BORDER SURVEILLANCE SYSTEM AND WEAPON ACTIVATION USING IOMT

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

J. DILEEP KUMAR (319126512021) **B. ADITYA PATRO** (319126512004)
K. SAI CHARAN (319126512026) **U. CHAITANYA TEJA** (319126512011)

Under the guidance
Of
Dr. N. SWATHI
Associate Professor, ECE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(permanently Affiliated to AU, Approved by AICTE & Accredited by NBA & NAAC)
Sangivalasa-531162, Bheemunipatnam Mandal, Visakhapatnam District(A.P)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC*)
Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA
ANITS

CERTIFICATE

This is to certify that the project report entitled **"BORDER SURVEILLANCE SYSTEM AND WEAPON ACTIVATION USING IOMT"** submitted by **J.DILEEP KUMAR(319126512021), B. ADITYA PATRO(319126512004), K. SAI CHARAN(319126512026), U. CHAITANYA TEJA(319126512011)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Anil Neerukonda Institute of technology and Sciences(A), Visakhapatnam is a record of bonafide work carried out under my guidance and supervision

Project Guide

Dr. N. SWATHI
Associate Professor
Department of ECE
ANITS

Associate Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department

DR. B. JAGADEESH
Professor & HOD
Department of ECE
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Dr. N. SWATHI**, Associate Professor, Department of Electronics and Communication Engineering, ANITS, for her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Prof. B. JAGADEESH**, Head of the Department of Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa** , for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of the Department of ECE, whose suggestions during reviews helped us in I accomplishment of our project. We would like to thank non-teaching staff of the Department of ECE, ANITS for providing great assistance in accomplishing our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. Last but not these, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

J. DILEEP KUMAR (319126512021)

B. ADITYA PATRO (319126512004)

K. SAI CHARAN (319126512026)

U. CHAITANYA TEJA (319126512011)

ABSTRACT

Our country India shares border with seven different countries so we absolutely need to keep an open eye 24/7.This long stretch of the border makes it really challenging and uses a lot of human resource which is not ideal. So, introducing a smart border surveillance system consisting of Intruder Detection System(IDS) and weapon activation using Internet of Things ,which can perform surveillance across the border at the hostile conditions ,without requiring any human assistance at the border.

The surveillance system detects any intrusions at the border like terrorist infiltrations, illegal movements , the system shares the intrusion information directly to the command control room. This system also warns any hostile body by activating the weapons by relaying the information to the command centre and follows their command. Advantages of this surveillance system is to eliminate the risk to border security forces and uses a lot of less manpower and also reduces the risk to border security forces. So, here designing the surveillance system which helps enhancing the security of our border especially at the terrain and extreme climatic conditions

CONTENTS

‘LIST OF FIGURES

LIST OF TABLES

CHAPTER -1

INTRODUCTION TO IOT

1.1 Introduction

The Internet of Things (IoT) has become a significant topic in the technology industry, policy debates, and engineering communities. It has also garnered attention from both specialized and general media. This technology enables various connected devices, systems, and sensors to utilize improvements in processing power, electronic miniaturization, and network connectivity to provide enhanced capabilities that were previously unattainable. The IoT ecosystem comprises an array of devices, including sensors, actuators, radio frequency identification tags, and mobile phones, among others. These devices use different addressing schemes to communicate with each other and collaborate towards common goals.

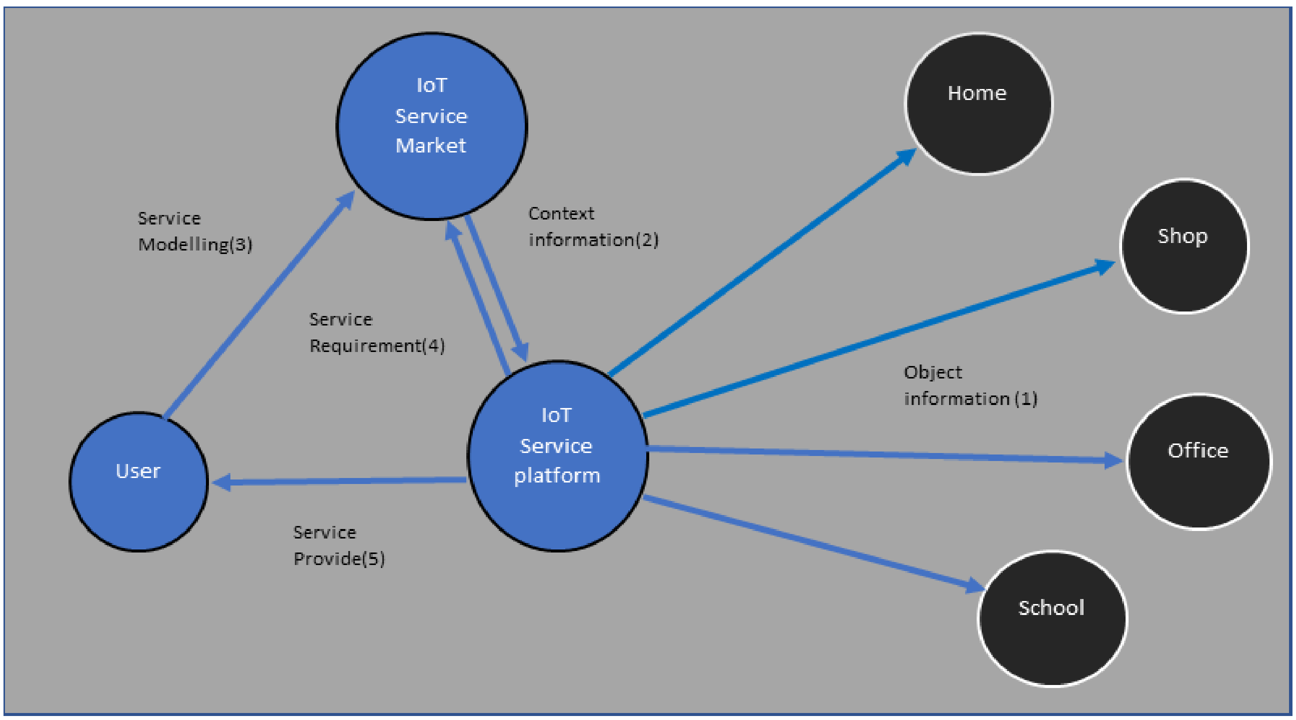


Figure 1.1 : IoT Service Model for Smart Space Management

The Internet of Things (IoT) is a technology that seeks to connect a vast array of objects with unique addresses, enabling them to communicate and work together seamlessly. This area is gaining increasing attention due to its technical, social, and commercial significance. The integration of powerful data analytics capabilities and Internet connectivity into ordinary objects such as consumer goods, durable goods, transportation equipment, industrial and utility components, and sensors has the potential to fundamentally transform the way we live, work, and play. According to projections, IoT could have a significant impact on the economy and the internet, with up to 100 billion IoT devices connected by 2025 and a potential economic impact of over $11 trillion.

What is the Internet of Things?

In 1999, Kevin Ashton, a prominent British technology pioneer, introduced the term "Internet of Things" (IoT) to describe a system that uses sensors to connect tangible objects to the Internet. The goal of Ashton's creation was to demonstrate the potential of linking Radio Frequency Identification (RFID) tags, which are commonly used in corporate supply chains, to the Internet, enabling them to be tracked and monitored with minimal human intervention. Nowadays, the phrase "Internet of Things" is frequently employed to describe situations where an assortment of devices, sensors, and everyday objects are interconnected to the Internet and possess computing capabilities. Although the idea of utilizing computers and networks to manage and monitor devices is relatively new,

A) **Ubiquitous Connectivity:** The prevalence of high-speed, affordable, and easily accessible network connections, including licensed and unlicensed wireless technologies, has made it possible for almost anything to be connected, leading to ubiquitous connectivity.
B) **Advances in Data Analytics**— Recent advancements in computing power, data storage, and cloud services have facilitated the development of new algorithms that can efficiently gather, correlate, and analyse vast amounts of data. These large and constantly evolving datasets have created new opportunities for extracting information and knowledge.
C) **Emergence of Cloud Computing** - Cloud computing empowers small and geographically dispersed devices to link up with potent back-end analytical and control capabilities by leveraging remote and networked computing resources for processing, managing, and storing data.

1.2 Definition and Example of IOT

IoT Definition: The term "Internet of Things" usually refers to scenarios where common items, including sensors, household objects, and other non-computing items, are interconnected to networks and possess computational abilities. This allows these devices to generate, exchange, and consume data without requiring human intervention. While there is no widely accepted definition for this term, it is a concept that deserves recognition

Definition by ITU-T

Looking at it from a wide perspective, the Internet of Things (IoT) can be perceived as a vision that has implications for both technology and society. In terms of technical standardization, the IoT can be regarded as a global infrastructure for the information society that enables sophisticated services by connecting objects using interoperable data collection, processing, and communication capabilities that already exist or are being developed. By fully utilizing objects, the IoT can provide services for a range of applications while still ensuring necessary privacy measures.

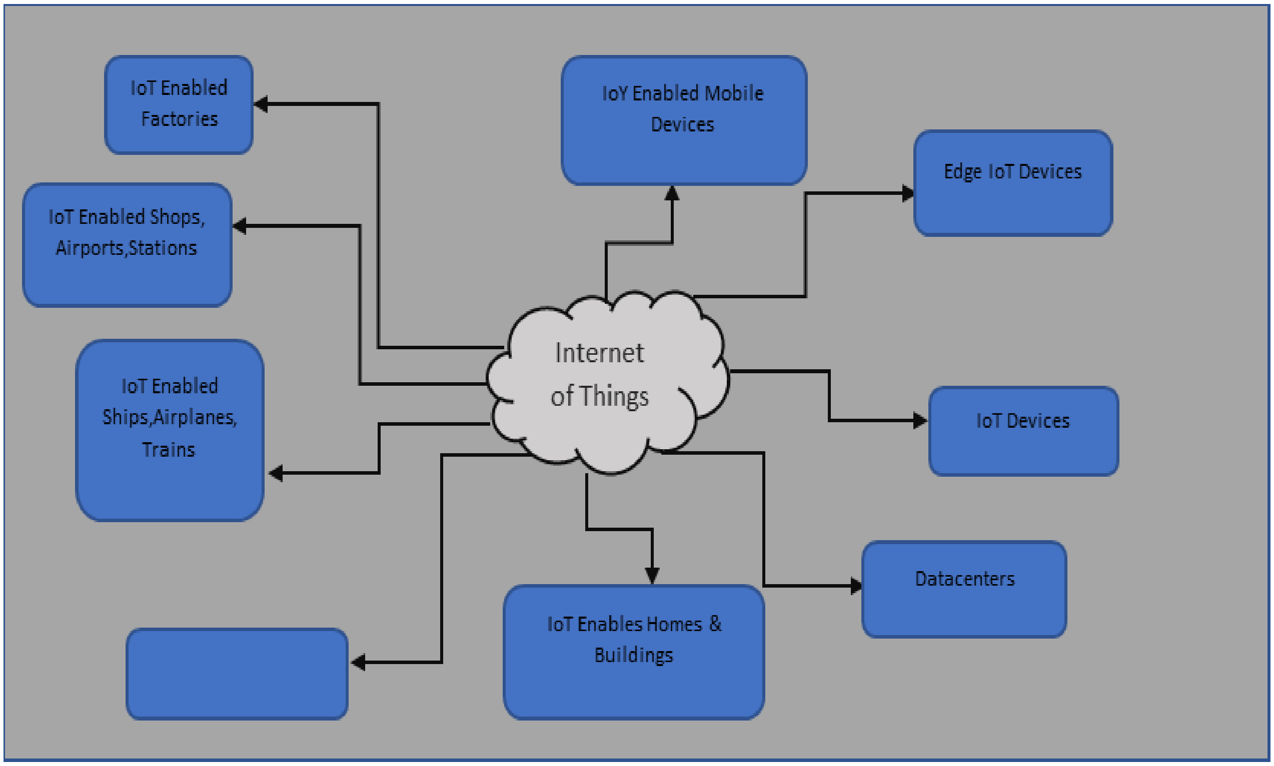


Figure 1.2: An Example of IoT

1.2.1 Elements of IOT

The Internet of Things (IoT) comprises of six main components, namely semantic, identification, sensing, computing, communication technologies, and computation.

A) Identification is essential for designating services and connecting their demand to those offerings.
 Ex: such as the electronic product code (EPC)

B) The act of collecting various data from interconnected objects and transmitting it to a database is known as sensing.
 Ex: temperature sensors

C) Communication technologies link disparate objects together to provide unique services.
 Ex: Wi-Fi, Bluetooth

D) This task is carried out through computation, hardware processors, and software programmes.

E) The IoT includes services for identity management, information aggregation, collaborative awareness, and ubiquitous use.

F) Semantic refers to the capacity to deduce knowledge in an informed manner to offer the necessary services.

