A

MAJOR PROJECT REPORT ON

MULTI THREAT SURVEILLANCE VEHICLE

Submitted in partial fulfilment of the requirement for the award of a degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

GOUNI PAVAN KALYAN	**218R1A04F4**
GUDIBOINA SUBHASH	**218R1A04F5**
JONAGANI SAI KIRAN	**218R1A04F6**
KALPAGURI AAKASH	**218R1A04F7**

Under the Esteemed Guidance of

Mr. R. SRIDHAR
Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)
Kandlakoya(V), Medchal(M), Telangana – 501401

(2024-2025)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA& NAAC)

Kandlakoya (V),Medchal (M),Telangana –501401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Major Project work entitled **"MULTI THREAT SURVEILLANCE VEHILCE"** is being submitted by **G.PAVAN KALYAN** bearing Roll No: **218R1A04F4, G.SUBHASH** bearing Roll No**: 218R1A04F5, J.SAI KIRAN** bearing Roll No: **218R1A04F6, K.AAKASH** bearing Roll No: **218R1A04F7** in B.Tech IV-II semester, Electronics and Communication Engineering is a record bonafide work carried out during the academic year 2024-25.

INTERNAL GUIDE:	**HEAD OF THE DEPARTMENT:**
Mr. R. SRIDHAR	Dr. SUMAN MISHRA
Assistant Professor	Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

We sincerely thank the management of our college **CMR Engineering College** for providing required facilities during our project work. We derive great pleasure in expressing our sincere gratitude to our Principal **Dr. A. S. Reddy** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA,** Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our major project coordinator **Dr. T. SATYANARAYANA**, Associate Professor, Department of ECE for the ideas that led to complete the project work and we also thank him for his continuous guidance, support and unfailing patience, throughout the course of this work. We sincerely thank our project internal guide **Mr. R. SRIDHAR,** Assistant Professor of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We here by declare that the project work entitled **"MUTLI THREAT SURVEILLANCE VEHICLE "** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2024-2025 and is submitted as Major project in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING FROM JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**

G. PAVANKALYAN	**218R1A04F4**
G. SUBHASH	**218R1A04F5**
J. SAIKIRAN	**218R1A04F6**
K. AAKASH	**218R1A04F7**

ABSTRACT

The Multi-Threat Surveillance Vehicle is an advanced autonomous and remotely controlled robotic platform engineered for real-time security monitoring and multi-hazard threat detection. At its core, the vehicle is powered by an ESP32 microcontroller that orchestrates a robust four-wheel drive system, granting the unit exceptional mobility and adaptability across varied and challenging terrains. This mobility is critical in applications where rapid deployment and versatile navigation are required.The vehicle is outfitted with an array of sensors designed to detect a wide spectrum of environmental hazards. Gas sensors from the MQ series provide continuous monitoring for potentially explosive or toxic gases, while PIR motion detectors serve as an early warning system against unauthorized intrusions. Complementing these are flame sensors, which are integral in early fire detection scenarios, and metal detectors that assist in identifying the presence of concealed metallic objects. This sensor suite is strategically integrated to create a layered security approach that enhances situational awareness and minimizes the risk of false positives.

A significant innovation of the system is its incorporation of the ESP32-CAM module, which enables real-time video streaming. This capability allows remote operators to conduct live surveillance, facilitating immediate decision-making based on visual data. Additionally, the vehicle employs a GPS module that continuously updates its position on a web-based interface, ensuring precise geolocation tracking. This dual approach of visual and locational monitoring makes the vehicle highly effective for dynamic security operations such as border patrol, military reconnaissance, and monitoring of hazardous environments. Powering the system is a 12V 2A rechargeable battery, selected for its ability to deliver sustained energy over extended operational periods. This power solution is critical in scenarios where the vehicle is expected to function autonomously for long durations, such as during extended reconnaissance missions or continuous security sweeps in remote areas.

The design and implementation of the Multi-Threat Surveillance Vehicle reflect a convergence of IoT and AI technologies. The use of smart sensor networks and data analytics not only allows for real-time threat detection but also facilitates predictive maintenance and operational optimizations. This integration enhances the overall reliability and effectiveness of the system, providing a robust solution capable of adapting to the evolving challenges in modern security operations.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page No

CHAPTER 1
INTRODUCTION
1.1 OVERVIEW OF THE PROJECT

This project proposes the development of an advanced Multi-Threat Surveillance Vehicle designed to monitor and respond to a variety of threats in real-time. The vehicle will utilize multiple threat-detection sensors such as gas, smoke, and metal detectors, integrated with a system that activates buzzer alerts and web alerts to notify operators about potential threats. The vehicle will also feature GPS location updates transmitted in real-time via a web platform, allowing for seamless monitoring and tracking of its movement and status.The surveillance vehicle is equipped with an ESP32 camera module that streams video in real-time, offering visual feedback to the operator. This feature enhances the surveillance capabilities, enabling operators to remotely monitor and assess the surrounding area. The vehicle's control system is powered by an Arduino Nano microcontroller, which manages sensor data, alerts, and communication with other systems.

The Arduino Nano ensures real-time decision-making and control of vehicle functions. The vehicle is mounted on a four-wheel robot platform, designed for high mobility across various terrains. The wheels are driven by motors controlled via the Arduino system, ensuring smooth movement and the ability to navigate rugged environments.12V 2A Battery: The surveillance vehicle operates on a 12V 2A battery, providing adequate power to run all systems, including the sensors, cameras, and motors, ensuring extended operational duration in the field. The proposed multi-threat surveillance vehicle integrates cutting-edge technologies for enhanced situational awareness, security, and real-time threat detection. This system provides a mobile, versatile solution for applications in law enforcement, military operations, or disaster response, offering both automated threat alerts and human-controlled monitoring. This vehicle is not only a powerful tool for home automation and industrial safety, but also has the potential for future enhancements like AI-driven navigation, IoT integration, and smart city applications. the system efficiently processes sensor data and controls the four-wheeled robotic platform, ensuring smooth mobility across diverse terrains. The proposed multi-threat surveillance vehicle integrates cutting-edge technologies for enhanced situational awareness, security, and real-time threat detection. With a 12V 2A battery supplying reliable power, the vehicle is capable of extended operation, making it ideal for applications in law enforcement, industrial safety, and disaster response.

1.2OBJECTIVE OF THE PROJECT

The primary objective of this project is to design and deploy an advanced Multi-Threat Surveillance Vehicle capable of autonomously detecting, monitoring, and responding to a wide range of environmental and security threats in real time. In today's world, where safety and security challenges are increasingly complex ranging from industrial accidents to terrorist threats there is a growing need for intelligent, mobile surveillance systems that can provide early warnings and rapid response capabilities. This project aims to address these challenges by integrating cutting-edge sensor technology, real-time data transmission, and automated alert systems into a single, robust platform. The surveillance vehicle will be equipped with multiple threat-detection sensors, including gas detectors (for toxic and combustible gases), smoke detectors (for fire hazards), and metal detectors (for identifying weapons or explosives). These sensors will continuously scan the environment, ensuring comprehensive threat coverage. Upon detecting a potential hazard, the system will instantly trigger on-site alarms (such as buzzers or flashing lights) to alert nearby personnel while simultaneously sending web-based notifications to a centralized monitoring station. This dual-alert mechanism ensures that both on-ground responders and remote operators are immediately informed, enabling swift action to mitigate risks.

