interface. As a result, such gesture-based interfaces can not only replace standard interface devices, but also be used to enhance their capabilities.

* 1. **Problem Statement**

In the early days, the only method to interact with a robot was to program, which took a lot of time and effort. The gesture-controlled robot was developed as science and robotics progressed. This robot will be controlled with a gyroscope sensor supported android device, the tilting of the device will be utilized as input for the robot's movement, which is the main idea behind this project. In this project, a system (robotic vehicle) will be developed which can identify human interaction and perform the objectives that have been set to it. The robotic vehicle will receive signals sent by the android device which will control its movement.

* 1. **Motivation**

The main processing unit consists of ESP8266 to perform specified operations. ESP8266 board is used as the main microprocessor of the device. It will run and will control the movement of the vehicle through the use of the motor driver who will control the wheels. The Programs are compiled using Arduino IDE and then the sketch is uploaded to the board using a pc and a USB cable. Speed variations can be achieved by making slight changes to the sketch. It is based on a 2-wheel drive and a caster wheel. The car is upgradable and can replace, upgrade and expand the input sensors and add other functionality by adding other modules.

1. **OBJECTIVE AND SCOPE**

The goal of this project is to create a robotic battery-powered automobile that can be operated wirelessly by gesture. While doing so, we will also need to:

* Analyze the energy consumption rate of the four motors that drive the wheels over a specified period of time. The motors that use the least amount of energy will be considered.
* Research the most efficient, low-cost, and high-memory-capacity electronic board.
* Research the most effective, low-cost, long-range, energy-efficient communication technologies.
* Analyze the sensors that will be on board to allow the robotic automobile to be easily maneuvered. For example, we may use similar sensors to pre-determine the terrain in which the robot would operate.

### RELATED WORKS

Similar research from the past are discussed in this chapter, and some of the methodologies and working concepts employed are reviewed.

* 1. **ROPE AND STRINGS** The early semi-autonomous cars were completely mechanical and were mostly utilized as children's toys. The toy car is comprised of wheels, a string, and a rope and can only move in one direction. When the user pulls back on the rope, tension is created on the string, and when released, the vehicle travels forward at a constant speed that decreases as the tension on the string decreases. Problems
     + The rope becomes slack and may be cut.
     + It covers a very small distance.
     + It can only travel in one direction.
  2. **STRINGS AND WINDER** Later, modifications were made, and a toy car capable of moving in both directions was developed. This version was similar to the rope and strings model except for the addition of a winder, which was necessary since the rope used in the previous version could only be pulled in one direction and was not particularly strong. The winder is rotated in the opposite direction that the user desires the vehicle to go. The user then releases the winder, and the vehicle goes at a high speed that decreases as the strain on the string decreases and ultimately stops.

Problems

* + - The winder is easily damaged.
    - It only travels a small distance.
  1. **STRINGS AND WHEELS** Further improvements were made to the previously described toy vehicles, and the usage of strings and wheels was discarded. The vehicle was constructed with just wheels and springs linked to the wheels, but previous versions had their string put only at the back wheels, which controlled the vehicle's motions. The springs in this variant were located at the front and rear wheels. The vehicle is dragged backward or pushed forward to travel ahead or backward at a constant speed that decreases as the tension on the string decreases. Problems
     + The wheels deteriorate due to excessive dragging.
     + It travels a short distance.
  2. **BATTERY-POWERED** Mechanical toy vehicles became obsolete, and electrical-powered toy vehicles were developed. The car was constructed with DC motors that controlled the wheels, a battery for power, and a switch to turn the vehicle on and off. The motors are connected to the power supply through connecting wires, and the power source is then connected to the switch. When the switch is turned on, the battery gives power to the motors that drive the wheels, allowing the vehicle to continue going ahead. If the vehicle hits a wall, the battery will continue to supply power to the motors until it reaches its limit, at which point it will stop. This depletes the battery. Problems
     + It cannot be used in every environment.
     + It is power consuming.

