Object Sensing and Identification using IOT

A Major Project Report

Submitted to

CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY

BHILAI (C.G.), INDIA

In partial fulfillment of requirement for the award of the Degree

of

Bachelor of Technology

in

Electronics & Telecommunication Engineering

by

Abhinav Sharma (BG1759)
Ayush Dewangan (BG1780)
Shubharangshu Chakraborty (BG1857)

Under the Guidance of
Prof. K.Uma
Associate Professor

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

BHILAI INSTITUTE OF TECHNOLOGY, BHILAI HOUSE, DURG (C.G.) -491001, INDIA

SESSION 2021-2022

D E C L A R A T I O N

I the undersigned solemnly declare that the report of the Project work entitled ***"Object Sensing and Identification using IOT"***, is based on my own work carried out during the course of my study under the supervision of Prof. K.Uma, Department of Electronics and Telecommunication Engineering, Bhilai Institute of Technology, Durg, Chhattisgarh.

I assert that the statements made, and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University/ deemed University of India or any other country. All help received, and citations used for the preparation of the Report have been duly acknowledged.

Abhinav Sharma
Roll No.: 300102818067
Enrollment No.: BG1759
Department of Electronics
& Telecommunication Engg.
BIT, Durg

Ayush Dewangan
Roll No.: 300102818055
Enrollment No.: BG1780
Department of Electronics
& Telecommunication Engg.
BIT, Durg

Shubharangshu Chakraborty
Roll No.: 300102818059
Enrollment No.: BG1757
Department of Electronics
& Telecommunication Engg.
BIT, Durg

CERTIFICATE BY THE SUPERVISOR

This is to certify that the report of the Project submitted is an outcome of the project work entitled ***"Object Sensing and Identification using IOT"***, carried out by

Abhinav Sharma bearing ***Roll No.: 300102818067*** & ***Enrollment No. : BG1759***
Ayush Dewangan bearing ***Roll No.: 300102818055*** & ***Enrollment No. : BG1780***
Shubharangshu Chakraborty bearing ***Roll No.: 300102818059*** & ***Enrollment No. : BG1857***

carried out under my guidance and supervision for the award of Degree, ***Bachelor of Technology in Electronics and Telecommunication Engineering*** of Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G.), India.

To the best of my knowledge and the Report

i. Embodies the work of the candidate himself,
ii. Has duly been completed,
iii. Fulfills the requirement of the ordinance relating to the B.E. Degree of the University and
iv. Is up to the desired standard for the purpose of which is submitted.

Signature of the Project Incharge
Prof. Pushpendra Singh
Assistant Professor
Department of Electronics
& Telecommunication Engg.
BIT, Durg

Signature of the Supervisor
Prof. K.Uma
Associate Professor
Department of Electronics
& Telecommunication Engg.
BIT, Durg

The project work as mentioned above is hereby being recommended and forwarded for examination and evaluation.

Signature of
Head of the Department
With Seal

CERTIFICATE BY THE EXAMINERS

This is certified that the project work entitled

"Object Sensing and Identification using IOT"

Submitted by

Abhinav Sharma bearing *Roll* ***No.: 300102818067*** & ***Enrollment No.: BG1759***
Ayush Dewangan bearing *Roll* ***No.: 300102818055*** & ***Enrollment No.: BG1780***
Shubharangshu Chakraborty bearing *Roll* ***No.: 300102818059*** & ***Enrollment No.: BG1857***

has been examined by the undersigned as a part of the examination for the award of the ***Bachelor of Technology Degree in Electronics and Telecommunication Engineering*** of Chhattisgarh Swami Vivekanand Technical University, Bhilai, (C.G.).

Internal Examiner — External Examiner

Date: — Date:

ACKNOWLEDGEMENT

With deep regards and profound respect, I avail this opportunity to express my deep sense of gratitude and indebtedness to **Prof. K.Uma**, Department of Electronics and Telecommunication Engineering, BIT Durg for his valuable guidance and support. I am deeply indebted for the valuable discussions at each phase of the project. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I express my sincere gratitude to **Dr. Arun Arora**, Director, **Dr. Mohan Kumar Gupta,** Principal and **Dr. Manisha Sharma,** Vice Principal, Bhilai Institute of Technology, Durg for providing adequate infrastructure to carry out present investigations and also motivating for research work, which has been a constant source of inspiration in completing this work.

I take immense pleasure to thank **Dr. Naveen Kumar Dewangan**, HOD (ETC), Bhilai Institute of Technology, Durg**,** for motivating to work in research direction and providing opportunities to connect with global research.

I whole heartedly extend my gratitude to **Prof. Pushpendra Singh**, Department of Electronics and Communication Engineering, BIT Durg for constant feedbacks and encouragements and endless support and help throughout this project work.

Lastly, I feel immensely moved in expressing my indebtedness to my revered parents whose sacrifice, guidance and blessings helped me to complete my work.

Abhinav Sharma

Ayush Dewangan

Shubharangshu Chakraborty

ABSTRACT

Object sensing and Identification using IOT is a project based on Computer Vision and Image processing. ESP-32 cam module is used to take input video feed this video feed is than send to IoT device (in our case laptop) where this image is compared to pre-existing image library (OpenCV) and the output is than displayed to laptop monitor.

Simultaneously Ultrasonic sensor sends the sound wave which is used to calculate distance from the system to the object which is to be identified.

Camera Module and Ultrasonic sensor are mounted on top of a Pan-Tilt servo motor which is used to widen the view point of the camera and ultrasonic module and thus increasing the field of view.

These all elements combine creates a module which is able to sense and identify unknown objects.

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Camera Radar Fusion
ESP	Espressif Systems
FT	Frequency-tuned
GPIO	General Purpose Input/output
GPU	Graphic Processing Unit
GND	Ground
IKB	Information Knowledge Base
IOT	Internet of Things
ISP	Internet Service Provider
OpenCV	Open Computer Vision
PCB	Printed Circuit Board
QS	Query by Shape
RX	Receiving Pin
SRAM	static random-access memory
TX	Transmitting pin
Wi-Fi	Wireless Fidelity

List of Figures

List of Tables

Table of Contents

Chapter 1
INTRODUCTION

1.1 Introduction to project

In this ever-expanding world security has been one of the prime concerns. To identify an object and to distinguish it from one other has been part of this process, with the advancement in the field of computer vision and image processing has enabled a us to better distinguish the objects digitally and to identify/recognize them.

Computer vision and Image processing along with the growing GPU & CPU power of modern-day computer has decreased the time consumption on this and thus making it possible to apply this concept on every day to day scenarios.

Robotics along with the efficient semiconductor designing and manufacturing has enabled to touch into whole another level of untapped potential of computer vision.

Nowadays Computer Vision and Image processing can be applied into live video feed and thus helping in monitoring system and replacing the conventional approaches.

Here in this work we have developed a system which takes live image feed using a camera module and then broadcast it to the IOT device where it gets processed and after comparing it with the pre-existing image models it identifies the object.

This camera module along with an ultra-sonic sensor is mounted on top of a Pan-Tilt servo motor, the pan-tilt servo facilitates the movement of the camera module and thus helping the system to cover a broader range of area. The pan servo helps in horizontal motion while the tilt servo helps in vertical motion. And with the help of pan-tilt servo we are calculating the angle at which the camera module is pointing towards.

The ultra-sonic sensor present in the pan-tilt servo is used to calculate the distance between the system and the object which is to be identified.

1.2 Purpose of this project

The key purpose of this project is to recognize and identify different objects in the live feed along with calculating an approx. distance and angle of the object which is to be identified from the system.

1.3 Objectives

1. To Sense multiple objects and able to identify them in the live feed.
2. To calculate distance of this objects from the system.
3. To rotate the camera module & ultra-sonic sensor horizontally & vertically.
4. To be able to perform this task with minimum system size and cost.

1.4 Functional Requirement

1. This system aims to improve the surveillance and monitoring system with the requirement of minimum human supervision.
2. This system can be easily modified to be able to use in different fields.
3. The system can be easily rotated to cover different angles of the area and thus enabling it to cover a wider area.

Chapter 2
Literature Survey

2.1 Literature Survey

Nobis.F.et al.in 2019 discussed about enhancing 2D object detection networks by fusing camera data and projected sparse radar data in the network layers. The proposed Camera Radar Fusion Net (CRF-Net) automatically learns at which level the fusion of the sensor data is most beneficial for the detection result. Additionally, they introduce BlackIn, a training strategy inspired by Dropout, which focuses the learning on a specific sensor type and showed that the fusion network can outperform a state-of-the-art image only network for two different datasets[1].

Zhao.Z.Q.et al.in 2019 provided a review on deep learning-based object detection frameworks. Their review begins with a brief introduction on the history of deep learning and its representative tool, namely Convolutional Neural Network (CNN). Then they focus on typical generic object detection architectures along with some modifications and useful tricks to improve detection performance further and distinct specific detection tasks exhibit different characteristics, they also briefly survey several specific tasks, including salient object detection, face detection and pedestrian detection. They also provided experimental analyses to compare various methods and drew some conclusions[2].

Yu.L.et al.in 2018 proposed a new Frequency-tuned (FT) algorithm for extracting target dynamic saliency information from a mixture of Gaussian models, aiming at the inconspicuous effect of the traditional Frequency-tuned (FT) algorithm saliency map and the significant "dilution" of feature map fusion. This algorithm makes innovative improvements from distance metrics and feature graphs. To solve the large computational complexity of traditional identification algorithms, the algorithm uses a Haar cascaded classifier with low computational complexity as a classification algorithm and uses OpenCV and Qt interface library to build an integrated multi-module system software platform to achieve single-target moving object detection and recognition. The experimental results show that the system has significant effects on the detection and recognition of single-target motion and has high accuracy, and it has a good engineering application prospect[3].