Another critical feature of this project is the integration of real-time GPS tracking, which will allow security teams to monitor the vehicle's exact location at all times. The GPS data will be transmitted to a web-based dashboard, providing operators with a live feed of the vehicle's movement, sensor readings, and threat alerts. This feature is particularly valuable in large-scale facilities, disaster zones, or high-security areas where constant surveillance is required. By leveraging IoT (Internet of Things) technology, the system ensures seamless communication between the vehicle and the command center, enhancing coordination and situational awareness.

The intended applications of this surveillance vehicle are vast, spanning industrial plants, military installations, public venues, and emergency response scenarios. In industrial settings, it can monitor for gas leaks or fires, preventing catastrophic accidents. In security-sensitive areas, it can detect unauthorized metallic objects, such as weapons or explosives, enhancing public safety. For urban environments, this technology could evolve into a smart patrolling unit, working during large events. The GPS data will be transmitted to a web-based dashboard, providing operators with a live feed of the vehicle's movement, sensor readings, and threat alerts.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING SYSTEM

Current surveillance and threat detection systems often relay on static sensors, manual patrols, or semi-automated monitoring solutions, which have several limitations in dynamic environments. Traditional security setups typically use fixed cameras, standalone gas detectors, or handheld metal scanners, requiring constant human supervision and intervention. While these systems provide basic monitoring capabilities, they lack real-time mobility, centralized intelligence, and automated response mechanisms, making them inefficient in rapidly evolving threat scenarios.

One major drawback of existing systems is their limited coverage area. Fixed sensors and CCTV cameras can only monitor predefined zones, leaving blind spots that adversaries or hazards can exploit. Additionally, most conventional detectors (such as smoke or gas sensors) operate in isolation, without integration into a unified alert system. This means that even if a threat is detected, the response depends on human operators noticing the alarm and taking action leading to potential delays in emergencies.

Another challenge is the absence of real-time tracking and data fusion. Many security vehicles used in patrols or industrial inspections rely on manual reporting rather than live GPS updates and automated threat logging. Without instant data transmission, command centers cannot accurately track the vehicle’s location or assess detected threats in real time. Furthermore, most existing solutions do not incorporate IoT-based cloud connectivity, preventing remote monitoring and data analysis for predictive threat assessment.

In high-risk environments like chemical plants, military zones, or disaster sites, these inefficiencies can have severe consequences. For example, a gas leak might go unnoticed until it spreads dangerously, or an unauthorized metallic object could evade detection due to the lack of an integrated scanning system. The reliance on human vigilance also increases the risk of human error, fatigue, or oversight, particularly in prolonged surveillance operations.While some advanced security robots and drones have been introduced in recent years, they are often cost- prohibitive, complex to operate, or limited to specific functions There remains a critical need for an affordable, mobile, and multi-functional threat-detection system that combines real-time sensor fusion, automated alerts, and live tracking into a single, user-friendly platform.

By leveraging modern IoT and AI- driven technologies, the proposed system seeks to overcome the shortcomings of traditional surveillance methods and set a new standard for intelligent threat monitoring.

2.2 PROPOSED SYSTEM

The Multi-Threat Surveillance Vehicle represents a next-generation security solution designed to overcome the limitations of conventional monitoring systems. Unlike static sensors or manual patrols, this intelligent mobile platform integrates advanced detection technologies, real time IoT connectivity, and automated response mechanisms into a unified, high-efficiency system.

At its core, the vehicle employs a multi-sensor fusion network combining gas detectors (for toxic/combustible gases), smoke/fire sensors, and metal detection systems to identify potential threats with high accuracy. When hazards are detected, the system triggers multi-stage alerts activating local alarms (buzzers/strobe lights) while simultaneously pushing instant notifications to a operator gadget.

A key innovation is the vehicle's autonomous GPS enabled tracking system, which continuously broadcasts its location and operational status to security personnel via an interactive web dashboard. This vehicle surveillance for different environments like industrial plants (gas/chemical monitoring), public venues (weapons detection), or disaster zones (radiation/air quality sensing).Where this vehicle followed with the operator-guided path.

2.3 EMBEDDED SYSTEMS

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end- user needs. Embedded systems control many devices in common use today. Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors.

For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure. In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points . A modern example of embedded system is shown in fig: 2.1.

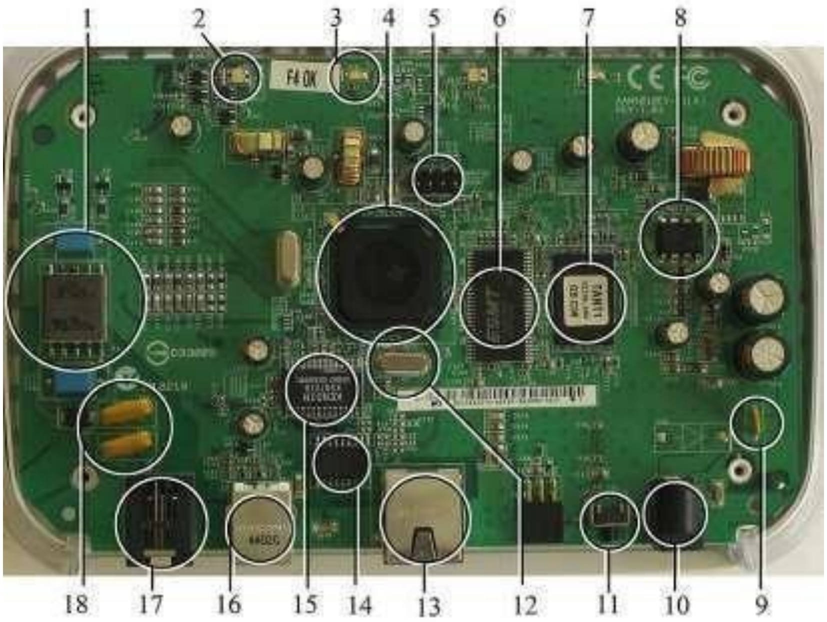


Fig 2.1:A Modern example of Embedded System

Labeled parts include microprocessor (4), RAM (6), flash memory (7).Embedded systems programming is not like normal PC programming. In many ways, programming for an embedded system is like programming PC 15 years ago.The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

2.3.1 History:

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit (IC) in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect real-time flight data. In 1965, Autonoetic, now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system.

