Android Controlled Solar Surveillance Robot with Camera Visualization

Submitted by

Rehan Mahmud
Student ID: BME1901017132

Arup Kumar
Student ID: BME1901017141

Ratan Md Safiul Azam
Student ID: BME1901017330

Sojib Kumar Das
Student ID: BME1901017078

Md Rysul Islam
Student ID: BME1901017088

Supervised by

MD. Ali Azam
Lecturer, Department of Mechanical Engineering
Sonargaon University (SU)

September, 2022

Declaration of Authorship

Bangladesh, hereby declare that the internship report titled "**Android Controlled Solar Surveillance Robot with Camera Visualization**" is prepared aster the completion of my internship at "**Android Controlled Solar Surveillance Robot with Camera Visualization**". We also declared that the internship report is prepared for an academic purpose and has not been submitted by me before of any degree.

We declare that the internship report embodies the result of own research work, perused under the supervision of **MD. Ali Azam**, Lecturer, Department of Mechanical Engineering Sonargaon University (SU)

We Also Confirm you this report from any counterfeit and no other student either B.Sc. Or other discipline have submitted partially or whole of it.

Signed:

Rehan Mahmud

Arup Kumar

Ratan Md Safiul Azam

Sojib Kumar Das

Md Rysul Islam

Supervised by

MD. Ali Azam

Date: 14.09.2022

Abstract

The main object of this project is to develop an **Android Controlled Solar Surveillance Robot with Camera Visualization**. This technology is not only easy to use but also helps to prevent misuses of energy. Remote control (key fob) transmitter is small and very light weight, which will work from a decent distance. In order to achieve this, an android app is interfaced to the micro-controller on transmitter side which sends Left, right, front, back commands to the receiver where robot are move according to the signal of micro-controller. Arduino IDE software has been used to compile some programs related to the micro-controller ATmega328. The robotics and automation industry which is ruled the sectors from manufacturing to household entertainments. It is widely used because of its simplicity and ability to modify to meet changes of needs. The robot along with camera can witlessly transmit real time video. This is kind of robot can be helpful for spying purpose in war fields. The ESP32 camera and the ultrasonic sensor are attached for security of this system. Camera will be captured the robot surrounding view and ultrasonic sensor detect obstacle in-front of the robot.

Keyword: Arduino Pro Mini, Ultrasonic Sensor, ESP-32 Camera, Surveillance Robot.

Acknowledgement

First we started in the name of almighty Allah. This thesis is accomplished under the supervision of **MD. Ali Azam**, Lecturer, Department of Mechanical, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for this consistent guidance, encouragement, helpful suggestion, constructive criticism and endless patience through the progress of this work. The successful completion of this thesis would not have been possible without his persistent motivation and continuous guidance.

The author are also grateful to **MD. Mostofa Hossain**, Head of the Department of Mechanical Engineering and all respect teachers of the Mechanical Engineering Department for their co-operation and significant help for completing the thesis work successfully.

[Authors]

Rehan Mahmud
Student ID: BME1901017132

Arup Kumar
Student ID: BME1901017141

Ratan Md Safiul Azam
Student ID: BME1901017330

Sojib Kumar Das
Student ID: BME1901017078

Md Rysul Islam
Student ID: BME1901017088

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Energy is an important component in economic infrastructure of a country. The Sources of conventional energy is limited. They are reducing day by day. By using solar vehicle we can save conventional energy sources and control pollution. The explosion of the fossil fuel in the motor gives power to the wheels. For the **solar car**, the sun's energy is converted to electricity with the help of the **solar** cells. It is very **important** for the **car** to be very efficient since its power source is the sun and it may not be present all the time. A solar car is a solar vehicle used for land transport. Solar cars are usually run on only power from the sun, although some models will supplement that power using a battery, or use solar panels to recharge batteries or run auxiliary systems for a car that mainly uses battery power. Solar cars combine technology typically used in the aerospace, bicycle, alternative energy and automotive industries. The design of a solar vehicle is severely limited by the amount of energy input into the car. Most solar cars have been built for the purpose of solar car races. Some prototypes have been designed for public use, although no cars primarily powered by the sun are available commercially.

Solar cars depend on a solar array that uses photovoltaic cells (PV cells) to convert sunlight into electricity. Unlike solar thermal energy which converts solar energy to heat, PV cells directly convert sunlight into electricity.[1] When sunlight (photons) strike PV cells, they excite electrons and allow them to flow, creating an electric current. PV cells are made of semiconductor materials such as silicon and alloys of indium, gallium and nitrogen. Crystalline silicon is the most common material used and has an efficiency rate of 15-20%. Solar cars can accomplish this through photovoltaic cells (PVC). PVCs are the components in solar paneling that convert the sun's energy to electricity. They're made up of semiconductors, usually made of silicon, that absorb the light. The sunlight's energy then frees electrons in the semiconductors, creating a flow of electrons. That flow generates the electricity that powers the battery or the specialized car motor in solar cars. For more details about solar energy, read How Solar Cells Work.

Solar power has great potential as an energy source for many different types of residential homes and businesses. Most people know that getting energy from the sun is a "clean", environmentally friendly and renewable way to generate energy. What most people don't know, however, is that solar power is affordable for many homeowners. With the advancement of technology, number of equipment and modern household appliances increases to make life easier and comfort. Operating them manually is a tedious job and again hectic sometimes. If one can control devices like TV, fan, light or a music system with a remote from a distance place just by pressing the button, life will become simpler. Home automation is becoming very common these days as technology advances to reduce manual work. To switch on or off the devices one has to move to the switch board which is inconvenient even for an able person. If all this manual work is replaced by a single remote control even the aged and disable person can do the task like a normal person.

1.1 Objectives:

In the course of design and installation of an **Android Controlled Solar Surveillance Robot with Camera Visualization**, the following criteria are set to fulfill its objectives:

- To study of the **Android Controlled Solar Surveillance Robot with Camera Visualization.**
- To design and construct of an **Android Controlled Solar Surveillance Robot with Camera Visualization.**
- To know the Ultrasonic Sensor's testing capabilities.

1.2 Organization of Book

This project book consists of five chapter. The first chapter contains the statement of the introduction, objectives of the study and the project organization. Chapter two contains literature review in details. Chapter three describes the project theory and working principle, , working principle and shows the complete prototype of the project that we have built.. Chapter four deals with the result and discussion. In the final chapter, we have conclusion and future work of the project and also about some aspects we had to overcome while doing the project and lastly, we gave the conclusion of the book.

Chapter 2

Literature Review

The general idea of controlling a device or a system is a well-known fact in today's technology. Using a remote controller to control a single device from a short distance has been used for a long era of time. However, recently people have been progressing in technology and have been able to control large equipment or devices from the maximum possible distance. These developments have decreased the industrial cost by reducing manpower and also increased comfort, efficiency, and accuracy mentioned by Md. Shahinoor et al., [2].

WiFi technology has been used for controlling purposes along with transmitting and receiving data. Radio frequency has been used to operate this technology. However, this technology is limited to short range and may not operate when there are large obstacles between the device and the remote. On the other hand, Amey Kelkar [3] discussed that WiFi stills remains in this modern era only to serve the purpose of transmitting data over smartphones. Along with Wi-fi, there is another technology called IR system which has been used to control large devices, even a vehicle. There have been a few implemented cars based on this system. This technology involves infrared rays to stay connected between the device and remote. Hence, in this case, large obstacles easily disconnected the system, which was a very dangerous issue, especially for cars.

Moreover, this system too has range limitations as WiFi mentioned by Amey Kelkar [4]. Another approach of controlling long-distance devices was made prior to other modern technologies for home and office applications, which involved telephone line systems and personal computers. In this system, DTMF technology was introduced to send the commands to the controlled device. The systems developed by B. Koyuncy [5], Coskun and H. Ardam [6], the monitored device is not movable, and the system structure is far complicated. Nowadays, the modern networking system GSM is being used to control devices where an operator's cell phone could be used as a remote controller. This paper is all about proposing a simpler design to control a vehicle via the GSM network.

The following section delivers an overview of all related approach in this project and additionally, proposes a simple design to implement. In one of the papers done by Sourangsu Banerji, [7], the design of the remote controlling car has been proposed using the GSM network and cell phone. In this design, a microcontroller has been interfaced with DTMF decoder and a motor driver, which is connected to the car wheels. DTMF system of this project helps the operator to input the command by pressing the keypad of the cell phone. This project overcomes the distance limitations problem and can be operated from anywhere in the world. Though microcontroller requires programming, it is still used in this system as various applications can be added to the system. On the other hand, the DTMF system complexes the circuitry because it needs an extra filter to separate the frequencies.

Another project was done by Arab Fakih, [8]. The design pattern is almost the same as the first one. The only difference is that, here, two wheels have been used instead of four. Besides, this paper offers various applications to add to the system to make the car more efficient. A few future scope also has been added to improve the design. The paper which was done by T.T. Oladimeji, [9] proposed a design to prevent a car from being hijacked. This system uses the GSM network along with operators' cell phones to avoid stealing. The design involves the GSM module, interface box and vehicle sound system. Although the purpose of this system is not to control the car, it ensures a strong and reliable security system of the vehicle. In a paper done by Amey Kelkar, [10], presented a simple design using GSM and cell phone interfacing with a microcontroller. In this design, an Arduino microcontroller has been used to receive the input from the DTMF decoder and drive the motor driver accordingly. The advantage of using Arduino is that various shields like a prototype, Wifi and Bluetooth shields are added in this microcontroller, which makes it more convenient.

This design also proposes various applications to implement in the car. The project was done by Pathik, et al., [11], a controlling design using GSM and DTMF decoder has been proposed. Unlike other designs, this design does not involve any microcontroller. This design is only to control the car and overcome distance limitations. In this project, instead of a microcontroller, a four relay circuit has been used through which the commands from the DTMF decoder will pass and drive the motor accordingly. Since no microcontroller has been used, no other application other than controlling the vehicle can be implemented using this design. Moreover, this design introduced solar energy as the power source of the car battery.

In the following research paper, a Vehicle Security and Entertainment System was developed using Raspberry Pi Tabassum, et al., [12] to monitor, track the vehicle, and to offer local entertainment system. Two embedded devices were used to split the entertainment system from the security system to provide isolation and safety. They developed a low-cost passenger safety system for vehicles to use in passenger buses, trains, and even cars. The development was economical and additional modules can be added. When a vehicle is stolen, the device will give an alert signal, through tracking device and camera, live data can be obtained from the vehicle to ensure the passenger's safety.

Chapter 3

Theory & Working Principle

3.1 System Description

In this system here we use a Arduino Nano Pro mini microcontroller for controlling this whole system. This system we use ultrasonic sensor to detect the car obstacle object and use a ESP-32 Camera to view surrounding view in-front of the car. Arduino Nano Micro-controller can get signal from sensor and camera and user can control the car with the help of Android mobile apps.

