VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA" , BELAGAVI-590018, KARNATAKA

Project Report
On

"PHONE-BOT"

Submitted in the partial fulfillment of the requirement for the award of degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

Submitted By

ABHISHEK ANAND	**1VA19CS002**
AMEENODDIN M	**1VA19CS004**
GAUTAM KUMAR	**1VA19CS017**
SUMAN KUMAR SAHU	**1VA19CS053**

Under the Guidance of
Prof. ASHWINI S S
Assistant Professor,
Dept. Of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SAI VIDYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi | Recognized by Govt. of Karnataka | Approved by AICTE, New Delhi)

Accredited by NBA New Delhi (CSE, ECE, ISE, MECH, CIVIL), NAAC-"A" Grade

RAJANUKUNTE, BENGALURU – 560 064

2022-23

SAI VIDYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi | Recognized by Govt. of Karnataka | Approved by AICTE, New Delhi)

Accredited by NBA New Delhi (CSE, ECE, ISE, MECH, CIVIL), NAAC-"A" Grade

Rajanukunte, Bengaluru- 560 064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Project work entitled **"PHONE-BOT"** carried out by **Mr. SUMAN KUMAR SAHU (1VA19CS053), Mr. GAUTAM KUMAR (1VA19CS017), Mr. ABHISHEK ANAND (1VA19CS002), Mr. AMEENODDIN M (1VA19CS004),** a bonafide students of **SAI VIDYA INSTITUTE OF TECHNOLOGY**, Bengaluru, in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, Belagavi during the year **2022-23.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

----------------------	----------------------------	------------------------
Prof. Ashwini S S	**Dr. Shantakumar B Patil**	**Dr. M S Ganesha Prasad**
Assistant Professor,	HOD,	Principal,
Dept. of CS&E, SVIT	Dept. of CS&E, SVIT	SVIT

External Viva:

	Name	**Signature**
1.	______________________________	______________________________
2.	______________________________	______________________________

ACKNOWLEDGEMENT

The completion of project brings with and sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion. First and foremost we wish to express our deep sincere feelings of gratitude to our Institution, **Sai Vidya Institute of Technology**, for providing mean opportunity to do our education.

We would like to thank the **Management** and **Dr. M R Holla,** Director, Sai Vidya Institute of Technology for providing the facilities.

We extend our deep sense of sincere gratitude to **Dr. M S Ganesha Prasad**, Principal, Sai Vidya Institute of Technology, Bengaluru, for having permitted to carry out the project work on **"PHONE-BOT"** successfully**.**

We are thankful to **Dr. A M Padma Reddy**, Director (A), Professor and Dean (Student affairs), Department of Computer Science and Engineering, Sai Vidya Institute of Technology, for his constant support and motivation.

We express our heartfelt sincere gratitude to **Dr. Shantakumar B Patil,** Professor and HOD, Department of Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for his valuable suggestions and support.

We express our sincere gratitude to **Prof. Ashwini S.S**, Assistant Professor, Project Guide, Department of Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for her constant support.

Finally, we would like to thank all the Teaching, Technical faculty and supporting staff members of Department of Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for their support.

ABHISHEK ANAND **1VA19CS002**

AMEENODDIN M **1VA19CS004**

GAUTAM KUMAR **1VA19CS017**

SUMAN KUMAR SAHU **1VA19CS053**

ABSTRACT

Current robots are either expensive or make significant compromises on sensory richness, computational power, and communication capabilities. We propose to leverage smartphones to equip robots with extensive sensor suites, powerful computational abilities, state-of-the-art communication channels, and access to a thriving software ecosystem. We design a small electric vehicle that costs $50 and serves as a robot body for standard Android smartphones. We develop a software stack that allows smartphones to use this body for mobile operation and demonstrate that the system is sufficiently powerful to support advanced robotics workloads such as person following and real-time autonomous navigation in unstructured environments. Controlled experiments demonstrate that the presented approach is robust across different smartphones and robot bodies.

In this work, we push further along the path to highly capable mobile robots that could be deployed at scale. Our key idea is to leverage smartphones. We are inspired in part by projects such as Google Cardboard: by plugging standard smartphones into cheap physical enclosures, these designs enabled millions of people to experience virtual reality for the first time. Can smartphones play a similar role in robotics? More than 60% of the world's population own smartphones. Commodity models now carry HD cameras, powerful CPUs and GPUs, advanced IMUs, GPS, Wi-Fi, Bluetooth, 4G modems, and more.

Modern smartphones are even equipped with dedicated AI chips for neural network inference, some of which already outperform common desktop processors.
Hardware components on custom robots are quickly outdated. In contrast, consumer-grade smartphones undergo generational renewal on an annual cadence, acquiring higher-resolution and higher-framerate cameras, faster processors, new sensors, and new communication interfaces.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Robots are expensive. Legged robots and industrial manipulators cost as much as luxury cars, and even the cheapest robots from Franka Emika or Clearpath cost at least $10K. Few academic labs can afford to experiment with robotics on the scale of tens or hundreds of robots. In this work, we push further along the path to highly capable mobile robots that could be deployed at scale. Our key idea is to leverage smartphones. We are inspired in part by projects such as Google Cardboard: by plugging standard smartphones into cheap physical enclosures, these designs enabled millions of people to experience virtual reality for the first time. Can smartphones play a similar role in robotics? More than 40% of the world's population own smartphones. Commodity models now carry HD cameras, powerful CPUs and GPUs, advanced IMUs, GPS, Wi-Fi, Bluetooth, 4G modems, and more. Modern smartphones are even equipped with dedicated AI chips for neural network inference, some of which already outperform common desktop processors. We develop and validate a design for a mobile robot that leverages a commodity smartphone for Sensing and computation. The smartphone acts as the robot's brain and sensory system. This brain is plugged into a cheap electromechanical body that costs less than $50. Using off-the-shelf smartphones as robot brains has numerous advantages beyond cost. Hardware components on custom robots are quickly outdated. In contrast, consumer-grade smartphones undergo generational renewal on an annual cadence, acquiring higher-resolution and higher-framerate cameras, faster processors, new sensors, and new communication interfaces. As a side effect, second-hand smartphones are sold cheaply, ready for a second life as a robot. In addition to the rapid advancement of hardware capabilities, smartphones benefit from a vibrant software ecosystem. Current robots are either very difficult for a normal people to afford or make significant compromises on the functionality of the robots such as sensors, environment interaction capabilities, communication and computational abilities, etc. We propose to build a robot equipped with smartphone which acts as the brain for the robot multiple abilities to interact with human and environment to create a powerful ecosystem for itself. We design a small electric robot that costs very less and serves as a robot body for standard Android smartphones. We develop a software's that allows smartphones to use the robot body for mobile operation and demonstrate the functionalities of the system and is sufficiently powerful to support workloads such as person following, autonomous navigation, obstacle avoidance in any environment.

1.1 PROBLEM STATEMENT

We develop and validate a design for a mobile robot that leverages a commodity smartphone for sensing and computation. The smartphone acts as the robot's brain and sensory system. This brain is plugged into a cheap electromechanical body.

1.2 SOLUTION OF THE PROBLEM

We develop and validate a design for a mobile robot that leverages a commodity smartphone for sensing and computation. The smartphone acts as the robot's brain and sensory system. This brain is plugged into a cheap electromechanical body that costs less than 6k rupees. Using off-the-shelf smartphones as robot brains has numerous advantages beyond cost.