It is widely recognized as the first mass- produced embedded system. When the Minuteman II went into production in 1966, the D-17B was replaced with the NS-17 missile guidance system, known for its high- volume use of integrated circuits. In 1968, the first embedded system for a vehicle was released; the Volkswagen 1600 used a microprocessor to control its electronic fuel injection system. By the late 1960s and early 1970s, the price of integrated circuits dropped, and usage surged.

The first microcontroller was developed by Texas Instruments in 1971. The TMS 1000 series, which became commercially available in 1974, contained a 4-bit processor, read-only memory (ROM) and random-access memory (RAM), and cost around $2 apiece in bulk orders. Also, in 1971, Intel released what is widely recognized as the first commercially available processor, the 4004. The 4-bit microprocessor was designed for use in calculators and small electronics, though it required eternal memory and support chips. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. The 8-bit Intel 8008, released in 1972, had 16 KB of memory; the Intel 8080 followed in 1974 with 64 KB of memory. The 8080's successor, x86 series, was released in 1978 and is still largely in use today.In 1987, the first embedded operating system, the real-time VxWorks, was released by Wind River, followed by Microsoft's Windows Embedded CE in 1996. By the late 1990s, the first embedded Linux products began to appear

2.3.2 Tools:

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the Unix world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point you will almost always find a compiler bug of some sort.Debugging tools are another issue. Since you can't always run general programs on your embedded processor, you can't always run a debugger on it. This makes fixing your program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if you stop on a breakpoint permanent equipment damage can occur.

2.3.3 Resources:

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means your programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt you. Luckily, this won't happen too often- use reasonably efficient algorithms to start, and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well.

Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, you will frequently sacrifice processor time for memory, rather than the reverse). It also means you can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether. To optimize performance on resource-constrained embedded systems, focus on minimizing memory usage and avoiding dynamic allocation, as heap fragmentation can cause instability. Manual loop unrolling and inline assembly may sometimes be necessary for time-critical sections, though they reduce portability. Finally, leveraging hardware peripherals (DMA, interrupts, and hardware accelerators) can offload the CPU.

2.3.4 Real time issues:

Embedded systems frequently control hardware, and must be able to respond to them in real time. Failure to do so could cause inaccuracy in measurements, or even damage hardware such as motors. This is made even more difficult by the lack of resources available. Almost all embedded systems need to be able to prioritize some tasks over others, and to be able to put off/skip low priority tasks such as UI in favor of high priority tasks like hardware control.

2.4 NEED FOR EMBEDDED SYSTEMS:

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So when implementing a new form of control, it's wiser to just buy the generic chip and write your own custom software for it.Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing your own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response.

The main elements that make embedded systems unique are its reliability and ease in debugging. Embedded systems have become an essential part of modern technology, revolutionizing various industries with their efficiency and versatility. These systems consist of a combination of hardware and software designed to perform specific tasks. Unlike general-purpose computers, embedded systems are dedicated to particular functions, making them more efficient, reliable, and cost-effective. The increasing affordability and advancements in microprocessors, microcontrollers, and FPGA chips have further accelerated the widespread adoption of embedded technology. One of the most significant applications of embedded systems is in consumer electronics. Devices such as smartphones, smart TVs, washing machines, digital cameras, and gaming consoles rely on embedded technology to enhance user experience and functionality. Smart home automation, including voice assistants and intelligent security systems, also utilizes embedded systems to enable seamless interaction and control. These systems consist of a combination of hardware and software designed to perform specific tasks.

Unlike general-purpose computers, embedded systems are dedicated to particular functions, making them more efficient, reliable, and cost-effective.In the automotive industry, embedded systems play a crucial role in vehicle safety, performance, and automation. They are used in engine control units (ECUs), anti-lock braking systems (ABS), airbag deployment, and advanced driver-assistance systems (ADAS). Electric and hybrid vehicles incorporate embedded controllers for battery management and energy efficiency, ensuring optimal performance and sustainability.Healthcare is another sector that greatly benefits from embedded systems. Medical devices such as pacemakers, MRI machines, glucose monitors, and robotic surgical tools rely on embedded technology for accurate diagnosis, treatment, and patient monitoring. Real-time monitoring devices enable continuous tracking of vital signs, allowing doctors to make timely medical decisions and improve patient outcomes.

Embedded systems also play a significant role in industrial automation. Manufacturing plants use embedded controllers to manage robotic arms, conveyor belts, and automated process control systems. With the integration of artificial intelligence (AI), embedded solutions enable predictive maintenance, reducing downtime and improving overall productivity. These systems ensure that industries run efficiently with minimal human intervention.In telecommunications, embedded technology is found in routers, modems, and signal processing units. These systems facilitate seamless communication by optimizing network performance and data transmission. The implementation of 5G networks and the Internet of Things (IoT) has further increased the demand for embedded systems, as they provide the backbone for connected devices and smart infrastructures.

The aerospace and defense industries also rely on embedded systems for mission-critical applications. Avionics systems in aircraft use embedded computers for navigation, autopilot, and communication, ensuring safe and efficient air travel. In military applications, embedded technology is used in radar systems, missile guidance, drones, and surveillance equipment to enhance security and defense capabilities.One of the main advantages of embedded systems is their ability to be customized for specific tasks. Instead of developing expensive custom-made chips, manufacturers can use generic microcontrollers and microprocessors while writing custom software to meet their requirements. This approach not only reduces costs but also allows for greater flexibility and adaptability in design.Additionally, embedded systems offer real-time performance, which is crucial in applications where immediate responses are required, such as healthcare monitoring and automotive safety.Avionics systems in aircraft use embedded computers for navigation, autopilot, and communication, ensuring safe and efficient air travel.The future of embedded systems is being shaped by emerging technologies such as AI, IoT, and edge computing. AIpowered embedded solutions are transforming industries by enabling smart automation, real-time analytics, and adaptive learning. IoT-connected embedded systems facilitate seamless communication between devices, driving advancements in smart cities, healthcare, and industrial automation. Moreover, edge computing allows embedded devices to process data locally rather than relying on cloud servers, enhancing security and reducing latency.

In conclusion, embedded systems have become the foundation of modern technology, providing intelligent and efficient solutions for various industries. Their adaptability, affordability, and integration with advanced technologies make them indispensable in today's world. As innovations in AI, IoT, and edge computing continue to evolve embedded systems. They remain at the forefront of technological advancements, shaping the future of automation, connectivity, and smart solutions.

2.4.1 Debugging:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.
- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.
- Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software(and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor).

Embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a logic analyzer, for instance.