Chen.H.et al.in 2018 presented a novel software-based simulation method for modern radar system in automotive radar application. The method provides target detection and tracking algorithm simulator and the simulation results are compared with the measurement results[4].

Deniziak.S.et al.in 2015 took use of Internet of Things (IOT). It consists of the set of smart objects with video sensors, controlled by the Shape Identification Cloud. They build the real-time system to recognize the objects based on their approximate shape and with the option to be used as a web service in a cloud. To monitor the environment the system uses robots equipped with video sensors as well as surveillance cameras capable of remote position control. For object identification task they use the Query by Shape (QS) method which decomposes objects into simple graphical primitives like lines, circles, ellipses etc. and then it identifies them in a shape database[5].

Heisele.B.et al.in 2006 They present a component-based system for object detection and identification. From a set of training images of a given object they extract many components which are clustered based on the similarity of their image features and their locations within the object image. The cluster centers build an initial set of component templates from which we select a subset for the final recognizer. The localization of the components is performed by normalized cross-correlation. Two types of components are used, gray value components and components consisting of the magnitudes of the gray value gradient.
In experiments he investigated how the component size, the number of the components, and the feature type affects the recognition performance. The system is compared to several state-of-the-art classifiers on three different data sets for object identification and detection[6].

Chapter 3

Problem Identification

In the conventional method of object identification, it faces a huge drawback of limited datasets which is overcome in over project with the use of a pre-existing library named as OpenCV which a pre-stored collection of multiple objects.

While some of these authors have proposed our approach none of them have implemented it together with ultrasonic sensor which has enabled us to also determine the distance of the objects.

Most of the proposed models by the above authors are can only identify a single object at a time not only that but also the camera module has a limited field of view.

Chapter 4

Methodology

4.1 System Design

Here the block diagram and flow chart diagram of the implemented system are described.

4.1.1 Block Diagram

In this project "**Object Sensing and Identification using IOT**" we developed a system that can detect and identify an object.

To identify the object, it compares the live feed of the object to the pre-recorded images from the library known as Open-CV.

The block diagram of the system is described below –

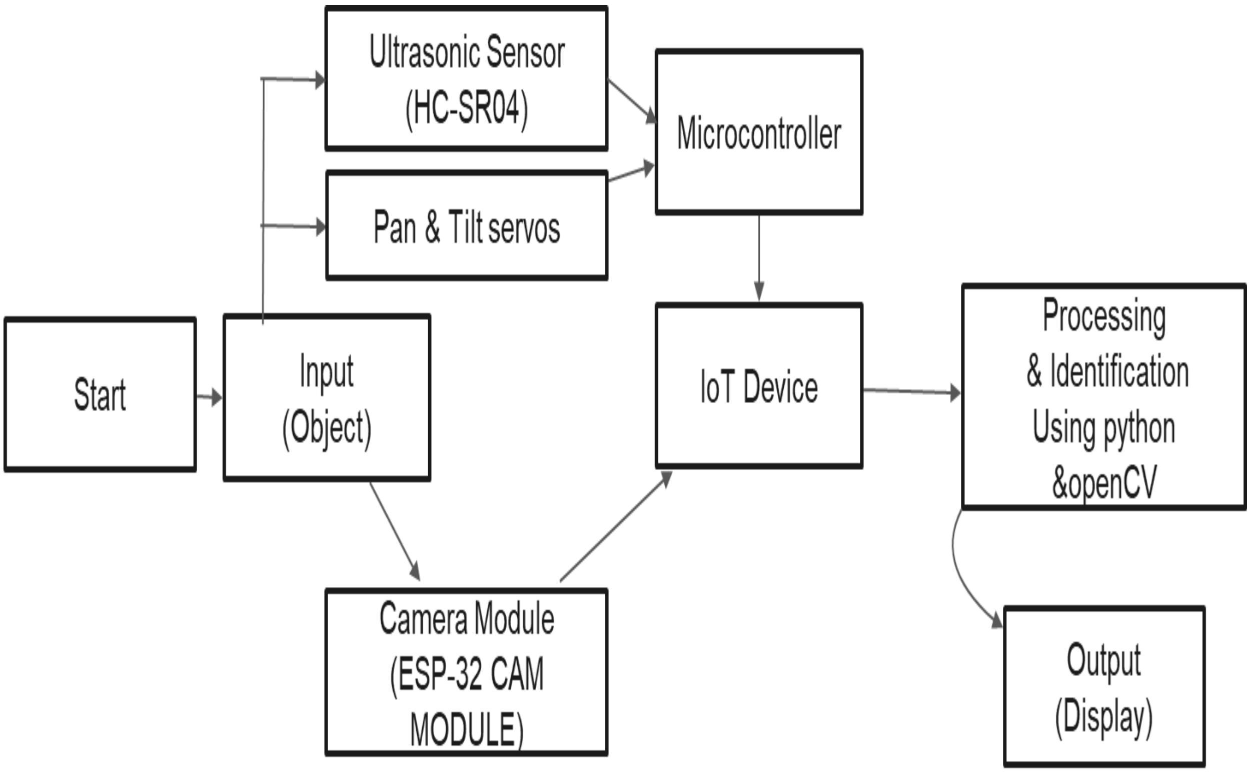


Figure 4.1 Block Diagram of Object Sensing and Identification

Description of elements of block diagram are –

1. **Input**– The object to be identifies is taken as the input when the system is turned on.
2. **Camera Module**– The camera module is used to take live video feed of the object to be identified and then transmit this data via inbuild Wi-Fi module to the IoT device.

3. **Ultrasonic Sensor**– This Ultrasonic Sensor transmit a sound signal as simultaneously as the camera module takes the input, these data is used to calculate the distance of object from the system.
4. **Pan-Tilt Servo motor**- This module has mounted Camera & Ultrasonic sensor on top of it is used to change viewing point of the camera module and ultrasonic sensor, it is controlled using keyboard keys in the IoT device.
5. **Microcontroller**- The microcontroller Atmega 328 is used as a microcontroller which stores the code of operation for the pan tilt servo and ultrasonic sensor.
6. **IoT device**– IoT stands for Internet of things, in this case we are using laptop as an IoT device to bridge the gap between the raw data from camera, ultra-sonic module and to control the Pan tilt servo and to produce the required result.
7. **Processing & Identification**– PyCharm is used to code the program for sensing and identification and to connect to the Open-CV library.
8. **Output**– The output is displayed in the computer monitor where the image identification and distance, angle of the object to the system is displayed.

4.1.2 Explanation of the connection

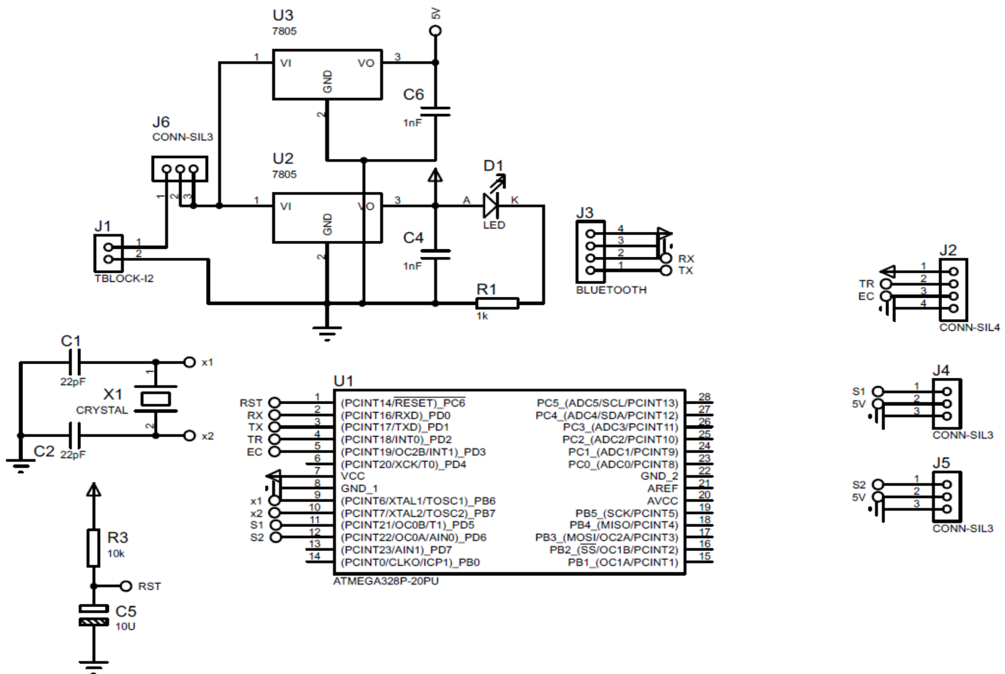


Figure 4.2 Proteus Block Diagram of the system

1. To create a webserver and to generate IP address we have given the 5V, GND, TX, RX to ESP32 cam using Arduino Uno and shorted the GND to GPIO 0 of the ESP32 cam module (only for the uploading the code) GPIO 0 determines whether the ESP32

is in flashing mode or not. This GPIO is internally connected to a pull-up 10k Ohm resistor. When GPIO 0 is connected to GND, the ESP32 goes into flashing mode and we can upload code to the board.

2. Then we have connected the data pin of the two servos for the pan and the tilt to digital 9 & 10 of the Arduino Uno board and provided the power supply from the battery.
3. Then we have connected trigger and echo pin of the Ultrasonic sensor to digital pin 7 & 8 respectively.
4. After completion of the code upload we have taken out the IC of the Arduino and placed in the external PCB and have done the connections.

4.2 Components required

4.2.1 ATMEGA 328A

The h1igh-performance Microchip 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, IKB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. Microcontroller Atmega 328A is a 28pin device in which ground is at pin 8 and 22. Supply is at pin 7, transmit and receive pins are at 2 and 3 respectively.