1.3 EXISTING TECHNIQUE

- The navigation is not autonomous but relies on user input and leverages a smartphone as a sensor suite for a wheeled robot. Our platform turns a smartphone into the brain of a fully autonomous robot with onboard sensing and computation.

  LIMITATIONS:
- Moreover, the proposed robot is costly without the smartphone and does not leverage recent advancements that enable onboard deep neural network inference or visual-inertial state estimation.
- The project is stale, with no updates to the software for last several years. The project use the smartphone as a remote control for teleoperation, offload data to a server for processing, or rely on commercial or outdated hardware and software.

1.4 PROPOSED TECHNIQUE

- *A BODY FOR A LOW-COST WHEELED ROBOT:*
  - A brain without a body cannot act.
  - In order to leverage the computational power and sensing capabilities of a smartphone, the robot needs an actuated body.
  - We are developing a body for a low-cost wheeled robot which only relies on readily available electronics.

- *MECHANICAL DESIGN:*
  - It consists of a bottom plate and a top cover and connectors that click into place. The bottom plate features the mounting points for the motors and electronics.
  - The four motors are fixed with eight screws. The motor controller and microcontroller attach to the bottom plate. There are appropriate openings for the indicator LEDs and grooves for the encoder disks mounted on the front wheels.
  - The top plate features a universal smartphone mount which uses two springs to adjust to different phones. There is also an opening for the USB cable that connects the smartphone to an Arduino microcontroller and grooves for the optical wheel odometer sensors.

- *ELECTRICAL DESIGN:*
  - We use the breakout board as motor controller. The two left motors are connected to one output and the two right motors to the other.
  - An Arduino Nano board is connected to the smartphone via its USB port, providing a serial communication link and power.
  - The Arduino, motors, indicator LEDs, speed sensors, and an optional ultrasonic sensor are simply plugged in. When building multiple robots, the PCB further reduces setup time and cost.

1.5 OBJECTIVE

Current robots are either expensive or make significant compromises on sensory richness, computational power, and communication capabilities. We propose to leverage smartphones to equip robots with extensive sensor suites, powerful computational abilities, state-of-the-art communication channels, and access to a thriving software ecosystem. We design a small electric vehicle that costs rupees 5k and serves as a robot body for standard Android smartphones. As person following and real-time autonomous navigation in unstructured environments. Controlled experiments demonstrate that the presented approach is robust across different smartphones and robot bodies.

1.6 SCOPE OF PROJECT

- *PERSON FOLLOWING:*

  We only consider detections of the person class and reject detections with a confidence below a threshold of 50% we track detections across frames, and pick the one with the highest confidence as the target.

- *AUTONOMOUS NAVIGATION:*

  The robots have to autonomously navigate through corridors in an office building without colliding. The driving policy receives high-level guidance in the form of indicator commands such as turn left / right at the next intersection

CHAPTER 2

LITERATURE SURVEY

We present an automatic robot control system in Android platform. This work is aims at designing and developing a robust, reliable, and correct software intensive system for pursuing top level, incomparable and preeminent research objectives by consolidating several techniques and methodologies to make our system more responsive, user friendly and more communicative with the robot.

2.1 ABSTRACT

Robots are machines to execute different task repeatedly with high precision. They are used in many situation like collecting information and studies about the hazardous sites which is too risky to send human inside. The proposed design has person following and obstacle avoidance from one place to another. A robotic arm is designed using Arduino to perform pick and place the objects via Bluetooth commands. The movement of robot is controlled by android based smart phone. The robot receives commands from the application. All the actions or the tasks are controlled by Arduino micro controller with a pair of motor drivers. The robot consist of four motors, both pair for the direction control and navigation.

2.2 INTRODUCTION

Robot is a machine that performs work to assist people. It deals with the design of robots manufacture and applications. Robots consist of five major components, they are- computer which acts as the brain, effectors which are arms and legs of the robot, actuators that activate physical action and sensors that receive information about the surrounding environment and mechanical fixtures which performs overall robot hardware. Robots play very important role in the modern world as they are involved in assisting labourer’s in industry to involving in human rescue operations under critical situations. Now-a days in the modern world, industries are focusing of unmanned or a computerized devices to operate to increase productivity and delivery of the final products with greater quality in a shorter period of time. These robots can be fast, accurate and with almost zero error in performing the task. It cost less to operate than a human labour to do the same task. The proposed robot model can place objects from source to destination.

The movement of robot is controlled using android based smart phone. The android device has a Bluetooth controller application installed. Robot will operate based on the Bluetooth commands. The overall actions of motor drivers, Bluetooth controller is controlled by Arduino microcontroller.

2.3 LITERATURE SURVEY

1. Mohamed et.al. Introduced a Pick and place robotic arm controlled by Computer vision. Here the robot picks the object at a specific orientation only. The gripper used here is a mechanical gripper. So it can't handle the object safely. Objects in a specific orientation is only picked up by the robotic arm.
2. T Yoshimi et.al. Introducing a system for picking up operation of thin objects by robotic arm with two fingered parallel gripper. Thin objects like paper and plastic cards are picked up by this robotic arm. The objects may slide down due to the use of parallel gripper. This method does not provide safety of the object.
3. Anish et.al introduced design and fabrication of pick and place robotic arm to be used in library. Here the robots pick the books from library and places it in the destination. RFID tags are used to identify the books. The system is capable of doing when it is a line following robot, each RFID has its own path and this makes the robot complex.
4. N F Begum et. Al introducing an autonomous android controlled robot design using wireless energy. In this system works according to voice commands or speech delivered by the user and the robotic arm is capable of picking the objects of any type orientation. RF technology is used so line of sight is a major constraint in communication.

2.3.1 ABOUT FUTURE SCOPE

As robots can be used in lot of rescue operations, the pick and place robot can be used as a rescue robot in the natural disaster by using a suitable gripper to lift the human. The gripper should be safe enough to lift the human. The detection of human is done by placing sensor to the grippers. So once when it is identified it has to intimate concerned people through GSM either through via a recorded phone call or SMS.

2.4 CONCLUSION

The main objective of the project is to control the pick and place robot using android or smart phone. The proposed model can be used in places where the work is done repeatedly. So it is used in industrial activities and hazardous operations. It can perform the programed task easily and safely in a short span of time.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

Following are the requirement specifications required to build our project.

3.1 SOFTWARE REQUIREMENTS

- ROBOT APP
- CONTROLLER APP
- ANDROID STUDIO
- ARDUINO IDE

3.2 HARDWARE REQUIREMENTS

- ROBOT BODY
- ARDUINO NANO V3
- EXPANSION SENSOR SHIELD
- JUMPER WIRES
- L9110S DUAL CHANNEL MOTOR DRIVER
- POWER BANK
- MOBILE STAND

3.3 FUNCTIONAL REQUIREMENTS

Functional requirements are a set of specifications and actions that describe what a system, product, or software application should do to meet the desired objectives and fulfill the needs of its users.

- **Features and functionality**: Functional requirements outline the specific features and capabilities that the system should provide. For example, in an e-commerce website, functional requirements could include user registration.
- **Inputs and outputs**: Functional requirements describe the expected inputs that the system will receive and the corresponding outputs it should produce. This includes user actions, data entry, system responses, and generated reports.