**IDENTIFICATION OF NEED**

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studies to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The System is viewed as a whole and the input to the system are identified. The outputs from the organization are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and Decisional variables, analysis and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be 9scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem area are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

**Feasibility Study**

Feasibility study is the detailed study expanded from the result of initial investigation. This is done by investigating the existing system in the area under investigation or generally ideas about a new system. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources.

Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Objective of feasibility study is to acquire a sense of the scope of the problem. It is carried out to select the best system that meets performance requirement.

Feasibility analysis involves the following steps:

From a project and appoint a project leader.

* Prepare system flowchart.
* Weigh system performance.
* Prepare and report final project directive to management.

The document provides the feasibility of the project that is being designed and lists various area that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

Economic Feasibility

Are there sufficient benefits in creating the system to make the costs acceptable? Or, are the costs of not creating the system so great that it is advisable to undertake the project? These are the important questions to be answered in economic feasibility. Proposed system will not face any economic constraint as it will be developed by students only. And it will help to automate manual processing, which will save time and money. This will provide economic benefits.

Operational Feasibility

Will the system be used if it is developed and implemented? Will there be resistance from users that will undermine the possible application benefits? When we look from this perspective, we don’t see any riskin implementing and making it operational. This project will help everyone, as it is planned to be more reliable, maintainable, affordable and producible.

Technical Feasibility

Can the work for the project be done with current equipment, existing software and hardware technology and available personnel? These are the questions which need to be answered to check Technical feasibility. Project will be developed with Arduino which is the microcontroller. There is basic requirement of hardware to run this application. This application can be accessed by using any device like (Personal Computers, Laptop and with some hand-held devices).

1. **WORK FLOW**

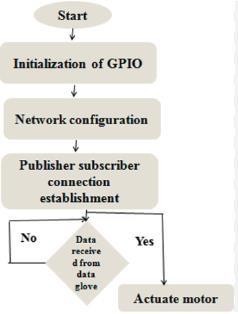
The system design is broken down into two main categories,

**Power Supply:** A power supply is an electronic device that supplies electric energy to an electrical load. The primary function of a power supply is to convert one form of electrical energy to another and, as a result, power supplies are sometimes referred to as electric power converters.

**Wi-Fi module:** It is small wireless serial communication module that can be connected with a Micro-Controller to receive and send data when connected with other Wi-Fi devices.

**Arduino IDE:** Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino consists of both a physical programmable circuit board and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

**Motor driver:** It is a small circuit that hoists the motor driving IC, and can control two motors at the same time. It controls the motor speed by pulse width modulation (PMW).



### HARDWARE & SOFTWARE REQUIREMENTS

**Software Requirements**

* **OS:** Windows 10 or above
* **Python 3.9**
* **Arduino IDE:** Arduino 1.8.9 or above

#### Computer Hardware Components

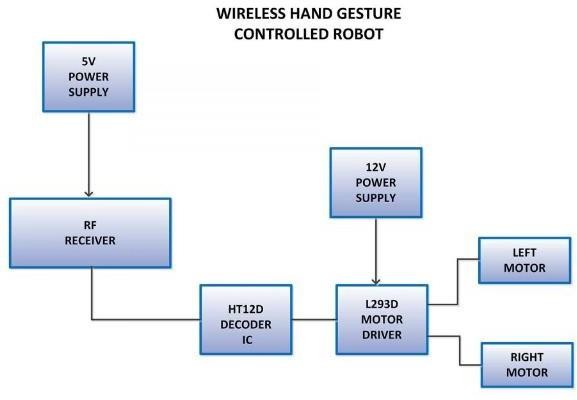
* **Processor:** i5 7th Generation.
* **Hard Disk:** 1 TB
* **Memory:** 8GB RAM
* **Mouse:** Any Standard.
* **Keyboard:** Any Standard.
* **Monitor:** Any color monitor.