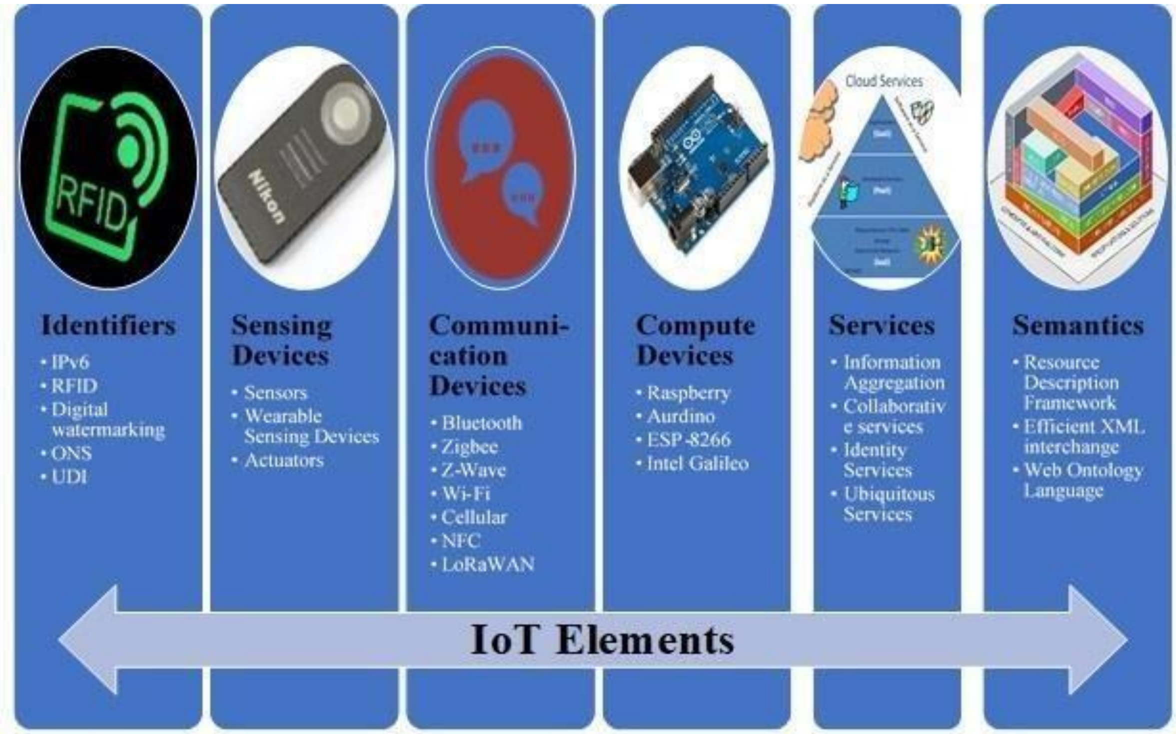


Figure 1.3: Elements of IoT

1.3 Applications of IOT

IoT technology finds applications across diverse domains. Here are some examples of IoT applications:

A) **Smart Home:** The use of IoT technology enhances the quality of life at home by enabling easy and remote monitoring and control of home appliances and systems.

B) **Industrial Automation:** Robotic gadgets automate production activities with the least amount of human intervention. Also, productivity is increased.

C) **Smart Healthcare:** By incorporating sensors and actuators into patients' bodies and their medications, healthcare applications perform better by monitoring and tracking patients. Using sensors, for instance, to collect and analyse patient body data before sending it to a processing facility.

D) **Smart Grid:** The increasing demand for power due to population growth can be addressed by power suppliers utilizing IoT technology to manage and control resources. As a result, buildings and homes may require a higher energy supply. Energy providers can enhance their services by connecting meters to their network and using data analytics to gather, monitor, and manage energy consumption in real-time.

E) **Smart City:** Making it quicker and more straightforward for city people to find relevant information improves the quality of life there. For instance, a number of networked systems intelligently give consumers the desired services (such as transportation, utilities, health, etc.) based on their needs.

F) **Internet of Things in health monitoring:** IoT technology offers a solution for power suppliers to manage and control resources in response to the growing energy demands caused by population expansion. Consequently, there is a need to increase the energy supply for buildings and homes. Energy providers can optimize their services by integrating meters with their network and utilizing data analytics to collect, track, and manage energy usage in real-time.

1.4 Pros and cons of IoT:

The benefits of IoT technology are numerous and may include:

a) The ability to access information on any device, from anywhere, at any time;

b) improved connectivity and communication between connected technological devices;

c) IoT technology can help businesses save time and costs by automating data transmission over network connections, thereby improving service quality and reducing the need for human intervention.

There are several drawbacks associated with IoT technology, which may include:

a) As the number of connected devices and shared information increases with IoT technology, the risk of hackers stealing private data also escalates.

b) As the number of IoT devices in businesses continues to rise, potentially reaching millions, the task of monitoring and collecting data from these devices becomes increasingly challenging.

c) It's likely that every linked device will become corrupted if the system has a problem.

d) It is challenging for devices from various manufacturers to connect with one another because there is no global standard for IoT compatibility.

1.5 IoT standards and frameworks:

Standards and frameworks play a vital role in the Internet of Things (IoT) realm, ensuring that IoT devices and systems can interoperate, communicate securely, and scale effectively. These guidelines, protocols, and specifications provide a foundation for seamless communication and interaction among diverse IoT devices. Prominent examples of IoT standards and frameworks include MQTT, CoAP, Zigbee, LoRaWAN, OPC UA, IoTivity, and Wi-Fi. It is crucial to carefully select the appropriate standard or framework that aligns with the specific requirements of an IoT use case to ensure smooth interoperability, communication, and security within the IoT ecosystem.

1.6 IoT security and privacy issues:

IoT (Internet of Things) is faced with unique challenges in terms of security and privacy, stemming from the interconnected nature of devices, data, and networks. Several common issues include device security, data security, network security, privacy concerns, supply chain security, lack of standards, and human factors. For instance, vulnerabilities in IoT devices may arise from weak authentication, default passwords, or outdated firmware. Data security risks can result from the massive volume of data generated and transmitted by IoT devices, leading to potential breaches or unauthorized access. Network security is crucial for protecting communication channels among IoT devices, and privacy concerns arise due to the collection and processing of personal and sensitive data. The complexity of supply chains in IoT can introduce security risks, and the lack of standardized security protocols can result in inconsistencies. Additionally, human factors such as error, negligence, and lack of awareness can impact IoT security. To effectively address these challenges, a comprehensive approach is needed, encompassing robust security measures

CHAPTER- 2

OVERVIEW OF THE PROJECT

2.1 Aim of the Project:

The project's primary objective is to offer low-cost, effective border monitoring and weapon activation if an intruder crosses the boundary.

2.1.1 Objectives:

1. To design and develop an automated border security system using IOT.
2. Monitoring and sensing the intruder infiltration.
3. Activating the alarm and message alert given to the control room.
4. Automated weapon activation when the intruder trespasses the border.

2.1.2 System statement of scope:

Our nation's border security is currently our top priority. Our soldiers are manning these boundaries. Monitoring in terrains like rivers, valleys, hilly areas, etc. is comparatively more difficult to guard and where it is very risky for our soldiers, but it can now be reduced thanks to IOT-based technology. The term pertains to safeguarding a nation's boundaries from the unlawful transportation of items, narcotics, armaments, and individuals. They make extraordinary sacrifices as a result of these evil factors. If we could have only spared half as many lives. In previous hotter periods, this might not have been feasible, but things are different now. A technologically nimbler world is one we are moving towards.

The purpose of the border surveillance systems that we developed was to keep an eye on activity near the borders and detect any suspicious activity. A set of planned tasks are carried out if anything happens that raises suspicion. It entails notifying the appropriate authorities. IDS (Intruder Detection Systems) are crucial to border surveillance. They are made to work in dangerous environments where they constantly watch out for, find, and pursue trespassers (moving targets). Knowing how difficult it is for people to pay close attention to the live video streaming that occurs continuously around-the-clock, A system for intrusion detection that has the ability to send out automatic alerts can be quite helpful. The top of a structure has a video surveillance camera fixed on it. whenever the microcontroller receives a signal from PIR sensors and sends an object detection message to the control room in case the PIR sensor senses any motion. The camera rotates 180 degrees and can be controlled from the control room, where they may view live video from the border.

2.1.3 Methodology:

The suggested solution can serve as a smart border monitoring system for our border security officers. This technology offers video surveillance in locations where human deployment is challenging. Several ultrasonic and pyroelectric infrared sensors (PIR) are fixed to the fencing to monitor the border region for any entry. Radar vision is displayed using ultrasonic sensors that are coupled to a stepper motor that spins 180 degrees both clockwise and anticlockwise.

Connectors are used to connect the Arduino board to the ultrasonic sensor. VCC, GND, and an Analog Out pin are its three pins. Among other connections, the Arduino board is linked to pins A0, GND, and VCC.

The ultrasonic sensor provides sensor data to the Arduino board via the attached A1 pin. The ATmega328p microprocessor serves as the basis for the Arduino UNO microcontroller board. The Arduino Uno is equipped with 6 analog inputs, a USB connection, and a total of 14 digital input/output pins. The processing software receives this data in serial, and the monitor displays the radar view. The ESP8266 Wi-Fi module with a patch antenna and 8 pins of Node MCU, which also includes a processor, is linked to the pir sensor. The RX and TX pins are utilised during data transmission and reception. With the help of the esp8266 board, the sensor data is uploaded to the cloud server.

The IFTTT Platform, which offers free cloud storage, has been employed as a cloud server in the aforementioned system. Sending an alert message, the video cameras are put far away to continuously monitor the border area, and if any movement is noticed in the space being watched by the sensors, the control room controls the camera locations. We can do with the aid of live video streaming. To the controller is sent a warning message. A sound system is in place to alert the invader and demand his surrender. After a couple of warnings, if the intruder still refuses to leave, an auto-combat system is triggered. To trigger the weapon, we are using IR sensors that are connected to an Arduino Uno. The Arduino Uno uses the sensor data to determine the direction of the intruder and then sends the command to the stepper motor, which is connected to the laser gun. In order to allow the closest military installation enough time to get ready for the necessary operations, it is crucial to send the incursion alarm there. Also, it is possible to watch the live video streaming, which is wirelessly transmitted to the control room.

2.1.4 System Context:

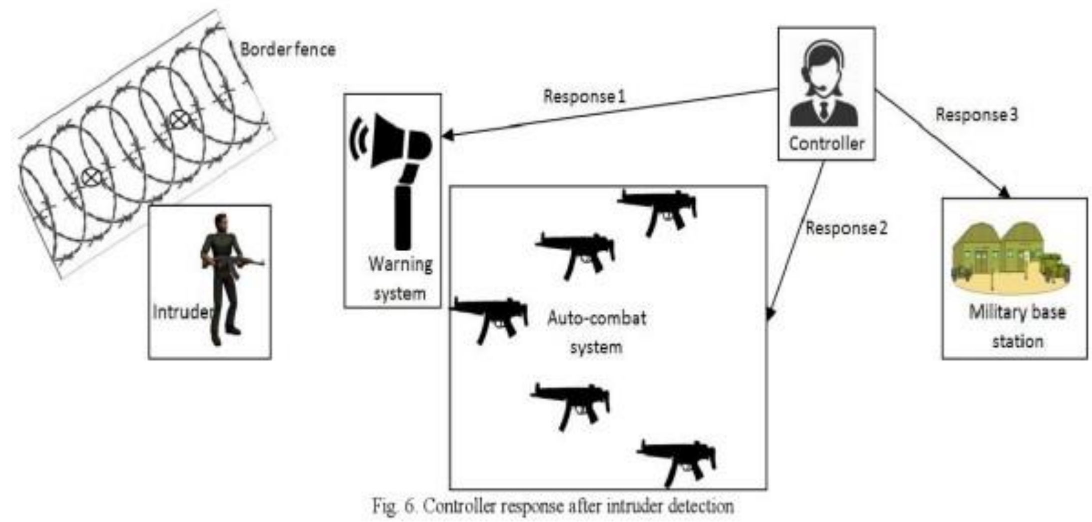


Figure 2.1: System Context Diagram

2.2 Block diagram:

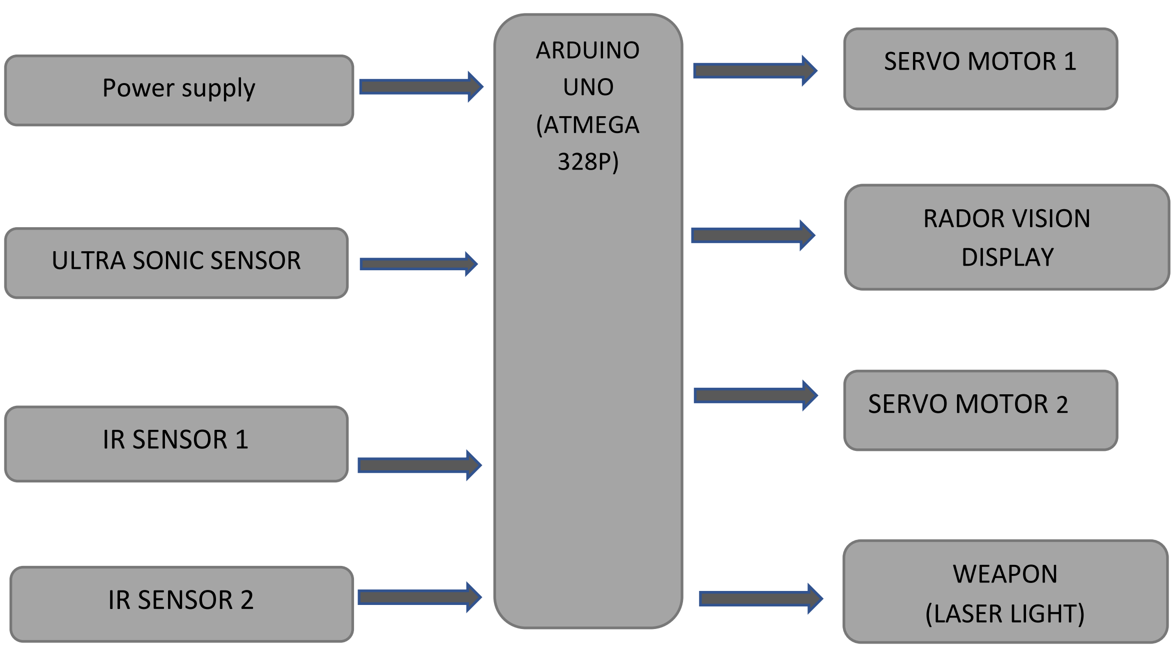


Figure 2.2.1: Block diagram of Radar vision & weapon activation

For the hardware design, we are utilising a node MCU that includes an esp8266 WIFI module and an Arduino uno microcontroller that both run at 5 volts. The Arduino Uno, which is connected to the digital pins in the Arduino, is used to interface with ultrasonic sensors and infrared sensors. The echo and trigger pins of the ultrasonic sensor are connected to the digital pins 6 and 7, respectively, and the two IR sensors have data pins that are connected to the digital pins 8 and 9, with the remaining ground and VCC pins being connected to the Arduino Uno.

Arduino controls the rotation of the first stepper motor, which rotates 180 degrees in both clockwise and anticlockwise directions, by connecting the control signal line to digital pin 4. When intruder tracking begins, the Arduino uno uses an IR sensor to rotate the second stepper motor, which is fitted with a laser light. The weapon is automatically triggered, and the laser light is pointed at the intruder.