- **User interactions**: Functional requirements define how users will interact with the system. This includes actions, workflows, and interfaces. For example, a social media application might have requirements related to user authentication.
- The required functionalities of our project are:
    - PERSON FOLLOWING
    - AUTONOMOUS NAVIGATION
    - OBSTACLE AVOIDANCE

3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements, also known as quality attributes or system qualities, describe the characteristics and qualities that a system should possess rather than its specific functionality. These requirements define how the system should behave, perform, and interact with its environment.

- **Performance**: Performance requirements specify the system's expected response times, throughput, and resource utilization. For example, a non-functional requirement could state that a web page must load within two seconds.
- **Reliability**: Reliability requirements define the system's ability to perform its intended functions accurately and consistently over time. This includes measures such as uptime, availability, fault tolerance, and error handling.
- **Security**: Security requirements address the system's ability to protect sensitive information, prevent unauthorized access, and ensure data integrity. This includes measures like authentication, authorization, encryption, and compliance with relevant security standards.
- The required functionalities are:

- REAL-TIME PERFORMANCE TRACKING
- THRESHOLD LEVEL CHECK
- SPEED STATS
- PATH VISUALIZATION

CHAPTER 4

SYSTEM DESIGN

The project aims in designing a robot that can be operated using Android App. The controlling of the Robot is done wirelessly through an Android smartphone one using the Bluetooth module feature. Here in the project, the android smartphone is used as a remote control for the Robot and have key applications. Bluetooth is an open standard specification for a radio frequency (RF)-based and have a short-range connectivity technology that promises to change the face of computing and wireless communication. The controlling device of the whole system is a Arduino. Bluetooth module, DC motors are interacted to the Arduino.

4.1 BASIC BLOCK DIAGRAM

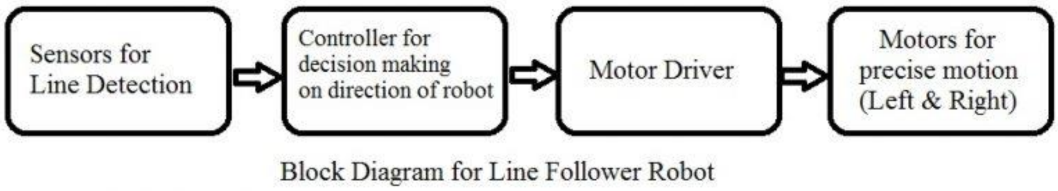


FIGURE 4.1

Sensors (IR Sensor): We have used IR Sensor Module as the line detecting sensor for the project. It consists of an IR LED and a Photo diode and some other components like comparator, LED etc.

Controller (Arduino UNO): Arduino UNO is the main controller in the project. The data from the sensors (IR Sensors) will be given to Arduino and it gives corresponding signals to the Motor Driver IC.

Motor Driver (L293D): L293D Motor Driver IC is used in this project to drive the motors of the robot. It receives signals from Arduino based on the information from the IR Sensors.

Motors (Geared Motors): We have used two geared motors at the rear of the line follower robot. These motors provide more torque than normal motors and can be used for carrying some load as well.

4.2 DEPLOYMENT DIAGRAM

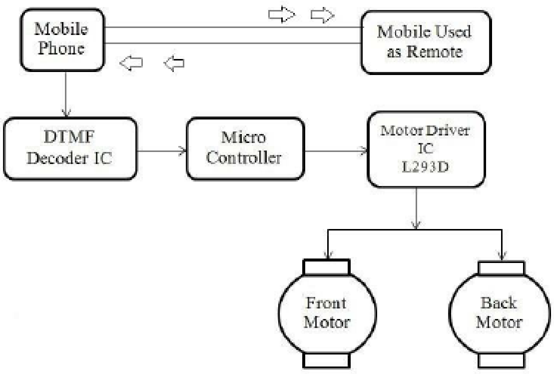


FIGURE 4.2

In this system there are some units as control unit, sensor unit, power supply unit etc. According to the suggested system, the robot is controlled by a smart phone from distance. Smart phone that is stacked on the robot will be kept in auto answer mode. After the generation of DTMF signal, decoder IC decodes the DTMF signal into its equivalent binary digit and this binary digits are dispatched to Arduino. Arduino is pre-programmed to take up decisions for the given input (Pressed button) and outputs the decision to motor driver IC. Finally, driver IC completes the command to drive the motors.

4.3 FLOWCHART

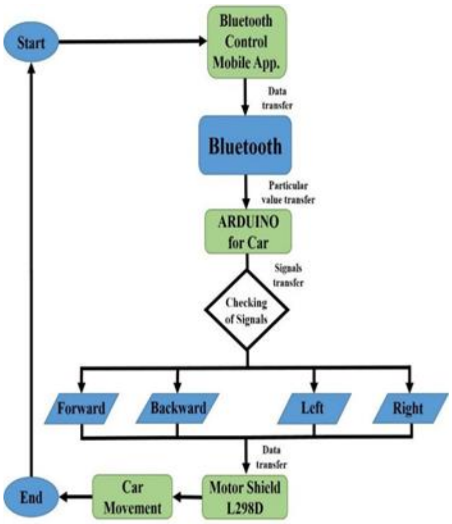


FIGURE 4.3

In this scenario, the robot car will move with the built-in Android application with different methods like touch button control (arrow and simple buttons). Firstly, with the touch buttons; after establishing the connection to the Android mobile application

with Arduino through the Bluetooth module, whenever the user presses any of the touch buttons in the application, the corresponding signal is sent to the
Arduino Uno. After receiving the signal, Arduino will check this with predefined instruction for moving forward, backward, left, right, and brake, then send the command to the motor module to move the robot in the corresponding direction.

4.4 STATE TRANSITION DIAGRAM

1. OBSTACLE AVOIDANCE

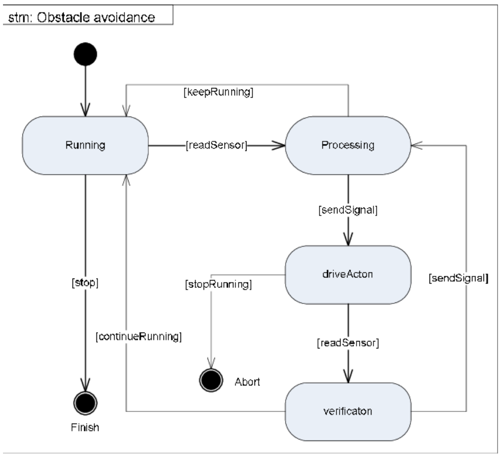


FIGURE 4.4

The aim of system design is to create overall models on extended detail level. These are cross domain as well as domain specific models - describing system states, sub-functions and structure. Models used are Activity and State Machine. Diagram describes UGV basic electronic components for drive control and feedback. Assembly inputs are power voltage, feedback momentum and external control signal carried by EMI bus (External Memory Interface). The output is PWM signal for motors and sensor signal for operator/external control device. System behaviour is described by State Machine diagram. Discrete states of a system and transactions between states with guard conditions define system behaviour in different level and situations. It is about system behaviour in obstacle avoidance situation on a high abstract level. The “Running” state is general state of normal system activity. The UGV can receive different control signals for different actions, but in this diagram all this is considered as “Running” state. When an unexpected obstacle is detected by the sensors of UGV, the system state changes to the “Processing” state where on board decision is made to carry out next action. Decision can be ether to keep running in normal mode or carry on next state “drive Acton” to avoid collision. “Verification” state verifies the results of drive action and decides following state change. This is general model of obstacle avoidance behaviour and more precise algorithms are composed for every state on following design cycles.