**Robot Hardware Parts**

* USB Cable
* **Wires:** Jumper Wires as required.
* **Sensors:** ESP8266
* **Others:** L298N Motor Driver, 9V DC Motor

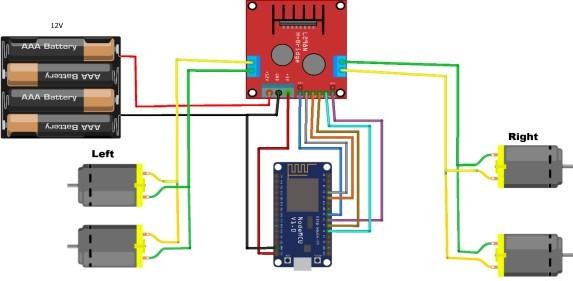
### SYSTEM ARCHITECTURE

* 1. **Block Diagram**

****

**Block Diagram**

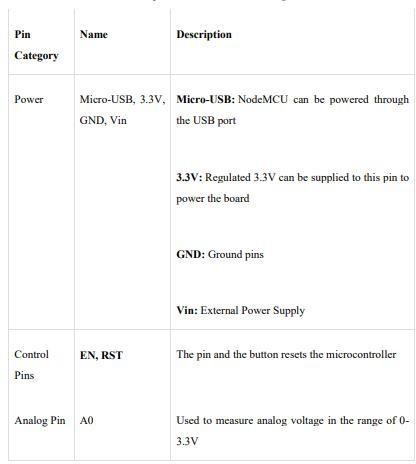
* 1. **Circuit Diagram and Description**

****

The circuit above has been designed to function according to the aim of this project which is wirelessly controlling the movement of a robotic car with the aid of gestures from the gyroscope sensor embedded in an android device. In the circuit above the motors are connected to the motor driver, the positive and negative of the two left motors are connected to the output 1 and 2 of the motor driver respectively, the positive and negative of the two right motors are connected to the output 3 and 4 of the motor driver respectively. The motor driver is connected to the node MCU, IN1 and IN2 of the motor driver are connected to pin 3 and 4 of the node MCU to control the left motors, IN3 and IN4 of the motor driver are connected to pin 7 and 8 of the node MCU to control the right motors, ENA and ENB of the motor driver are connected to pin 5 and 6 of the node MCU to control the speed and direction of the motors. The vin pin of the node MCU is connected to the 5v outlet of the motor driver, the ground pin of the node MCU is connected to the ground outlet of the motor driver and negative of the battery and 12v of the motor driver to the positive of the battery.

##### ESP8266 Wi-Fi Module

The Node MCU is a micro-controller capable of communicating wirelessly with other devices. The Node MCU ESP8266 development board comes with the ESP-12E module containing the ESP8266 chip which is a Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. It has high processing power with in-built Wi-Fi / Bluetooth. It also has a Deep Sleep Operating features make it ideal for IoT projects. Since an IoT device has to operate 24 hours a day and 7 days a week, this Deep Sleep feature wakes the device whenever there is a command, therefore helps save power consumption. NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.





L298n Motor Driver

L298N is a dual full-bridge motor driver module for DC and stepper motors. It controls two DC motors' speed and direction. This module contains a dual-channel H-Bridge motor driver IC. This module controls DC motor speed and direction using two methods. PWM controls speed, and H-Bridge controls rotation. These modules may control two DC motors or one stepper. This module has an L298 motor driver IC and 78M05 5V regulator. L298N Module controls up to 4 DC motors or 2 with directional and speed control.

#### DC Motors

****

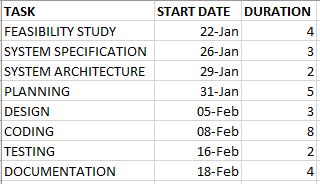
The Electric motors took a revolutionary change in last few decades. Now a days, almost every mechanical movement needs an electrical motor to drive our everyday life. Electrical motors are not only used for mechanical movement but also produce and store mechanical energy. Now in various purposes, hundreds of devices are accomplished in our regular life by an electric motor. We use motor in automobiles, robot, hand power tools, fan, washings machine, food blender even in medical science. In this way, we use two DC motor in our Spy Car.