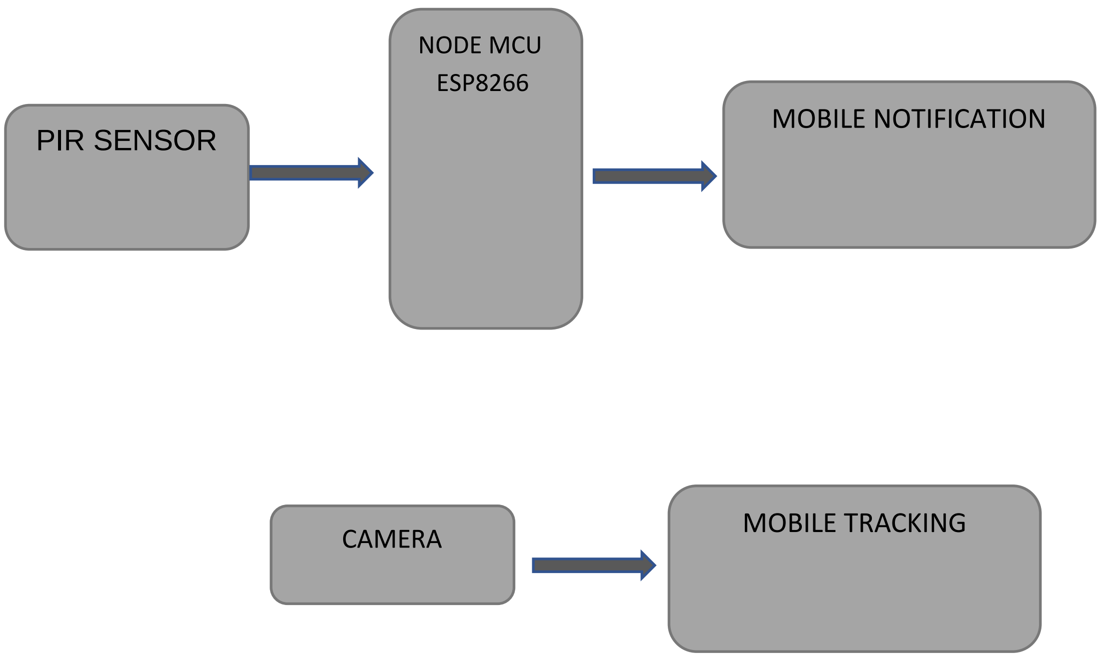


Figure 2.2.2: Block diagram of Alert Notifications

Every time the PIR sensor detects movement, we are using the Node MCU, which is connected to the sensor. Via IFTTT, the alert message is sent to the control room.

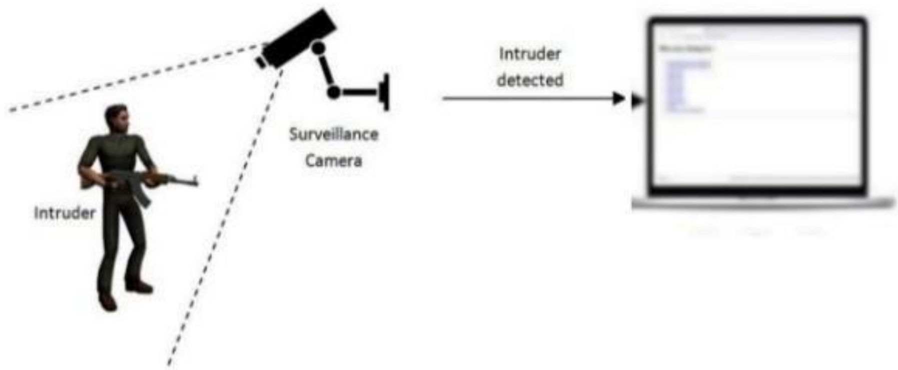


Figure 2.2.3: Video Surveillance

They can then turn on the camera, which will immediately start streaming live video to our phone.

CHAPTER-3
MICROPROCESSOR

3.1. Introduction:

The term "microprocessor" refers to a versatile central processing unit (CPU) used in digital computers that can perform a range of tasks. It is often referred to as a "computer on a chip" as it includes arithmetic and logic units, a program counter, a stack pointer, working registers, a clock timing circuit, and an interrupt circuit. However, to form a complete computer, additional components are needed such as memory (usually in the form of RAM and ROM), memory decoders, oscillators, and a variety of input/output (I/O) devices such as parallel and serial data ports, as well as specialized devices like interrupt handlers and counters. The microprocessor's main feature is its "general-purpose" architecture, which allows for a system of any size to be built around it due to its flexible hardware design. It reads data and processes it through complex calculations before storing the results onto a mass storage medium. The microprocessor's programs are stored on the mass storage device and are loaded into RAM when required by the user. Some programs are also stored in the ROM, but these are typically small, fixed programs used to operate peripherals and other fixed devices connected to the system.

3.2 Microcontroller:

A microcontroller is a chip that contains only a minimal number of embedded devices. Despite having a CPU, ALU, PC, SP, registers, RAM, ROM, I/O ports, and timers like a typical computer, it is much smaller and more portable. The microcontroller is specifically designed to perform a single function, regulating a single system, unlike a general-purpose computer.

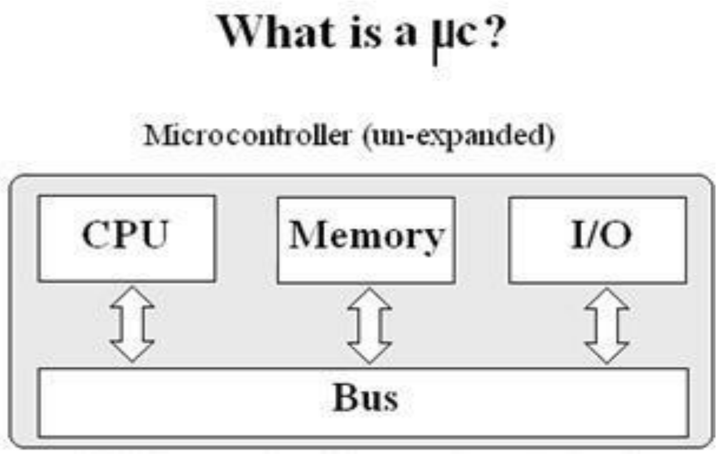


Figure3.1: Microcontroller Block Diagram

Most microcontrollers come with additional components, such as a timer module that enables the microcontroller to perform specific actions for a set duration of time. They also typically include a serial I/O port, which allows for connection to other microcontrollers, computers, and external devices. Additionally, microcontrollers often feature an ADC, which enables the processing of analog input data.

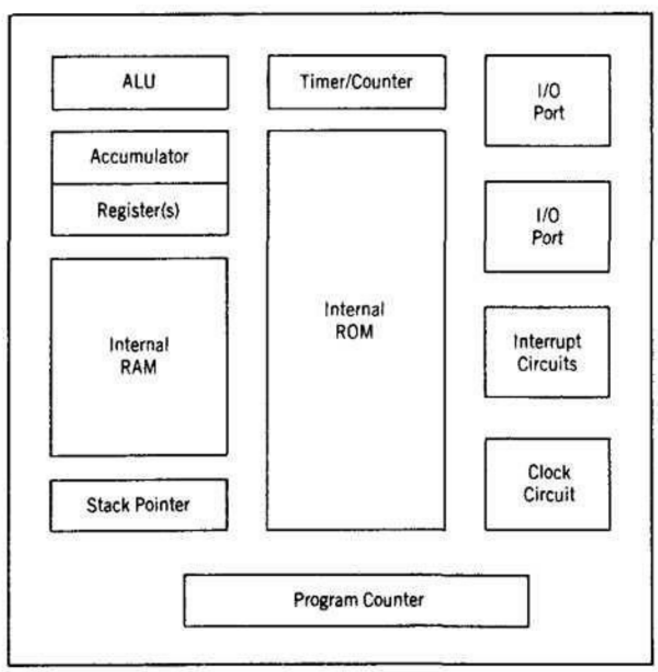


Figure3.2: Functional Block Diagram of Microcontroller

3.3 Comparision b/w Microcontroller & Microprocessor:

The past three decades have brought about technological advancements that have transformed how we perceive and interact with the world around us. Intel Corp.'s 4-bit 4004 chip, released in 1971, made the microprocessor commercially accessible and marked a significant milestone in computer design and integrated circuit manufacturing.

Today, it is widely recognized that programmable smart devices, such as VCRs, clock radios, washers, dryers, video games, cell phones, microwaves, TVs, vehicle toys, vending machines, copiers, and elevators, all rely on microchip technology. Companies understand that to remain competitive in the microchip era, their products or manufacturing equipment must be "smart." Microcontrollers, which are computer-on-a-chip devices, have all the functions of a microprocessor, as well as additional components, including ROM, RAM, parallel and serial I/O, counters, and a clock circuit, required to create a complete computer.

While microcontrollers are general-purpose devices like microprocessors, they have the added capability of reading data and performing simple calculations based on that data and their surroundings. They are often used in conjunction with fixed programs stored in ROM to control machine operations, with the program remaining constant throughout the system's use.

3.4 Criteria of Choosing Microcontroller:

When selecting a microcontroller for a project, the main priority is its ability to perform the necessary tasks quickly and cost-effectively. It is essential to determine whether an 8-bit, 16-bit, or 32-bit microcontroller is most suitable and can handle the computing requirements of the project. Several factors must be considered, such as the maximum operating speed of the microcontroller, its form factor (40-pin DIP, QPF, etc.), power consumption (especially important for battery-powered devices), RAM and ROM capacities, the number of timers and I/O pins available, ease of updating to a more efficient or power-saving version, and cost per unit. The microcontroller's utilization in the final product is dependent on its cost.

Differences between suppliers indicate that there may be a manufacturer involved in the production of the microcontroller, in addition to the designer.

The 89C51 microcontroller, initially created by Intel, is now manufactured by multiple companies such as Atmel, Dallas-Semiconductor, Siemens, and Philips. To ensure their products are reliable and easily accessible, companies like Motorola, Zilog, and Microchip Technologies have dedicated significant resources to make their microcontrollers single-sourced and mature. More recently, these companies have started providing the ASIC library cell for the microcontroller.

3.5 Advantages of Microcontroller:

With the help of switches, buttons, sensors, LCD displays, relays, and other I/O devices, their carefully selected and powerful electronics can autonomously manage a wide range of processes and machinery, including industrial automation, voltage, temperature, engines, and more.

3.6 ATMEL ATMEGA328P

3.6.1 Introduction:

The ATmega328P is a part of the Atmel 8-bit microcontroller series, with each member having unique specifications, including RAM, ROM, I/O ports, and more. They come in packages with a varying number of pins, ranging from as few as eight to more than one hundred. The ATmega328P was chosen for the EE459 course due to the availability of chips and software for development, and its dual-in-line packaging with 28 pins that can fit in existing IC sockets. It also has enough TTL compatible I/O pins (21) and FLASH memory for fast and easy reprogramming, making it suitable for most EE459L project tasks. The following brief instructions are intended to help set up the 328P, with additional resources available on the EE459 website for using C programming language systems. For more information, refer to the detailed Atmel datasheet or programming manual.

3.7 ARDUINO UNO:

The Arduino Uno is widely used and highly recommended for those just starting out with microcontrollers. It is a board that features the ATmega328P microcontroller and includes 14 digital input/output pins.

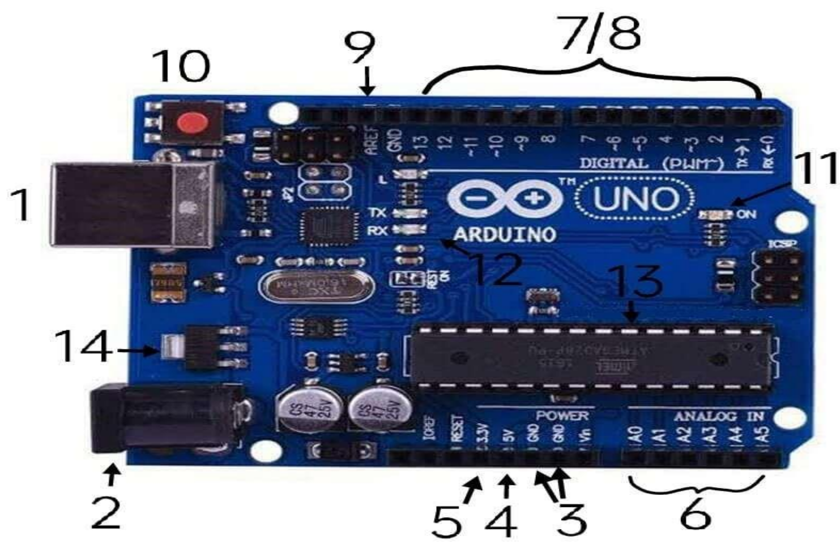


Figure3.7: Arduino

The Arduino Uno comes equipped with a variety of features, including a USB connector, power jack, ICSP header, six analogue inputs, six pins that can be utilized as PWM outputs, and a reset button. Additionally, it has a 16MHz quartz crystal and is fully supported with all the necessary components for operation.

3.7.1 Arduino Functional Blocks

- Microcontroller
- Input/Output (I/O) Pins
- Power Supply
- Clock
- Reset Button
- Crystal Oscillator
- Serial Communication Interface
- Sensors
- Actuators
- Memory

Main IC

The small black square with multiple metal legs is called an integrated circuit, or IC, which is considered the "brain" of the Arduino. The primary integrated circuit used in the Arduino may vary, depending on the type of board being used, but is often sourced from ATMEL's ATmega series of ICs. In the Arduino UNO, for example, the ATmega328 microcontroller is utilized, which boasts the following peripheral features:

- 8-bit RISC microcontroller architecture.
- Clock speed of up to 20 MHz.
- 32KB of flash memory for program storage.
- 1KB of EEPROM for non-volatile data storage.
- 2KB of SRAM for temporary data storage.
- 23 I/O pins for communication with external devices.
- Supports UART, SPI, and I2C communication interfaces.
- 10-bit ADC with up to 8 channels for analog signal conversion.
- Multiple timers/counters for timing and PWM generation.
- Power-saving modes for reduced power consumption.
- Wide operating voltage range (1.8V to 5.5V).
- Available in DIP and QFN package options.