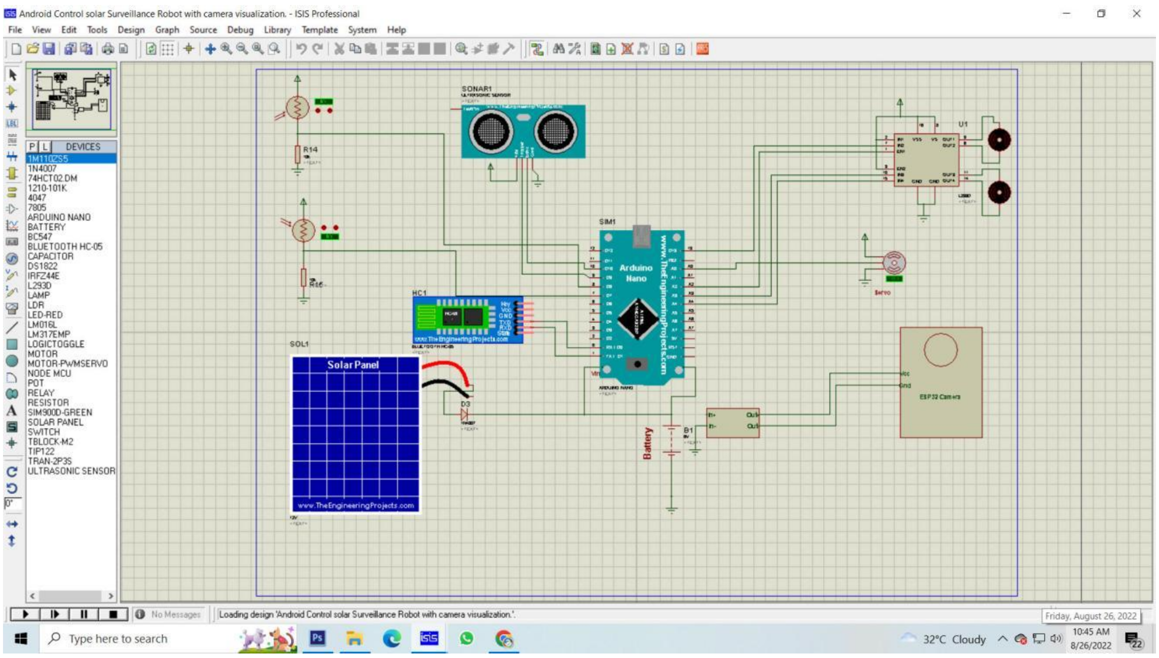

Figure 3.1: System Structure Circuit Design in Proteus

3.2 Block Diagram

In our project, we have set up an **Android Controlled Solar Surveillance Robot with Camera Visualization**. Here ultrasonic sensor will able to detect in-front of car obstacle and view the robot in-front of view with the help of ESP-32 Camera. Here the main power supply is DC 5 Volt from battery. In this circuit we have used an Arduino Nano Pro Mini for controlling this system.

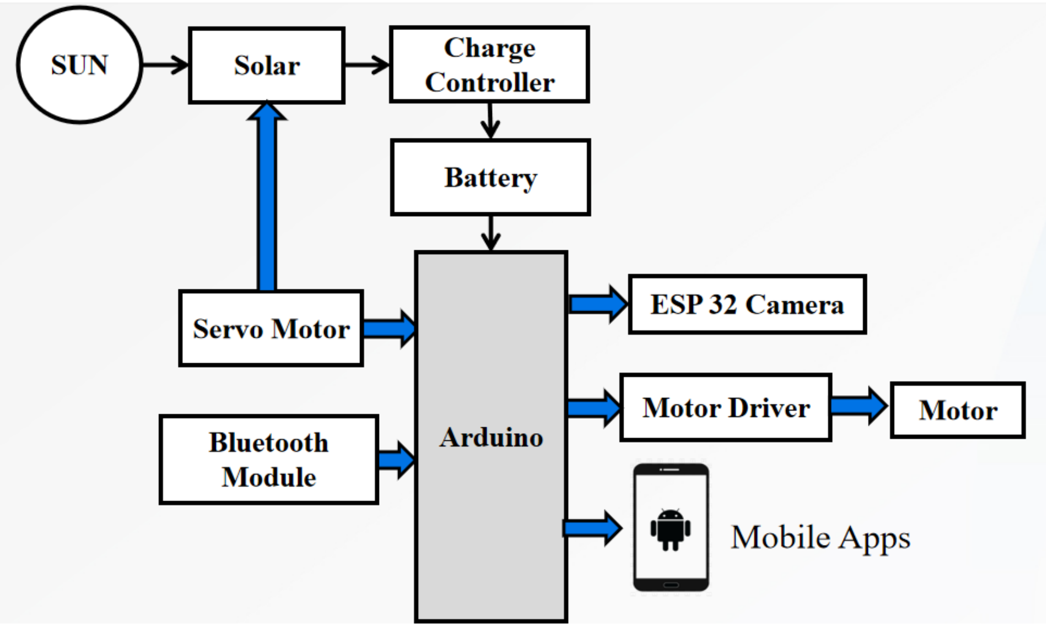

Figure 3.2: Block Diagram of the **Android Controlled Solar Surveillance Robot with Camera Visualization.**

3.3 Circuit Diagram

Arduino Nano Pro Mini has been used as main controller for this project. Here we have connected all of our instrument through wire.

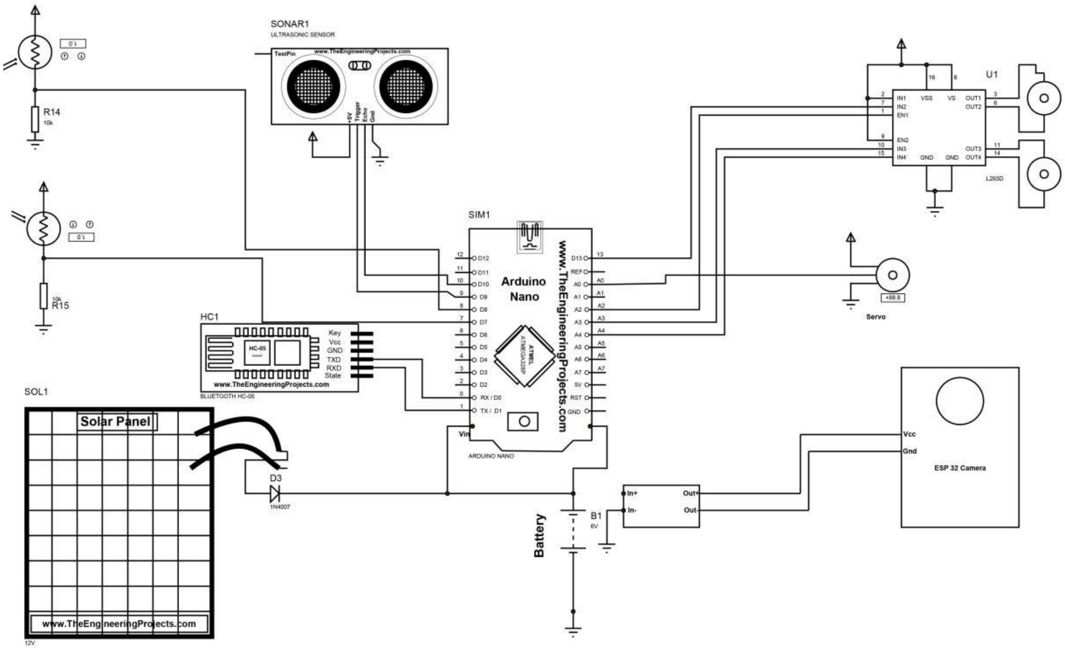

Figure 3.3: Circuit diagram of **Android Controlled Solar Surveillance Robot with Camera Visualization.**

3.4 Working Principle

Android apps Bluetooth controlled solar panel based robotic vehicle can be used for observing an area and can be used for security purposes. In this solar panel project, **Android Controlled Solar Surveillance Robot with Camera Visualization**. Ultrasonic sensor is mounted the front of the robot for detect obstacle in-front of the robot. This robotic vehicle movement can be controlled using android apps technology for remote operation. With the help of these android apps buttons, the receiver is able to receive commands. These commands that are sent are used to control the movement of the robot which gives instructions for either to move the robot forward, backward, left or right etc. It uses the Atmega 328 series of a micro-controller to achieve its desired operation. This robot car has a camera that can be used as security surveillance and solar panel for charging the battery. The wireless camera will be streaming live on web browser.

3.5 Complete Project Prototype Image :

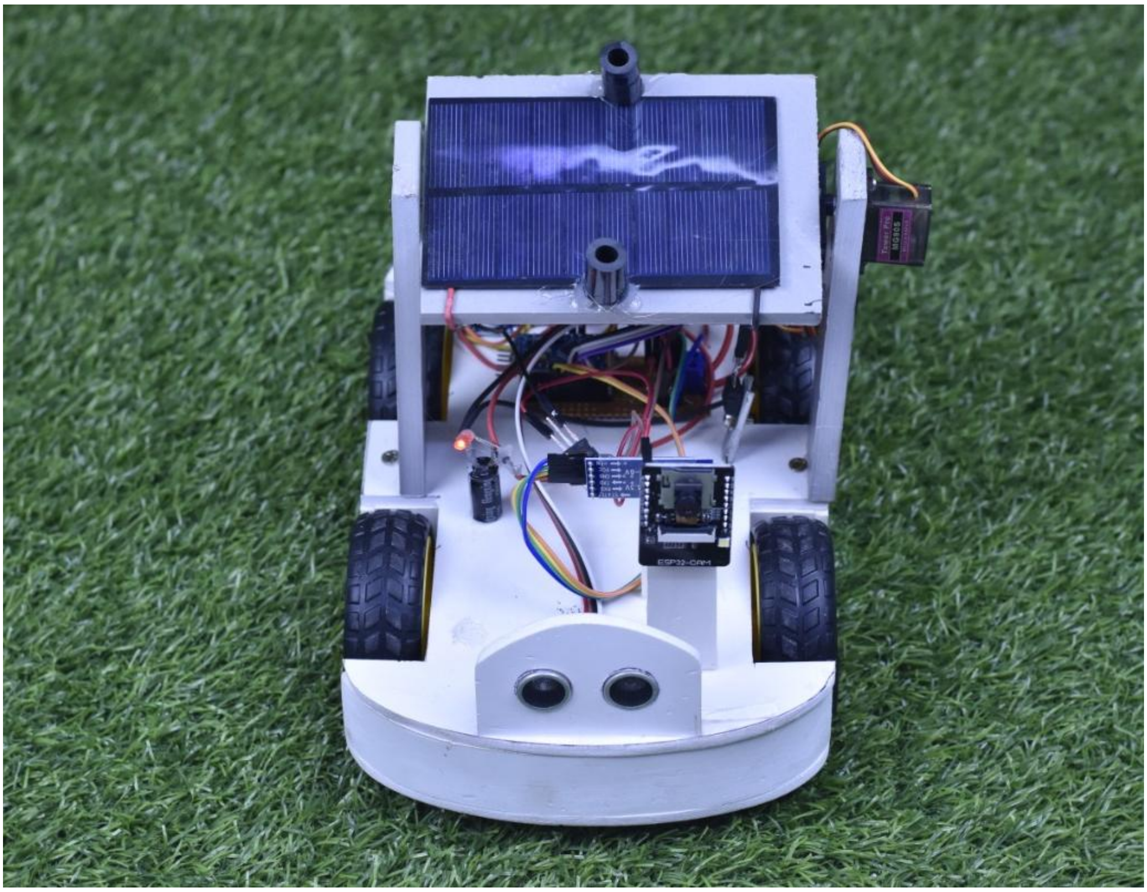

Figure 3.4: Project Prototype Image (Front View)