2. PERSON FOLLOWING:

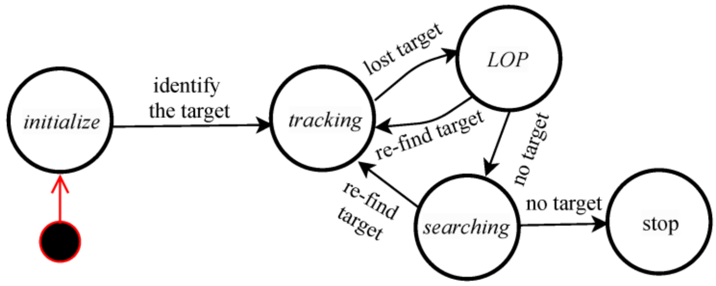


FIGURE 4.5

The proposed person following system is shown in the state-machine diagram. This diagram explains the robotic behavior, which tells all the different states of a robot to achieve efficient tracking and autonomous navigation in dynamic environments. The person tracking may sometimes fail to track the person. However, the robot must be able to recover from the failure to keep track of the person during the following. Therefore, three major states are defined, i.e., tracking, last observed position (LOP), and searching, among which the robot can change its operation during, in addition to the initialization and stop states.

INITIALIZATION STATE:

In this state, the starting position of a robot [0,0,0][0,0,0] and its quaternion [1,0,0,0][1,0,0,0] are referenced to the base coordinate axes, where the starting vector [0,0,0][0,0,0] represent [x0,y0,z0][x0,y0,z0] and the quater represents [orientation, roll, pitch, yaw]. In navigation, only 2D plane is needed, namely the orientation (θr) and coordinates (xr, yr) of the mobile robot. The starting state starts from the run moment of the code to the first frame in the tracking state.

TRACKING STATE:

Humans have several s behavioral and physical features, such as fingerprints, iris, face and voice. These features help brain processes to readily differentiate a certain person from others.

Most of these features have already been implemented in other fields like security. However, applying some of these features like iris and fingerprint in the robot field poses are some challenges such as requiring human action from very close distances with expensive and advanced devices for identifying a person. However, the identification of a person by height, width, and clothing color is possible with affordable devices. The most important part of humans is arguably the eye, which detects objects by their sizes, color, widths, and heights, among other features. Similarly, robots can use these sensors to identify objects in the world. The framework of the method of human tracking using an SSD model and a point cloud. We executed a novel architecture that involves the following steps: First, the SSD was used to frame bounded boxes for person detection via CNN in a 2D image. Second, 2D coordinates are changed to 3D coordinates in world space by reducing the point cloud to the volume of the person body based on the output of the detector for estimating distances. Meanwhile, the color feature is extracted from the region of interest-based also on the output of the SSD detector. Third, the distance and labelled colour are obtained to identify and track the target person. Finally, this information is then sent from a workstation, which publishes the people information as a talker to a robot computer, which subscribes to this information as a listener.

HUMAN DETECTION:

To enable the robot to detect humans around it, CNNs were used by the SSD model as a detector. This technique showed excellent performance in the detection of people in real-time, thus overcoming its counterparts in detection speed. The SSD uses an end-to-end training method to improve both the speed and accuracy; it is more accurate than You Only Look Once (YOLO) and faster than Faster R-CNN (regions with CNN) with some data sets. However, these techniques require pre-training data. The details of Mobile-Nets and SSD are extensive and beyond the scope of this work. Therefore, we briefly describe them here. In short, SSD achieves a balance between accuracy and speed. To detect objects using 2D RGB image sequences, in which the input size of the image is 300×300300×300, only a single feed-forward process is required, followed by a non-maximum suppression step to produce the final detection. There are several different layers producing multi-scale feature maps and many aspect ratios with multiple scales that predict object detection. Each bounding box predicts the shape of the object with confidence, where every shape is either a rectangle or a square depending on the shape of the predicted object. Therefore, every bounding box has four values in the pixel unit.

4.5 SEQUENCE DIAGRAM

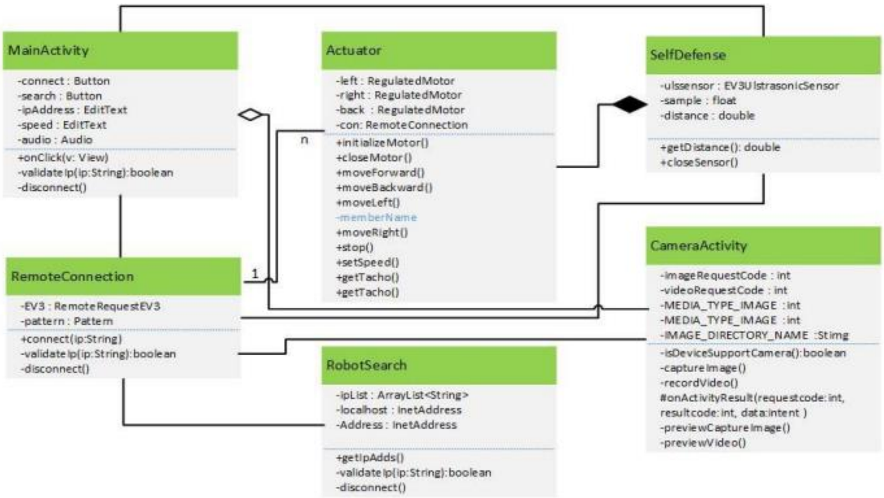


FIGURE 4.6

It shows the static structure of the Robot Control System by presenting the classes along with their properties and methods. It offers a compact view of overall implementation of the system. Moreover it provides the visibility and scope of each property of the class. In addition to this, it exhibits the dependency of one class on another by demonstrating the coherent relationship among the classes. Remote Connection, Self-defence, and Robot Search classes are associated with the Main Activity class. Besides, Main Activity is aggregated with the Camera Activity class and exposes an aggregation relationship between those two classes. The Self-defence class is composed with the Actuator class because of its entire action dependency on that class. This class diagram basically represents the dependencies and the relationship among the classes build for Robot Control System.

4.6 ACTIVITY DIAGRAM

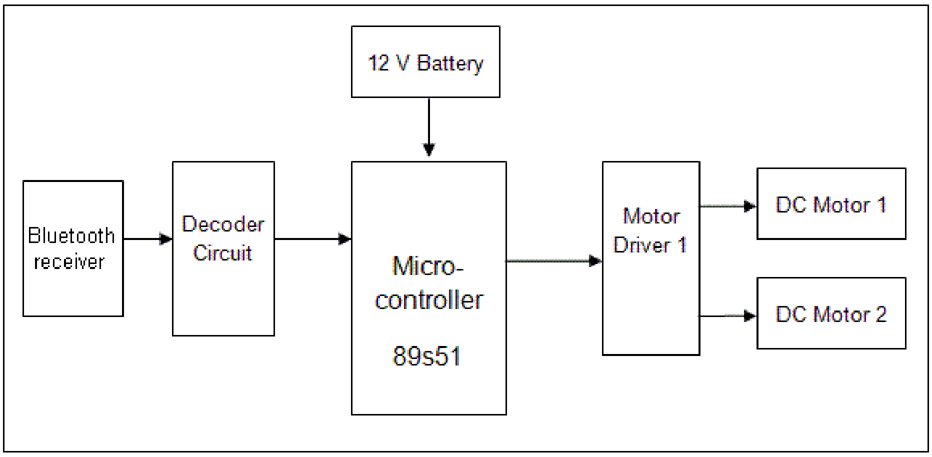


FIGURE 4.7

Circuit diagram for Bluetooth controlled Robot is shown in above figure. A Motor driver is connected to Arduino to run the Robot. Motor driver's input pins 2, 7, 10 and 15 are connected to Arduino's digital pin number 12, 11, 10 and 9 respectively. Here we have used two DC motors to driver Robot in which one motor is connected at output pin of motor driver 3 and 6 and another motor is connected at 11 and 14. A 6 volt Battery is also used to power the motor driver for driving motors. Bluetooth module's RX and TX pins are directly connected at TX and RX of Arduino. And vcc and ground pin of Bluetooth module is connected at +5 volt and ground of Arduino. And a 9 volt battery is used for power the circuit at Arduino's VIN pin.