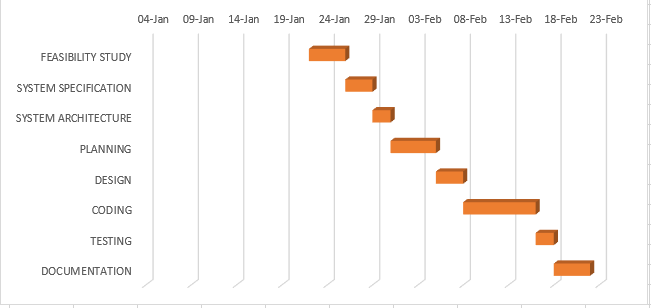
#### Rechargeable Battery

Rechargeable Li Po Battery is very effective in the field of Robotics and power management system. It can store energy and produce DC voltage supply. We use 2 cell 3.7 Volt battery that gives us almost 7.4V and 2400mA.

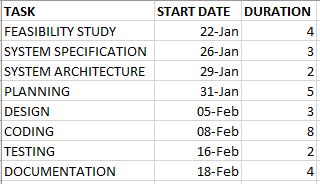
### SYSTEM DESIGN

* 1. **Gantt Chart**

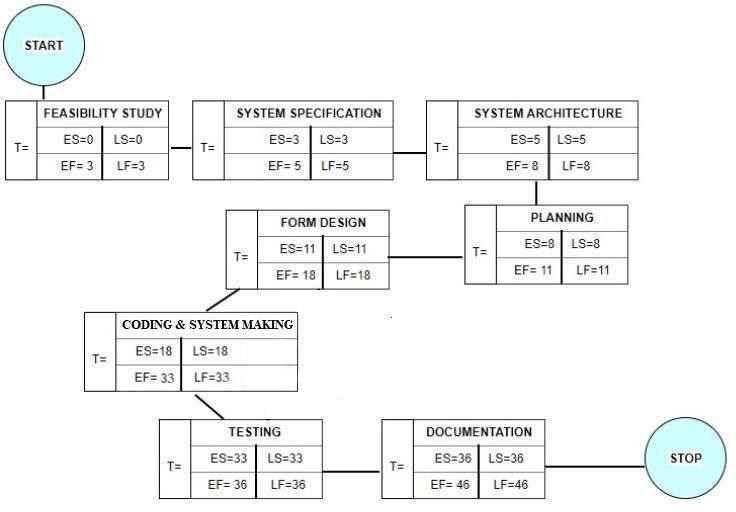
****

****

* 1. **Pert chart**

****

**Pert chart Data**

****

**Pert Chart**

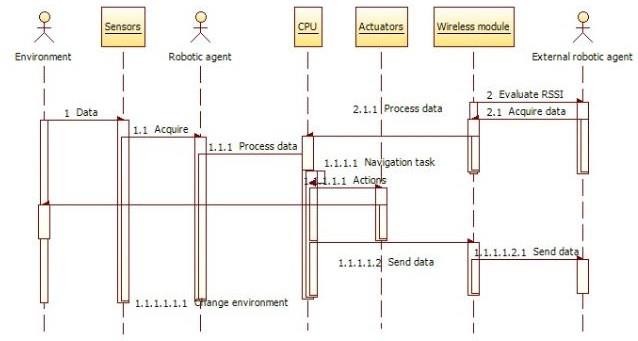
* 1. **Sequence Diagram**

Sequence diagrams can be useful reference diagrams for businesses and other organizations. Try drawing a sequence diagram to:

* Represent the details of a UML use case.
* Model the logic of a sophisticated procedure, function, or operation.
* See how tasks are moved between objects or components of a process.
* Plan and understand the detailed functionality of an existing or future scenario.

Popular Sequence Diagram Uses

* **Usage Scenario** – A usage scenario is a diagram of how your system could potentially be used. It’s a great way to make sure that you have worked through the logic of every usage scenario for the system.
* **Method Logic** - Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to Usage Scenario - A usage scenario is a diagram of how your system could potentially be used. It's a great explore the logic of any function, procedure, or complex process.