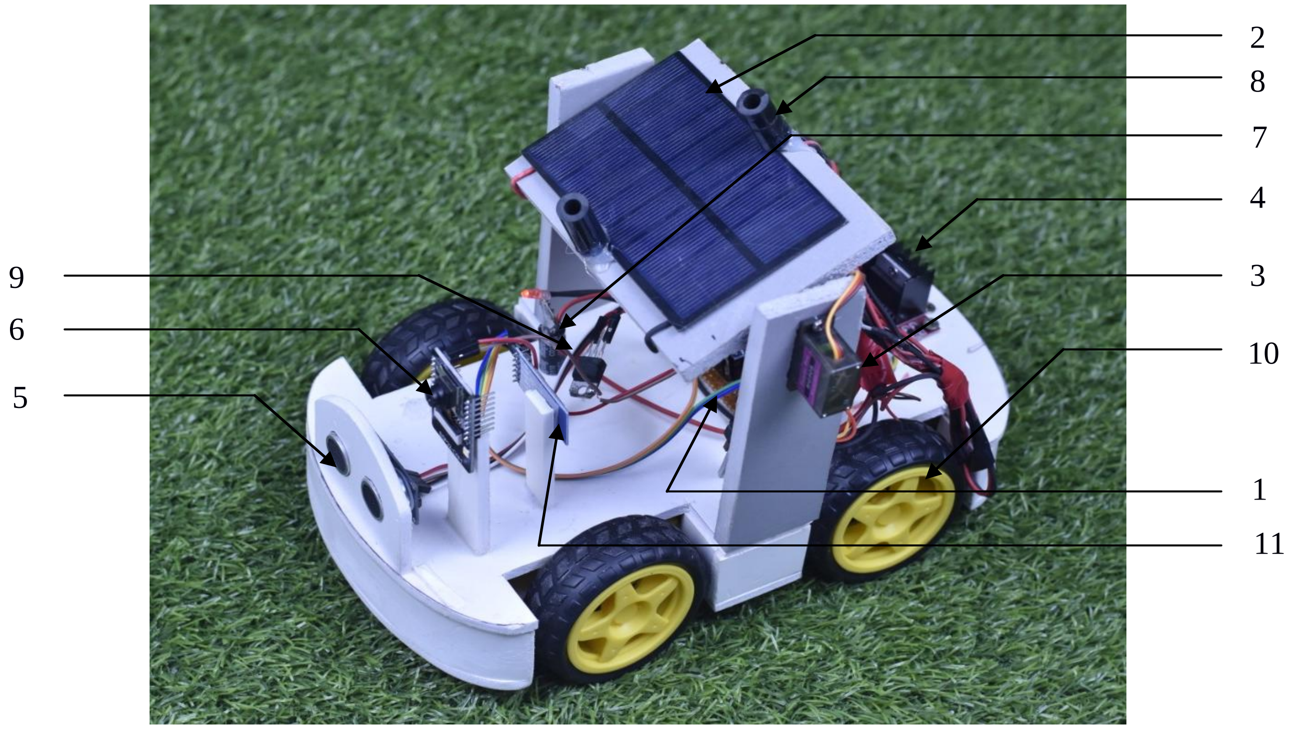

Figure 3.5: Project Prototype Image (Side View)

3.6 Components List:

1. Arduino Nano
2. Solar Panel
3. Servo Motor
4. Motor Driver
5. Ultrasonic Sensor
6. ESP-32 Camera
7. Battery
8. LDR Sensor
9. Voltage Regulator IC
10. Wheel
11. Bluetooth Module

Internally we used these components

1. Gear Motor
2. Resistor
3. Capacitor

3.7 Arduino Pro Mini

The Arduino Pro Mini is intended for **semi-permanent installation in objects or exhibitions**. The board comes without pre-mounted headers, allowing the use of various types of connectors or direct soldering of wires. The pin layout is compatible with the Arduino Mini. There are two version of the Pro Mini.

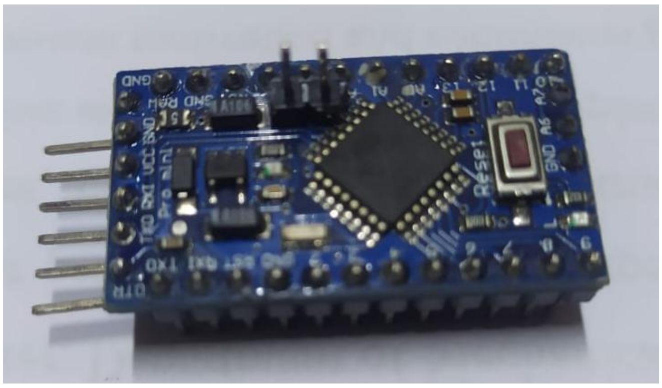

Figure 3.6: Arduino Pro Mini

Overview

The **Arduino Pro Mini** is a microcontroller board based on the ATmega328P.It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. A six pin header can be connected to an FTDI cable or Sparkfun breakout board to provide USB power and communication to the board.The Arduino Pro Mini is intended for semi-permanent installation in objects or exhibitions. The board comes without pre-mounted headers, allowing the use of various types of connectors or direct soldering of wires. The pin layout is compatible with the Arduino Mini.

Summary

- Microcontroller ATmega168
- Operating Voltage: 3.3V or 5V (depending on model)
- Input Voltage: 3.35 -12 V (3.3V model) or 5 - 12 V (5V model)
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6

- DC Current per I/O Pin: 40 mA
- Flash Memory: 16 KB (of which 2 KB used by bootloader)
- SRAM: 1 KB
- EEPROM: 512 bytes
- Clock Speed: 8 MHz (3.3V model) or 16 MHz (5V model)

Power

The Arduino Pro Mini can be powered with an FTDI cable or breakout board connected to its six pin header, or with a regulated 3.3V or 5V supply (depending on the model) on the Vcc pin or an unregulated supply on the RAW pin.

The power pins are as follows:

- RAW. For supplying a raw (unregulated) voltage to the board.
- VCC. The regulated 3.3 or 5 volt supply.
- GND. Ground pins.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader). It has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library).

Input Output

Each of the 14 digital pins on the Pro Mini can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 3.3 or 5 volts (depending on the model). Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the TX-0 and RX-1 pins of the six pin header.

- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Pro Mini has 6 analog inputs, each of which provide 10 bits if

resolution (i.e. 1024 different values). Four of them are on the headers on the edge of the board; two (inputs 4 and 5) on holes in the interior of the board. The analog inputs measure from ground to VCC. Additionally, some pins have specialized functionality:

- I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library.

There is another pin on the board:

- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

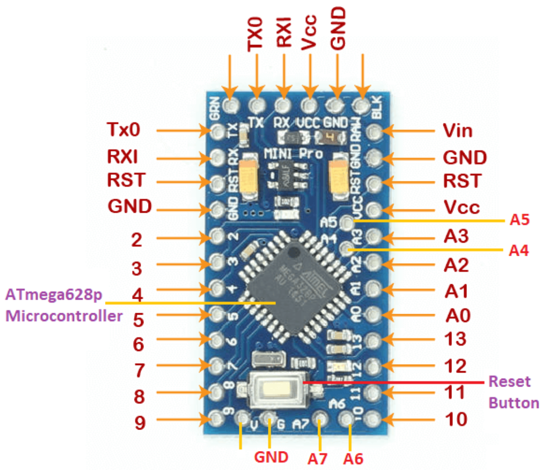

Figure 3.7: Micro Controller pin out

Communication

The Arduino Pro Mini has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 provides UART TTL serial communication, which is available on digital pins 0 (RX) and 1 (TX). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board via a USB connection.

A Software Serial library allows for serial communication on any of the Pro Mini's digital pins.The ATmega168 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the reference for details. To use the SPI communication, please see the ATmega168 datasheet.

Programming

The Arduino Pro Mini can be programmed with the Arduino software (download). For details, see the reference and tutorials. The ATmega168 on the Arduino Pro Mini comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the ATmega168 with an external programmer; see these instructions for details

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Pro Mini is designed in a way that allows it to be reset by software running on a connected computer. One of the pins on the six-pin header is connected to the reset line of the ATmega168 via a 100 nanofarad capacitor. This pin connects to one of the hardware flow control lines of the USB-to-serial convertor connected to the header: RTS when using an FTDI cable, DTR when using the Sparkfun breakout board. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of the reset line can be well-coordinated with the start of the upload.

This setup has other implications. When the Pro Mini is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Pro. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

3.8 HC-05 Bluetooth Module

Bluetooth is a wireless technology standard for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves in the industrial, scientific and medical radio bands, from 2.400 to 2.485 GHz, and building personal area networks.

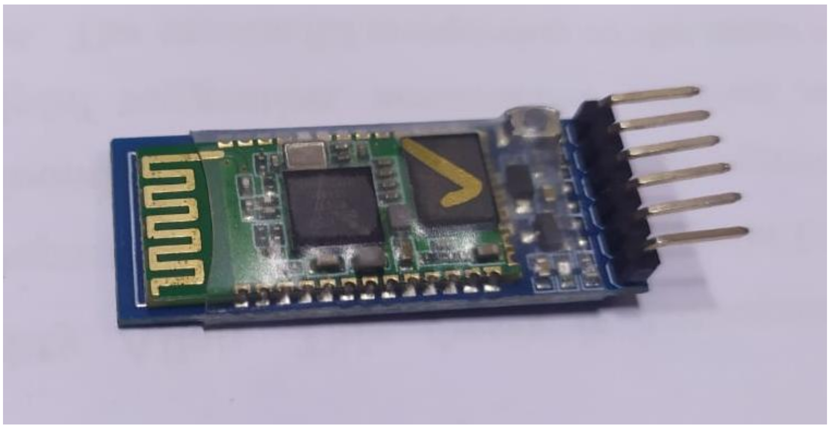

Figure 3.8: Bluetooth Module HC-05

HC-05 Technical Specifications

- Serial Bluetooth module for Arduino and other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible

- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

Applications

The HC-05 is a very cool module that can add two-way (full dual) wireless functionality to your projects. You can use this module to communicate between two microcontrollers, such as Arduino, or to communicate with any device that has Bluetooth functionality, such as a phone or laptop. We can also configure module defaults using command mode. So if you are looking for a wireless module capable of transferring data from your computer or mobile phone to microcontroller or vice versa, then this module may be the right choice for you. But don't expect this module to transfer multimedia like pictures or songs; For this you need to look for the CSR 8645 module.