3.7.2 POWER

The power pins on the Arduino Uno board are as follows:

- Vin for external power input.
- 5V for regulated 5V output.
- 3.3V for regulated 3.3V output.
- GND for common ground reference.
- Vcc for separate positive supply voltage (specific components/modules)

3.7.3 MEMORY

The Atmega328 has a RAM capacity of 32 KB, with a bootloader taking up about 0.5 KB of space. Additionally, it features 2 KB of SRAM and 1 KB of EEPROM, which can be utilized by the EEPROM library for reading and writing purposes.

3.7.4 PIN DESCRIPTION

The Arduino Uno board features 14 digital pins that can function as both inputs and outputs through the use of the pin Mode (), digital Write(), and digital Read() methods, and operate at 5 volts. Each pin is equipped with an internal pull-up resistor, ranging from 20-50k ohm, which is not connected by default. These pins can send or receive 20 mA of current under normal conditions, and it is essential to prevent any I/O pin from exceeding its maximum current draw to avoid long-term harm to the microcontroller. The ICSP header uses digital pins 11, 12, and 13 as MOSI, MISO, and SCK connections, respectively (Pins 17, 18, & 19 on an Atmega 168). It is necessary to use caution while connecting low-impedance loads to these pins.

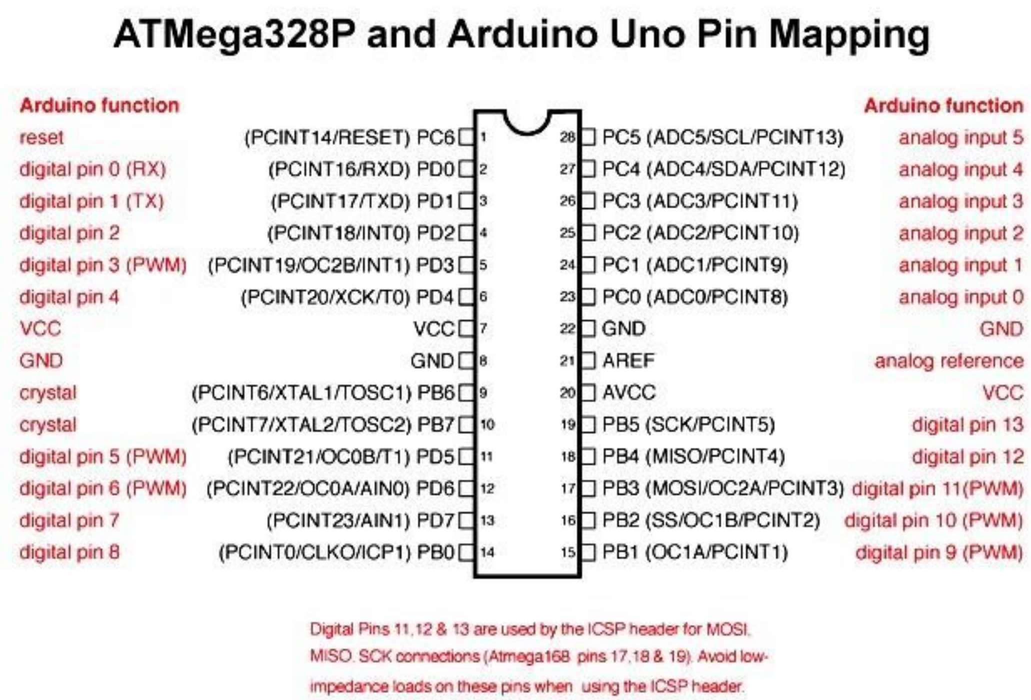


Figure 3.7.4: Atmega 328P Pin Diagram

- Some of the Uno's pins are dedicated to specific functions:
- Pin 0 (RX) reserved for receiving serial data through UART communication.
- Pin 1 (TX) dedicated for transmitting serial data through UART communication.
- Pins 2 and 3 designated as interrupt pins for external interrupts.
- Pins 4, 5, 6, 7, 8, 9, 10, and 11 exclusively used for PWM (Pulse Width Modulation) output.
- Pins A4 and A5 assigned for I2C communication (SDA and SCL respectively).
- Pins A6 and A7 solely for analog input without digital I/O capabilities.
- Pin 13 reserved for built-in LED, typically used for status indication.
- Pins A0 to A5 designated for analog input with 10-bit ADC functionality.

3.7.5 Communication:

- UART (Universal Asynchronous Receiver-Transmitter) for serial communication.
- I2C (Inter-Integrated Circuit) for two-wire serial communication.
- SPI (Serial Peripheral Interface) for synchronous serial communication.
- Digital I/O pins for custom digital communication interfaces.
- Wireless communication using modules or shields (e.g. Wi-Fi, Bluetooth, Zigbee, LoRa).
- Choosing the appropriate communication method based on project requirements.
- Using relevant software libraries and protocols for proper communication.

3.7.6 Reset Button

Similar to the original Nintendo, the Arduino also has a reset button (10) that can be pressed to restart any code currently running on the board. When pressed, the reset pin is briefly connected to the ground. This feature can be very useful if you need to test your code multiple times, especially if it doesn't have a loop. However, unlike the original Nintendo, simply blowing on the Arduino won't solve any issues.

3.7.7 Power Led Indicator

There is a small LED located on the Arduino circuit board near the word "ON" and situated beneath and to the right of the word "UNO" (11). This LED is designed to illuminate every time the Arduino is connected to a power source. If the LED fails to light up, it is likely that there is an issue with the circuit. It is advisable to conduct a circuit check immediately.

3.7.8 TX/RX LED

TX and RX are common abbreviations used in electronics to denote the pins used for serial communication, where TX represents transmit and RX represents receive. On the Arduino UNO board, these labels are present twice: once next to the digital pins 0 and 1 and again next to the indication LEDs for TX and RX. The LEDs next to these labels are helpful visual indicators that show when the board is receiving or transmitting data, such as when uploading new software onto the board.

3.7.9 Voltage Regulator

It is not recommended or possible to modify the voltage regulator on an Arduino board. However, understanding its function and presence can be beneficial. The voltage regulator serves to control the amount of voltage allowed to enter the Arduino, much like a gatekeeper that restricts excess electricity from entering the circuit. It's essential to note that the Arduino has its limitations, and it's not advisable to connect it to anything with a voltage higher than 20 volts.

3.8 Warnings

On the Uno, there is a polyfuse that protects the USB ports of your computer from shorts and overcurrent. Although most computers have built-in protections, the fuse provides an extra layer of safety. If the USB port is supplied with more than 500 mA, the fuse will automatically disconnect the connection until the overload or short circuit is resolved. It can be reset on the Uno.

3.9 Applications

The primary aim of Arduino is to simplify the process of incorporating electronics into cross-disciplinary projects, making it accessible to designers, artists, and individuals interested in building interactive environments or objects. By utilizing various sensors, Arduino can gather information about its surroundings and control motors, lights, and other actuators to influence them.

CHAPTER – 4
HARDWARE COMPONENTS

4.1 Arduino Uno:

The microcontroller that powers the Arduino Uno board is based on the ATmega328P. The gadget has six analogue inputs, a 16 MHz quartz crystal, 14 digital I/O pins, six of which can be used as Output pin, a USB port, a power jack, an ICSP header, and a reset button. There is also a USB port. All of the components needed for operating the micro - controller are part of the package; all you have to do is connect it to a machine using an USB cable or use an AC to DC adapter. The device does have a 32-bit flash drives to retain this same number of instructions, especially in comparison to a 2 KB Main memory and a 1 KB EEPROM. The operational voltage of the gadget is 5V, which implies that the devices for the board's microcontroller as well as any connected components function at this voltage. Nonetheless, the suggested input voltage ranges from 7V to 12V, whereas the input voltage ranges from 6V to 20V.

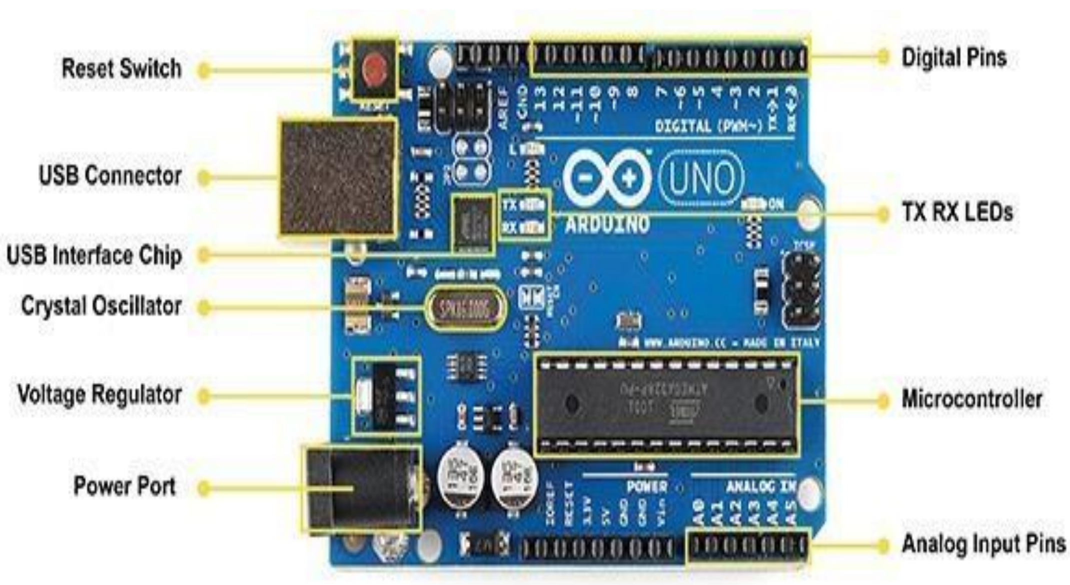


Fig 4.1: Arduino Uno Board

4.1.1 Pin Diagram:

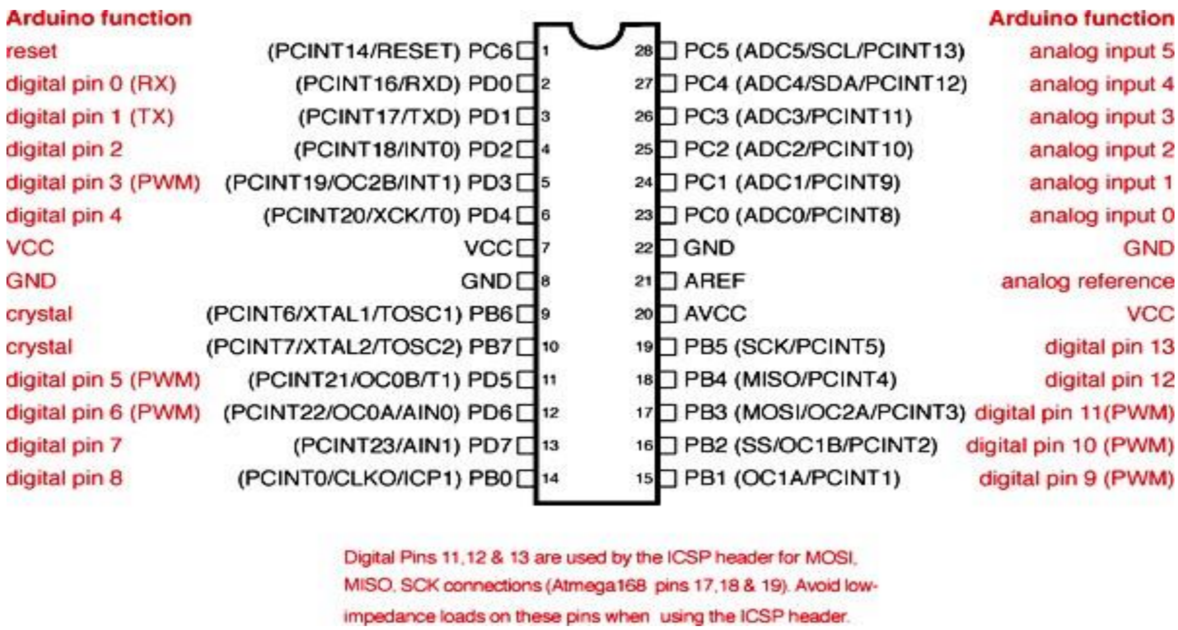


Fig 4.1.1: Arduino Pin diagram

Pin Category	**Pin Name**	**Details**
Power	Vin, 3.3V, 5V, GND	Vin: When using an external source, the Arduino's input voltage 5V: powered by a regulated power source is a microcontroller 3.3V: 3.3V supply generated by the onboard voltage reg. The maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.

Analo g Pins	A0 – A5	Used to provide analog input in the range of 0-5V

Table 4.1: Pin Configuration-1

Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.

Table 4.2: Pin Configuration-2

4.1.2 Description of Arduino Uno Board:

Arduino is a single-board computer designed to simplify the operation of interactive devices and their surroundings, and it is based on an 8-bit Atmel AVR microprocessor. The Arduino Uno board, powered by an ATmega328 microcontroller, is a popular version of Arduino. It features 14 digital input/output pins, a 16 MHz ceramic resonator, an ICSP header, a USB port, 6 analog inputs, a power jack, and a reset button, providing all the necessary microcontroller support for easy connectivity to a computer via USB. Unlike other boards, the Arduino Uno does not have the FTDI USB-to-serial driver chip; instead, it uses a Flash drive engine Atmega16U2 (up to R2 version) for USB communication. The Arduino Uno R3 version is commonly used in designs. Arduino systems can work independently or in conjunction with computer software, powered by a USB connection and operating at a 16 MHz ceramic resonant frequency. For larger projects, additional micro SD/SD card storage can be easily added.

4.1.3Applications:

- **a** a component and framework parametric modeling in circuit boards
- **b** Projects that necessitate multiple I/O interfaces and communications.

4.1.4Features of the Arduino UNO:

- Digital Input/Output (I/O) Pins: The Arduino Uno has 14 digital I/O pins, which can be configured as either input or output pins to connect and control external devices such as sensors, actuators, LEDs, and motors.
- Analog Inputs: The Arduino Uno has 6 analog input pins, labeled A0 to A5, which can read analog signals from sensors and other analog devices, allowing for precise analog measurements.
- 16 MHz Ceramic Resonator: The Arduino Uno operates at a resonant frequency of 16 MHz, which determines the clock speed of the microcontroller and affects the overall. Power Jack: The Arduino Uno can be powered using an external power supply connected to the power jack, providing an alternative power source to the USB connection.
- Reset Button: The Arduino Uno has a reset button that can be pressed to restart the microcontroller and reset the board, useful for restarting the program or troubleshooting.
- Micro SD/SD Card Storage: The Arduino Uno can be easily expanded with a micro SD/SD card module, providing additional storage space for data logging or other storage needs.