2.4.2 Reliability:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them if an error occurs. Therefore the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

- The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
- The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.
- The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.
- A variety of techniques are used, sometimes in combination, to recover from errors both software bugs such as memory leaks, and also soft errors in the hardware:
- Watchdog timer that resets the computer unless the software periodically notifies the watchdog
- Subsystems with redundant spares that can be switched over to
- software "limp modes" that provide partial function

- Designing with a Trusted Computing Base (TCB) architecture ensures a highly secure & reliable system environment
- An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software.
- This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
- Immunity Aware Programming

2.5 EXPLANATION OF EMBEDDED SYSTEMS:

2.5.1 Software architecture:

There are several different types of software architecture in common use.

Simple Control Loop: In this design, the software simply has a loop. The loop calls subroutines, each of which manages a part of the hardware or software.

Interrupt Controlled System: Some embedded systems are predominantly interrupt controlled. This means that tasks performed by the system are triggered by different kinds of events. An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte. These kinds of systems are used if event handlers need low latency and the event handlers are short and simple.

Usually these kinds of systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

Cooperative Multitasking: A non-preemptive multitasking system is very similar to the simple control loop scheme, except that the loop is hidden in an API. The programmer defines a series of tasks, and each task gets its own environment to “run” in. When a task is idle, it calls an idle routine, usually called “pause”, “wait”, “yield”, “nop” (stands for no operation), etc

Primitive Multitasking: In this type of system, a low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt). This is the level at which the system is generally considered to have an "operating system" kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel any code can potentially damage the data of another task (except in larger systems using an MMU) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronization strategy, such as message queues, semaphores or a non-blocking synchronization scheme. Because of these complexities, it is common for organizations to buy a real-time operating system, allowing the application programmers to concentrate on device functionality rather than operating system services, at least for large systems; smaller systems often cannot afford the overhead associated with a generic real time system, due to limitations regarding memory size, performance, and/or battery life.

Microkernels And Exokernels: A microkernel is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc.

In general, microkernels succeed when the task switching and intertask communication is fast, and fail when they are slow. Exokernels communicate efficiently by normal subroutine calls. The hardware and all the software in the system are available to, and extensible by application programmers. Microkernels prioritize stability and modularity at the cost of performance due to frequent IPC between servers and the kernel. Optimizations like L4's synchronous IPC reduce overhead but don't eliminate it.

2.5.2 Stand alone embedded system:

These systems takes the input in the form of electrical signals from transducers or commands from human beings such as pressing of a button etc.., process them and produces desired output. This entire process of taking input, processing it and giving output is done in standalone mode. Such embedded systems comes under stand alone embedded systems. Eg: microwave oven, air conditioner etc. These embedded systems follow a relative dead line time period i.e.., if the task is not done in a particular time that will not cause damage to the equipment.

2.5.3 Real-time embedded systems:

Embedded systems which are used to perform a specific task or operation in a specific time period those systems are called as real-time embedded systems. There are two types of real-time embedded systems.Hard Real-time embedded systems: These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment.Eg: consider a system in which we have to open a valve within 30 milliseconds. If this valve is not opened in 30 ms this may cause damage to the entire equipment. So in such cases we use embedded systems for doing automatic operations.

Soft Real Time embedded systems: These embedded systems follow a relative dead line time period i.e.., if the task is not done in a particular time that will not cause damage to the equipment. Eg: Consider a TV remote control system, if the remote control takes a few milliseconds delay it will not cause damage either to the TV or to the remote control. These systems which will not cause damage when they are not operated at considerable time period those systems comes under soft real-time embedded systems.

2.5.4 Network communication embedded systems:

A wide range network interfacing communication is provided by using embedded systems. Consider a web camera that is connected to the computer with internet can be used to spread communication like sending pictures, images, videos etc.., to another computer with internet connection throughout anywhere in the world. Consider a web camera that is connected at the door lock. Whenever a person comes near the door, it captures the image of a person and sends to the desktop of your computer which is connected to internet. : These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment .Eg: consider a system in which we have to open a valve within 30 milliseconds. If this valve is not opened in 30 ms this may cause damage to the entire equipment. This gives an alerting message with image on to the desktop of your computer, and then you can open the door lock just by clicking the mouse. Fig: 2.2 show the network communications in embedded systems. These embedded systems follow an absolute dead line time period i.e.., if the tasking is not done in a particular time period then there is a cause of damage to the entire equipment. This entire process of taking input, processing it and giving output is done in standalone mode.

Fig 2.2: Network communication embedded systems

2.5.5 Different types of processing units:

- The central processing unit (c.p.u) can be any one of the following microprocessor, microcontroller, digital signal processing.
- Among these Microcontroller is of low cost processor and one of the main advantage of microcontrollers is, the components such as memory, serial communication interfaces, analog to digital converters etc.., all these are built on a single chip. The numbers of external components that are connected to it are very less according to the application.
- Microprocessors are more powerful than microcontrollers. They are used in major applications with a number of tasking requirements. But the microprocessor requires many external components like memory, serial communication, hard disk, input output ports etc.., so the power consumption is also very high when compared to microcontrollers.
- Digital signal processing is used mainly for the applications that particularly involved with processing of signals. Each processing unit type represents a different balance of flexibility, power efficiency, and computational capability.
- Emerging architectures include neuromorphic processors that mimic biological neural networks for efficient AI processing, and quantum processing units (QPUs) that leverage quantum mechanics for solving complex problems.

2.6 APPLICATIONS OF EMBEDDED SYSTEMS:

2.6.1 Consumer applications:

At home we use a number of embedded systems which include microwave oven, remote control, vcd players, dvd players, camera etc….

Fig 2.3: Automatic coffee makes equipment

2.6.2 Office automation:

We use systems like fax machine, modem, printer etc…

Fig 2.4: Fax machine

Fig 2.5: Printing machine

2.6.3 Industrial automation:

Today a lot of industries are using embedded systems for process control. In industries we design the embedded systems to perform a specific operation like monitoring temperature, pressure, humidity ,voltage, current etc.., and basing on these monitored levels we do control other devices, we can send information to a centralized monitoring station.

Fig 2.6: Robot

In critical industries where human presence is avoided there we can use robot**s** which are programmed to do a specific operation. These robots are programmed with advanced algorithms to handle tasks that may be too dangerous for humans. For example, in nuclear power plants, robots can be deployed for maintenance and inspection in radioactive environments. Similarly, in deep-sea exploration, robotic systems can operate in extreme pressure conditions where human survival is impossible. In the medical field, robotic-assisted surgeries are revolutionizing healthcare by ensuring precision and reducing human errors.Moreover, in space exploration, robots like rovers and drones collect data from planets where humans cannot yet travel. Industrial automation in manufacturing plants also relies on robots to perform repetitive tasks with high accuracy and speed.

Overall, robots are transforming various industries by enhancing efficiency, safety, and productivity.In disaster response, robots assist in search and rescue operations, reaching areas inaccessible to humans after earthquakes or fires. Agriculture is also evolving with robotic farming systems that automate planting, harvesting, and monitoring crop health. With continuous technological advancements, robots are reshaping industries, ensuring safety, innovation, and efficiency while expanding human capabilities beyond natural limitations. Robots are programmable machines designed to perform tasks autonomously or with minimal human intervention, capable of executing complex actions that often mimic biological movements. Comprised of key components including sensors for environmental perception, actuators for physical movement, and an intelligent control system for processing and decision-making, modern robots represent the convergence of mechanical engineering.