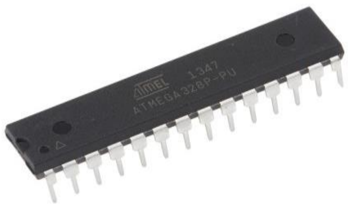


Figure 4.3 Atmega 328 IC

4.2.2 Features of AVR Atmega 328

1. High Performance, Low Power Atmel AVR 8-Bit Microcontroller Family.
2. Advanced RISC Architecture.
3. High Endurance, non-Volatile Memory Segments.
4. Peripheral Features.
5. Two 8-Bit Timer/Counters with Separate Presale and Compare Model

6. One 16 Bit Timer/Counter with Separate Presale and Compare Models.
7. Capture Mode.
8. Real Time Counter with Separate Oscillator.
9. Special Microcontroller Features
10. Power on Reset and Programmable Brown-Out Detection.
11. Internal Calibrated Oscillator.
12. External and Internal Interrupt Sources
13. I/O and Packages-23 Programmable I/O Lines.
14. 32-pin AVR microcontroller
15. 1 comparator
16. Flash program memory type

Table 4.1 Features of Atmega 328

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	332
CPU Speed (MIPS)	20
RAM Bytes	2048
Data EEPROM (bytes)	1024
Digital Communication Peripherals	1-UART, 2-SPI, 1-12C
Capture/Compare/PWM Peripherals	1 Input Capture, 1 CCP, 6PWM
Timer	2 * 8-bit, 1*16-bit
Comparators	1
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	32

4.2.3 Pin Configuration

As stated before, Twenty of the pins functions as I/O ports. This means they can function as an input to the circuit or as output. Whether they are input, or output is set in the software.

Fourteen of the pins are digital pins, of which Six can function to give PWM output. Six of the pins are for analog input/output.

Two of the pins are for the crystal oscillator. This is to provide a clock pulse for the Atmega chip. A clock pulse is needed for synchronization so that communication can occur in synchrony between the Atmega chip and a device that it is connected to.

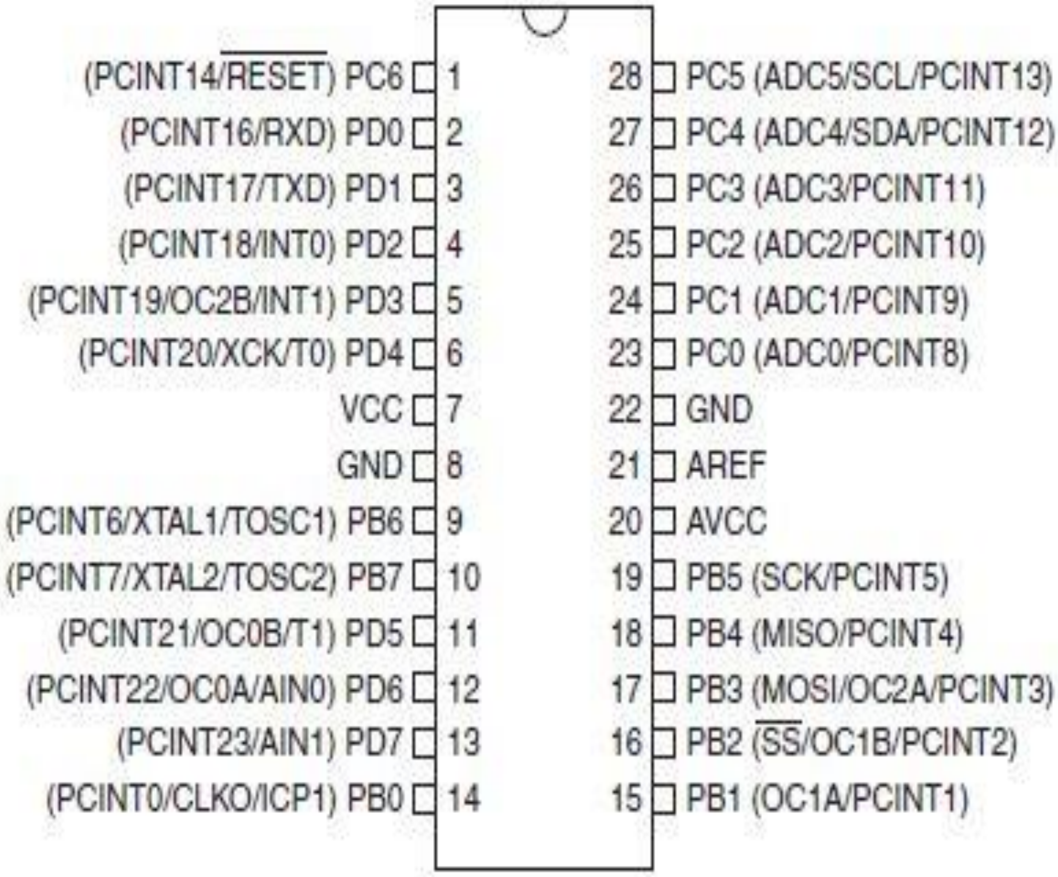


Figure 4.4 Pin Configuration of Atmega328

The chip needs power so Two of the pins, VCC and GND, provide it power so that it can operate. The Atmega328 is a low-power chip, so it only needs between 1.8-5.5V of power to operate.

The Atmega328 chip has an analog-to-digital converter (ADC) inside of it. This must be or else the Atmega328 wouldn't be capable of interpreting analog signals. Because there is an ADC, the chip can interpret analog input, which is why the chip has six pins for analog input. The ADC has three pins set aside for it to function- AVCC, AREF, and GND. AVCC is the power supply, positive voltage, that for the ADC. The ADC needs its own power supply to work. GND is the power supply ground. AREF is the reference voltage that the ADC uses to convert an analog signal to its corresponding digital value. Analog voltages higher than the reference voltage will be assigned to a digital value of 1, while analog voltages below the reference voltage will be assigned the digital value of 0. Since the ADC for the Atmega328 is a 10-bit ADC, meaning it produces a 10-bit digital value, it converts an analog signal to its digital value, with the AREF value being a reference for which digital values are high or low. Thus, a portrait of an analog signal is shown by this digital value, thus, it is its digital correspondent value.

The last pin is the RESET pin. This allows a program to be rerun and start over. And this sums up the pin out of an Atmega328 chip.

4.2.2 ESP-32 Cam module

ESP32-CAM is a small size, low power consumption camera module based on ESP32-based development board with onboard camera. It is an ideal solution for IoT application, prototypes constructions and DIY projects.

The board integrates Wi-Fi, traditional Bluetooth and low power BLE, with 2 high-performance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz.

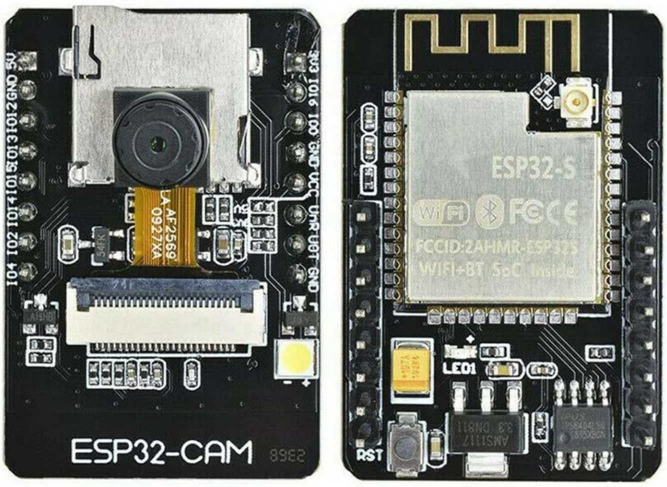


Figure 4.5 ESP-32 CAM MODULE

Fully compliant with Wi-Fi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to another host MCUs to add networking capabilities to existing devices.

ESP32-CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications.

Pin Schematics:

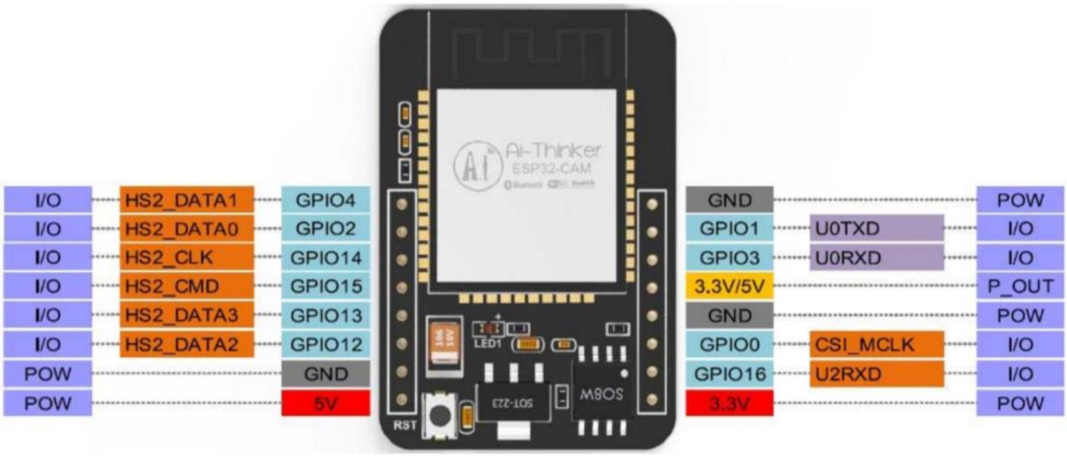


Figure 4.6 Schematic Diagram of ESP-32 cam module

4.2.3 Pan-Tilt Servo Motor

The Pan-Tilt servo is perfect to provide 2-D motion to the project. The pan-tilt can rotate roughly 180° from side-to-side and can tilt up and down around 150°.The motors are given power of 5 V 2 A from a battery source and are being controlled using keyboards of the laptop.