4.7 DATAFLOW DIAGRAM

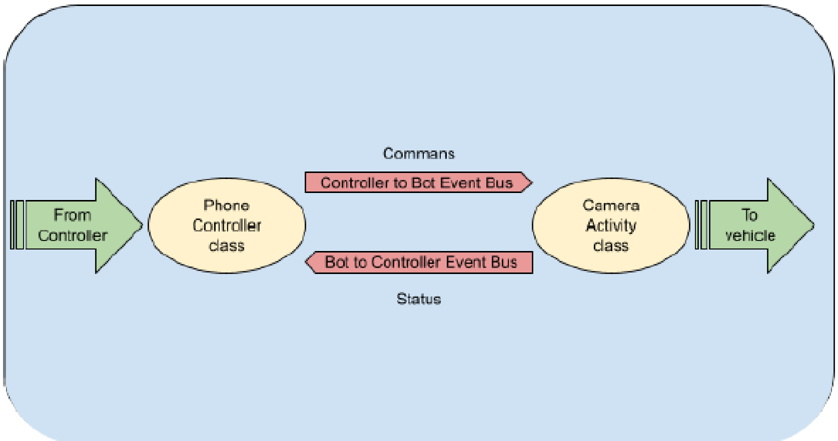


FIGURE 4.8

The robot app provides two channels of communication between the main Camera Activity class and the Phone Controller class. The Phone Controller class handles the interface to the Controller connection. In this case, it uses the Nearby Connection class to receive and send data to the remote controller. Once data is received, the Phone Controller class sends an RX event to the Camera Activity via the ControllerToBotEventBus. The ControllerToBotEventBus class handles all the details, so to send command event is really simple: ControllerToBotEventBus.*emitEvent* (new JSONObject ({command: "LOGS"})); to receive a status events from the Camera Activity class is also simple. Subscribe to status events like this: BotToControllerEventBus.*getProcessor*().subscribe(event -> handleThisEvent(event)); In our case, we simply send back the status information to the controller using the Nearby Connection class: BotToControllerEventBus.*getProcessor*().subscribe(event -> send(event)); where the send(event) functions send the data to the controller.

Note that using event buses makes it easy for other remote devices to issue commands to the robot, without the need to change any code in the Camera Activity and Network Activity classes. For example, if we are adding a web client for controlling the robot, we can create a class similar to the Phone Controller to receive commands from the web server in any format, then just create and emit an event and send it via the ControllerToBotEventBus. The rest of the app can be treated as a black box.

CHAPTER 5

IMPLEMENTATION

5.1 SOFTWARE IMPLEMENTATION

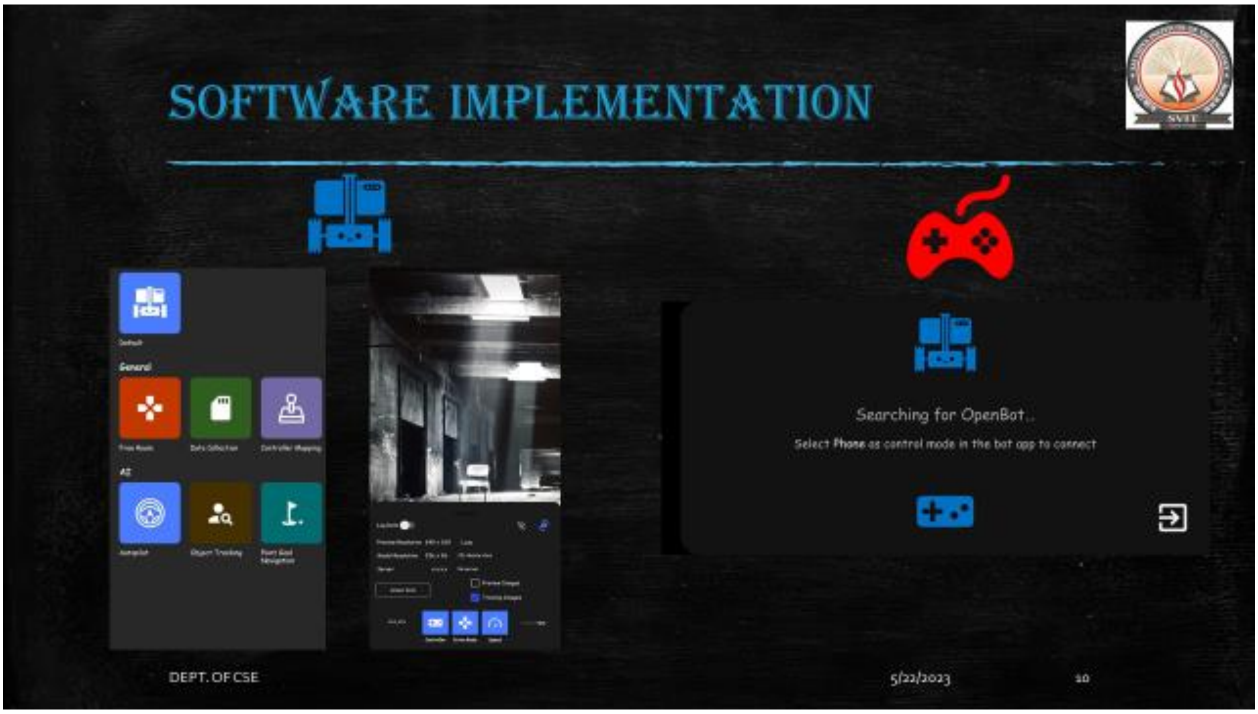


FIGURE 5.1

- We have created two apps in Android Studio IDE using Java Language.
- Using software implementation we can implement the following functions:
    - Person Following
    - Autonomous Navigation
    - Obstacle Avoidance
    - Real-Time Data Collection
    - Path Following
- For software implementation we have created two apps:
    - Robot Application
    - Controller Application
- We used Arduino IDE to flash the Arduino code into our Arduino ATMEGA382P.