4.2 Node MCU (ESP8266):

The NodeMCU ESP8266 is a widely used development board that is based on the ESP8266 Wi-Fi module. It provides an affordable and convenient way to add Wi-Fi connectivity to various projects, making it popular for Internet of Things (IoT) applications. Notable features of the NodeMCU ESP8266 board include its microcontroller based on the ESP8266 chip, built-in Wi-Fi connectivity, GPIO pins for versatile interfacing, support for Lua programming language, USB interface for easy programming and communication, compatibility with popular IDEs, OTA firmware updates for remote updates, breadboard-friendly design, and low power consumption due to the ESP8266 chip's design. Overall, the NodeMCU ESP8266 board is a cost-effective and user-friendly solution for incorporating Wi-Fi connectivity into IoT projects, making it a popular choice among hobbyists, makers, and developers.

4.2.1 Node MCU Pinout and Functions Explained:

- D0 (GPIO16): Digital input/output pin. Typically used for GPIO operations, such as interfacing with digital sensors or actuators.
- D1 (GPIO5): Digital input/output pin. Typically used for GPIO operations, such as interfacing with digital sensors or actuators.
- D2 (GPIO4): Digital input/output pin. Typically used for GPIO operations, such as interfacing with digital sensors or actuators.
- D3 (GPIO0): Digital input/output pin. Additionally, this pin serves as a boot mode selection pin during power-up or reset. It should be pulled high (to 3.3V) for normal operation, and pulled low (to GND) to enter programming mode.
- D4 (GPIO2): Digital input/output pin. Additionally, this pin serves as a boot mode selection pin during power-up or reset. It should be pulled high (to 3.3V) for normal operation, and pulled low (to GND) to enter programming mode.
- D5 (GPIO14): Digital input/output pin. Typically used for GPIO operations, such as interfacing with digital sensors or actuators.
- D6 (GPIO12): Digital input/output pin. Typically used for GPIO operations, such as interfacing with digital sensors or actuators.
- D7 (GPIO13): Digital input/output pin. Typically used for GPIO operations, such as interfacing with digital sensors or actuators.
- D8 (GPIO15): Digital input/output pin. Additionally, this pin serves as a boot mode selection pin during power-up or reset. It should be pulled high (to 3.3V) for normal operation, and

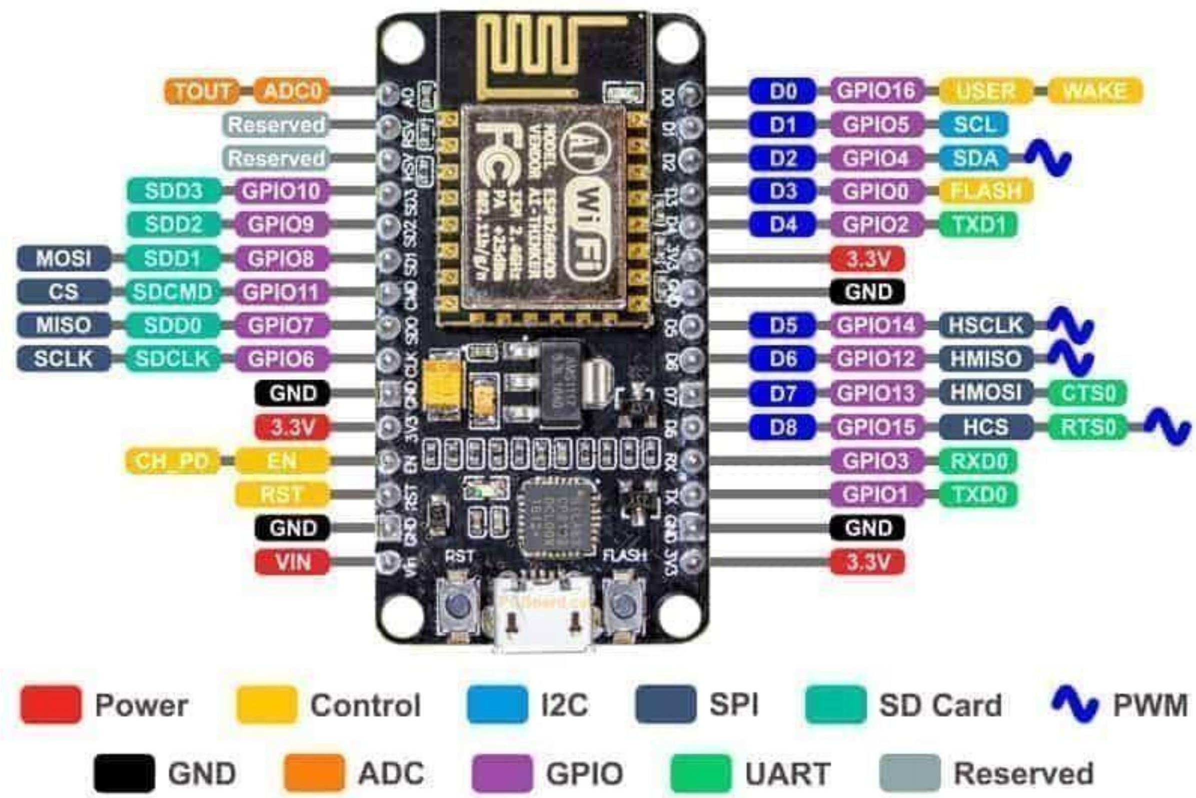


Fig 4.2.1: Node MCU pinout

4.2.2 Node MCU ESP8266 Specifications & Features:

- Microcontroller: ESP8266EX with 32-bit RISC CPU running at 80 MHz
- Operating Voltage: 3.3V
- Digital I/O Pins: 11
- Analog Input Pins: 1 (3.2V max input voltage)
- Flash Memory: 4MB (32Mbit) for program storage
- Wi-Fi Connectivity: 802.11 b/g/n (2.4 GHz)
- Wireless Protocols: TCP/IP, UDP, HTTP, MQTT, etc.
- Interfaces: UART, SPI, I2C, GPIO, ADC
- Onboard USB to Serial Converter: CH340G or CP2102 for programming and debugging
- Built-in LED: Pin D0 (GPIO16) for status indication
- Programming Environment: Arduino IDE, NodeMCU Lua, and other frameworks
- Power Input: Micro USB or external 3.3V DC power source

4.2.3 Applications

a) Prototyping of IoT devices
b) Low power battery operated applications
c) Network projects

d) Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

4.3. Servo Motor

1. A servo motor is an electric motor used for precise and controlled motion.
2. It can rotate to specific angles, making it suitable for tasks that require accurate positioning.
3. Servo motors are typically controlled using a pulse-width modulation (PWM) signal.
4. They have a rotation range of 0 to 180 degrees, although higher-end ones may have extended ranges.
5. Servo motors usually have built-in feedback mechanisms for position feedback.
6. They deliver high torque output for exerting force or torque on mechanical components.
7. Servo motors have specific voltage and current ratings.
8. They come in various sizes and form factors, such as micro, mini, and standard sizes.
9. Servo motors have standard mounting options for easy installation.
10. Some advanced servo motors may offer programmable features for adjustable speed, acceleration, and position limits.

Figure 4.3: Servo motor

4.3.1 Working Principle of Servo Motor :

The working principle of a servo motor involves the use of a feedback mechanism to achieve precise and controlled motion. It is typically controlled using a pulse-width modulation (PWM) signal, and has a built-in feedback mechanism, such as a potentiometer or an encoder, that provides position feedback to the motor controller. This feedback allows the controller to accurately monitor the actual position of the motor shaft and make adjustments to minimize any position error. The motor's internal components, such as the rotor and stator, work together to convert electrical energy into mechanical energy, driving the motor shaft to rotate to the desired position. This closed-loop control system allows servo motors to achieve high precision and control in applications that require accurate positioning

4.4 PIR Sensor

A Passive Infrared (PIR) sensor is a motion sensor that detects changes in infrared radiation emitted by objects within its field of view. It typically consists of a pyroelectric sensor that generates an electric charge when exposed to temperature changes. PIR sensors have a limited detection range and are designed to detect motion, such as a moving person or an animal. They can be used in various applications such as security systems, lighting control, and other devices. PIR sensors have adjustable sensitivity and delay settings and can be powered by batteries or low-voltage sources. However, they may produce false triggers due to temperature changes or insufficient heat emissions from smaller objects. Understanding the basics of PIR sensors can help in their effective use in motion detection applications.

Figure 4.4: PIR Sensor

4.4.1. Operation:

The operation of a Passive Infrared (PIR) sensor involves the detection of changes in infrared radiation emitted by objects within its detection range. When an object moves within the sensor's field of view, it emits heat in the form of infrared radiation, which is detected by the pyroelectric sensor in the PIR sensor. The detection circuit in the PIR sensor processes the signals from the pyroelectric sensor, analyzing the amplitude, duration, and frequency of the detected signal to determine if it indicates motion. If motion is detected, the PIR sensor generates an output signal, which can be used to trigger actions such as turning on lights, sounding an alarm, or activating other devices. PIR sensors often have adjustable sensitivity and delay settings, allowing for customization of the sensor's performance. Additionally, PIR sensors typically have a reset time during which they are inactive to avoid continuous triggering. Power requirements for PIR sensors are typically low, making them suitable for battery-powered devices. Understanding the operational principles of PIR sensors is essential in designing effective motion detection systems for various applications

4.4.2. Interfacing PIR With Node MCU:

Because the Node MCU uses PIR sensors as digital outputs, all you have to do is set the pin to high (detected) or low (detcted) (not detected). The photo shows how simple the links are. The pinout of each module may differ, so double-check! A typical DC input rated voltage is 3-5 volts.

The following are the circuit connections:

1. Connect the HC-Vcc SR501's pin to the +3v pin of a Node MCU.

2. Connect the HC-output SR501 pin to the Digital pin D7 of the Node MCU.

3. Connect the HC-GND SR501's pin to the Ground pin of the Node MCU (GND).

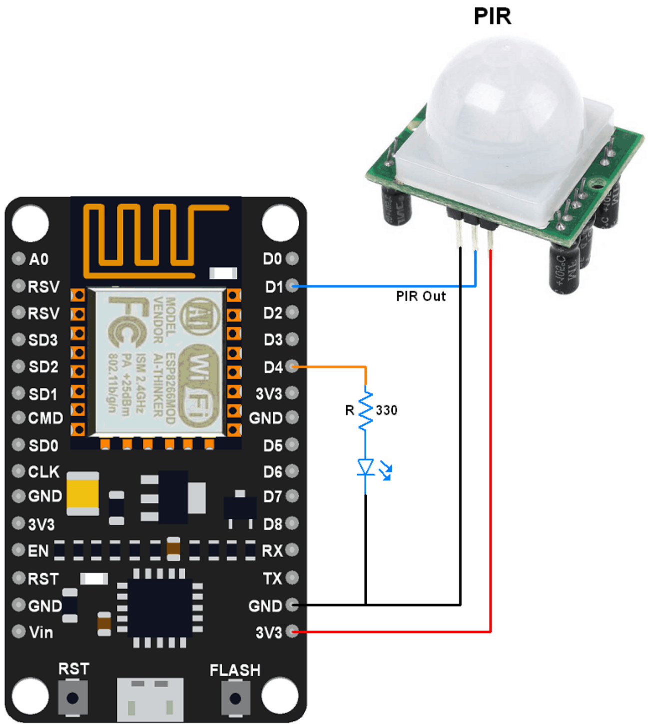


Figure 4.4.2 : Interfacing PIR with Node MCU

4.5 Ultrasonic Sensor

An ultrasonic sensor is a device that uses sound waves to detect objects, measure distances, or detect motion. It consists of a transmitter that emits high-frequency sound waves and a receiver that detects the echoes of these waves as they bounce off objects. By measuring the time taken for the sound waves to travel to the object and back, the sensor calculates the distance to the object. Ultrasonic sensors are widely used in various applications, such as distance measurement, obstacle detection, and motion detection, due to their accuracy, reliability, and versatility

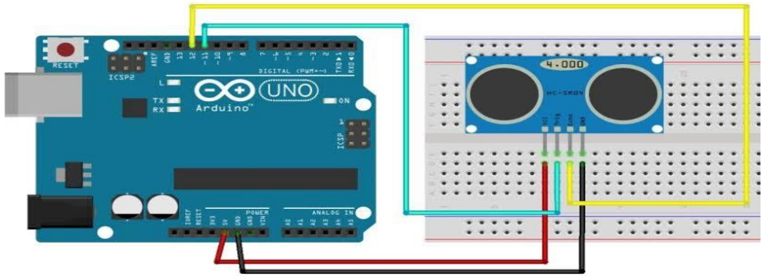


Figure 4.5: Ultrasonic Sensor

4.5.1 Working Principle

The operation of an ultrasonic sensor is based on the emission and detection of sound waves. It typically consists of a transmitter that emits high-frequency sound waves, and a receiver that detects the echoes of these waves as they bounce off objects. The emitted sound waves travel through the air and bounce back when they encounter an object in the sensor's field of view. The receiver then detects the echoes of these sound waves, and measures the time taken for the waves to travel to the object and back. Using the speed of sound in air as a reference, the sensor calculates the distance to the object based on the time taken for the sound waves to travel. This distance measurement allows the sensor to determine the presence, distance, or motion of objects in its surroundings.

4.5.2 Specifications

Understanding the specifications of an ultrasonic sensor aids in understanding the reliable approximations of distance measurements.