2.6.4 Computer networking:

Embedded systems are used as bridges routers etc..

Fig 2.7: Computer networking

2.6.5 Tele communications:

Cell phones, web cameras etc.

Fig 2.8 Cell Phone **Fig 2.9 Web Camera**

CHAPTER 3

HARDWARE REQUIREMENTS

3.1 SCHEMATIC DIAGRAM

In this chapter, schematic diagram and interfacing of each module is considered**.**

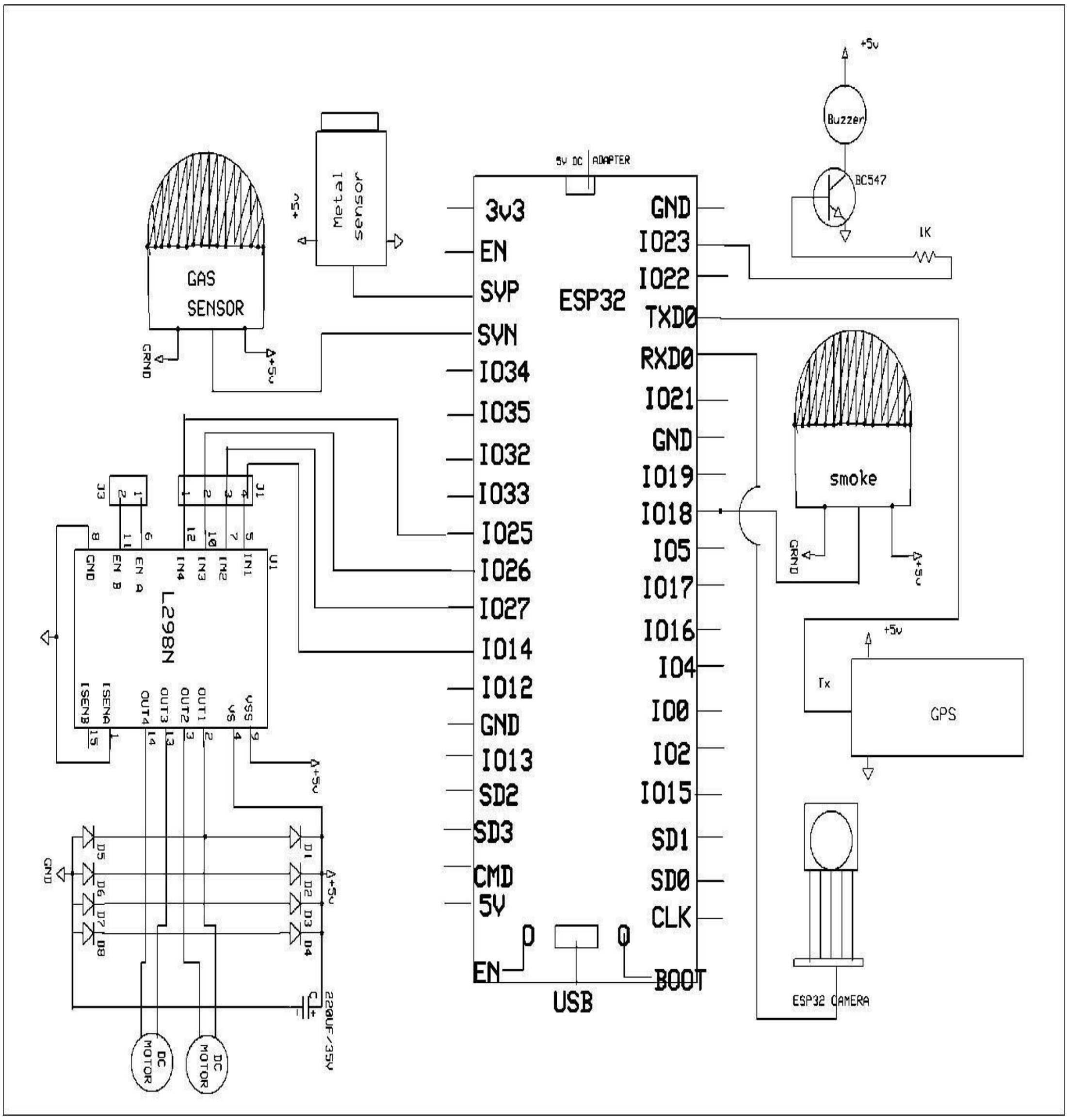


Fig 3.1 Schematic diagram of Multi Threat Surveillance Vehicle

The above schematic diagram explains the interfacing section of each component with micro controller and input output modules.

3.2 HARDWARE COMPONENTS:

1. **ESP32 Microcontroller:**

○ Acts as the brain of the system. ○ Controls various sensors and peripherals. ○ Communicates with other modules via GPIO pins.

2. **Gas Sensor (MQ Series):** ○ Detects hazardous gases in the environment. ○ Provides an analog output to the ESP32 for monitoring. ○ Used for chemical or gas leak detection.
3. **Smoke Sensor:** ○ Detects smoke and possible fire hazards. ○ Provides an output signal when smoke is detected.
4. **GPS Module:** ○ Tracks the real-time location of the robot. ○ Sends location data via a **TX** (Transmit) pin.
5. **Buzzer Alert System:** ○ A buzzer is connected via a **BC547 transistor** for alerting. ○ Activated when a gas leak, smoke, or another threat is detected.
6. **ESP32-CAM Module:** ○ Used for **real-time video streaming** and surveillance. ○ Helps in remote monitoring of the robot's surroundings.
7. **L298N Motor Driver :** ○ Controls the movement of the robot's wheels or legs. ○ Allows forward, backward, left, and right motion

This system is a smart robot using an ESP32 microcontroller as its core. It integrates multiple sensors and modules for safety and surveillance:

- Gas & Smoke Sensors: Detect hazardous gases and smoke, sending signals to the ESP32.
- GPS Module: Provides real-time location tracking.
- Buzzer Alert System: Sounds an alarm when a threat is detected.
- ESP32-CAM: Enables real-time video streaming for remote monitoring. L298N Motor

Driver: Controls the robot’s movement in all direct.

CHAPTER 4
SOFTWARE DESCRIPTION

4.1 SOFTWARE

How to Upload Code to the ESP32 Module

Uploading (or "dumping") code into an ESP32 module requires a few steps using **Arduino IDE** or **Platform IO in VS Code**. Below is a step-by-step guide for both methods:

1. Using Arduino IDE

Step 1: Install Arduino IDE and ESP32 Board Support

1. Download and install Arduino IDE from Arduino's official website.
2. Open Arduino IDE and go to:

○ File → Preferences ○ In the Additional Board Manager URLs field, add this URL: https://raw.githubusercontent.com/espressif/arduino-esp32/gpages/package_esp32_index.json

- Click OK.