5.2 HARDWARE IMPLEMENTATION

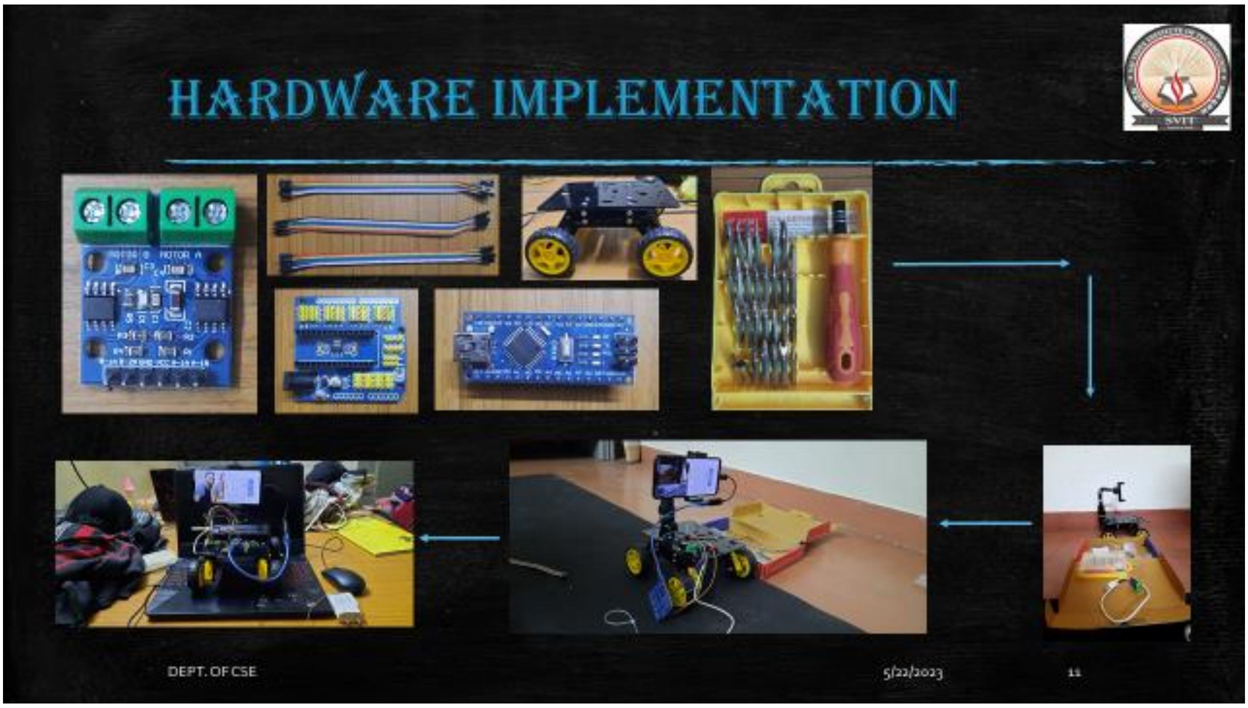


FIGURE 5.2

- For implementing the hardware part we assembled each part and connected it with each other using jumper wires
- Following is the list of Hardware used to make our robot:
    - Robot Body
    - Arduino Nano V3
    - Expansion Sensor Shield Board
    - Jumper Wires
    - L9110s Dual Channel Motor Driver
    - Power Bank
    - Mobile Stand

CHAPTER 6

CONCLUSION

Our work aims to address two key challenges in robotics: accessibility and scalability. Smartphones are becoming more powerful by the year.
We are developing a combination of hardware and software that turns smartphones into robots. The resulting robots are inexpensive but capable.
Smartphones point to many possibilities for robotics that we have not yet explored. For example, smartphones also provide a microphone, speaker, and screen, which are not commonly found on existing navigation robots.

These may enable research and applications at the junction of human robot interaction and natural language processing.
This work aims to address two key challenges in robotics: accessibility and scalability. Smartphones are ubiquitous and are becoming more powerful by the year. We have developed a combination of hardware and software that turns smartphones into robots.

The resulting robots are inexpensive but capable. Our experiments have shown that a rupees robot body powered by a smartphone is capable of person following and real-time autonomous navigation.

REFERENCES

[1] B. Balaji, S. Mallya, S. Genc, S. Gupta, L. Dirac, V. Khare, G. Roy, T. Sun, Y. Tao, B. Townsend, et al. Deep racer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning, 2019.

[2]Y. Cao, Z. Xu, F. Li, W. Zheng, K. Huo, and K. Ramani. V. ra: An in-situ visual authoring system for robot iot task planning with augmented reality, 2019.

[3]C. Delaunay. Botiful, social telepresence robot for android, botiful- telepresencerobot -for-android, Accessed: 2020-06-20.

[4] S. Consolvo et al., "Activity Sensing in the Wild: A Field Trial of Ubifit Garden," Proc. 26th Annual ACM SIGCHI Conf. Human Factors Comp. Sys., 2018.

[5] E. Miluzzo et al., "Sensing meets Mobile Social Networks: The Design, Implementation, and Evaluation of the CenceMe Application," Proc. 6th ACM SenSys, 2018.

[6] M. Mun et al., "Peir, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research," Proc. 7th ACM MobiSys, 2019.

[7] A. Thiagarajan et al., "VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones," Proc. 7th ACM SenSys, Berkeley, CA, Nov. 2019.

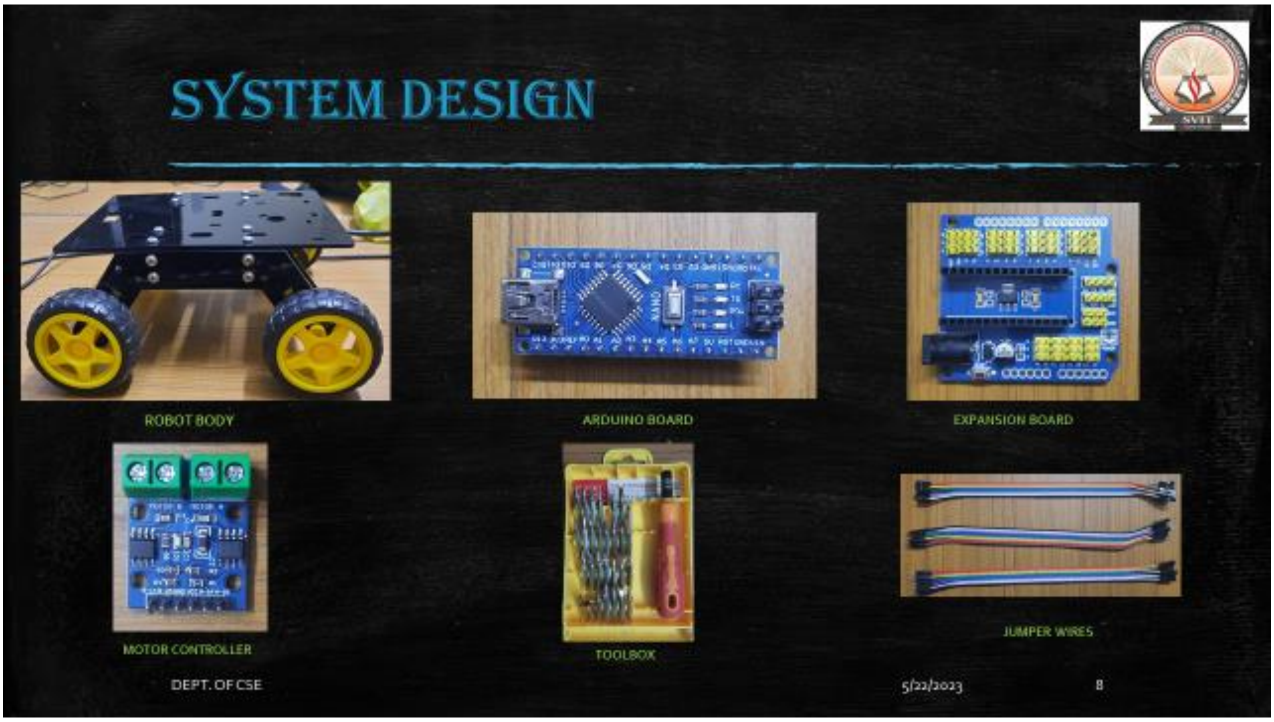
[8] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, et al. Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In ICRA, 2019.