- Operating Frequency: 20 kHz to 200 kHz
- Detection Range: Few centimeters to several meters
- Detection Angle: 15 degrees to 180 degrees
- Accuracy: +/- 1 cm
- Output Signal: Analog or digital
- Input Voltage: 3V to 5V or higher
- Operating Temperature: -20°C to 60°C
- Housing Material: Plastic or metal
- Interface: UART, I2C, SPI, etc.
- Dimensions: Length, width, height

4.5.3. Interfacing Ultrasonic With Arduino

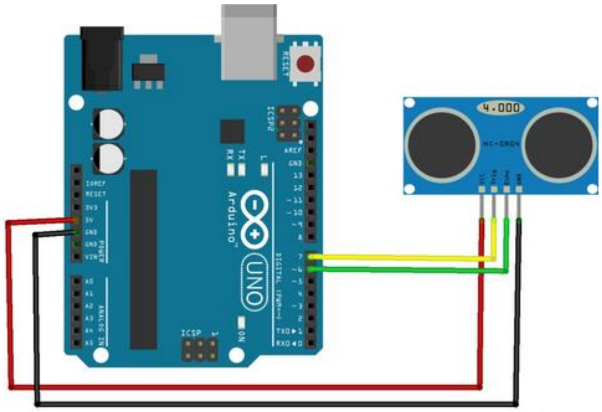


Figure 4.5.3:Interfacing Ultrasonic With Arduino

1. The ultrasonic sensor has four terminals connected as follows: +5V, Trigger, Echo, and GND.
2. Connect the +5V pin on your Arduino board to the +5V pin.
3. Connect Trigger to digital pin 7 on your Arduino board.
4. Connect Echo to digital pin 6 on your Arduino board.
5. Connect GND to GND on Arduino.

4.6 IR Sensor

An infrared proximity detector, also known as an IR Sensor, is an electronic device that emits infrared light to detect some aspect of its surroundings and can be used to detect the movement of an object. Because this detector is unreliable, it can only detect infrared radiation.

4.6.1 Working Principle:

An infrared (IR) sensor is an electronic device that detects and measures infrared radiation, which is a type of electromagnetic radiation with longer wavelengths than visible light. IR sensors work based on different principles depending on their type and application, such as detecting changes in emitted or reflected IR radiation, using a photodiode or phototransistor to detect incoming IR radiation, or detecting changes in temperature caused by objects. The specific working principle of an IR sensor depends on its type and design, but they all involve the detection and measurement of IR radiation using different mechanisms. IR sensors are commonly used in various applications such as motion detection, temperature measurement, proximity sensing, and remote control systems.

4.6.2 IR Transmitter:

An IR transmitter is a device that emits infrared radiation, typically using an LED or laser diode, to wirelessly communicate or control other devices. It works by converting an electrical signal into IR radiation, which is transmitted as pulses or continuous waves. IR transmitters are commonly used in remote control systems for consumer electronics, industrial automation, security systems, and communication devices. The emitted IR radiation can be modulated with different frequencies or codes for communication and control purposes. IR transmitters are affordable, efficient, and widely used for wireless communication and remote control applications

Figure 4.6.2 IR Transmitter

4.6.3 IR Receiver

An IR receiver is a device that detects infrared signals from an IR transmitter and converts them into electrical signals. It is commonly used in remote control systems, communication systems, and proximity sensors. Typical specifications include operating frequency, supply voltage, output voltage levels, output signal format, sensitivity, detection angle, package type, operating temperature, and package dimensions. Refer to the manufacturer's datasheet for accurate specifications.

Figure 4.6.3 IR Receiver

4.6.4 Applications:

1. Night vision devices
2. Radiation Thermometers

4.6.5 Interfacing IR With Arduino

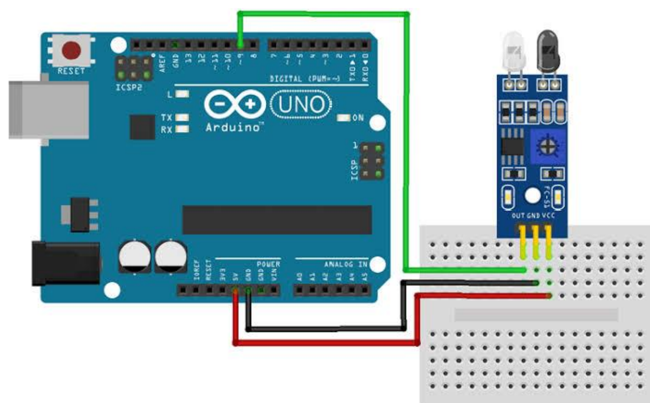


Figure 4.6.5. Interfacing IR with Arduino

It is very simple to connect the IR detector to any microcontroller. This detector, as we know, generates a digital signal that is extremely simple to process. There are two ways to do this: first, you can always check the harborage in a horizonless circle to see when it keeps changing from high to low, or you can do it with an intrude if your design is complicated. Connect floor to earth and energy the IR with 5V or 3.3V. Connect the affair to the D9 digital leg. To connect the IR detector module to the Arduino uno board, we simply used a manly to womanly Jumper line.

4.7. Camera

a) CP-E25A 2MP Wi-Fi PT Camera - 10 Meter.
b) 1/4.5" 2MP CMOS Image Sensor (0.5644 cm)
c) 3.6mm Lens
d) Wi-Fi Support
e) H.264 Video Compression
f) Support SD Card 128GB
g) IR Distance of 10 Mtr.
h) Works with Alexa
i) Works with Google Assistant

Figure 4.7 Camera

A Summary of the System The reasons for the preface of the EzyKam Series by CP PLUS are to increase your security and ensure your safety at all times. The EzyKam Series of seductive cameras are easy to install because they don't require any wiring and work as a Wi-Fi-connected draw and play surveillance bias with full HD view. In terms of CP PLUS CP PLUS, the brand on its way to leading the world in security and surveillance assiduity, continues to flood the market with the most dependable range of products that have been designed precisely and methodically to automate the entire surveillance and security process, no matter how large your demesne is. CP PLUS was conceived as a public assistance action.

4.7.1 Specifications

Table 4.7.1: Camera Specifications

Image Sensor	2MP CMOS Image Sensor(0.5644 cm)
Min. Illumination	0.01 Lux
Max Frame Rate	20fps
Screen Resolution	1920 x 1080
Image Compression	H.264
Lens	3.6 mm
Angle of View	85 degrees
Pan Range	Pan : 355 degree, Tilt: 75 degree
IR	IR LEDs upto 10 meters

CHAPTER – 5
SOFTWARES USED

5.1Arduino Programming

The Arduino board can be utilized to start or stop a motor, turn an LED on or off, establish a connection to the cloud, and perform various other tasks by reading analog or digital input signals from sensors. The control of the board's functionality is achieved through the uploading of software, where a set of instructions is sent to the microcontroller on the Arduino board using the Arduino Integrated Development Environment (IDE). Furthermore, the Arduino IDE employs a simplified version of C++ to facilitate learning for novice programmers. The standard form factor of Arduino boards provides a user-friendly container that encapsulates the microcontroller's functions for easier utilization.

5.1.1 Arduino Pin Diagram

The Arduino UNO Board, with the specification of pins, is shown below:

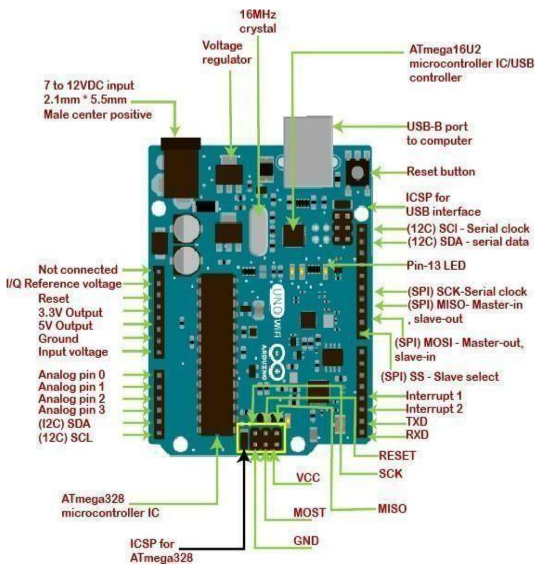


Figure 5.1.1 Arduino Pin Diagram

5.2 Programming

5.2.1 BASICS:

Coding Screen:

The group of statements in the setup and loop blocks are enclosed in curly braces. We can develop a number of statements, depending on the coding requirements for a certain project.

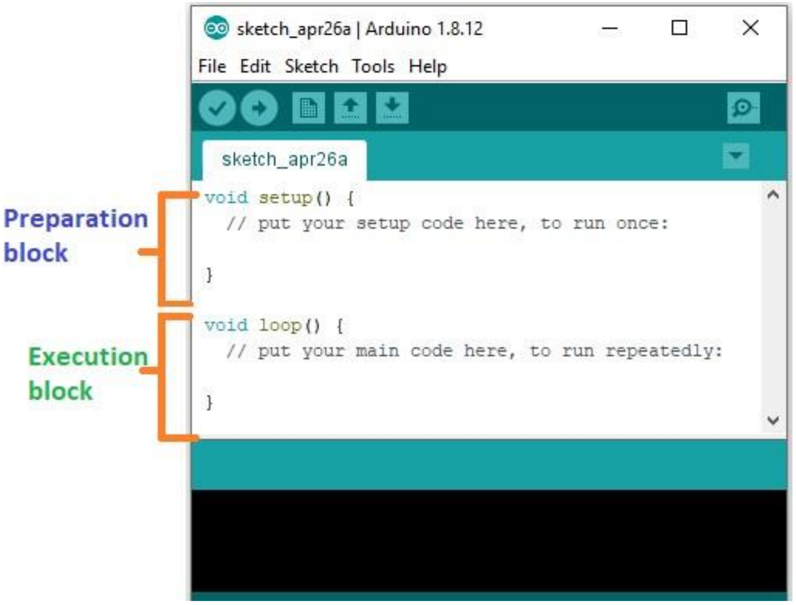


Figure 5.2 Programming

The "setup()" function is the initial code that is executed in an Arduino sketch. It is responsible for initializing pin modes, libraries, variables, and other settings during the setup phase, which occurs when the software is uploaded to the Arduino board or when the board is powered on or reset.

The "loop()" function is where statements are repeatedly executed in a loop. The code inside the curly brackets will be iterated based on the value of variables or conditions specified in the code.

The "pinMode()" function is used to configure a specific pin as an input or output. Its syntax is: pinMode(pin, mode)

where "pin" represents the pin number to be configured, and "mode" specifies whether the pin will be set as an input or output.

Including the "setup()" function at the beginning of an Arduino sketch is a standard practice to initialize the setup phase when the application starts. It is executed only once during the initial setup or when the board is powered on or reset.

Digital Write (): To set a pin's value to High or Low, use the digital Write () function. HIGH: It determines the voltage's value. It will set the value of 5V for the 5V board and the value of 3.3V for the 3.3V board. LOW: It makes the value equal to 0. (GND).

The LED may not light up brightly if the pinMode is not set to OUTPUT. Digital Write (pin, value HIGH/LOW) is the syntax.

The HIGH/LOW value of the digital pin is read from it using the digital Read () function, and set using the digital Write () function. pin: The defined variable or the PIN might be specified. Delay (): The blocking delay () function pauses a programme from doing a job for the number of milliseconds supplied.

Syntax:

The conventions that must be adhered to for the Arduino software to be successfully uploaded to the board are referred to as syntax. Arduino's syntax is comparable to English grammar. It indicates that in order for our code to successfully compile and run, the rules must be obeyed. Our computer software might build and execute, but it might do so with certain defects if we violate those criteria.

Functions: In Arduino, a function is a collection of several lines of code.

When a function has finished running, it often returns a value. But, because there is a void in this case, the method does not return anything. The void keywords are present in front of the names of the setup and loop functions. Curly brackets are used to separate the numerous lines of code that make up a function. Every closing curly bracket in the code needs to match the preceding one. We can write our own functions as well, which will be covered in more detail in the course.

White spaces and tabs preceding code statements are ignored by Arduino.

5.2.2 Syntax and Programme Flow

Syntax: Arduino syntax refers to the guidelines that must be followed for the Arduino software to be successfully uploaded to the board. Arduino's syntax is comparable to English grammar. It indicates that in order for our code to successfully compile and run, the rules must be obeyed. Our computer software might build and execute, but it might do so with certain defects if we violate those criteria.

White spaces and tabs preceding code statements are ignored by Arduino.

Moreover, spaces inside brackets, commas, blank lines, etc. are all ignored by the Arduino compiler.

Uses of Parenthesis ():

a) It designates a function by using notation like void setup () and void loop ( ).

b) The brackets denote the function's inputs for the parameter.

c) In mathematical operations, it is also used to alter the sequence of operations.

d) Semicolon; In both C and C++, this character serves as a statement terminator.

e) If a semicolon is missing from any of the statements, the compiler will flag the statement as incorrect. To make the code easier to read, it is advised to separate each statement with a semicolon and write it on a separate line

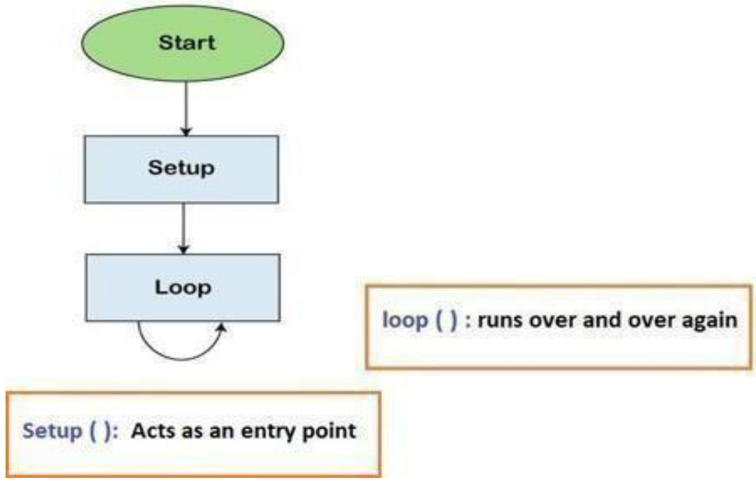


Figure 5.2.2 Program Flow chart

5.2.3 Serial Functions:

Serial.begin(): This function in Arduino sets the baud rate for serial data transfer, determining the data rate in bits per second (bps). The default baud rate is 9600 bps, with other options available. It configures data, stop, and parity bits for serial communication.