**Figure: Sequence Diagram**

* 1. **Activity Diagram**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

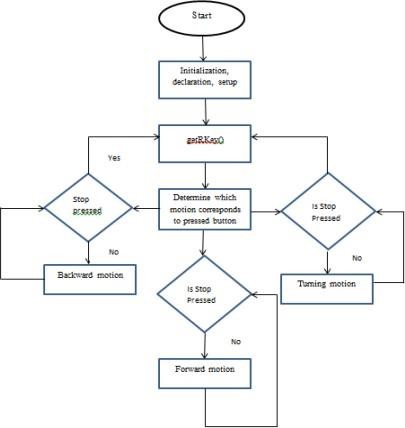
Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purpose of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

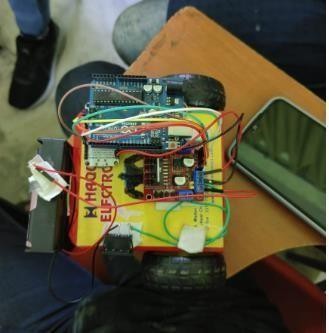
Where to Use Activity Diagrams?

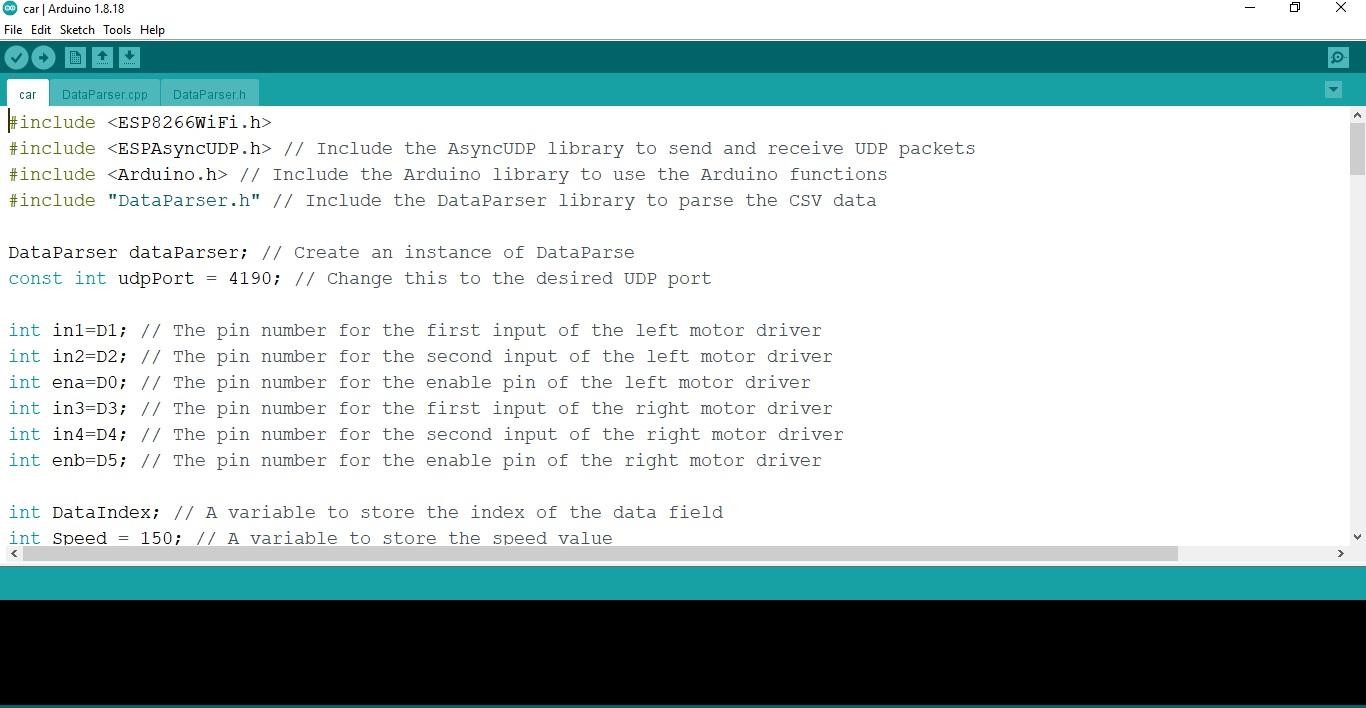
The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not includes messages. Activity diagram is suitable for modeling the activity flow of the system. An application can has multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.



**Figure: Activity Diagram**

* 1. **Snapshots**

****



### IMPLEMENTATION & TESTING:

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" of product validation. It should be thorough enough to be useful but not so thorough that no one outside the test group will read it.