1. Wireless communication between two microcontrollers
2. Communicate with Laptop, Desktops and mobile phones
3. Data Logging application
4. Consumer applications
5. Wireless Robots
6. Home Automation

How to Use the HC-05 Bluetooth Module

The HC-05 has two operating modes, one is the data mode which allows it to send and receive data from other Bluetooth devices and the other is the command mode where the default device settings can be changed. We can operate the device in either of these two modes using the PIN key described in the PIN description. Connecting the HC-05 module to micro-controllers is very easy because it operates using the Serial Port Protocol (SPP).

Power the module with only + 5V and connect the RX pin of the module to the TX of the MCU and the TX pin of the module to the RX of the MCU as shown in the figure below.

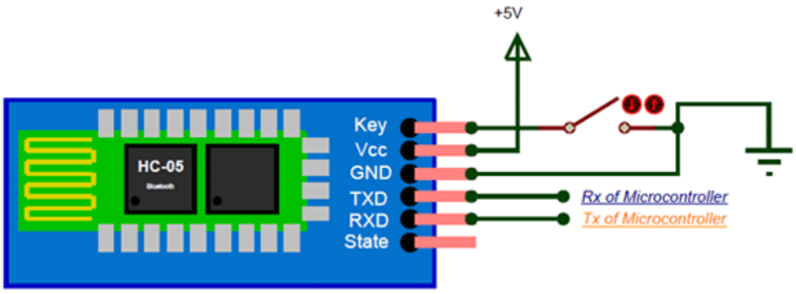

Figure: 3.9 HC-05 Bluetooth Module

During power-up, the key pin can be used to enter the command mode. If this pin is not used, it will enter the data mode by default. After the input module is activated, you should be able to detect the Bluetooth device as "HC-05" and then as the default password. Use 1234 to connect and start communicating with it.

Pin Configuration of HC-05 Bluetooth Module

Pin Number	**Pin Name**	**Description**
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth

6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module • Blink once in 2 sec: Module has entered Command Mode • Repeated Blinking: Waiting for connection in Data Mode • Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

3.9 Solar Panel

A solar panel is a set of solar photovoltaic modules electrically connected and mounted on a supporting structure. A photovoltaic module is a packaged, connected assembly of solar cells. The solar panel can be used as a component of a larger photovoltaic system to generate and supply electricity in commercial and residential applications. Each module is rated by its DC output power under standard test conditions (STC), and typically ranges from 100 to 320 watts. The efficiency of a module determines the area of a module given the same rated output - an 8% efficient 230 watt module will have twice the area of a 16% efficient 230 watt module. A single solar module can produce only a limited amount of power; most installations contain multiple modules. A photovoltaic system typically includes a panel or an array of solar modules, an inverter, and sometimes a battery and/or solar tracker and interconnection wiring.

Figure 3.10: Solar Panel

Solar cell modules produce electricity only when the sun is shining. They do not store energy, therefore to ensure flow of electricity when the sun is not shinning, it is necessary to store some of the energy produced. The most obvious solution is to use batteries, which chemically store electric energy. Batteries are groups of electro chemical cells (devices that convert chemical energy to electrical energy) connected in series.

Specifications of Solar Panel

- Size: 4.4" x 5.4" / 110mm x 140mm
- Weight: 3 ounces / 90 grams
- Cell type: Mono crystal line
- Cell efficiency: 19%+
- 2.27 Watts Peak Power
- Technical drawing

Dimensions:

- Length: 111.86mm/4.40in
- Width: 135.83mm/5.34in
- Thickness (without screws): 4.72mm/0.18in
- Thickness (with screws): 9.82mm/0.38in

3.10 Servo Motor

A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft. The motor is paired with some type of position encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

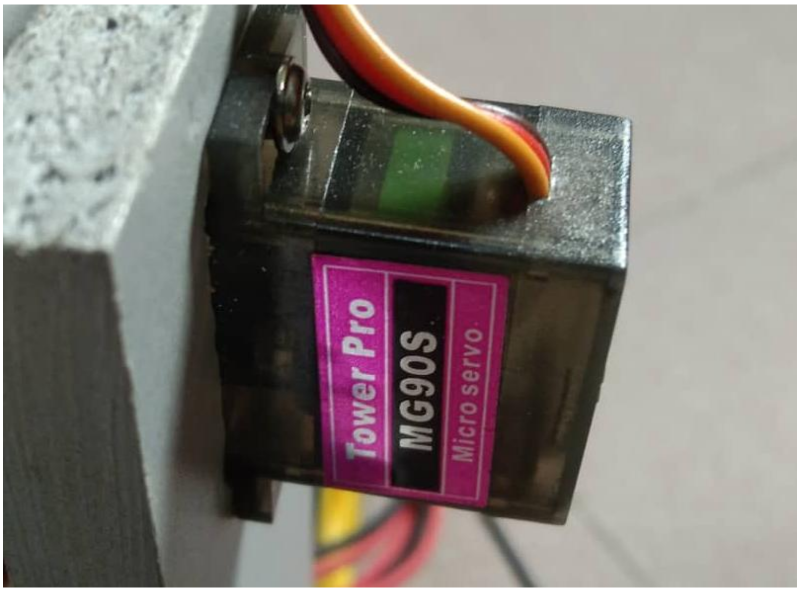

Figure 3.11: Servo Motor

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped). This type of servomotor is not widely used in industrial motion control, but it forms the basis of the simple and cheap servos used for radio-controlled models. More sophisticated servomotors use optical rotary encoders to measure the speed of the output shaft and a variable-speed drive to control the motor speed. Both of these enhancements, usually in combination with a PID control algorithm, allow the servomotor to be brought to its commanded position more quickly and more precisely, with less overshooting.

3.11 LDR Sensor

The LDR Sensor Module is used to detect the presence of light / measuring the intensity of light. The output of the module goes high in the presence of light and it becomes low in the absence of light. The sensitivity of the signal detection can be adjusted using potentiometer.

Features:

- Adjustable sensitivity (via blue digital potentiometer adjustment)
- Operating voltage 3.3V-5V
- Output Type: Analog voltage output -A0
- Digital switching outputs (0 and 1) -D0
- With fixed bolt hole for easy installation
- Small board PCB size: 3cm * 1.6cm
- Power indicator (red) and the digital switch output indicator (green)
- Using LM393 comparator chip, stable

Pin outs:

- External 3.3V-5V VCC
- External GND GND
- DO digital output interface, a small plate (0 and 1)
- AO small board analog output interface
- Can detect ambient brightness and light intensity

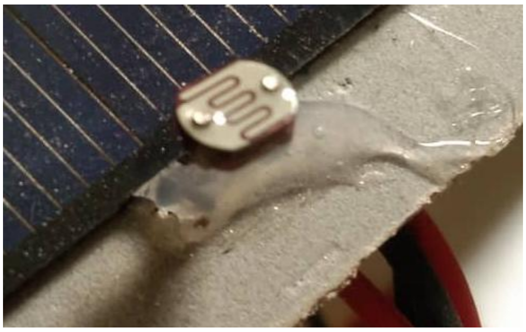

Figure 3.12: LDR Sensor

3.12 Battery

A lithium battery is the first battery to use a metal lithium anode. This type of battery is also called a lithium metal battery. They differ from other batteries in charging density and high unit cost.

Figure 3.13: 3.7V Battery

Product Specification

Output Voltage	3.7 V
Product Quality	Lithium-ion
Battery Capacity	2200mAh
Weight	45g
Model Number	ICR 18650

3.13 DC Gear Motor

Gear motors are mechanisms that adjust the speed of electric motors, leading them to operate at a certain speed. They are composed of a series of gears that make up a kinematic chain, working on a set of rotary parts.

Their main purpose is to allow the reduction from an initial high speed to a lower one without negatively affecting the mechanism. In addition to this adjustment, a gear motor is in charge of adjusting the mechanical power of a system.

Figure 3.14: Gear Motor

Mechanical components that complement gear motors

Gear motors are composed of an electric motor and gears, which form the kinematic chain – the fundamental component of the gear ratio.

Kinematic chain

A motor's speed reducer is composed of a speed reducer and its gears. This speed reducer is basically a variable speed drive that allows for the speed to be reduced and increased at the output shaft.

Gears

Gears are toothed wheels made of metal or plastic (and new materials with each passing day) that transmit motion when meshing with each other. They are defined by their number of teeth and their size. In addition, they may have straight-cut or helical teeth.

Motors

The five types of motors that see the most use in gear motors are:

- Brushed motors, with brushes normally made out of carbon. They are bidirectional and may be used with DC or AC. They have a service life of about 3000 hours.
- Asynchronous motors, which are brushless single-direction motors. They are highly limited.
- Synchronous brushless motors, which may be single-direction or bidirectional. They have a constant speed if the frequency of the power source is stable.
- Brushless DC motors that use a driver and can attain high speeds.
- Stepper DC brushless motors. They can be positioned with an average precision of 7.5º.

3.14 L293D Motor Driver IC

The **L293D** is a popular 16-Pin **Motor Driver** IC. As the name suggests it is mainly used to drive **motors**. A single **L293D** IC is capable of running two DC **motors** at the same time; also the direction of these two **motors** can be controlled independently.

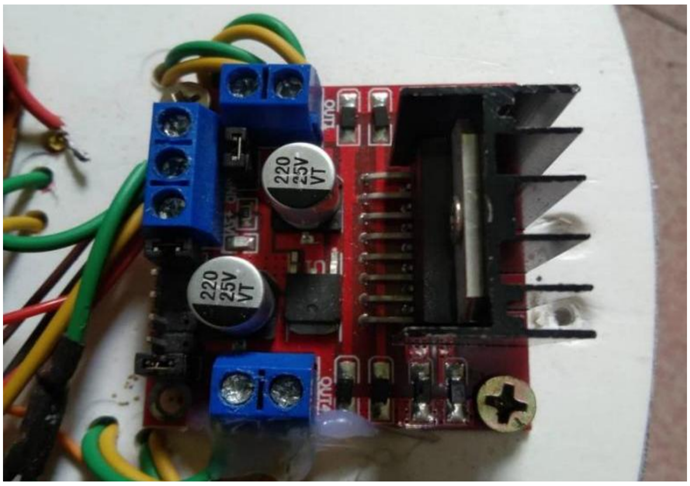

Figure 3.15: Motor driver IC L293D

Working Process:

L293D IC is a typical **Motor Driver** IC which allows the DC **motor** to drive on any direction. This IC consists of 16-pins which are used to control a set of two DC **motors** instantaneously in any direction. It means, by using a **L293D** IC we can control two DC **motors**.

Features

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available

L293D Pin Configuration

Pin Number	**Pin Name**	**Description**
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)

9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

Use of a L293D Motor Driver IC:

Using this L293D motor driver IC is very simple. The IC works on the principle of **Half H-Bridge**, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.