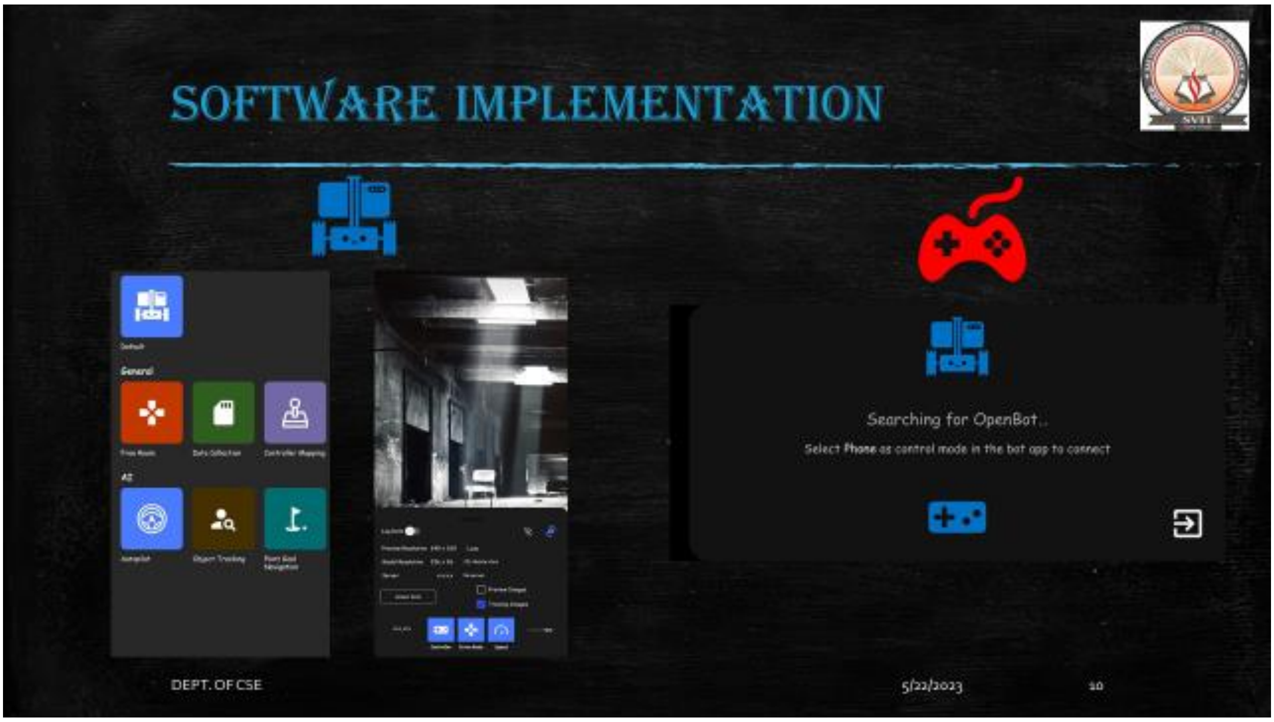
[9] F. Riedo, M. Chevalier, S. Magnenat, and F. Mondada. Thymio ii, a robot that grows wiser with children. In IEEE Workshop on Advanced Robotics and its Social Impacts, 2020.

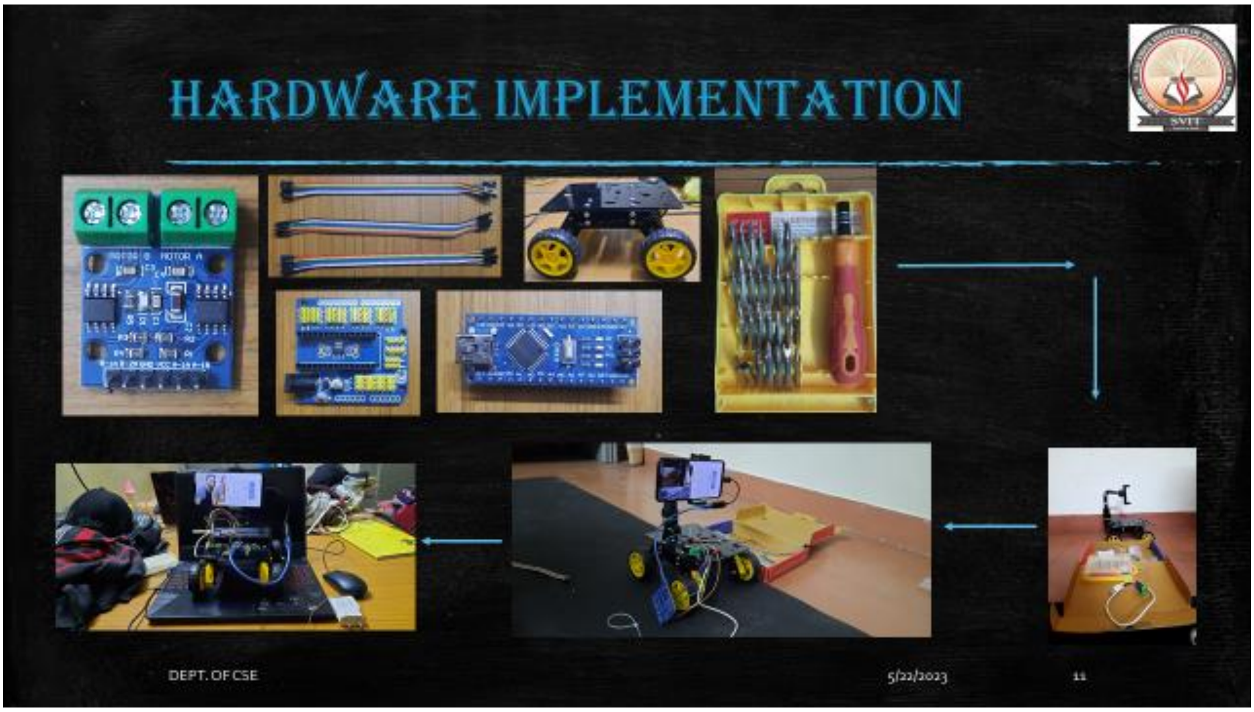
[10] Muhammad, Mirza, "WG11 Android based Surveillance Robot Control System Using Socket Programming with Implementation", 2015, International Journal Of Multidisciplinary Sciences And Engineering, Vol. 6, No. 3, March 2015