OBJECTIVES OF TESTING

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user- friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

PROCESS OVERVIEW

The following represents the overall flow of the testing process:

* Identify the requirements to be tested. All test cases shall be derived using the current program specifications.
* Identify which particular test(s) will be used to test each module.
* Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
* Identify the expected results for each test.
* Document the test case configuration, test data, and expected results.
* Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
* Successful unit testing is required before the unit is eligible for component integration/system testing.
* Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.

TEST CASES

A test case is a document that describes an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data. Requirement expected results. The process of developing test cases can help find problems in the requirements or design of an application since it requires completely thinking through the operations of the application.

TESTING STEPS

* **Unit Testing:** Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.
* **Integration Testing:** Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

VALIDATION

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exist.

1. The function or performance characteristics confirm to specification and are accepted.
2. A deviation from specification is uncovered and a deficiency lists is created.
3. Proposed system under consideration has been tested by using validation test and found to be working satisfactorily.

|  |  |
| --- | --- |
| **Tested By:** | Mrudu Malay Kundu |
| **Test Type** | Unit Testing |
| **Test Case Number** | 1 |
| **Test Case Name** | Testing Car Control via Camera |
| **Item(s) to be tested** | |
| 1 | * The user should control the car via Hand Gesture and can steer and go forwards and backwards * Laptop gives a Wi-Fi signal to the car and from the signal, the car works. * F is used for forwarding car movement. B is used for a backward car moment. Similarly, Other characters are used for other directions which are mentioned in the Arduino code |
| **Specifications** | |
| **Input**   * Signals from Phone | **Expected Output/Result**   * Car moves according to the hand gesture |

WHITE BOX TESTING

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

BLACK BOX TESTING

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

SYSTEM TESTING

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity.

OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hardcopy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

USER ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

* 1. Input Screen Design.
  2. Output Screen Design.
  3. Format of reports and other outputs.

INTEGRATION TESTING

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

* Whether the software matches with the specification (i.e. process base) to verify the product.
* Whether this software in one client what wants (i.e. product base) to validate the product.
* Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behavior if exists. Later all the module was integrated and flat file was generated.

FUNCTIONAL TESTING

These are the points concerned during the stress test:

* Nominal input: character is in putted in the place of digits and the system has to flash the message “Data error".
* Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

### SYSTEM SECURITY MEASURES

DATABASE SECURITY

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So, you should follow some security measures. We have used security levels in database level at system level.

SYSTEM SECURITY

Robotics can involves adding internet connectivity to a system of interrelated computing devices, mechanical and digital machines, objects, animals and/or people. Each "thing" is provided a unique identifier and the ability to automatically transfer data over a network. Allowing devices to connect to the internet opens them up to a number of serious vulnerabilities if they are not properly protected.

LIMITATIONS

The L298N is specified to work with 7-12V , so we are using a bigger battery with 7.2V or 11.1V. When the new Power source is connected, everything turns on and you can connect your phone with the Wi-Fi device, but if the power source is below or above it then this might cause the motor driver to burn.

1. **COST ESTIMATION**

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software

The necessary steps in this model are:

* 1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
  2. Determine a set of 15 multiplying factors from various attributes of the project.
  3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below

Ei=a\*(KDLOC)b

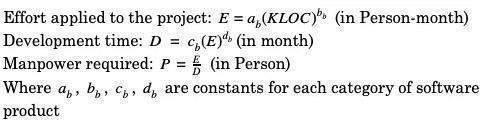
The value of the constant a and b are depending on the project type. In COCOMO, projects are categorized into three types:

1. Organic
2. Semidetached
3. Embedded
4. **Organic:** A development project can be treated of the organic type, if the project deals with developing a well- understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple e-learning portal, and data processing systems.
5. **Semidetached:** A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex e-learning portal.
6. **Embedded:** A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. For Example: ATM, Air Traffic control.

For three product categories, Bohem provides a different set of expression to predict effort (in a unit of person month) and development time from the size of estimation in KLOC (Kilo Line of code) efforts estimation considers the productivity loss due to holidays, weekly off, coffee breaks, etc.