3. Now, install ESP32 board support:

- Go to Tools → Board → Board Manager
- Search for ESP32
- Click Install on "esp32 by Espressif Systems"

Step 2: Connect ESP32 to Your Computer

1. Use a USB to Micro-USB cable to connect the ESP32 board to your PC.
2. Ensure you have the correct USB drivers installed: ○ For CP2102 USB-to-serial chips, download drivers from Silicon Labs. ○ For CH340 chips, download drivers from WCH official site.

Step 3: Select the Correct Board and Port

1. Go to Tools → Board, then select your ESP32 board (e.g., "ESP32 Dev Module").
2. Go to Tools → Port, then select the COM port associated with the ESP32. Step 4: Write or Open Your Code

- Open an Arduino sketch (or create a new one). • Example Blink LED Code for ESP32:

```
#define LED_BUILTIN 2 // On-board
LED for ESP32 void setup() {
pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUIL
  TIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUIL
  TIN, LOW); delay(1000);
}
```

Step 5: Put ESP32 in Boot Mode (if needed)

- Some ESP32 boards require manual boot mode activation.
- If the upload fails, press and hold the BOOT button on the ESP32 while uploading the

Code.. • Release it when you see “Connecting…” in the Arduino IDE.

Step 6: Upload the Code

- Click the Upload (Arrow) button in Arduino IDE.
- Wait for "Done uploading" message.
- If successful, your ESP32 is now running the uploaded code!

4.1 Procedural steps for compilation, simulation and dumping:

This board is based on the Arduino ESP32 Board Package, that is derived from the original ESP32 Board Package. It provides a rich set of examples to access the various features on your board, which is accessed directly through the IDE. So by this IDE we can directly dump the code to microcontroller so that we can run the specific Embedded system. Developers benefit from an extensive collection of pre-written example codes covering common use cases like wireless communication, sensor interfacing, and power management. The seamless integration allows users to transition smoothly from basic Arduino sketches to more advanced ESP32specific functionalities without changing development environments.

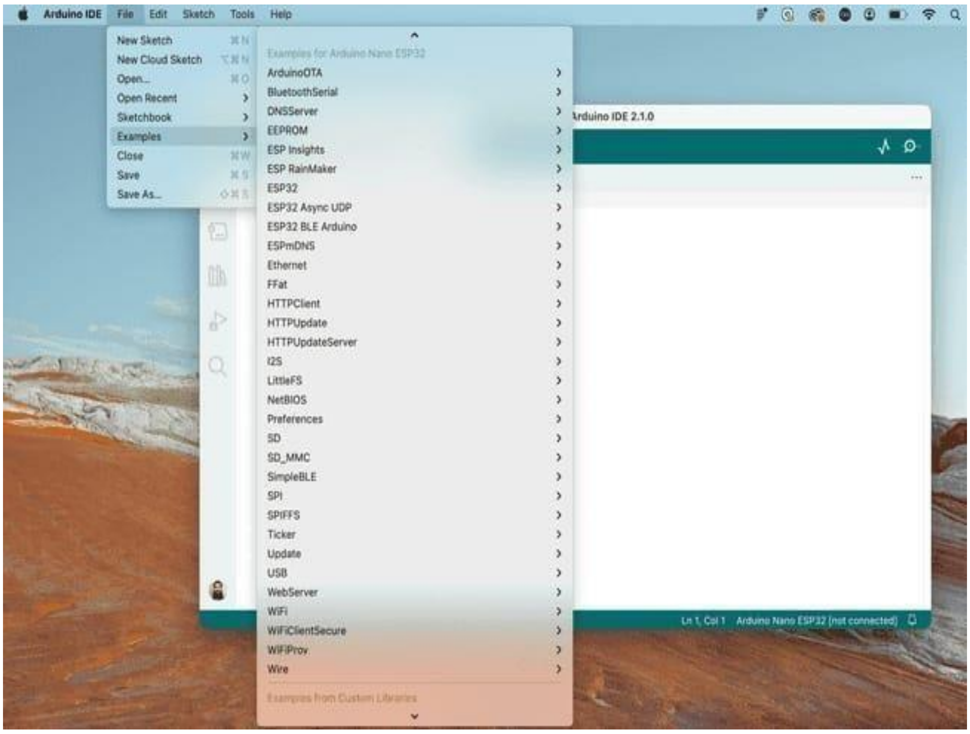


Fig 4.1 Software IDE

This development board leverages the Arduino ESP32 Board Package, which builds upon the foundational ESP32 Board Package while adding enhanced compatibility with the Arduino ecosystem. By supporting the familiar Arduino IDE, it significantly lowers the barrier to entry for both beginners and experienced developers working with ESP32-based projects. The package includes optimized libraries that fully utilize the ESP32's dual-core processor, WiFi/Bluetooth capabilities, and rich peripheral set while maintaining Arduino's signature simplicity. Developers benefit from an extensive collection of pre-written example codes covering common use cases like wireless communication, sensor interfacing, and power management. The seamless integration allows users to transition smoothly from basic Arduino sketches to more advanced ESP32-specific functionalities without changing development environments.

The package maintains backward compatibility with most existing Arduino libraries while adding ESP32-specific enhancements for improved performance. Automatic driver installation and board detection simplify the setup process compared to working with native ESP-IDF toolchains.

CHAPTER 5

WORKING MODEL AND COMPONENTS

5.1 BLOCK DIAGRAM:

The block diagram of the project and design aspect of independent modules are considered. Block diagram is shown in fig: 5.1:

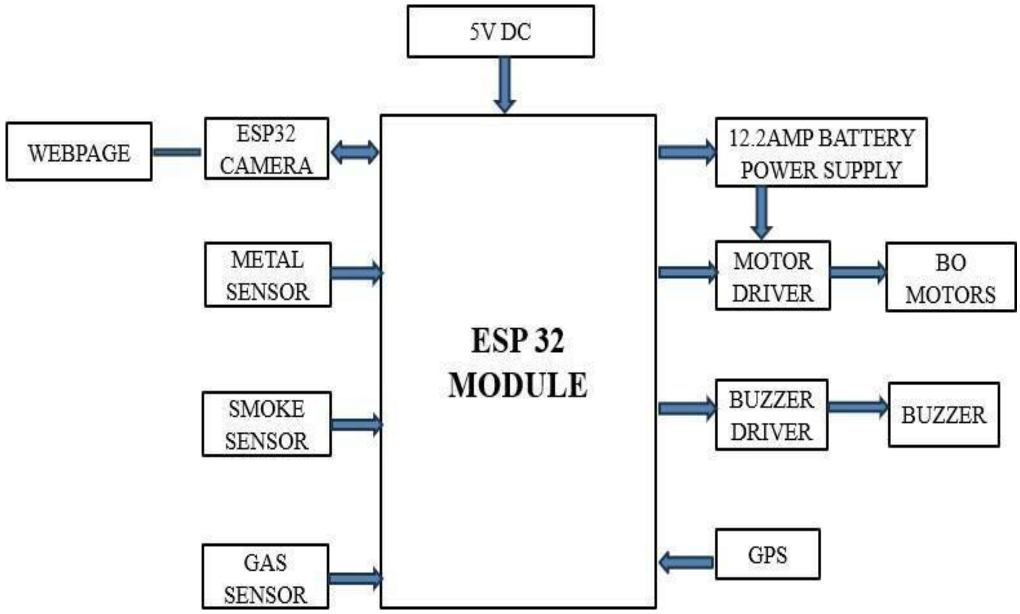


Fig 5.1: Block diagram of Multi Threat Surveillance Vehicle

The main blocks of this project are:

- ESP32 Module
- DC motor with L293D motor driver.
- ESP32 Camera.
- GPS.
- 12v,2amp Rechargeable Battery.
- GAS sensor.
- Smoke sensor.
- Metal sensor.
- Buzzer

5.1 WORKING

The multi-threat surveillance vehicle is an advanced robotic system designed to detect and respond to environmental hazards using integrated sensors for metal, smoke, and gas detection. Equipped with an ESP32 camera, it provides real time video streaming of its surroundings, while a GPS module transmits its exact location to users. The vehicle operates over Wi-Fi, enabling seamless data transmission from sensors to a user interface, where live metrics and camera feeds are displayed. Users can remotely control the vehicle's movement via a web application, with commands relayed through the ESP32 microcontroller. The system's core components ESP32CAM, metal sensor, smoke sensor, gas sensor, GPS, buzzer, and DC motors are interfaced with the ESP32 microcontroller, which processes inputs and coordinates responses.

Power is supplied by a 12V 2A battery, stepped down to 5V using an LM2596 buck converter to operate the microcontroller, camera, sensors, and GPS. When the user presses directional controls on the web interface, the ESP32 camera receives the command and forwards it to the microcontroller, which adjusts the DC motors via an L298N motor driver. Autonomous hazard detection triggers an immediate stop, activating the buzzer for local alerts and sending the GPS coordinates to the user's device. The ESP32's embedded C program orchestrates these functions, ensuring real-time threat analysis and response. Metal detection safeguards against concealed weapons or explosives, while smoke and gas sensors prevent fire or toxic exposure risks. The live video feed enhances situational awareness, allowing operators to navigate complex environments remotely.

The L298N driver ensures precise motor control, enabling smooth traversal across varied terrains. Sensor data is logged and displayed on the web dashboard, providing actionable insights. The buzzer's alarm is supplemented by visual alerts on the interface, ensuring no threat goes unnoticed. GPS updates are mapped in real time, aiding in rapid deployment during emergencies. The ESP32's dual-core processor handles concurrent tasks sensor polling, motor control, and data transmission—without latency. Encrypted Wi-Fi ensures secure communication between the vehicle and control interface. The metal sensor's sensitivity is adjustable to minimize false positives in cluttered environments. Smoke detection leverages particulate analysis to distinguish between fire and ambient dust. Gas sensors monitor for combustible or toxic leaks, triggering evacuation protocols if thresholds are exceeded. The buck converter's stable 5V output safeguards sensitive electronics from voltage fluctuations. Motor driver diagnostics prevent overheating during prolonged use. The embedded C firmware optimizes resource usage, prioritizing critical interrupts like hazard detection. Remote diagnostics via the web app include battery status and signal strength indicators. Field tests validate the system's reliability in low-connectivity areas using offline data caching. The vehicle's compact frame houses components securely, with cable management reducing interference. Future iterations may integrate solar charging or swarm coordination for large-area surveillance. By merging hardware robustness with software intelligence, this project delivers a scalable solution for modern security and safety challenges.

5.2.1 ESP32 Camera:

Fig 5.2.1 ESP32 Camera

ESP32 Features :

- ESP32-CAM :
- The smallest 802.11b/g/n Wi-Fi BT SoC module.
- Low power 32-bit CPU, can also serve the application processor.
- Up to 160MHz clock speed, summary computing power up to 600 DMIPS.
- Built-in 520 KB SRAM, external 4MPSRAM.
- Supports UART/SPI/I2C/PWM/ADC/DAC.
- Support OV2640 and OV7670 cameras, built-in flash lamp.
- Support image Wi-Fi upload.
- Supports TF card.
- Supports multiple sleep modes.

- Embedded Lwip and Free RTOS.
- Supports STA/AP/STA+AP operation mode.
- Support Smart Config/AirKiss technology.
- Support for serial port local and remote firmware upgrades (FOTA)

Specifications :

- Wireless Module: ESP32-S WiFi 802.11 b/g/n + Bluetooth 4.2 LE module with PCB antenna, connector, 32Mbit SPI flash, 4MBit PSRAM.
- External Storage: micro SD card slotup to 4GB.
- Camera
- FPC connector.
- Support for OV2640 (sold with a board) or OV7670 cameras.
- Image Format: JPEG(OV2640 support only), BMP, grayscale.
- LED flashlight.
- Expansion: 16x through-holes with UART, SPI, I2C, PWM.
- Misc: Reset button.
- Power Supply: 5V via pin header.
- Power Consumption.
  - Flash LED off: 180mA @ 5V.
  - Flash LED on to maximum brightness: 310mA @ 5V.
  - Deep-sleep: 6mA @ 5V min.
  - Modem-sleep: 20mA @ 5V min.
  - Light-sleep: 6.7mA @ 5V min.
- Dimensions (ESP32): 40 x 27 x 12 (LxWxH) mm.
- Temperature Range: Operating: -20 ~ 85 ; storage: -40 ~ 90 @ <90%RH.

The ESP32 module offers a range of expansion options, including 16 through-holes supporting UART, SPI, I2C, and PWM communication protocols, making it highly versatile for various applications. A reset button is also included for easy system reboot and troubleshooting. The module operates on a 5V power supply, which is provided via a pin header. In terms of power consumption, it draws 180mA at 5V when the flash LED is off, while turning the flash LED to maximum brightness increases consumption to 310mA at

5V. In different low-power modes, the module efficiently conserves energy, consuming as little as 6mA at 5V in deep sleep, 20mA at 5V in modem sleep, and 6.7mA at 5V in light sleep mode.