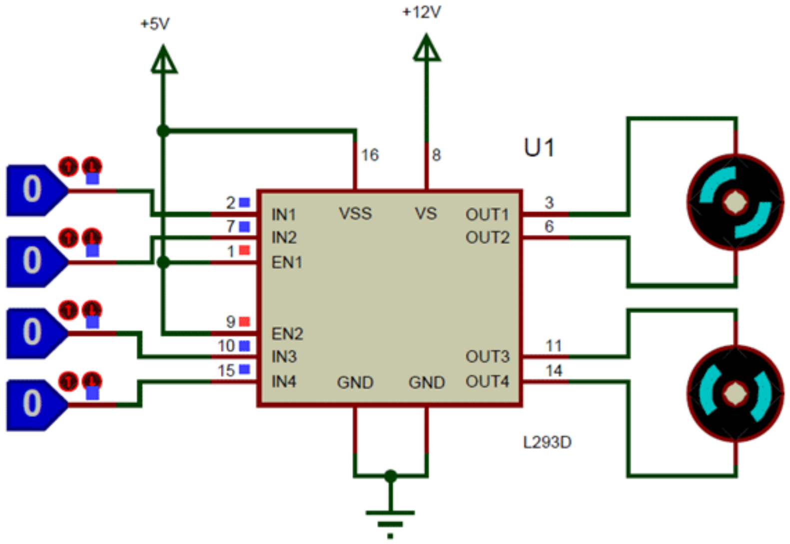

Figure 3.16: L293D circuit Diagram

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor.

Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

3.15 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below-

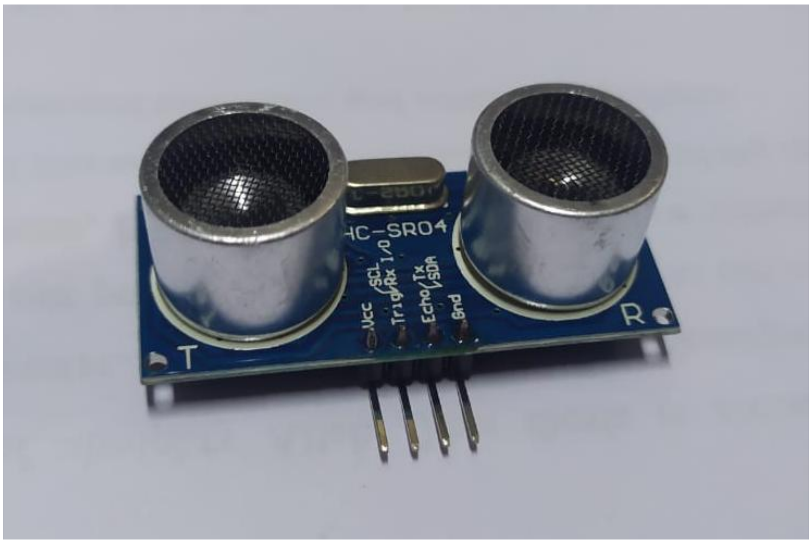

Figure 3.17: Ultrasonic Sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave, we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used. Power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor. The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information, the distance is measured as explained in the above heading.

Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

Ultrasonic Sensor Pin Configuration

Pin Number	**Pin Name**	**Description**
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

3.16 ESP32- CAM

The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera and microSD card socket. It’s inexpensive and easy to use, and is perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition.

Figure 3.18: ESP-32 CAM Pin out diagram

ESP32-CAM Specifications

The ESP32-CAM is based upon the ESP32-S module, so it shares the same specifications. It has the following features:

- 802.11b/g/n Wi-Fi
- Bluetooth 4.2 with BLE
- UART, SPI, I2C and PWM interfaces

- Clock speed up to 160 MHz
- Computing power up to 600 DMIPS
- 520 KB SRAM plus 4 MB PSRAM
- Supports WiFi Image Upload
- Multiple Sleep modes
- Firmware Over the Air (FOTA) upgrades possible
- 9 GPIO ports
- Built-in Flash LED

3.17 5V Regulator IC

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

7805 IC Rating

- Input voltage range 7V- 35V
- Current rating Ic = 1A
- Output voltage range VMax=5.2V ,VMin=4.8V

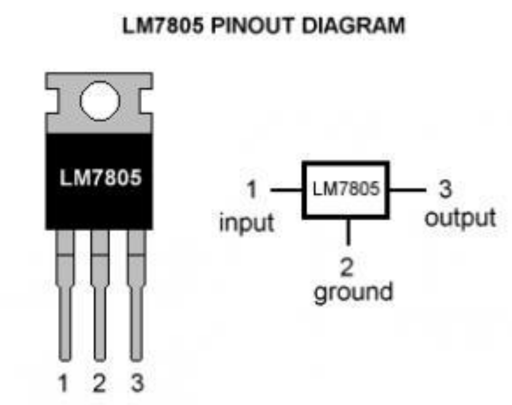

Figure 3.19: 5V Regulator IC

3.18 Resistor

A **resistor** is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

Two typical schematic diagram symbols are as follows:

(a) resistor, (b) rheostat (variable resistor), and (c) potentiometer

IEC resistor symbol

One common scheme is the RKM code following IEC 60062. It avoids using a decimal separator and replaces the decimal separator with a letter loosely associated with SI prefixes corresponding with the part's resistance. For example, *8K2* as part marking code, in a circuit diagram or in a bill of materials (BOM) indicates a resistor value of 8.2 kΩ. Additional zeros imply a tighter tolerance, for example *15M0* for three significant digits. When the value can be expressed without the need for a prefix (that is, multiplicator 1), an "R" is used instead of the decimal separator. For example, *1R2* indicates 1.2 Ω, and *18R* indicates 18 Ω.

Specifications:

- Resistance: 220 Ohms
- Power (Watts): 0.25W, 1/4W
- Temperature Coefficient: 350ppm/Celcius
- Tolerance: +/- 5%
- Case: Axial
- Size: 0.094" Dia x 0.248" L (2.40mm x 6.30mm)

3.19 Arduino Software

The smart microcontroller unit named as Arduino Nano can be programmed with the Arduino software. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the microcontroller on your board). The IC used named as ATmega328 on the Arduino Nano comes pre burned with a boot loader that allows you to upload new code to it without the use of an external hardware programme.

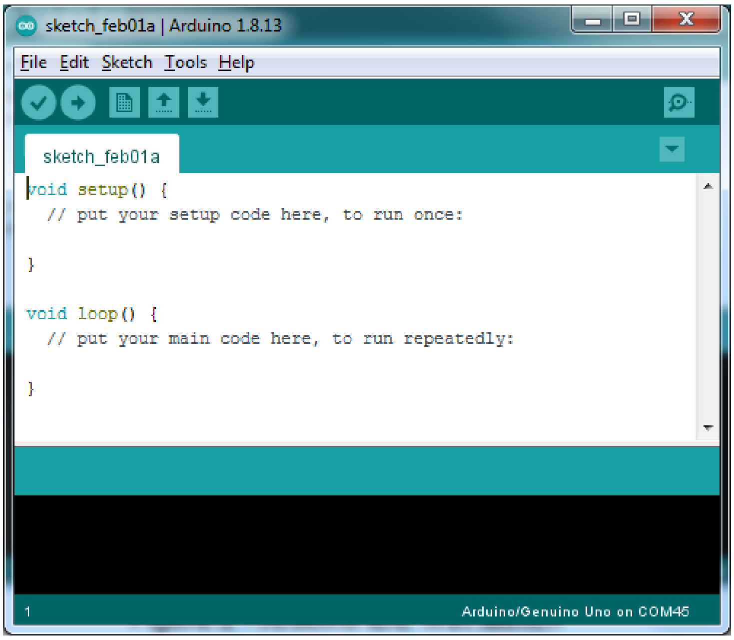

Figure 3.20: Arduino Software Interface IDE.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In

Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by: On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino Nano is one of the latest smart microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Find in Reference

This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

3.20 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in [13]. Schematic Capture support followed in [14], with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in [15] and microcontroller simulation then arrived in Proteus in [16]. Shape based auto routing was added in [17] and [18] saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in [19] and MCAD import/export was included in [20]. Support for high speed design was added in [21]. Feature led product releases are typically biannual, while maintenance based service packs are released as required.

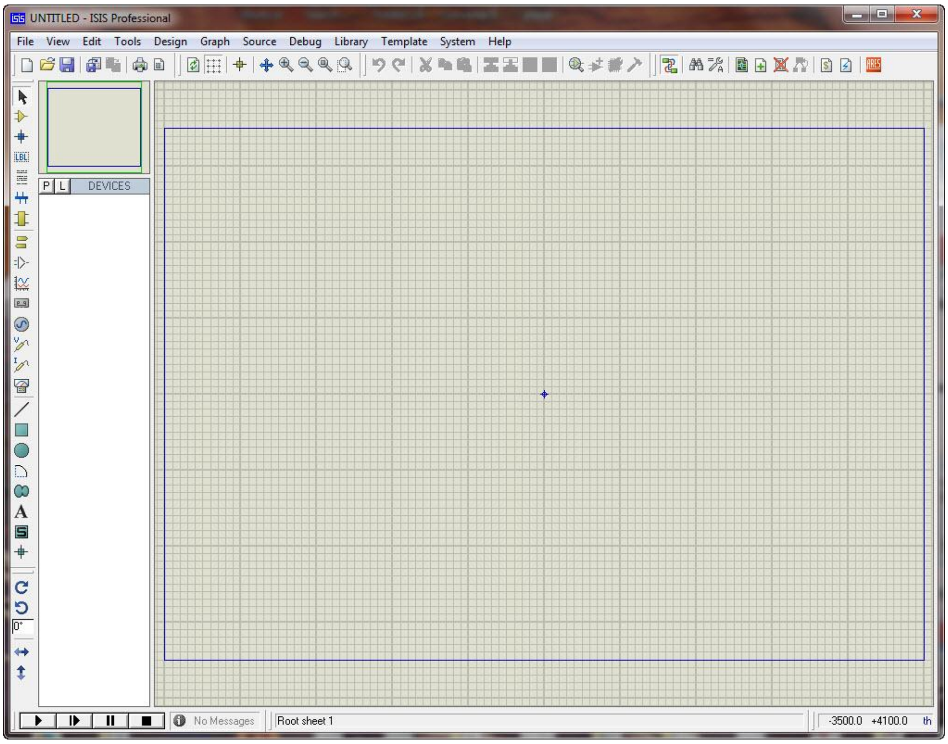

Figure 3.21: Proteus Software Interface.

3.21 Methodology

- Creating an idea for Design and construction of **Android Controlled Solar Surveillance Robot with Camera Visualization**. And designing a block diagram & circuit diagram to know which components need to construct it.
- Collecting the all components and programming for the micro-controller to controlled the system.
- Setting all components in a PCB board & soldering. Then assembling the all block in a board and finally run the system & checking.

Chapter 4

Test & Result

4.1 Result

This chapter contains the results obtained and discussion about the full project. Our project is **Android Controlled Solar Surveillance Robot with Camera Visualization**. In our project making we used PVC boards for total hardware making. After finally completing this project, we ran it & we observed the output of this project.We can see that it is working well as expected. After making our project we observe it very careful. It works as we desire. Our project give output perfectly and all equipment are work perfectly. We check how much it works and we get perfect output from this project.

- ❑ Finally, we have completed our project successfully & check our project its run accurately according to our objective.
- ❑ At first, we start our system.
- ❑ Then it was control by Bluetooth Module with the help of android apps.
- ❑ Then the robot was detect the obstacle and show the robot in-front view in browser.