According to Boehm, software cost estimation should be done through three stages:

The following expressions give the basic COCOMO estimation model:





### CONCLUSION

The objective of this project which is to control a robotic car using gestures was achieved with just a few problems along the way. The robot is showing proper responses whenever the device is tilted in any direction. The car moves forward whenever the device is tilted downward, moves backward whenever the device is tilted upward, moves left and right whenever the device is tilted left and right respectively and stops whenever the device is parallel to the surface of the earth. The connection between the app and the node MCU is fast, and the robotic car can be controlled from a long range. The problems encountered were with soldering some of the parts to the vero board and bad connecting wires, which we overcame. This project as therefore proved that an android device can be used to control a robot wirelessly.

### FUTURE SCOPE AND FURTHER ENHANCEMENTS

We can modify the rover as per our requirement. If we want to use this for surveillance purpose so we can implement the camera or thermal sensors along with others sensors. Again, once develop the Android application, the robot can respond to intuitive forum gesture based on the accelerometer sensor data, respond to your voice, at a click of a button, or at the swipe on the touchscreen. Further if we want to improve the accuracy of the rover, we can implement sonar sensors so controlling rover from remote place can be possible. We can also implement GPS system so it can be semi- autonomous. The proposed work can be enhanced with the help of more security function like passwords and so on.

### REFERENCES

1. J.S. Bridle, “Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition,” Neurocomputing—Algorithms, Architectures and Applications, F. Fogelman-Soulie and J. Herault, eds., NATO ASI Series F68, Berlin: Springer-Verlag, pp. 227-236, 1989. (Book style with paper title and editor)
2. W.-K. Chen, Linear Networks and Systems. Belmont, Calif.: Wadsworth, pp. 123-135, 1993. (Book style)
3. H. Poor, “A Hypertext History of Multiuser Dimensions,” MUD History, <http://www.ccs.neu.edu/home/pb/mud-> history.html. 1986. (URL link \*include year)
4. K. Elissa, “An Overview of Decision Theory," unpublished. (Unplublished manuscript) [5] R. Nicole, "The Last Word on Decision Theory," J. Computer Vision, submitted for publication. (Pending publication)
5. C. J. Kaufman, Rocky Mountain Research Laboratories, Boulder, Colo., personal communication, 1992. (Personal communication)
6. D.S. Coming and O.G. Staadt, "Velocity-Aligned Discrete Oriented Polytopes for Dynamic Collision Detection," IEEE Trans. Visualization and Computer Graphics, vol. 14, no. 1, pp. 1-12, Jan/Feb 2008, doi:10.1109/TVCG.2007.70405. (IEEE Transactions )
7. S.P. Bingulac, “On the Compatibility of Adaptive Controllers,” Proc. Fourth Ann. Allerton Conf. Circuits and Systems Theory, pp. 8-16, 1994. (Conference proceedings)
8. H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient Scheduling Focusing on the Duality of MPL Representation,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS ’07), pp. 57-64, Apr. 2007, doi:10.1109/SCIS.2007.367670. (Conference proceedings)
9. J. Williams, “Narrow-Band Analyzer,” PhD dissertation, Dept. of Electrical Eng., Harvard Univ., Cambridge, Mass., 1993. (Thesis or dissertation)
10. E.E. Reber, R.L. Michell, and C.J. Carter, “Oxygen Absorption in the Earth’s Atmosphere,” Technical Report TR-0200 (420-46)-3, Aerospace Corp., Los Angeles, Calif., Nov. 1988. (Technical report with report number)
11. L. Hubert and P. Arabie, “Comparing Partitions,” J. Classification, vol. 2, no. 4, pp. 193-218, Apr. 1985. (Journal or magazine citation)
12. R.J. Vidmar, “On the Use of Atmospheric Plasmas as Electromagnetic Reflectors,” IEEE Trans. Plasma Science, vol. 21, no. 3, pp. 876-880, available at [http://www.halcyon.com/pub/journals/21ps03-vidmar,](http://www.halcyon.com/pub/journals/21ps03-vidmar) Aug. 1992. (URL for Transaction, journal, or magazine)