Figure 4.7 Pan-Tilt Servo Motor

Here the red cable is for VCC, brown for ground and yellow for write for both servos respectively. In this project we are also using Pan-Tilt to calculate the angle at which the camera module is aiming towards.

4.2.4 Ultrasonic Sensor [HC-SR04]

The HCSR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent noncontact range detection with high accuracy and stable readings. From 2cm to 400 cm. Its operation is not affected by sunlight or black material like Sharp rangefinders are. It comes complete with ultrasonic transmitter and receiver module.

Features:

1. Power Supply: +5V DC
2. Quiescent Current: <2mA
3. Working Current: 15mA
4. Effectual Angle: <15°
5. Ranging Distance: 2cm – 400 cm /1" - 13ft
6. Resolution: 0.3 cm
7. Measuring Angle: 30 degree
8. Trigger Input Pulse width: 10uS
9. Dimension: 45mm x 20mm x 15mm

Figure 4.8 Ultra-Sonic Sensor HC-SR04

Pin Layout of HC-SR04-

VCC = +5VDC;

Trig = Trigger input of Sensor;

Echo = Echo output of Sensor;

GND = GND

4.2.5 Crystal Oscillator

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a precise frequency.

An electronic circuit that is used to generate an electrical signal of precise frequency by utilizing the vibrating crystal's mechanical resonance made of piezoelectric material. There are different types of piezoelectric resonators, but typically, quartz crystal is used in these types of oscillators. The atoms, molecules, ions are packed in an order in three spatial dimensions with repeating pattern form a solid that can be called as a crystal. The crystal can be made by almost any object that is made of elastic material by using appropriate electrical transducers. As every object consists of natural resonant frequency of vibration, steel consists of high speed of sound and is also very elastic.

Figure 4.9 Crystal Oscillator

Thus, steel is frequently used instead of quartz in mechanical filters. This resonant frequency depends on different parameters such as sire, elasticity, speed of sound, and shape of the crystal. In general, the shape of high frequency crystals is simple rectangular plate and the shape of low frequency crystals is tuning fork shape. The most common type of piezoelectric resonator used in the quartz so oscillator circuits incorporating them known as crystal oscillators, but other piezoelectric materials including polycrystalline ceramics are used in similar circuits. Quartz crystals are manufactured for frequencies from a few tens to kilohertz to hundreds of megahertz.

4.2.6 Voltage Regulator

A voltage regulator is designed to automatically maintain a constant voltage level. A voltage regulator may be a simple feed-forward design or may include negative feedback It may use in electromechanical mechanism, or electronic components. Depending on the design, it may be used regulate one or more AC or DC voltages. Electronic voltage regulators are found in devices such as computer power supplies where they stabilize the DC voltages used by the processer and other dements. There are several types of voltage regulators that range from very affordable to very efficient. The most affordable and often the easiest type of voltage regulator to use are linear voltage regulators. Linear regulators come in a couple of types, are very compact, and used often in low voltage, low power systems,

Switching regulators are much more efficient than linear voltage regulators, but they are harder to work with and more expensive. This automatic regulation of the output voltage level is handled by various feedback techniques, some as simple as a Zener diode while others include complex feedback topologies that can improve performance, reliability, efficiency, and add other features Be boosting output voltage above the input voltage to the voltage regulator.

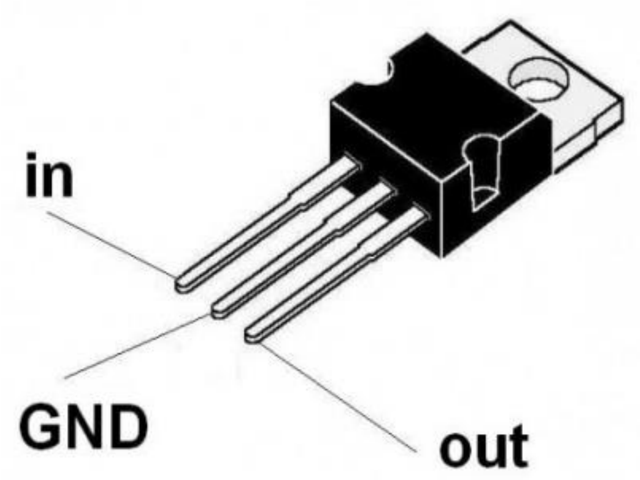


Figure 4.10 Voltage Regulator

In this project we are using voltage regulator to convert 7.4 Volts 2 Amp DC to 5 volts 2 Amp DC.

4.2.7 USB to TTL [CP2102 MODULES]

CP2102 Module is used for Communicate PC to the microcontroller easily. It directs 5V/3V level UART at (RX/TX) pins and create virtual serial COM port on PC using which we can control/communicate with the microcontroller.

Figure 4.11 USB to TTL [Cp2102 module]

This module has 6 pin breakout which includes

1. TXD = Transmit Output - Connect to Receive Pin (RXD) of Micro controller.
   This pin is TX pin of CP2102 on board.
2. RXD = Receive Input - Connect to Transmit Pin (TXD) of Micro controller.
   This pin is RX pin of CP2102 on board.
3. GND = should be common to microcontroller ground.

4. 3V3 = Optional output to power external circuit up to 50mA.
5. 5V = Optional output to power external circuit up to 500mA
6. DTR/RST = Optional output pin to reset external microcontrollers like Arduino.

This module can be used to upload Arduino sketches onto your Arduino loaded AVR devices over RXD, TXD & DTR/RST pins of module.

4.2.8 LED [Light Emitting Diodes]

The lighting emitting diode is a p-n junction diode. It is a specially doped diode and made up of a special type of semiconductors. When the light emits in the forward biased, then it is called as a light emitting diode.

The light emitting diode simply known as a diode. When the diode is forward biased, then the electrons & holes are moving fast across the junction and they are combining constantly, removing one another out.

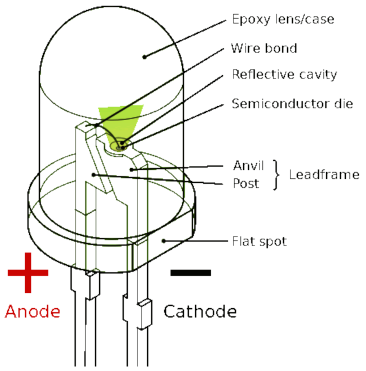


Figure 4.12 LED Generic Diagram

Soon after the electrons are moving from the n-type to the p-type silicon, it combines with the holes, then it disappears. Hence it makes the complete atom & more stable and it gives the little bunt of energy in the form of a tiny packet or photon of light.

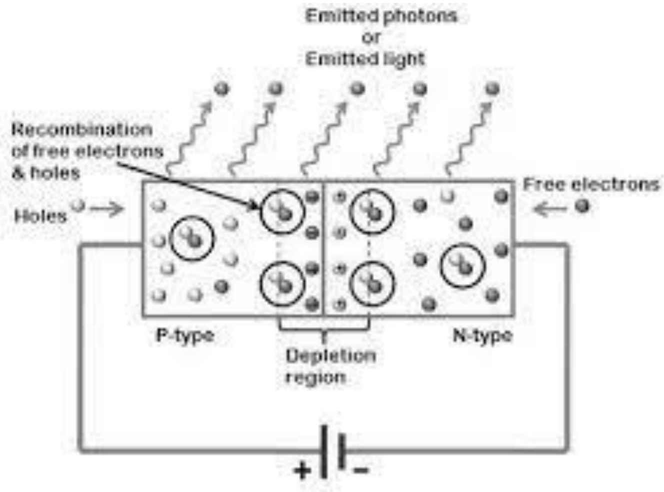


Figure 4.13 Working diagram of LED

The above diagram shows how the light emitting diode works and the step by step process of the diagram.

1. From the diagram, we can observe that the N-type silicon is in red color and it contains the electrons, they are indicated by the black circles.
2. The P-type silicon is in the blue color and it contains holes, they are indicated by the white circles.
3. The power supply across the p-n junction makes the diode forward blasted and pushing the electrons from n-type to p-type. Pushing the holes in the opposite direction.
4. Electron and holes at the junction are combined.
5. The photons are given off as the electrons and holes are recombined.

4.2.9 Capacitors

A capacitor or condenser is a passive electronic component consisting of a pair of conductors separated by a dielectric. When a voltage potential difference exists between the conductors, an electric field is present in the dielectric This field stores energy and produce a mechanical force between the plates. The effect is greatest between wide, flat, parallel, narrowly separated conductors.

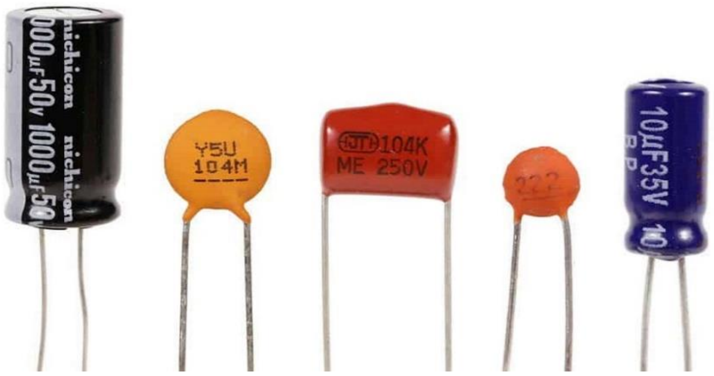


Figure 4.14 Types of Capacitor

In this project we are using 2 types of capacitor –

1. Electrolytic Capacitor – This Capacitor is used to provide clock pulse to Atmega 328.
2. Ceramic Capacitor – It is used to remove noise after the rectifier and to provide a noise free signal.