We calculated the surface length of each object in accordance with our test requirements, as detailed below. Our robot's ultrasonic sensor signals any object in front of it along its center.

Specification:

- To determine the range at which the object could be detected, we experimented with various rectangular surface measurements.

Below, you can see a picture showing the precise distance at which the robot is detecting the rectangular object.

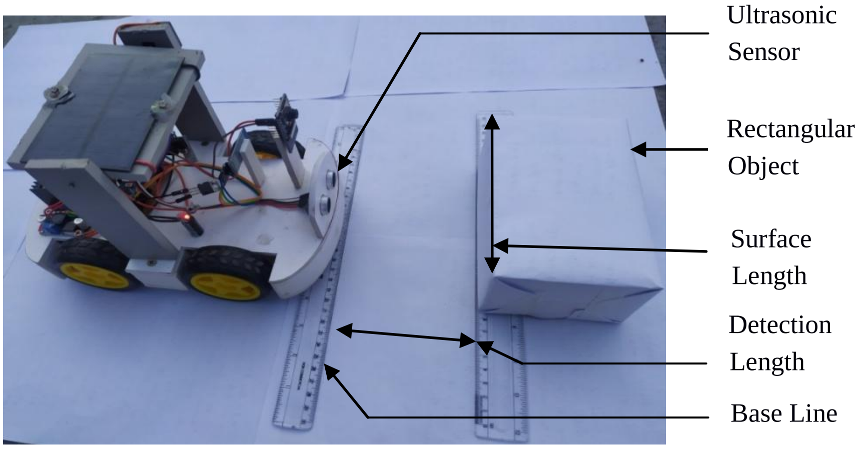

Figure: 4.1: Robot detect the Rectangular Object

Table of Rectangular object:

SL No.	Object	Surface Length (cm)	Detection Length (cm)	Distance Between Ultrasonic Sensor Eyeball (cm)	Angle (°)
1	Rectangular	15.00	10.50	7.00	33.69
2		9.50	9.30	7.00	36.96
3		6.50	8.50	7.00	39.47

Two graphs have been utilized in this instance; one will calculate the value of detection with regard to the object's length, and the other will show the angle at which the object will be detected.

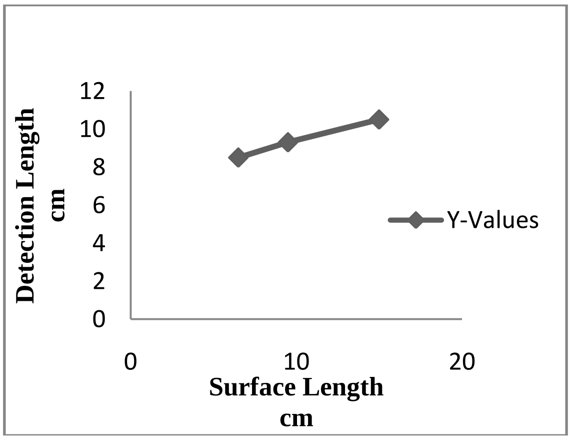

Figure: 4.2: Relation between Detection Length & Surface Length.

The value of the detection line rises with increasing surface length. If the surface area grows, the ultrasonic sensor can detect objects more quickly. Through this graph, we can also see that the detection value falls along with a surface's quality.

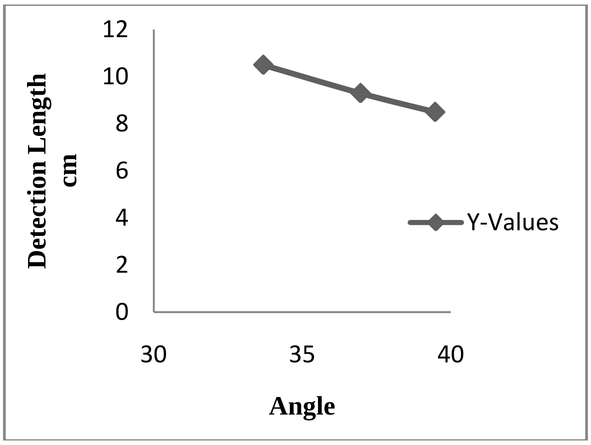

Figure: 4.3: Relation between Angle & Detection Length

The ultrasonic sensor has two eyes that are 14 cm apart. The amount of angle produced on the surface of the item reduces as the ultrasonic sensor's object detecting range grows. The angle value also decreases in proportion to the detection value that the ultrasonic sensor detects.

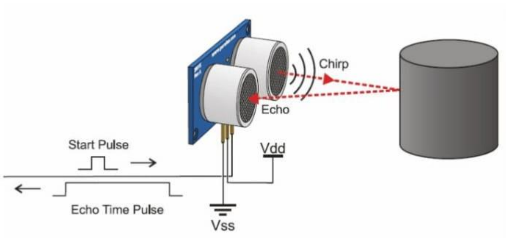

Figure: 4.4: Working of Ultrasonic Sensor.

A sound wave at a frequency outside the range of human hearing is sent out by ultrasonic sensors to function. To receive and transmit the ultrasonic sound, the sensor's transducer functions as a microphone.

Formula used to calculate the angle:

$$\theta = \tan^{-1}\frac{Half\ \ distance\ \ between\ \ two\ Ultrasonic\ \ Eyeball\ \ (l)}{Detection\ \ Length\ (l)}$$

- To determine the range at which the object could be detected, we experimented with the Round shape object surface measurements.

Below, you can see a picture showing the precise distance at which the robot is detecting the Round shape object.

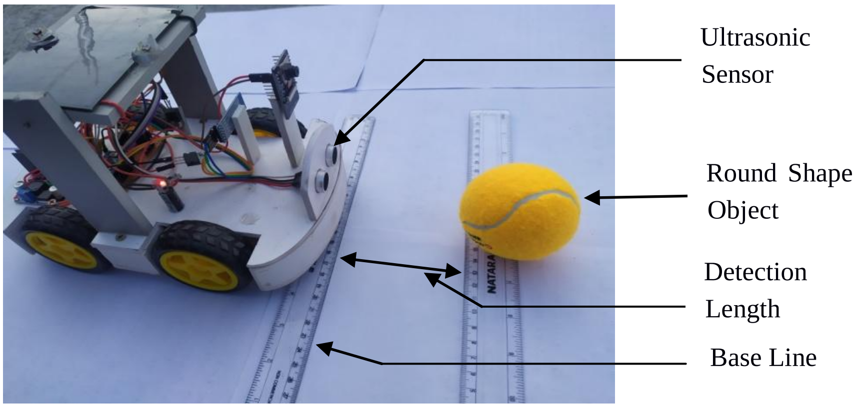

Figure: 4.5: Robot detect the Round object.

Table of Round object:

SL No.	Object	Surface Radius (cm)	Detection Length (cm)	Distance Between Ultrasonic Sensor Eyeball (cm)	Angle (°)
1	Table Tennis	2.00	8.70	7.00	38.82
2	Golf Ball	2.13	9.40	7.00	36.67
3	Tennis Ball	3.20	10.20	7.00	34.46
4	Cricket Ball	3.60	10.30	7.00	32.20

Two graphs have been utilized in this instance; one will calculate the value of detection with regard to the object's length, and the other will show the angle at which the object will be detected.

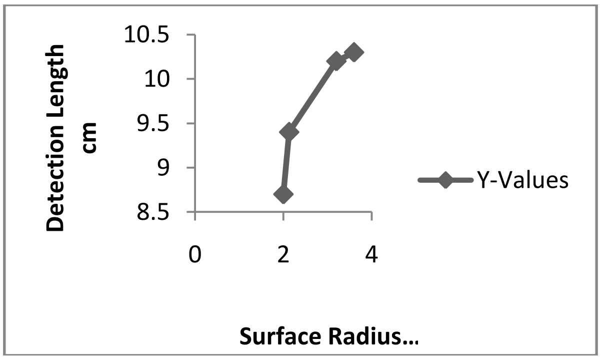

Figure: 4.6: Relation between Surface Radius & Detection Length.

Increased surface length leads to a higher value for the detection line. The ultrasonic sensor can find items faster as the surface area increases. We can also see from this graph that as a surface's quality declines, so does the detection value.
.

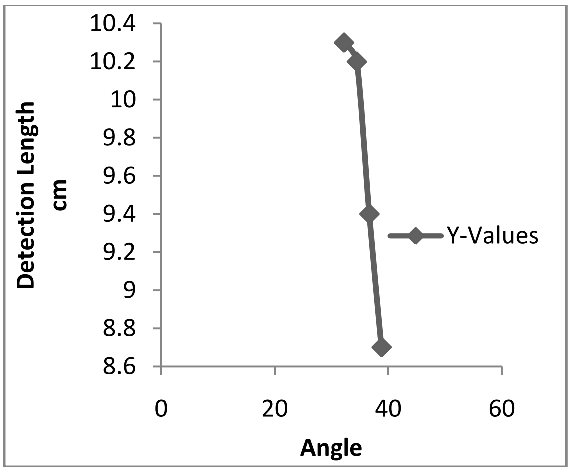

Figure: 4.7: Relation between Detection Length & Angle.

There are two 14 cm apart eyes on the ultrasonic sensor. As the object detecting range of the ultrasonic sensor increases, the amount of angle created on the surface of the object decreases. Additionally, in direct proportion to the detection value that the ultrasonic sensor senses, the angle value lowers.

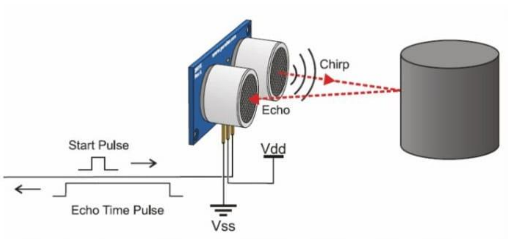

Figure: 4.8: Working of Ultrasonic Sensor.

A sound wave at a frequency outside the range of human hearing is sent out by ultrasonic sensors to function. To receive and transmit the ultrasonic sound, the sensor's transducer functions as a microphone.

Formula used to calculate the angle:

$$\theta = \tan^{-1} \frac{Half\ \ distance\ \ between\ \ two\ Ultrasonic\ \ Eyeball\ \ (l)}{Detection\ \ Length\ (l)}$$

- To determine the range at which the object could be detected, we experimented with the Cylindrical object surface measurements.

Below, you can see a picture showing the precise distance at which the robot is detecting the Cylindrical object.