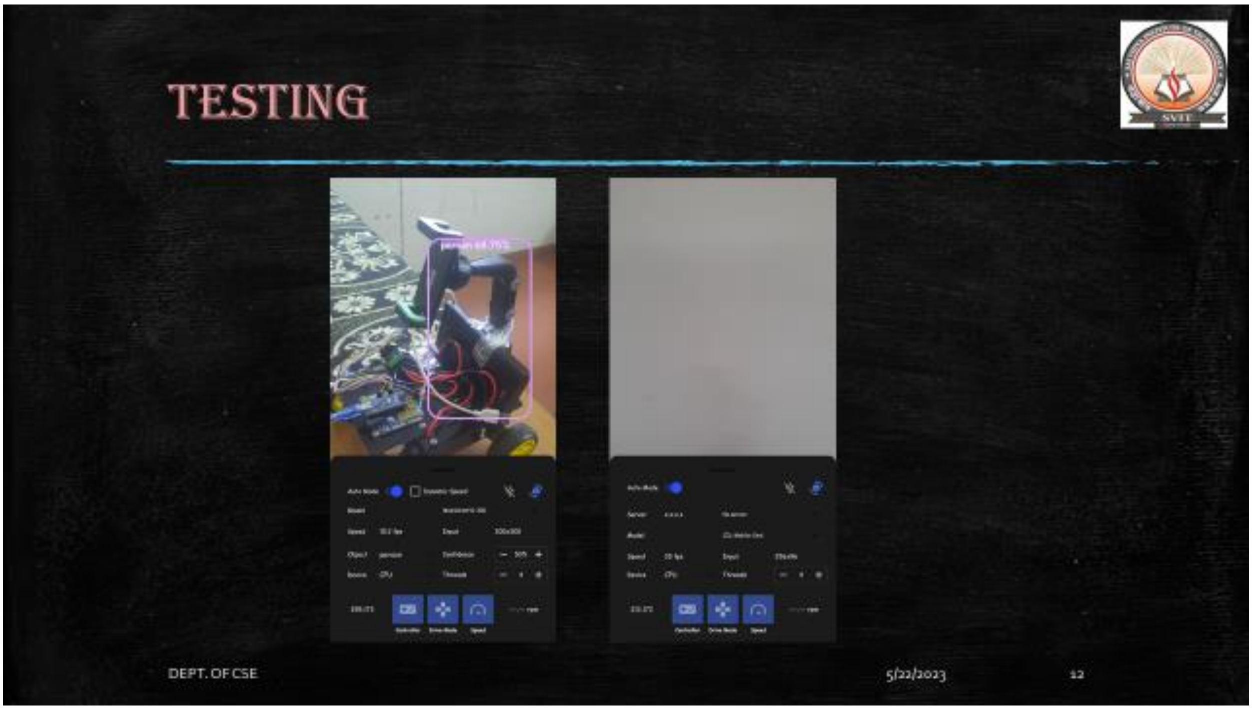
APPENDIX A

SNAPSHOTS






HARDWARE IMPLEMENTATION
DEPT. OF CSE
5/22/2023
11


TESTING
DEPT. OF CSE
5/22/2023
12

```
#elif (PHONEBOT == VEHICLE)
const String robot_type = "VEHICLE";
#define HAS_VOLTAGE_DIVIDER 0
const float VOLTAGE_DIVIDER_FACTOR = (20 + 10) / 10;
const float VOLTAGE_MIN = 2.5f;
const float VOLTAGE_LOW = 9.0f;
const float VOLTAGE_MAX = 12.6f;
#define HAS_INDICATORS 1
#define HAS_SONAR 0
#define SONAR_MEDIAN 0
#define HAS_SPEED_SENSORS_FRONT 0
#define HAS_SPEED_SENSORS_BACK 0
#define HAS_OLED 0
#define HAS_LEDS_FRONT 0
#define HAS_LEDS_BACK 0
#define HAS_LEDS_STATUS 0
const int PIN_PWM_L1 = 5;
const int PIN_PWM_L2 = 6;
const int PIN_PWM_R1 = 9;
const int PIN_PWM_R2 = 10;
const int PIN_SPEED_LF = 2;
const int PIN_SPEED_RF = 3; // 3
const int PIN_VIN = A7;
const int PIN_TRIGGER = 12;
const int PIN_ECHO = 11;
const int PIN_LED_LI = 4;
const int PIN_LED_RI = 7;

#endif
```

```
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
  super.onViewCreated(view, savedInstanceState);

  binding.confidenceValue.setText((int) (MINIMUM_CONFIDENCE_TF_OD_API * 100) + "%");

  binding.plusConfidence.setOnClickListener(
      v -> {
        String trimConfValue = binding.confidenceValue.getText().toString().trim();
        int confValue = Integer.parseInt(trimConfValue.substring(0, trimConfValue.length() - 1));
        if (confValue >= 95) return;
        confValue += 5;
        binding.confidenceValue.setText(confValue + "%");
        MINIMUM_CONFIDENCE_TF_OD_API = confValue / 100f;
      });
  binding.minusConfidence.setOnClickListener(
      v -> {
        String trimConfValue = binding.confidenceValue.getText().toString().trim();
        int confValue = Integer.parseInt(trimConfValue.substring(0, trimConfValue.length() - 1));
        if (confValue <= 5) return;
        confValue -= 5;
        binding.confidenceValue.setText(confValue + "%");
        MINIMUM_CONFIDENCE_TF_OD_API = confValue / 100f;
      });

  binding.controllerContainer.speedInfo.setText("{---,---} rpm");

  if (vehicle.getConnectionType().equals("USB")) {
    binding.usbToggle.setVisibility(View.VISIBLE);
    binding.bleToggle.setVisibility(View.GONE);
  } else if (vehicle.getConnectionType().equals("Bluetooth")) {
    binding.bleToggle.setVisibility(View.VISIBLE);
    binding.usbToggle.setVisibility(View.GONE);
  }
```

The Report is Generated by DrillBit Plagiarism Detection Software

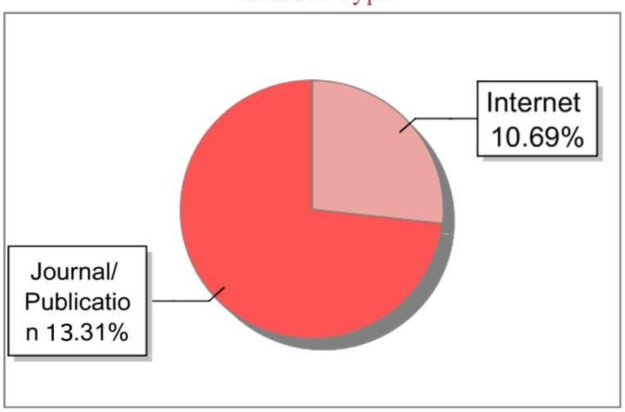
Submission Information

Author Name	ABHISHEK ANAND
Title	PHONE-BOT
Paper/Submission ID	747462
Submission Date	2023-05-20 23:47:11
Total Pages	18
Document type	Project Work

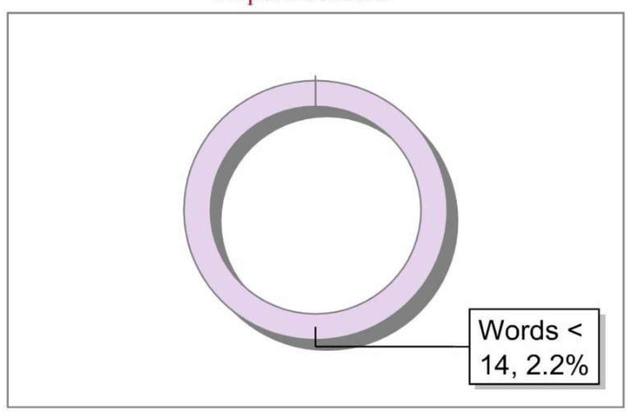
Result Information

Similarity **24%**

Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Excluded
Excluded Source	**0 %**
Excluded Phrases	Excluded

A Unique QR Code use to View/Download/Share Pdf File