4.2.10 Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines.

Figure 4.15 Resistor

In this project we used resistor with two different values which are of 1k ohm & 10k ohm. The 10k ohm resistor is used to restrict current flow in clock pulse while the 1k ohm is used after the rectifier to remove extra current.

4.3 Hardware & Software Testing

4.3.1 Continuity test

In electronics, a continuity test is the checking of an electric circuit to see if current flows (that it is in fact a complete circuit). A continuity test is performed by placing a small voltage (wired in series with an LED or noise-producing component such as a piezoelectric speaker) across the chosen path If electron flow is inhibited by broken conductors, damaged components, or excessive resistance, the circuit is "open".

Devices that can be used to perform continuity tests include multimeters which measure current and specialized continuity testers which are cheaper, more basic devices, generally with a simple light bulb that lights up when current flows.

An important application is the continuity test of a bundle of wires to find the two ends belonging to a one of these wires, there will be a negligible resistance between the "right" ends, and only between the "right" ends. This test is the performed just after the hardware soldering and configuration has been completed.

This test aims at finding any electrical open paths in the circuit after the soldering wrong and rough handling of the PCB, improper usage of soldering iron, components failure and presence of bugs in the circuit diagram. We use a multi-meter to perform this test. We keep the multimeter to the ground and connected both the terminal across the path that needs to be checked.

If there is continuation, then you will hear the beep sound.

4.3.2 Power on test

This test is performed to check whether the voltage at different terminals is according to the requirements or not. We take a multi meter and put it in voltage mode. Remember that this test is performed without Microcontroller. Firstly, we check the output of the transformer, whether we get the required 5 volts.

Figure 4.16 Multimeter

Then we supply this voltage to the power supply circuit. Note that we do this test without microcontroller because if there is any excessive voltage, this may lead to damaging the controller.

We check for the input to the voltage regulator. Similarly, we check for other terminal for the required voltage. In this way we can assure that the voltage at all the terminals is as per the requirement.

4.3.3 Arduino

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards can read inputs - light on a sensor, a finger on a button, or a twitter message - and turn it into an output - activating a motor, turning on an LED. publishing something online Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits.

These may connect with add-on modules termed shields. Multiple, and possibly stacked shields may be individually addressable via an PC serial bus.

Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs. such as the Lily Pad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Figure 4.17 Arduino Uno

4.3.4 Ultiboard

National Instruments Electronics Workbench Group

The National Instruments Electronics Workbench Group (formerly Electronics Workbench) equips professional PCB designers with schematic capture, interactive simulation, board layout, and integrated test tools.

NI Ultiboard and NI Multisim form a complete platform to design, validate and layout printed circuit boards. The Ultiboard environment accelerates PCB design with automated functionality while maintaining precision with manual control. Together, Multisim and Ultiboard are a complex circuit design solution you can get to optimize the circuit's performance. Ultiboard complement it with PCB layout and routing capabilities that get a prototype into your hands as quickly and accurately as possible.

Easy-to-Use Design Environment

The NI Ultiboard interface enables efficient layout and routing of PCB designs, Integration with NI Multisim allows seamless transfer of schematics to layout. The customizable environment ensures accessibility to desired features for immediate productivity. Tools such as the spreadsheet view. toolbox and design wizards easily manage, control and define any board layout. Part placement and copper routes are optimized to either allow full control for precise definition of critical parts or automation for design completion. The 3D preview renders a completed, populated board to gain a virtual perspective before production,

Ultiboard exports and produces industry standard format such as Gerber and DXF to take a final, optimized board to prototype and manufacture. The Ultiboard product editions (Full, Power Pro) provide a complete set of tools for professional PCB layout:

1. Intuitive, user-friendly design environment
2. Integrated spreadsheet view for managing design constraints, part placement, and copper routing
3. Easy-to-use design tools optimized for speed or precise control
4. Export to industry standard file formats such as Gerber and DXF for prototype and manufacturing.

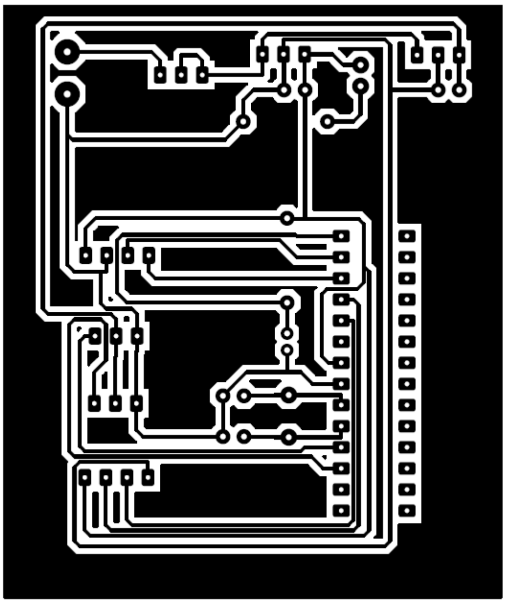


Figure 4.18. Ultiboard Simulation of the project

4.3.5 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language.
It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

4.3.6 OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

4.4 Process

The process is about the steps involved in the working of the system i.e. how it is turned on and various steps which are involved.

1. Firstly, we will have to make sure that we have all the parts, components, devices and batteries for power supply.

2. Then we have used Arduino UNO board and its IC ATmega328P microcontroller to interface with ESP32 CAM module through the Arduino IDE to create a webserver for the camera output.

3. From the web server we have generated an IP address from the serial monitor of the Arduino IDE and then through which we can access the output of the camera using our IOT devices (i.e. laptop).

4. Then we have interfaced our PAN TILT servo with the microcontroller and used Arduino IDE for that and from here we can control our PAN TILT servo through the serial monitor.

5. And we have also interfaced the Ultrasonic sensor with the microcontroller which will provide the distance of the object, the horizontal and the vertical angle.

6. Then we have installed python IDE (i.e. PyCharm)

7. To automate all this and integrate the OpenCV for object detection & identification we have used Python.

8. After this we have putted the camera IP address into the Python code and key mapped the PAN TILT servo to control that from the laptop's keyboard.

9. After this connecting the USB to TTL converter in USB port of the IOT device (i.e. laptop).

10. Then after running the python code a popup window will appear that will include the output of the camera which is detecting the objects.

11. By providing the keyboard input we can control the PAN TILT direction i.e. by pressing the key 'W','S','A' & 'D' from the keyboard it moves UP, DOWN, LEFT & RIGHT respectively.

12. And by pressing the input 'R' from the keyboard it will show the distance, horizontal and the vertical angle.

Chapter 5

Result and Discussion

5.1Result

- The Final Product is shown if Figure 5.1

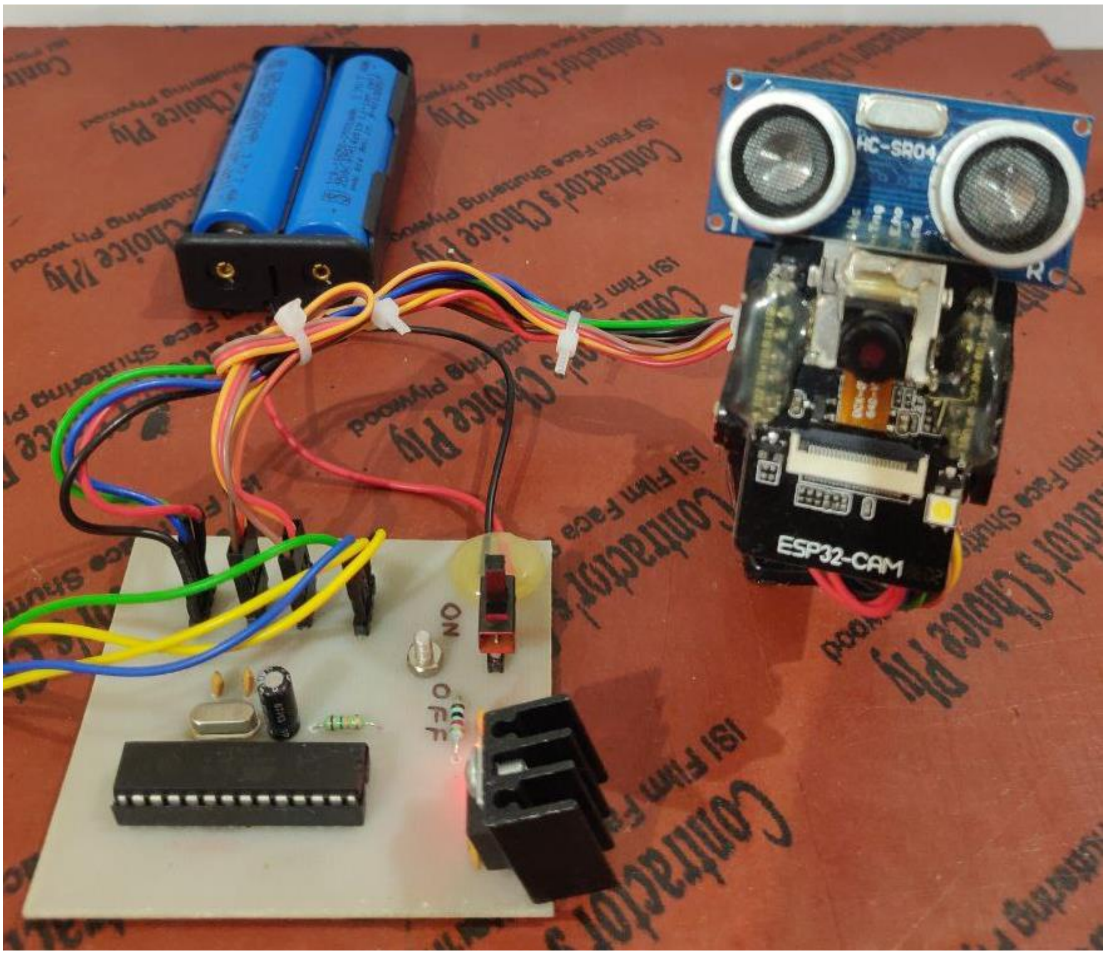


Figure 5.1 Product Design

- Keyboard of IoT device (Laptop) is used to control Pan-Tilt Servo Motor as shown in Figure 5.2

  W = Up;

  S = Down;

  A = Left;

  D = Right;

  R = Angle at which Ultrasonic sensor is pointing towards.
- Every press results in 10 degree of shift.