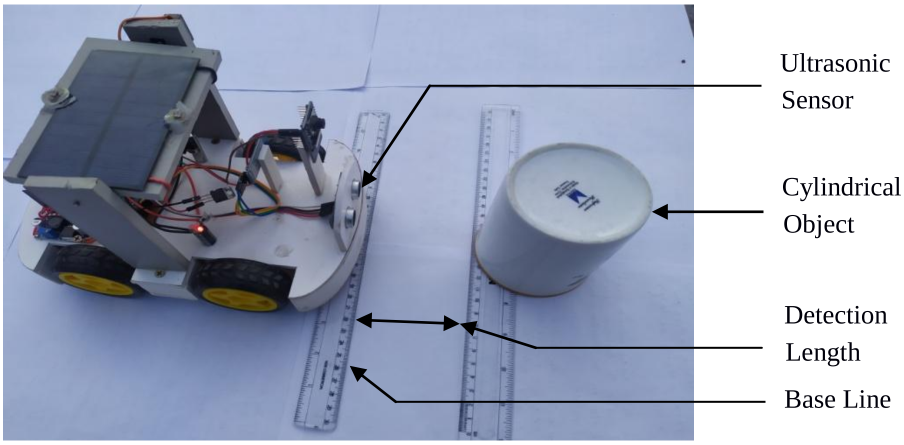

Figure: 4.9: Robot detect the Cylindrical object

Table of Cylindrical object:

SL No.	Object	Surface Radius (cm)	Detection Length (cm)	Distance Between Ultrasonic Sensor Eyeball (cm)	Angle (°)
1	Cylindrical	2.48	9.60	7.00	36.09
2		3.50	10	7.00	34.99
3		4.13	10.5	7.00	33.69

Two graphs have been utilized in this instance; one will calculate the value of detection with regard to the object's length, and the other will show the angle at which the object will be detected.

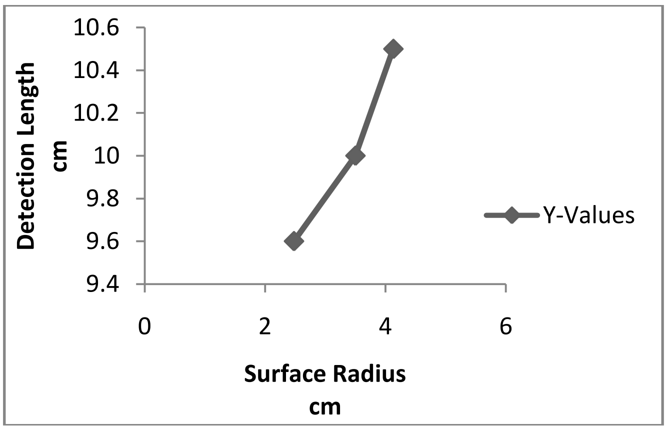

Figure: 4.10: Relation between Surface Radius & Detection Length.

The value of the detection line increases with increasing surface length. As the surface area rises, the ultrasonic sensor can locate objects more quickly. Additionally, this graph demonstrates that the detection value decreases as a surface's quality increases.

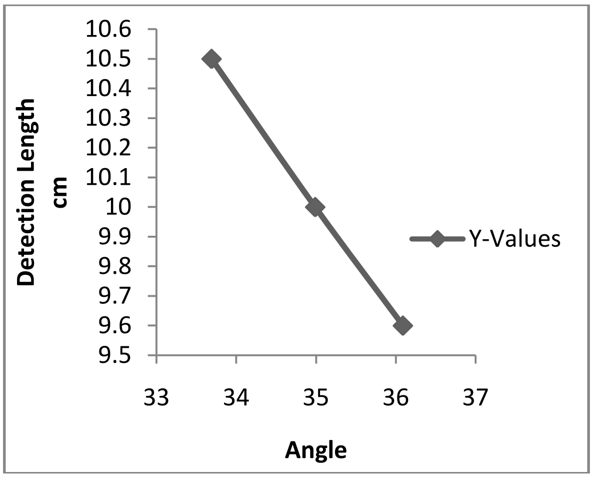

Figure: 4.11: Relation between Detection Length & Angle.

There are two 14 cm apart eyes on the ultrasonic sensor. As the object detecting range of the ultrasonic sensor increases, the amount of angle created on the surface of the object decreases. Additionally, in direct proportion to the detection value that the ultrasonic sensor senses, the angle value lowers.

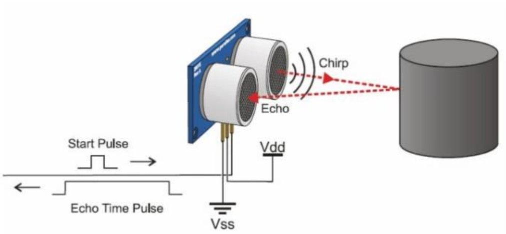

Figure: 4.12: Working of Ultrasonic Sensor.

A sound wave at a frequency outside the range of human hearing is sent out by ultrasonic sensors to function. To receive and transmit the ultrasonic sound, the sensor's transducer functions as a microphone.

Formula used to calculate the angle:

$$\theta = \tan^{-1} \frac{Half\ \ distance\ \ between\ \ two\ Ultrasonic\ \ Eyeball\ \ (l)}{Detection\ \ Lengt\,h\ (l)}$$

- To determine the range at which the object could be detected, we experimented with the Triangular object surface measurements.

Below, you can see a picture showing the precise distance at which the robot is detecting the Cylindrical object.

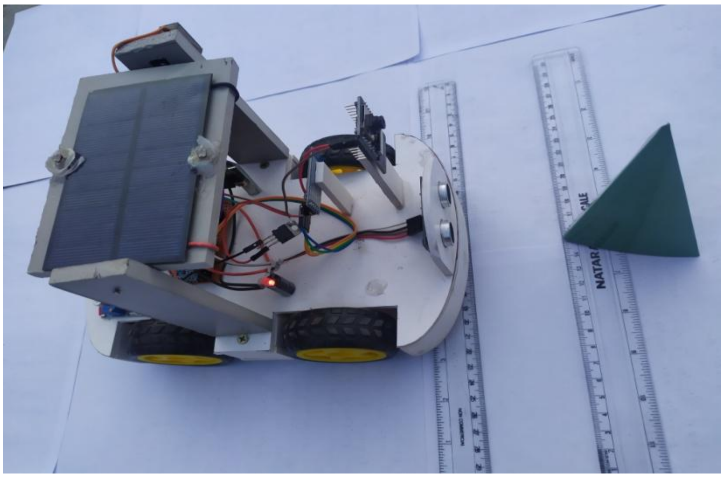

Figure: 4.13: Robot detect the Angular object.

Table of Triangular object:

SL No.	Object	Surface Length (cm)	Detection Length (cm)	Distance Between Ultrasonic Sensor Eyeball (cm)	Angle (°)
1	Triangular	8.00	8.5	7.00	39.26
2		12.00	9.2	7.00	37.26
3		15.00	11.00	7.00	32.47

Two graphs have been utilized in this instance; one will calculate the value of detection with regard to the object's length, and the other will show the angle at which the object will be detected.

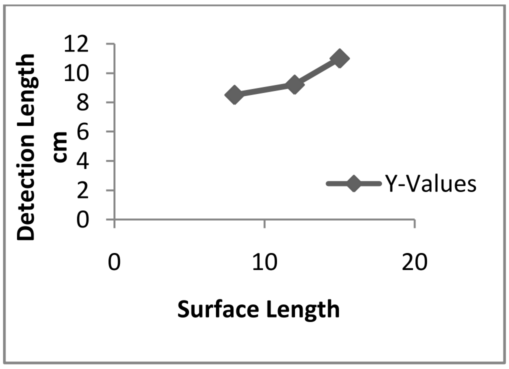

Figure: 4.14: Relation between Surface Length & Detection Length.

As surface length increases, the value of the detection line rises. The ultrasonic sensor is faster at locating objects as the surface area increases. This graph also shows that the detection value falls off as a surface's quality rises.

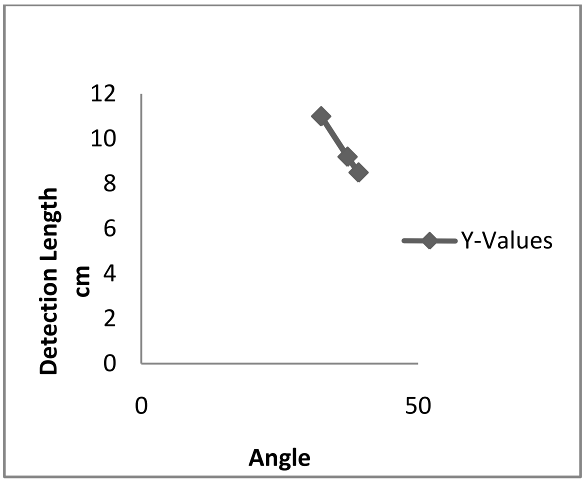

Figure: 4.15: Relation between Detection Length & Angle.

There are two 14 cm apart eyes on the ultrasonic sensor. As the object detecting range of the ultrasonic sensor increases, the amount of angle created on the surface of the object decreases. Additionally, in direct proportion to the detection value that the ultrasonic sensor senses, the angle value lowers.

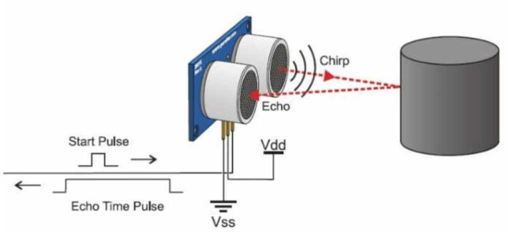

Figure: 4.16: Working of Ultrasonic Sensor.

A sound wave at a frequency outside the range of human hearing is sent out by ultrasonic sensors to function. To receive and transmit the ultrasonic sound, the sensor's transducer functions as a microphone.

Formula used to calculate the angle:

$$\theta = \tan^{-1} \frac{Half\ \ distance\ \ between\ \ two\ Ultrasonic\ \ Eyeball\ \ (l)}{Detection\ \ Length\ (l)}$$

4.2 Cost Analysis

Sl.no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)
1	Arduino	Pro Mini	1	1150	1150
2	Battery	3.7V	2	150	300
3	Ultrasonic Sensor	HC-SR04	1	450	450
5	Camera	ESP-32	1	1850	1850
6	Solar	Mini	2	1100	2200
7	Servo Motor		1	750	750
8	Motor Driver	L298N	1	420	420
9	LDR Sensor	DC 5V	2	75	150
10	Gear Motor	DC	4	250	1000
11	Bluetooth Module	HC 05	1	850	850
12	Others			3450	3450
				Total	12570/=

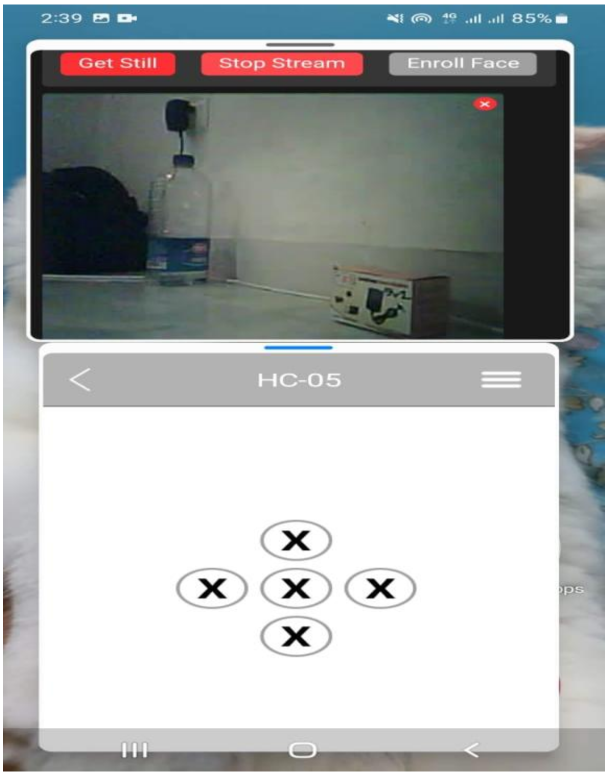

Figure: 4.17: Android Apps Camera View.