Serial.print(): The Arduino function used for transmitting data from Arduino to the serial port. The data is transmitted in ASCII format, which is human-readable. Each digit of the number is printed using ASCII characters. The printed information can be viewed on the serial monitor in the toolbar's right corner. The "Serial" object represents the serial port, and the "print()" function returns the number of written bytes. It can be used to print values of any data type.

analogRead(): This Arduino function reads the value from the analog to digital converter (ADC) of the Arduino board. The ADC translates the input voltage to a value between 0 and 1023, relative to the operating voltage (either 3.3V or 5V, depending on the board). It takes approximately 100 microseconds (0.0001 seconds) to read an analog input signal on boards such as UNO, Mega, Mini, and Nano, allowing for a maximum reading rate of around 10,000 times per second.

Different Arduino boards have varying operating voltages and resolutions. For instance, the Arduino UNO, Mini, Mega, Nano, Leonardo, and Micro operate at 5V with a 10-bit resolution. In contrast, the MKR family of boards, Arduino Due, and Zero operate at 3V with a 12-bit resolution.

5.2.4 Other Functions

1. Time Functions

millis(): This function returns the number of milliseconds that have elapsed since the Arduino board started running. It is a 32-bit unsigned long integer that can handle time durations of up to 49.7 days. It is often used for timing events or creating delays in code.

micros(): Similar to millis(), this function returns the number of microseconds that have elapsed since the Arduino board started running. It is also a 32-bit unsigned long integer, but has a shorter overflow time of approximately 71.6 minutes.
delay(ms): This function suspends the execution of the Arduino code for a specified number of milliseconds. It is a simple way to create delays in code, but it blocks the entire program during the delay, so it should be used with caution in time-sensitive applications.

5.2.5 Libraries

Standard Libraries: Arduino comes with a set of standard libraries that are included by default in the Arduino IDE (Integrated Development Environment). These libraries provide basic functionalities such as controlling digital and analog pins, reading sensors, working with timers, etc. Some examples of standard libraries are Wire for I2C communication, Serial for serial communication, EEPROM for reading and writing to EEPROM, etc.
Third-party Libraries: There are thousands of third-party libraries available for Arduino, which are created by the Arduino community and other developers. These libraries provide additional functionalities that are not available in the standard libraries. Third-party libraries can be easily downloaded and installed in the Arduino IDE using the Library Manager. Examples of third-party libraries include Adafruit NeoPixel for controlling RGB LEDs, DHT for reading DHT temperature and humidity sensors, Servo for controlling servo motors, etc.

5.3: IFTTT App

An automation tool called IFTTT was introduced in 2010. You can use it to develop applets, which are simple scripts or scenarios that can be used with a wide range of services and goods. Applets can be started manually or automatically in accordance with predefined criteria and conditions.
If This Then That, or IFTTT, is a platform that connects numerous systems and solutions, allowing them to interact without the need for specialised knowledge. When constructing conditional workflows, IFTTT is a viable option because it integrates with the majority of the web services you already use, including Gmail and Twitter in addition to smart home gadgets.
IFTTT applets are based on circumstances, or triggers, so when an event occurs, the script of your choice is automatically launched. IFTTT, for example, may automatically tweet your Instagram photographs, saving you from having to do it many times. Similar to how you can switch on your sprinklers when it's not raining or automatically turn off your Philips Hue lights when you leave the house. It's the same when songs or files are automatically added to your Spotify playlist or Dropbox.
IFTTT applets are based on circumstances, or triggers, so when an event occurs, the script of your choice is automatically launched. IFTTT, for example, may automatically tweet your Instagram photographs, saving you from having to do it many times. Similar to how you can switch on your sprinklers when it's not raining or automatically turn off your Philips Hue lights

when you leave the house. It's the same when songs or files are automatically added to your Spotify playlist or Dropbox.

If you want to create your own applet and customize it a bit further, that's possible and only takes a few clicks:

a) Access IFTTT.com.

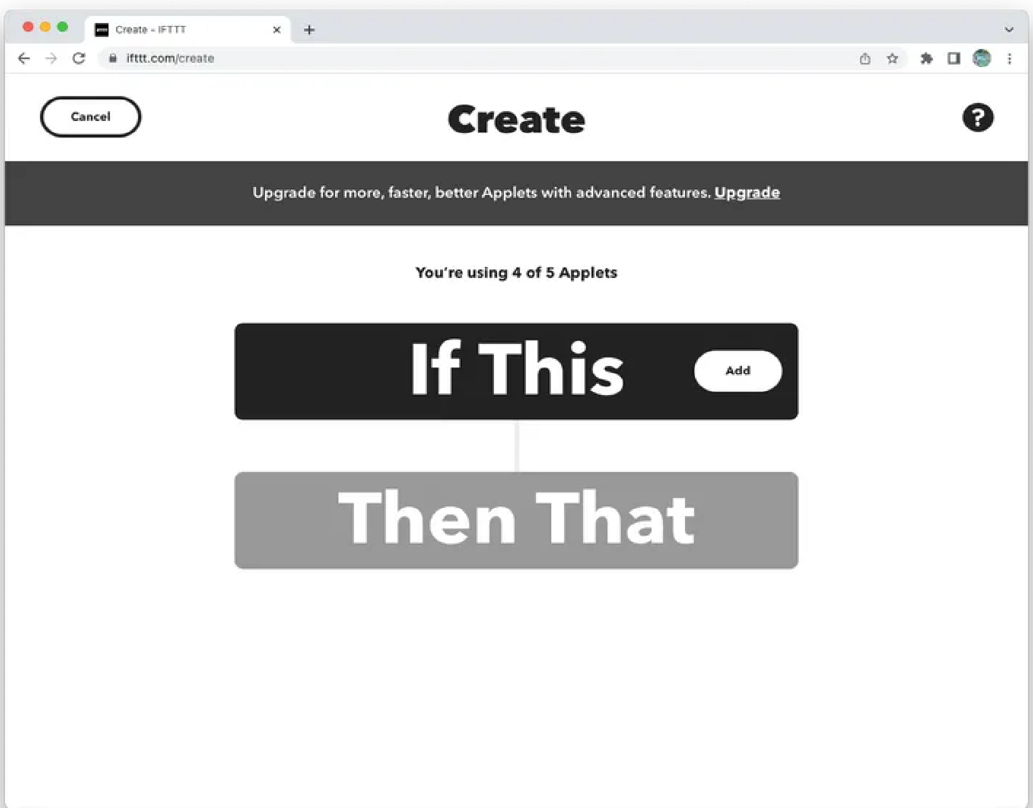
b) In the top-right corner, select Create.

c) If This is where you want to define your trigger, click Add. Then, configure what happens next by doing the same with Then That.

d) To specify a certain wait period before the action is executed, click the Plus symbol located between If and Then. If you want to arm your security system automatically with a longer delay, this can be helpful.

e) The scope of the free membership ends here. You can add more actions to the "Then" section and execute multiple actions with a single trigger if you have a Pro membership.

f) Click Continue when you're



finished.

Figure 5.3 IFTTT

g) Next, configure what happens by doing the same with Then That.

h)Adding a specific wait period before starting the action can be done by selecting the Plus icon next to If and Then. If you want to arm your security system automatically with a longer delay, this can be helpful.

i) The free membership is limited until this point. You can add more actions to the "Then" section and execute multiple actions with a single trigger if you have a Pro membership.

f) Click Continue once you're done.

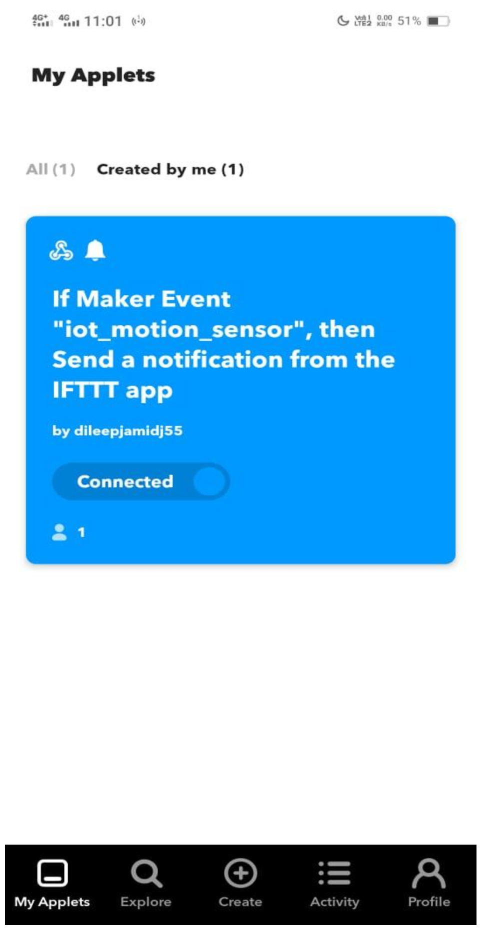


Figure 5.3.1 Created Applet

In this way we created our own applet for sending notification ,whether there is any obstacle detected

5.4: Processing 4 software

Since its inception in 2001, Processing has been a valuable resource for students, artists, designers, researchers, and hobbyists, promoting software literacy in the visual arts and technology. It has gained popularity as a cost-effective alternative to proprietary software tools due to its open-source nature, making it accessible to educational institutions and individual learners. One of the unique features of Processing is its active community participation, where contributors exchange programs, contribute code, and develop libraries, tools, and modes to enhance its capabilities. As a result, over a hundred libraries have been created, spanning diverse domains such as networking, computer vision, data visualization, and music creation, among others. This collaborative approach has fueled the growth of Processing and made it a widely used software for learning, prototyping, and creative exploration

5.4.1 Overview:

Processing is a popular open-source programming language and environment that enables artists, designers, and creative coders to create interactive graphics and multimedia applications. The latest version, Processing 4, introduces new features and enhancements, including a transition from Java to a JavaScript-based language called p5.js for improved performance and cross-platform compatibility. The p5.js language retains the simplicity and ease of use of the original Processing language, while offering more flexibility and integration with web technologies. Additionally, Processing 4 includes updated libraries, tools, and documentation for creating graphics, animations, and interactive applications, making it a powerful and accessible platform for creative coding and visual arts projects.

5.4.2 Processing Development Environment (PDE)

The Processing Development Environment (PDE) simplifies the process of writing Processing software. Programs are created using the Text Editor and can be executed by clicking the Run button. In the context of Processing, computer applications are referred to as sketches, and they are stored in the Sketchbook folder on the user's PC.

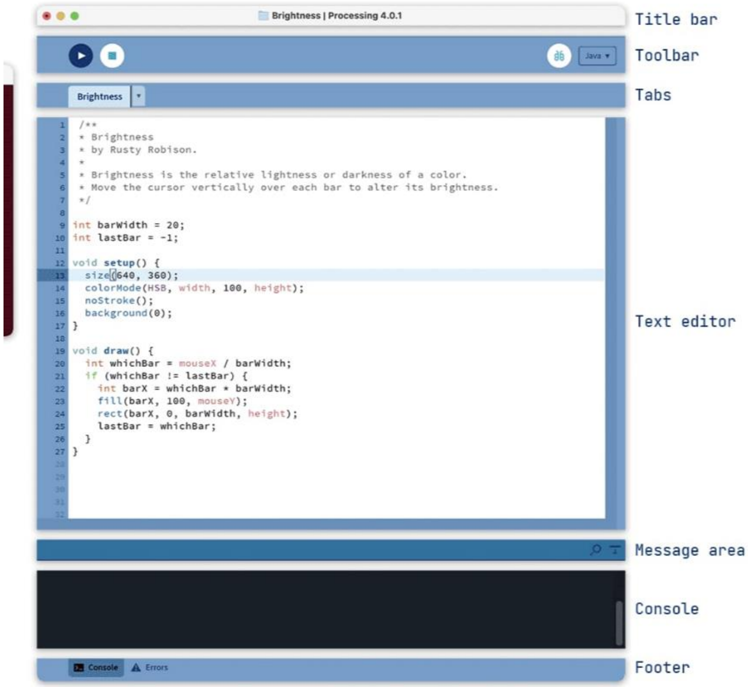


Figure 5.4.2 PDE

The Processing Development Environment (PDE) comprises multiple components, including a text editor for coding, a message area, a text console, file tabs for file management, a toolbar with buttons for common actions, and menus with varying options depending on the mode being used, such as the default Java mode.

In the context of Processing, software projects are referred to as sketches, which are created using the text editor. The text editor offers functionalities like text search and replacement. The message area provides feedback and reports issues during saving and exporting. The console displays output from Processing drawings, including error warnings and output from prints() and println() statements. It's important to note that the console is not optimized for high-speed, real-time output and is better suited for slower output.

The toolbar in PDE includes buttons for starting and stopping sketches, allowing users to run or pause their programs as needed. When a sketch is run, the code is compiled, and a new display window is opened in Java mode.