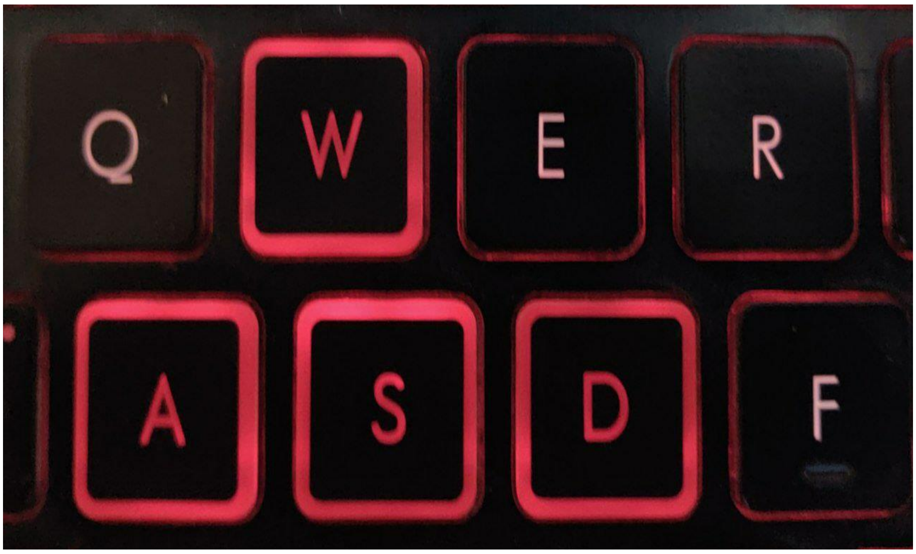


Figure 5.2 Keys of IoT to control Pan-Tilt Servo motor

- Movement in Pan result's in change in Horizontal Angle, while change in Tilt servo result in change in vertical angle.
- Figure 5.3 shows Pan-Tilt servo motor with mounted ultrasonic sensor and ESP cam module.

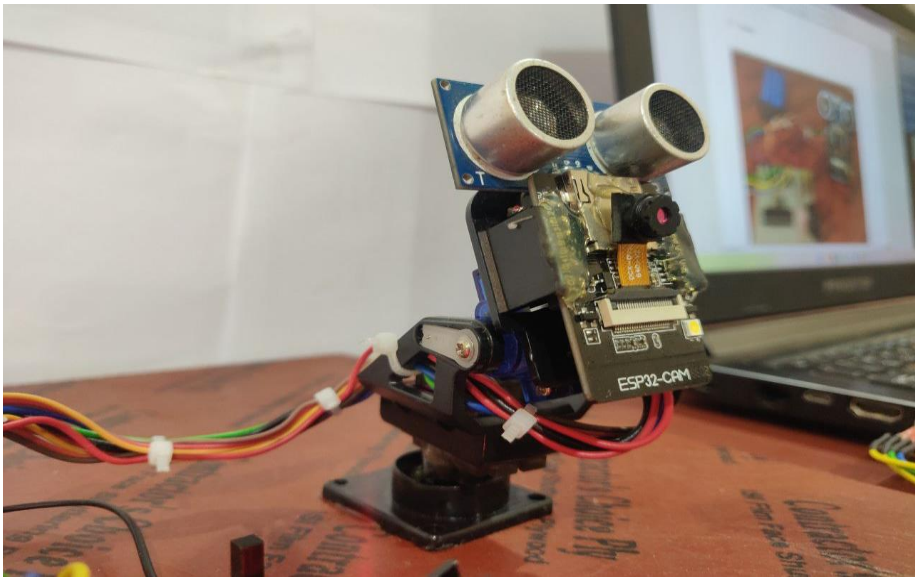


Figure 5.3 Pan-Tilt servo motor

- For working of the final system, the switch in the PCB must be turned ON.
- USB to TTL must be connected to the IoT device (laptop) as shown in Figure 5.4.

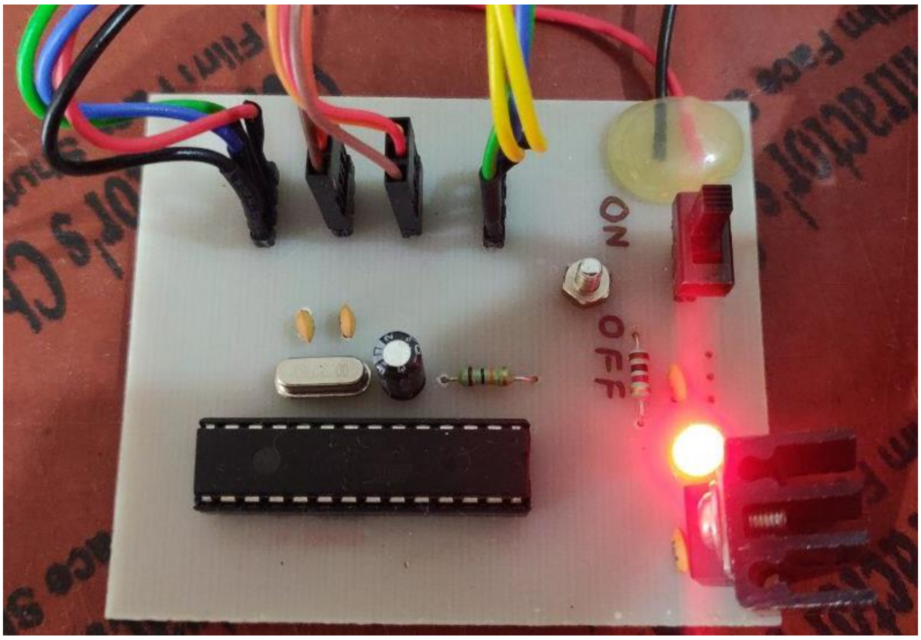
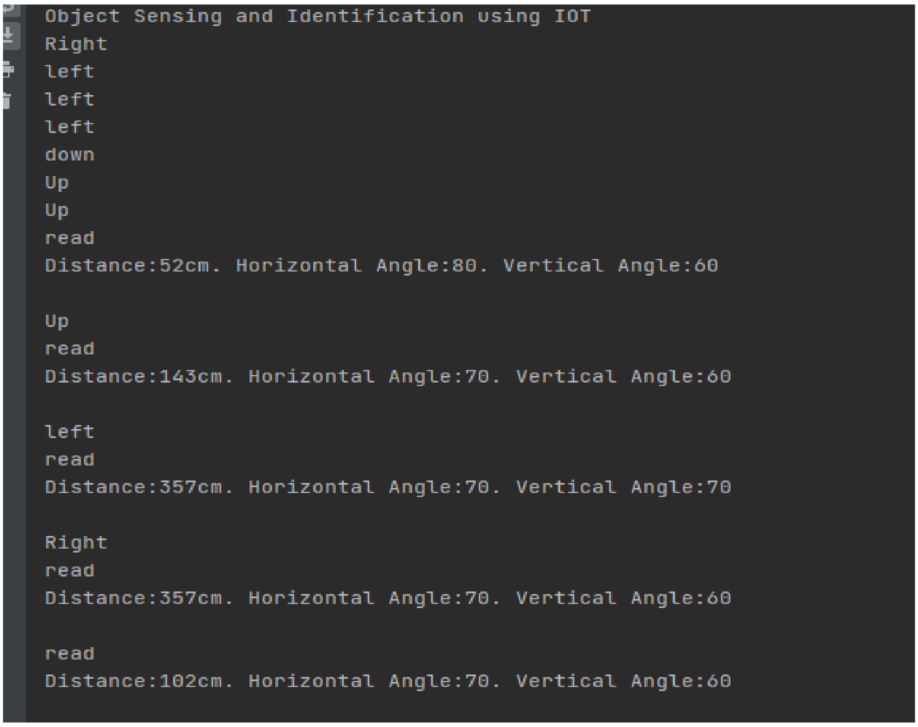


Figure 5.4 PCB layout of the system

- For getting the Angle and distance reading “R” key must be pressed in the console of PyCharm.
- Figure 5.5 represents console output of ultrasonic sensor for objects at different distance.



```
Object Sensing and Identification using IOT
Right
left
left
left
down
Up
Up
read
Distance:52cm. Horizontal Angle:80. Vertical Angle:60

Up
read
Distance:143cm. Horizontal Angle:70. Vertical Angle:60

left
read
Distance:357cm. Horizontal Angle:70. Vertical Angle:70

Right
read
Distance:357cm. Horizontal Angle:70. Vertical Angle:60

read
Distance:102cm. Horizontal Angle:70. Vertical Angle:60
```

Figure 5.5 Distance by Ultrasonic sensor on console for multiple distance objects

- Output of ESP 32 cam is shown after object identification using OpenCV is shown on output panel in PyCharm as shown in Figure 5.6 and 5.7.
- Figure 5.6 shows multiple object identification of same class.