In this figure 4.17 our system demonstrates how simple it is to use the Camera ESP-32 to capture the items in our rearview.

Additionally, the use of five buttons—forward, backward, left, right, and the stop button—on the mobile device is shown here. We may directly (live streaming) see the surroundings through the mobile device screen when the vehicle is moving. This is the result part photo here we attach.

4.3 Advantage

There are certainly many advantages of our project and some of the major ones have been given below:

- Can detect the obstacle in-front of the robot.
- Can show the surrounding view near the robot.
- Can control the robot with the help of android apps control by using Bluetooth module.
- This project is easy to use

- Cost effective
- User-friendly
- Automatic and requires less human interaction
- The whole system consumes very little energy.

4.4 Limitation

Though this project has many advantages but it has some limitations as well and they are listed below:

- Used cheap Chinese products for the prototype so there's some processing delay present in the circuit
- This project can now be only used for small scale purposes.
- In foggy weather solar can't store power from sun.

4.5 Application

This project has applications in many fields due its necessity. We have selected a few of them and they are given below:

- It can be used in any fire incident to inspection that place.
- It can be used in any industry work monitoring.

Chapter 5

Conclusion

The solar-powered surveillance robot was built with locally accessible components, and tests were conducted to evaluate its performance with various types of objects. This effectiveness test will aid in the development of a more accurate model because the surveillance robot must withstand a variety of objects. The following conclusions can be derived on the testing and significant information regarding the robot's design are shown in below:

- The robot's platform is made of locally accessible PCB board, which has sufficient strength and a low density that contributes to the creation of a less-weighted construction with a reliable support.
- Through the ultrasonic sensor, we have successfully located any object at a specific distance.
- When an object is present, the analogue output measurements are at their lowest, yet they still consistently deviate by 8 to 12 cm. Consequently, even under these conditions and with the lowest possible output reading, an efficient algorithm can be created.
- The value of the detection line increases as the object's weight increases, and the value of the detection line drops as the object's weight value declines.
- Following object detection, the robot starts to move away from the source approximately 65cm to 70 cm.
- The optimal platform to train the device for any type of item is 10 cm from, which is the best range for object source detection.

- The readings are obtained from the robot at various distances, and it is calibrated to stop at a secure distance from the source.

- For the purpose of quick object recognition, we mounted a camera here, directly assisting us in our security operations.

On our system, we read **Android Controlled Solar Surveillance Robot with Camera Visualization.** It will help to detect obstacle on any crucial condition more accurately and fast. We have successfully implemented the working of the surveillance robot controlled using Android Apps with Bluetooth Module device. The robot is successfully controlled using the remote application. Even the real time video and image view feel is successfully achieved using the camera.

5.1 Future Scope

The model can be improved by making some changes in the hardware and components. Some suggestions are given below-

- In future development this project can be develop by more sensor & alarm system.
- In future we will add fire extinguishing system.
- In future we will add GSM system.

Reference

[1] Li, C. R., Lin, Y. C., Hung, M. W., Yang, C. C., Tsai, H. Y., Chang, Y. J., & Hsiao, W. T. (2018, March). Integrating temperature, humidity, and optical aerosol sensors for a wireless module for three-dimensional space monitoring. In Sensors Applications Symposium (SAS), 2018 IEEE (pp. 1-4). IEEE.

[2] Bae, S. D., Kim, G. S., Chu, S. J., & Mok, Y. K. (2018). U.S. Patent No. 9,868,591. Washington, DC: U.S. Patent and Trademark Office.

[3] Pinter, M., Wright, T. C., Reynolds, H. N., Lai, F., & Wang, Y. (2018). U.S. Patent No. 9,974,612. Washington, DC: U.S. Patent and Trademark Office.

[4] Jiang, Z., Yang, C., He, W., & Ju, Z. (2017, July). Advanced tele-operated robot interface for teaching by demonstration. In Machine Learning and Cybernetics (ICMLC), 2017 International Conference on (Vol. 1, pp. 290-298). IEEE.

[5] Garrett, B., & Anderson, K. (2018). Drone methodologies: Taking flight in human and physical geography. Transactions of the Institute of British Geographers.

[6] Lehner, B. A. E., Mazzotta, D. G., Teeney, L., Spina, F., Filosa, A., Pou, A. C., ... & Soriano, P. L. (2017). Human Assisted Robotic Vehicle Studies-A conceptual end-to-end mission architecture. Acta Astronautica, 140, 380-387.

[7] Uddin, M. S., Gianni, M., & Lab, A. (2017, July). Long range robot teleoperation system based on internet of things. In Computer and Communication Systems (ICCCS), 2017 2nd International Conference on (pp. 163-167). IEEE.

[8] Piumatti, G., Lupetti, M. L., & Lamberti, F. (2017). Human-Robot Interaction.

[9] Belk, R. (2017). The Soul and the Machine: Humanlike Machines and Machinelike Humans. ACR North American Advances.

[10] Hazarika, S. M., & Dixit, U. S. (2018). Robotics: History, Trends, and Future Directions. Introduction to Mechanical Engineering, 213-239.

[11] Wronka, C. M., & Dunnigan, M. W. (2006). Internet remote control interface for a multipurpose robotic arm. International Journal of Advanced Robotic Systems, 3(2), 27.

[12] Minamide, A., Takemata, K., Naoe, N., Yamada, H., & Hoon, P. S. (2008, March). Development of a long-distance-controlled robot system for engineering education. In Wireless, Mobile, and Ubiquitous Technology in Education, 2008. WMUTE 2008. Fifth IEEE International Conference on (pp. 179-181). IEEE.

Appendix

Micro-Controller Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27 , 16,2);
#include <Servo.h>
float input_voltage = 0.0;
float temp=0.0;
Servo myservo;
// RemoteXY select connection mode and include library
#define REMOTEXY_MODE__SOFTSERIAL
#include <SoftwareSerial.h>

#include <RemoteXY.h>

#define REMOTEXY_SERIAL_RX 2
#define REMOTEXY_SERIAL_TX 3
#define REMOTEXY_SERIAL_SPEED 9600

int motor1 = 4;
int motor2 = 5;
int motor3 = 6;
int motor4 = 7;
long duration;
long distance;
#define trigPin 8
#define echoPin 10
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =   // 60 bytes
  { 255,5,0,0,0,53,0,16,31,1,1,0,25,33,12,12,31,24,88,0,
```

```
  1,0,25,61,12,12,31,24,88,0,1,0,11,47,12,12,31,24,88,0,
  1,0,39,47,12,12,31,24,88,0,1,0,25,47,12,12,31,24,88,0 };

// this structure defines all the variables and events of your control interface
struct {

    // input variables
  uint8_t forward; // =1 if button pressed, else =0
  uint8_t back; // =1 if button pressed, else =0
  uint8_t left; // =1 if button pressed, else =0
  uint8_t right; // =1 if button pressed, else =0
  uint8_t stop; // =1 if button pressed, else =0

    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else =0

} RemoteXY;
#pragma pack(pop)



/////////////////////////////////////////////
//           END RemoteXY include          //
/////////////////////////////////////////////

void setup()
{
  RemoteXY_Init ();
     lcd.init();
  lcd.backlight();
  Serial.begin(9600);            //Set rate for communicating with phone
 pinMode(A0, INPUT);       //Set relay1 as an output
pinMode(A1, INPUT);       //Set relay2 as an output

    myservo.attach(9);  // attaches the servo on GIO2 to the servo object
```

```
 pinMode(motor1,OUTPUT);
   pinMode(motor2,OUTPUT);
    pinMode(motor3,OUTPUT);
     pinMode(motor4,OUTPUT);
     pinMode(trigPin, OUTPUT);
       pinMode(echoPin, INPUT);

       }

void loop()
{
  RemoteXY_Handler ();

 int lightvalue1 = analogRead(A0);
   int lightvalue2 = analogRead(A1);
ultrasonic();
if(lightvalue1 > 500 && lightvalue2 < 500){
    myservo.attach(9);  // attaches the servo on GIO2 to the servo object
      myservo.write(140);
delay(1000);

}
 if(lightvalue1 < 500 && lightvalue2 > 500){
     myservo.attach(9);  // attaches the servo on GIO2 to the servo object
     myservo.write(50);
delay(1000);

  }
 if(lightvalue1 > 1000 && lightvalue2 > 1000){
     myservo.detach();  // attaches the servo on GIO2 to the servo object
delay(100);

  }
  Serial.print("lightvalue1: ");
```

```
      Serial.print(lightvalue1);
      Serial.print("   lightvalue2: ");
      Serial.println(lightvalue2);



  if (RemoteXY.forward!=0) {
  Serial.print("Forward");
forward();
}
  else {
    /*  button not pressed */
  }

    if (RemoteXY.back!=0) {
   Serial.print("Reverse");
backward();
  }
  else {
    /*  button not pressed */
  }

    if (RemoteXY.left!=0) {
 Serial.print("Left");
left();
}
  else {
    /*  button not pressed */
  }

    if (RemoteXY.right!=0) {
  Serial.print("Right");
right();
}
  else {
```

```
    /*  button not pressed */
  }
    if (RemoteXY.stop!=0) {
  Serial.print("Stop");
stop1();
}
  else {
    /*  button not pressed */
  }
}

void forward()
{
            digitalWrite(motor1, HIGH);
            digitalWrite(motor2, LOW);
            digitalWrite(motor3, HIGH);
            digitalWrite(motor4, LOW);

}

void backward()
{
            digitalWrite(motor1, LOW);
            digitalWrite(motor2, HIGH);
            digitalWrite(motor3, LOW);
            digitalWrite(motor4, HIGH);

}

void right()
{
            digitalWrite(motor1, LOW);
            digitalWrite(motor2, HIGH);
            digitalWrite(motor3, HIGH);
```

```
        digitalWrite(motor4, LOW);

}

void left()
{
        digitalWrite(motor1, HIGH);
        digitalWrite(motor2, LOW);
        digitalWrite(motor3, LOW);
        digitalWrite(motor4, HIGH);


}


void stop1()
{
        digitalWrite(motor1, LOW);
        digitalWrite(motor2, LOW);
        digitalWrite(motor3, LOW);
        digitalWrite(motor4, LOW);
}



void ultrasonic(){
   digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);


  // Measure the response from the HC-SR04 Echo Pin
```

```
   duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

Serial.println(distance);
  delay(20);



  if(distance < 10 && distance > 3 ){
stop1();
delay(500);
backward(); //Move Right
delay(1000);
  stop1();

  }

}

//////////////////////////// camera ///////////////////////////////////////////////

#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
```

```
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "abcde";
const char* password = "12345678";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
```

```
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

   sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
```

```
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);


#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```