CHAPTER- 6
RESULTS &CONCLUSION

6.1 Results:

The sensors are interfaced with the Arduino uno board and also with the esp8266 board

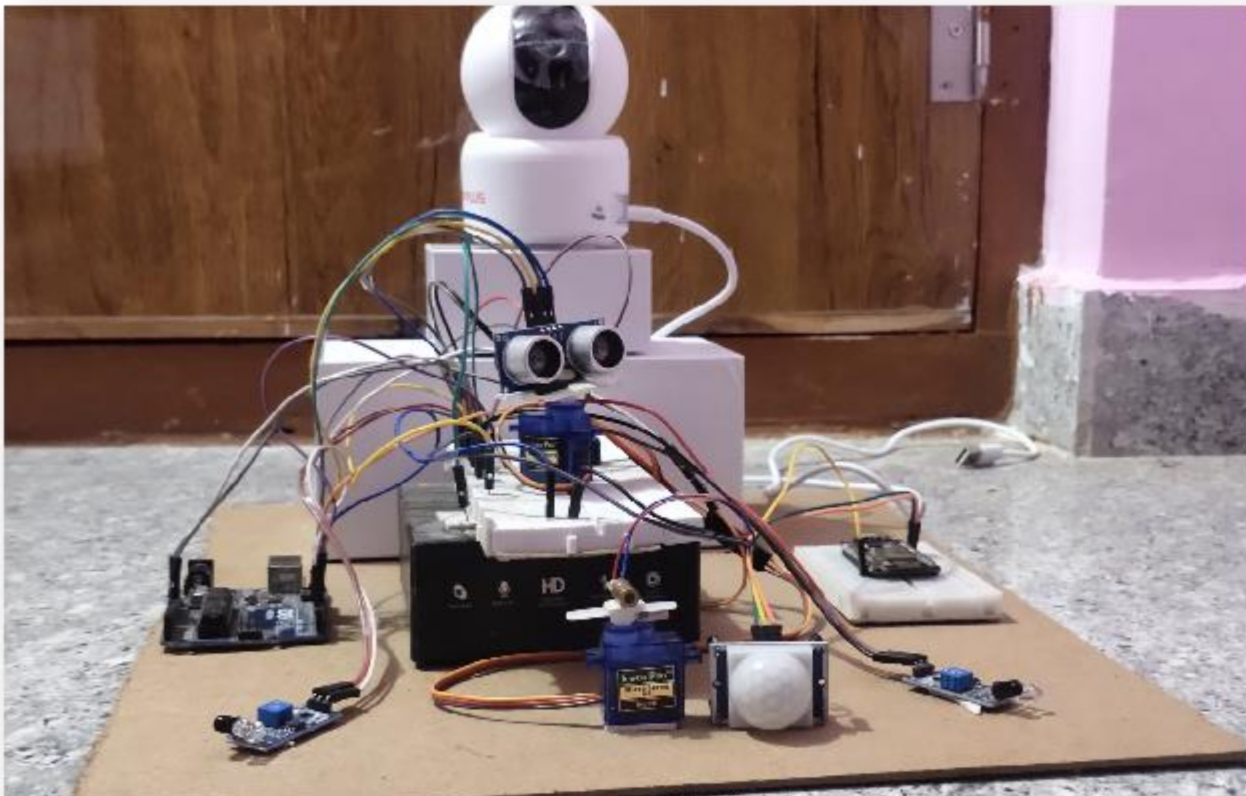


Fig 6.1.1 Front view of the project

The output from ultrasonic sensor is displayed as radar vision in the monitor , the data is serially transferred to the processing software to plot the radar vision whether the objected in the range or not.

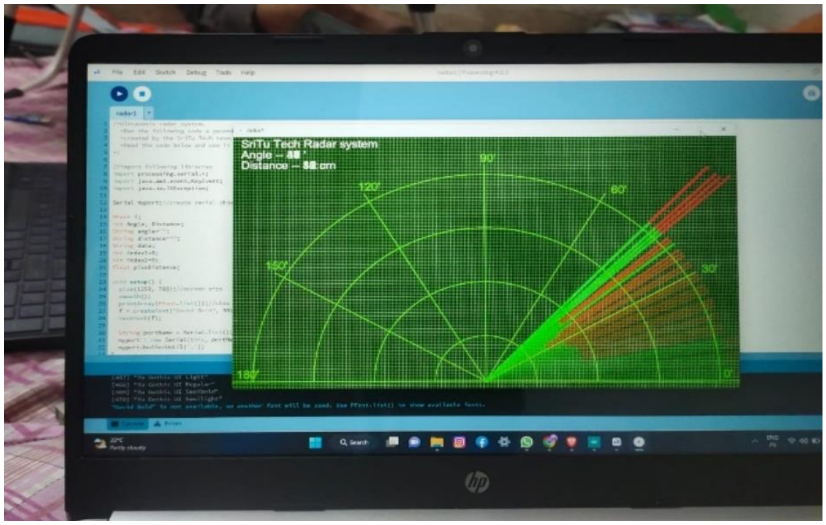


Fig 6.1.2 Radar vision when intrusion is happened

The IFTTT notifications are sent to the user mobile phone . and the live video streaming observed on the mobile phone.

Intruder tracking is done by the IR sensors and laser(weapon) will be activated.

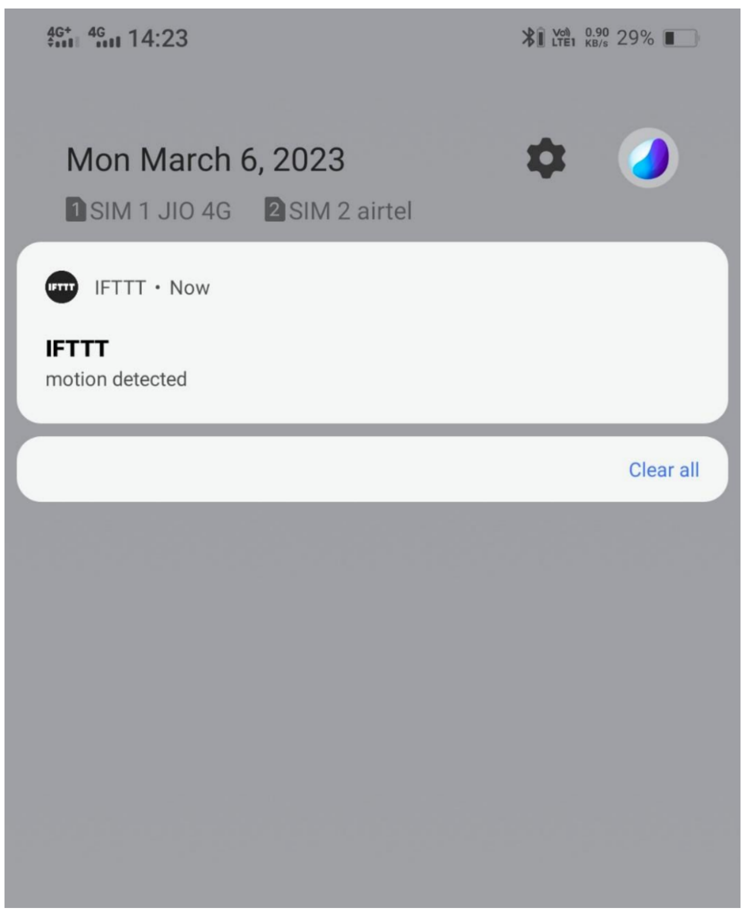


Fig 6.1.3 Alert notification sent to the mobile

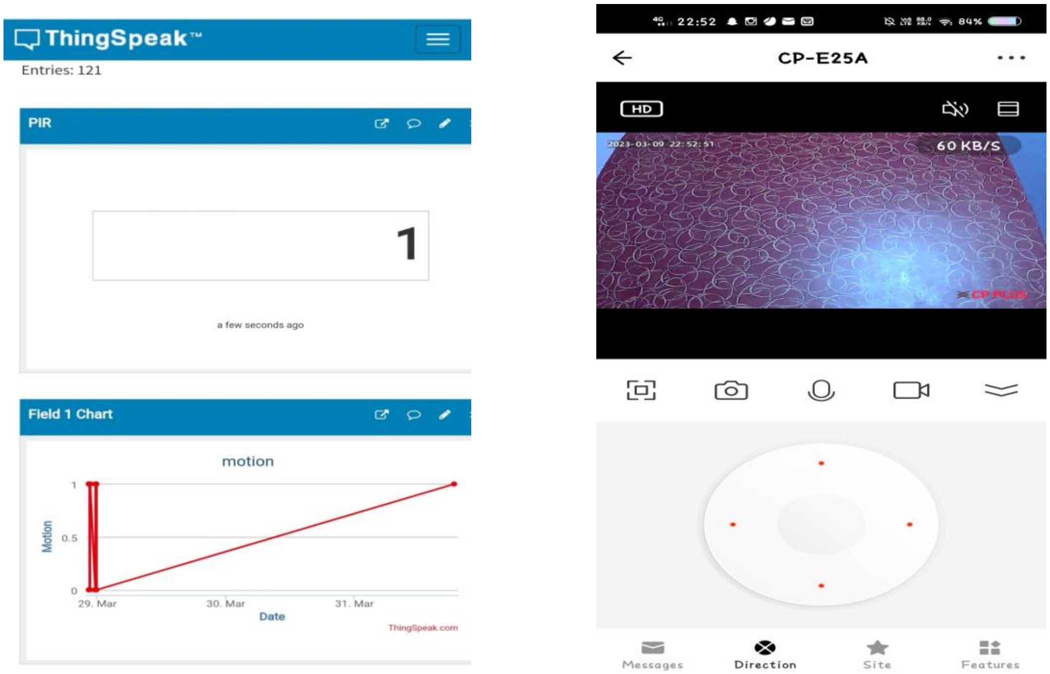


Fig 6.1.4 Triggering in Thingspeak & camera output

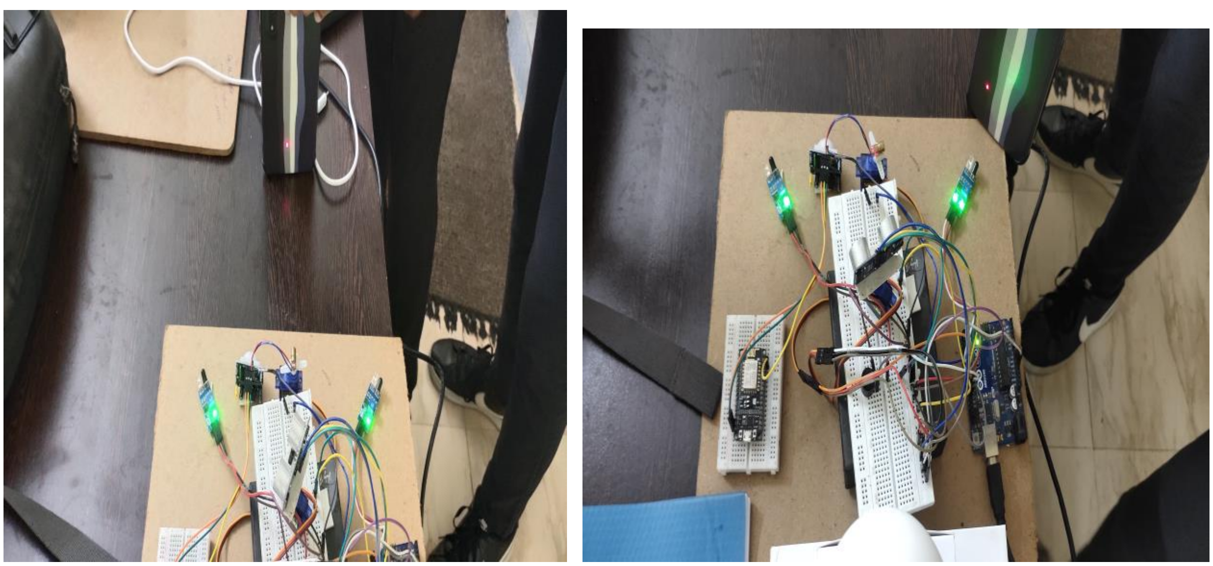


Fig 6.1.5 Weapon Activation

6.2 Conclusions

The soldier's monitoring attempts are stopped. Since this project automatically detects and senses intrusions. The real heroes of any nation are its soldiers. The project also aims to maintain peace at the borders and reduce tensions between the two countries. The suggested system prevents trespassers or antisocial individuals from attempting to enter the nation without the military's prior consent or with malice in their hearts. Now there is a functional connection between the sensors and microcontrollers. The project's viability as a viable solution to the issues with border monitoring by sensors and cameras using the internet of military things by 24 hours a day is supported by all experimental and observational data.

This project will reduce the manpower in the border areas and also reduces the risk of military people. Finally the implementation of such a system in real world can helps in enhancing the security in the boarders with less man power in the hostile regions.

REFERENCES :

[1] The Border S system employing Arduino, ultrasonic sensors, and IOT in 2020 was developed by Mr. Kanoujiya Akhilesh, Mr. Jha Ratnakar, Mr. Rayal Sandesh, and Mr. Khan Adnaan.

[2] Singh, Dushyant, and Kushwaha, Dharmendar. "Automatic Intruder Combat System: A Way to Smart Border Surveillance", Defence Science Journal, 2016.

[3] "Integrated human target detection, identification and tracking for surveillance applications," 6th IEEE International Conference Intelligent Systems, 2012, by Harish Bhaskar.

[4] "Border surveillance and intruder detection using wireless sensor networks: A brief survey," International Conference on Communication and Signal Processing, Chennai, 2019. Arjun, D., Indukala, P. K., and Menon, K. A. U.

[5] "Automatic Intruder Combat System: A Way to Smart Border Surveillance", Defence Science Journal, 67(1), pp. 50, 2016.

[6] Implementation of Spy Robot for A Surveillance System Using Internet Protocol of Raspberry Pi by Ghanem Osman Elhaj Abdalla and T. Veeramanikandasamy was presented at Recent Trends In Electronics Information & Communication Technology on May 19–20, 2017, in India.

[7] Frank Reinert and Jennifer Sander presented their system, "A Computer-Aided Assistance System for Resource-Optimal Sensor Scheduling in Intelligence, Surveillance, and Reconnaissance on Multi sensor Fusion and Integration for Intelligent Systems (MFI)" at the Kongresshaus Baden-Baden, Germany, from September 19–21, 2016.

[8] Frank Reinert, Jennifer Sander, Supporting Sensor Scheduling in Intelligence on ICICCS 2017.

[9] Syed Ali Imran Quadri and P. Sathish presented a paper titled "IoT Based Home Automation and Surveillance System" at the 2017 ICICCS conference.

[10] A Survey on IoT based Road Traffic Surveillance and Accident Detection System (A Smart Way to Handle Traffic and Concerned Problems) on International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017] by Rickin Patel, Vipul K. Dabhi, and Harshadkumar B. Prajapati

[11] MilNova: An Approach to the IoT Solution based on Model-Driven Engineering for the Military Health Monitoring in CCNC, by Rolando P. Reyes Ch, Manolo Paredes Calder'on, and Luis Montoya, published in 2015

[12] Mr. Chaitanya Vijaykumar Mahamuni, "A Military Surveillance System based on Wireless Sensor Networks with Extended Coverage Life" presented at the 2016 International Conference on Global Trends in Signal Processing, Information Computing, and Communication.

[13]2016 IEEE Prognostics and System Health Management Conference (PHMChengdu), Liang Cao, Guo Zheng, and Yong Shen, Research on Design of Military Ammunition Container Monitoring System Based on IoT. ISBN: 978-1-5090-2778-1**.**/16/$31.00 ©2016 IEEE.