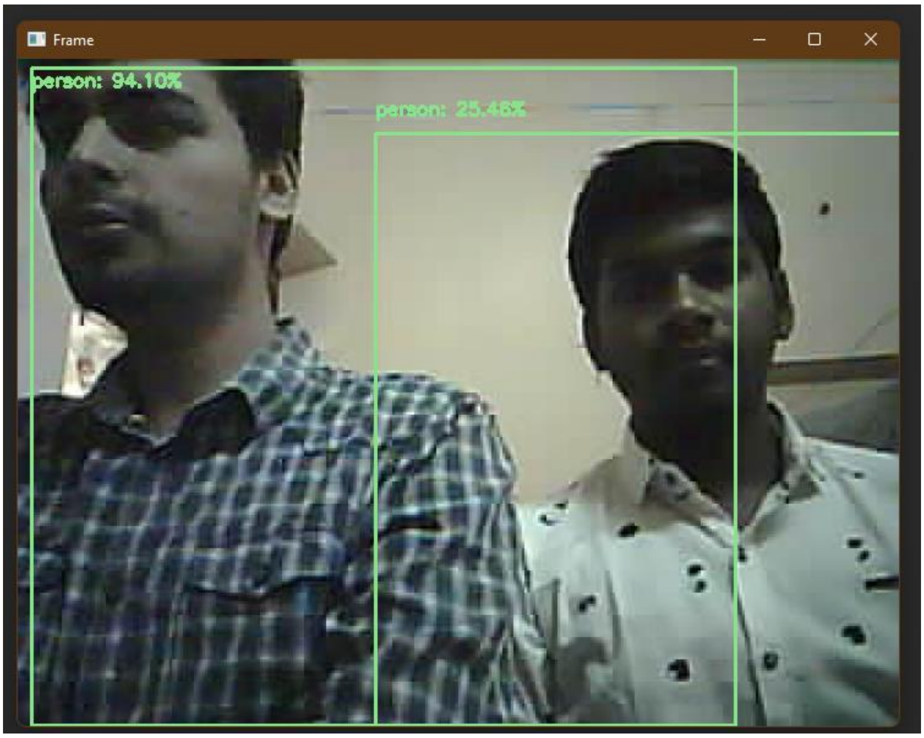


Figure 5.6 Multiple Subject Identification of Same Class

- Figure 5.7 shows multiple object identification of different class.

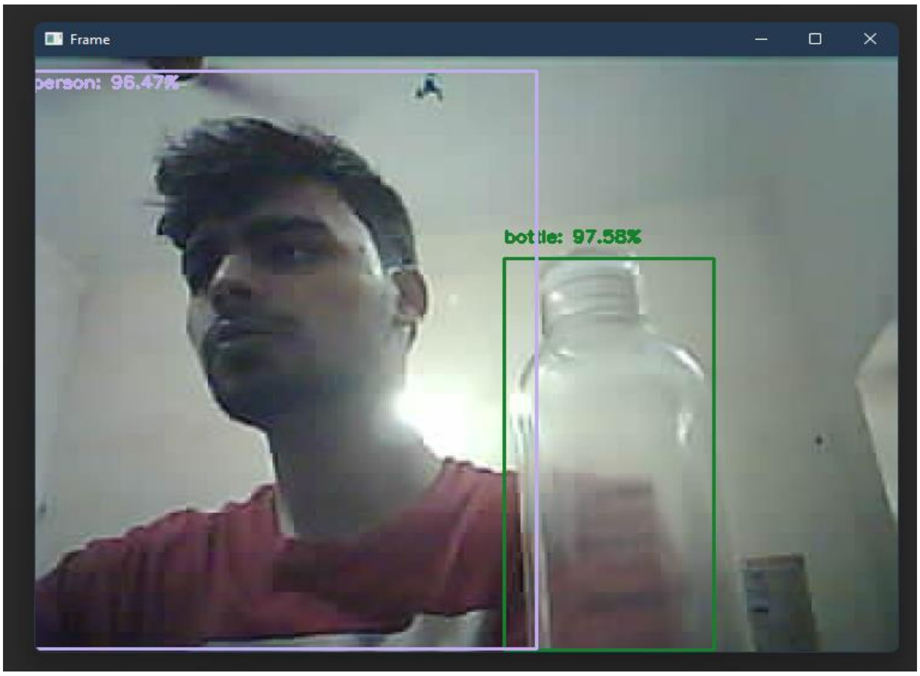


Figure 5.7 Multiple object identification of different class

- Initially when the power is turned on the Pan-Tilt servo Calibrates itself to 90 degree both vertically and horizontally.

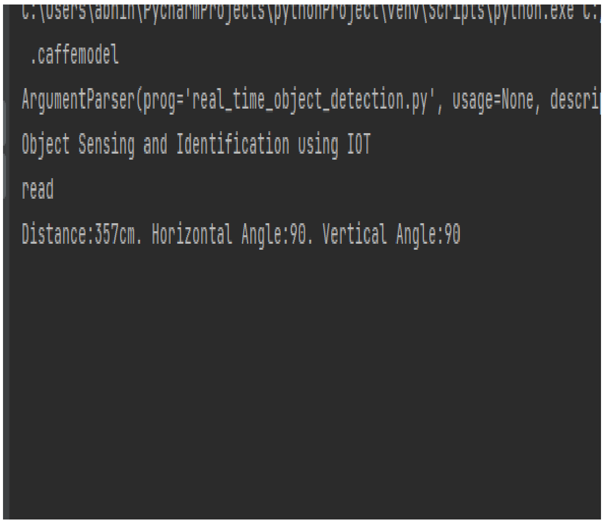


Figure 5.8 Initial Starting Position

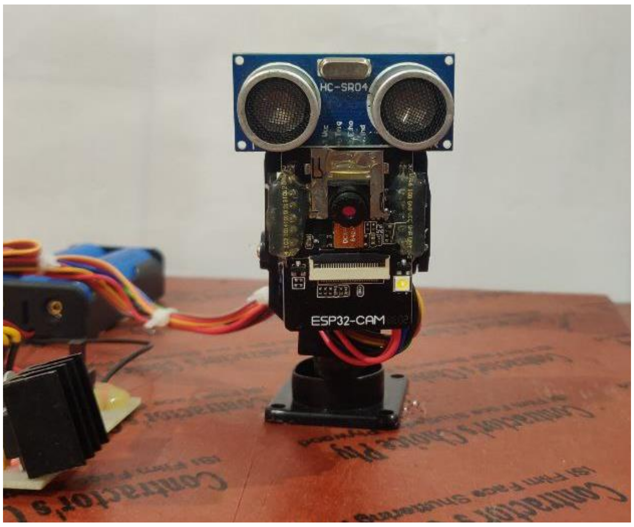


Figure 5.9 Initial Start Position of Pan-Tilt servo

- In the console of PyCharm the Input command to shift down for Tilt servo and output of Ultrasonic sensor is given in Figure 5.10.
- The response of Tilt Servo motor for the input given in Figure 5.10 is shown in Figure 5.11.

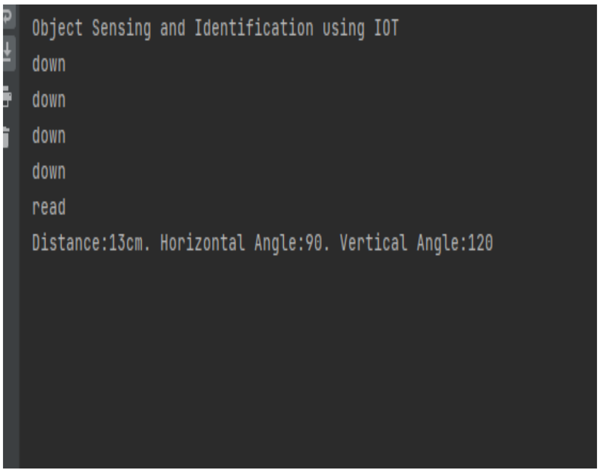


Figure 5.10 Input at console for down shift

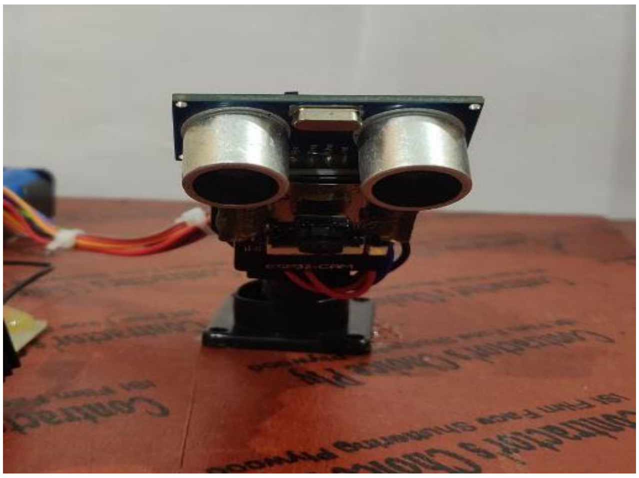


Figure 5.11 output of Pan-Tilt w.r.t figure 5.10

- Input for left shift of Pan servo motor while the Tilt servo is at down position is given and than the reading taken by Ultrasonic sensor is taken as shown in Figure 5.12.

- Output movement for dervo motor for the input given at console in Figure 5.12 is shown in Figure 5.13.

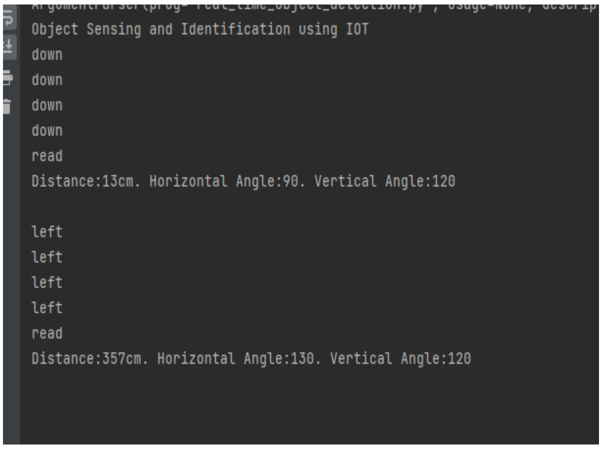


Figure 5.12. Input at console for left shift

Figure 5.13. output w.r.t figure 5.12

- Pan motor is given input to turn right while tilt motor is in down posotion and than output of ultrasonic sensor is taken as shown in Figure 5.14.
- The output of the console as shown in Figure 5.14 in the Pan tilt servo is shown in figure 5.15.

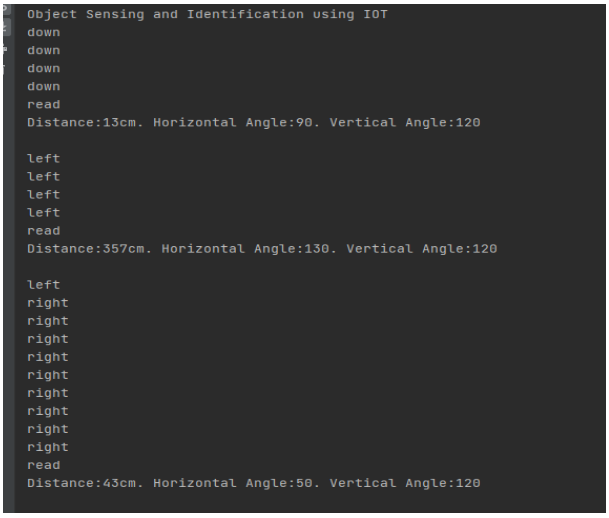


Figure 5.14 Input at console for right shift

Figure 5.15 output w.r.t figure 4.14.

- Tilt motor is given command to shift Up and the outpur of ultrasonic senser is taken as shown in Figure 5.16.
- Output of the command given at the servo is given in figure 5.17

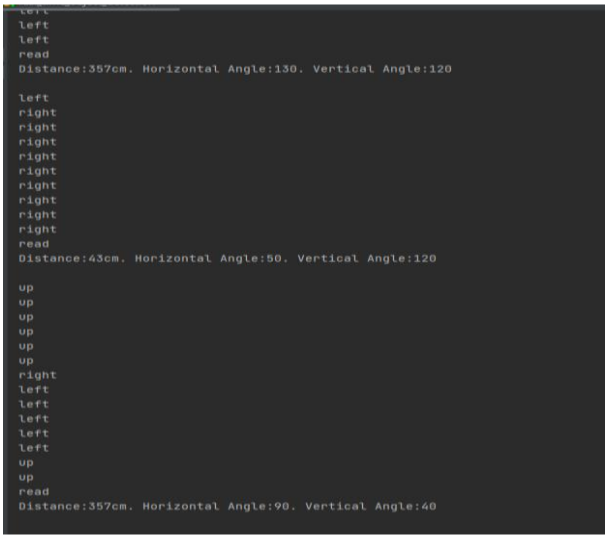


Figure 5.16. Input at Console for up shift

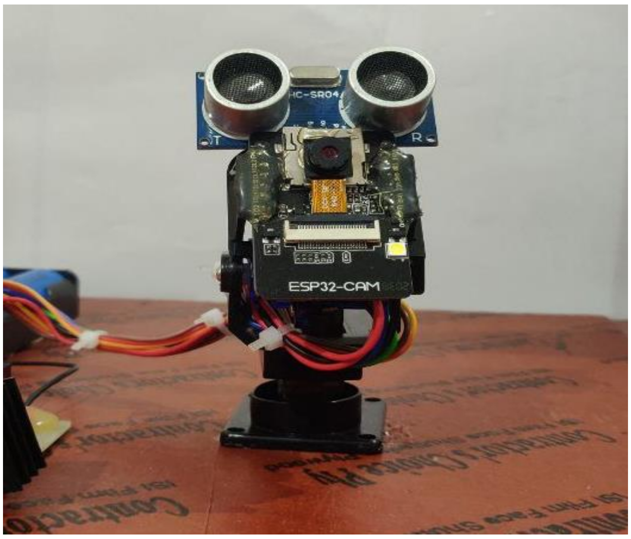


Figure 5.17. output at console w.r.t Figure 5.16.

Chapter – 6

Conclusion and Future Scope

6.1 Conclusion

- This project provides a module which helps identify object and calculate its distance from the system.
- The camera module and ultra-sonic sensor are mounted on top of pan-tilt servo motor which can be adjusted (both horizontally and vertically) according to the requirement.

6.2 Future Scope

- **Use in autonomous driving-**
  Radar/Lidar System can be added for wide object sensing and identification in autonomous driving.
- **Security Alert system-**
  The system can be programmed to alert the user when a specific object is detected.
- **Military application-**
  The system can be implemented for sensing ground & aerial objects and help to identify them.

Reference

[1] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz and Markus Lienkamp “A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection” IEEE 2019 Sensor Data Fusion: Trends, Solutions, Applications Conference (SDF) - Bonn, Germany 2019.

[2] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu and Xindong Wu “Object Detection with Deep Learning: A Review” IEEE Neural Network and learning system 2019.

[3] Lijun Yu, Weijie Sun, Hui Wang, Qiang Wang and Chaoda Liu “The Design of Single Moving Object Detection and Recognition System Based on OpenCV” IEEE International Conference on Mechatronics and Automation (ICMA) - Changchun, Jilin, China 2018.

[4] Huanlei Chen, Meisong Tong and Libo Huang “Target Detection and Tracking Algorithm Simulation for Automotive Millimeterwave Radar Based on Systematic Softeware-based Radar System” IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting - Boston, MA, USA 2018.

[5] Stanisław Deniziak, Tomasz Michno, and Pawel Pieta “IoT-Based Smart Monitoring System Using Automatic Shape Identification” Advances in Intelligent Systems and Computing Proceedings of the Federated Conference on Software Development and Object Technologies Volume 511 2015.

[6] Bernd Heisele, Ivaylo Riskov and Christian Morgenstern “Components for Object Detection and Identification” Toward Category-Level Object Recognition Volume 4170 2006.

Appendix

Code in Aurdino :

```
const int trigPin = 7; long duration; int distance; const int echoPin = 6;
#include <Servo.h>
Servo myservo;
Servo myservo1;
int val = 90; int val1 = 90;
void setup() {
// put your setup code here, to run once:
Serial.begin(9600);
myservo.attach(9);
myservo1.attach(10);
myservo.write(val); myservo1.write(val1);
}
void loop()
{
calculateDistance();
Serial.print("|| Distance= "); Serial.print(distance); Serial.println("Cms  ||");
if (Serial.available() > 0)
{
char a = Serial.read();
if (a == 'W')
{ if (val < 180)
{
val = val + 10;
myservo.write(val);
}
```

```
}
if (a == 'S')
{ if (val > 0)
{
val = val - 10;
myservo.write(val);
}
}
if (a == 'D')
{ if (val1 < 180)
{
val1 = val1 + 10;
myservo1.write(val1);
}
}
if (a == 'A')
{ if (val > 0)
{
val1 = val1 - 10;
myservo1.write(val1);
}
}
}
delay(50);
}
```

- Code for Capturing ESP-32 Cam Video Feed

```
import cv2
from urllib.request import urlopen
import numpy as np

url = r'http://192.168.1.70/capture'
while True:
    img_resp = urlopen(url)
    imgnp = np.asarray(bytearray(img_resp.read()), dtype="uint8")
    img = cv2.imdecode(imgnp, -1)
    cv2.imshow("Camera", img)
    if cv2.waitKey(1) == 'q':
        break
```

- Code for Object Identification and Servo Control

```
# USAGE
# python real_time_object_detection.py --prototxt
MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.caffemodel

# import the necessary packages
from imutils.video import VideoStream
import datetime
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2
from urllib.request import urlopen
import numpy as np
import pyttsx3

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
print(ap)
ap.add_argument("-p", "--prototxt", required=True,help="path to C"
                                                       "affe 'deploy' prototxt
file")
ap.add_argument("-m", "--model", required=True,help="path to Caffe pre-trained
model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,help="minimum
probability to filter weak detections")
print('hari')
args = vars(ap.parse_args())



# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
   "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
   "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
   "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
url = r'http://192.168.1.70/capture'
# loop over the frames from the video stream
while True:
   # grab the frame from the threaded video stream and resize it
   # to have a maximum width of 400 pixels
```

```
    img_resp = urlopen(url)
    imgnp = np.asarray(bytearray(img_resp.read()), dtype="uint8")
    frame = cv2.imdecode(imgnp, -1)
    frame = imutils.resize(frame, width=720)

    # grab the frame dimensions and convert it to a blob
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
        0.007843, (300, 300), 127.5)

    # pass the blob through the network and obtain the detections and
    # predictions
    net.setInput(blob)
    detections = net.forward()

    # loop over the detections
    for i in np.arange(0, detections.shape[2]):
        # extract the confidence (i.e., probability) associated with
        # the prediction
        confidence = detections[0, 0, i, 2]

        # filter out weak detections by ensuring the `confidence` is
        # greater than the minimum confidence
        if confidence > args["confidence"]:
            # extract the index of the class label from the
            # `detections`, then compute the (x, y)-coordinates of
            # the bounding box for the object
            idx = int(detections[0, 0, i, 1])
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")

            # draw the prediction on the frame
            label = "{}: {:.2f}%".format(CLASSES[idx],
                confidence * 100)
            cv2.rectangle(frame, (startX, startY), (endX, endY),
                COLORS[idx], 2)
            y = startY - 15 if startY - 15 > 15 else startY + 15
            cv2.putText(frame, label, (startX, y),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
            print(CLASSES[idx])


    # show the output frame
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    # if the `q` key was pressed, break from the loop
    if key == ord("q"):
        break


# do a bit of cleanup
cv2.destroyAllWindows()
```