5.2.2 ESP-32 Module:

Fig: 5.2.2 ESP-32 Module

The ESP32 is a powerful and versatile microcontroller developed by Espressif Systems, widely used in IoT, smart devices, and embedded systems. It features a dual-core Xtensa LX6 processor, running at up to 240 MHz, with built-in Wi-Fi (802.11 b/g/n) and Bluetooth 4.2 (including BLE) for seamless wireless communication. The module is designed for high performance and low power consumption, making it ideal for battery-powered applications. It supports various peripherals, including GPIOs, SPI, I2C, UART, ADC, DAC, and PWM, allowing easy integration with sensors, displays, and actuators. With FreeRTOS support, the ESP32 enables multitasking, making it suitable for real-time applications.

One of the standout features of the ESP32 is its security capabilities, including AES encryption, secure boot, and hardware acceleration for data protection. It also supports Over-the-Air (OTA) updates, enabling remote firmware upgrades. The module comes with a built-in capacitive touch sensor, making it useful for touch-based interfaces. Developers benefit from extensive support through the Arduino IDE, MicroPython, and ESP-IDF, offering flexibility in coding and development. Its affordability, open-source nature, and large community support make the ESP32 an excellent choice for DIY projects, industrial applications, and advanced IoT systems.

At the core of this module is the ESP32-D0WDQ6 chip. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The user may also power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is future proof: using Wi-Fi allows a large physical range and direct connection to the internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 µA, making it suitable for battery powered and wearable electronics applications. ESP32 supports a data rate of up to 150 Mbps, and 20.5 dBm output power at the antenna to ensure the widest physical range.

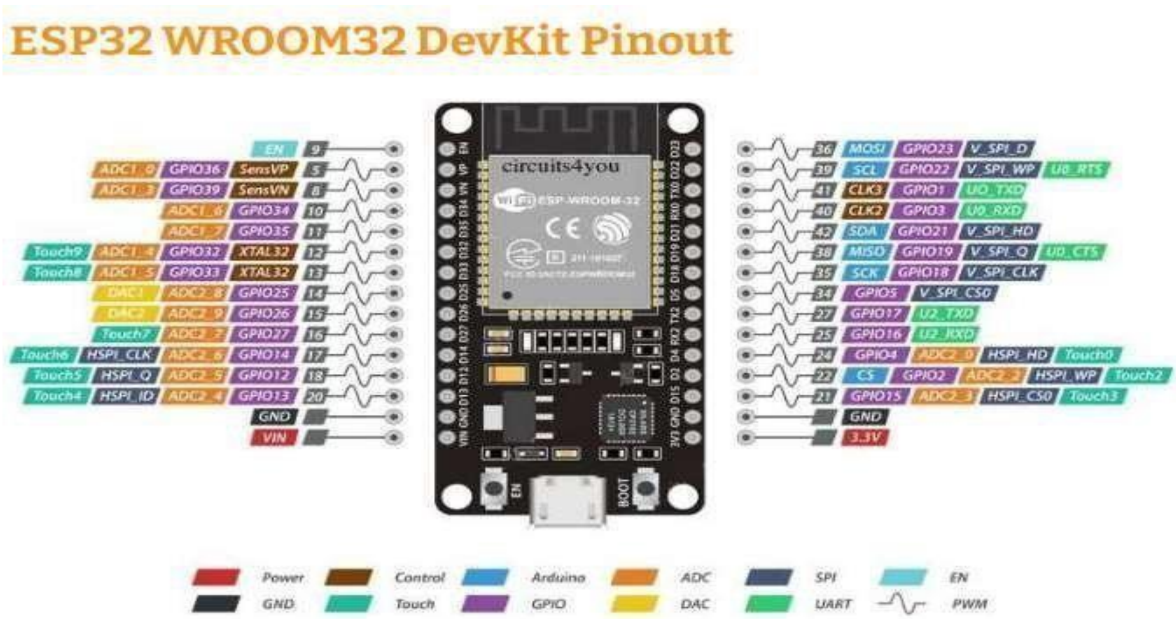


Fig 5.2.3 ESP32 WROOM32 DEVKIT POINT

ESP32 Peripherals Features

- 18 Analog-to-Digital Converter (ADC) channels
- 10 Capacitive sensing GPIOs
- 3 UART interfaces

- 3 SPI interfaces
- 2 I2C interfaces
- 16 PWM output channels
- 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces

GPIO Pins

ESP32 Wroom32 DevKit has total 25 GPIOs out of that few pins are Input only Pins,

Input Only Pins

- GPIO 34
- GPIO 35
- GPIO 36
- GPIO 39

Not all pins have input pullup, you need external pullup on these pins when using as input pullup.

Pins with internal pull up INPUT_PULLUP

- GPIO14
- GPIO16
- GPIO17
- GPIO18
- GPIO19
- GPIO21
- GPIO22
- GPIO23

Pins without internal pull up

- GPIO13
- GPIO25
- GPIO26
- GPIO27
- GPIO32
- GPIO33

Key Features

- Dual-Core Processor
- Built-in Wi-Fi
- Bluetooth 4.2 & BLE
- Ultra-Low Power Consumption
- Multiple GPIOs
- High-Speed SPI, I2C, and UART
- ADC & DAC Support
- Embedded Flash & SRAM
- Secure Boot & Encryption
- Support for FreeRTOS
- Capacitive Touch Sensor
- PWM Support
- Camera & Display Compatibility
- Over-the-Air (OTA) Updates
- Affordable & Open-Source

5.2.3 Battery power supply:

12v 1amh battery is used to give the power supply of the robot.

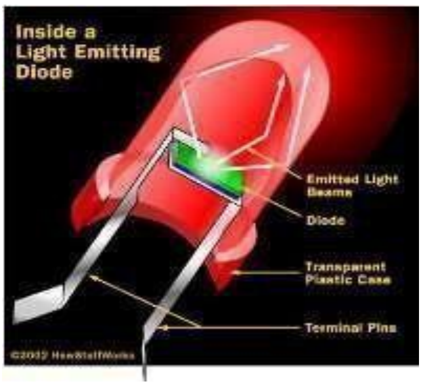
Fig 5.2.4 Battery

The surveillance vehicle is powered by a robust 12V 2A rechargeable lithium-ion battery that serves as the primary energy source for all its components. This battery pack is specifically chosen for its ability to deliver consistent power to both high-current components like the DC motors and sensitive electronics simultaneously. To ensure stable operation, the system incorporates an LM2596 buck converter that efficiently steps down the 12V supply to a regulated 5V output, which safely powers the ESP32 microcontroller, camera module, GPS, and all connected sensors. The power management system includes intelligent features like low voltage detection that automatically safeguards the battery from deep discharge, significantly extending its lifespan.

Designed for field operations, the compact and lightweight battery allows for quick swaps or recharging between missions. Careful power distribution architecture prevents voltage fluctuations even when the motors draw peak current during operation. The current configuration provides several hours of continuous operation, with potential for future enhancements like integrated solar charging panels that could further extend mission duration in outdoor environments.

5.2.4 LED:

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness. The internal structure and parts of a led are shown in figures 3.4.1 and 3.4.2 respectively.



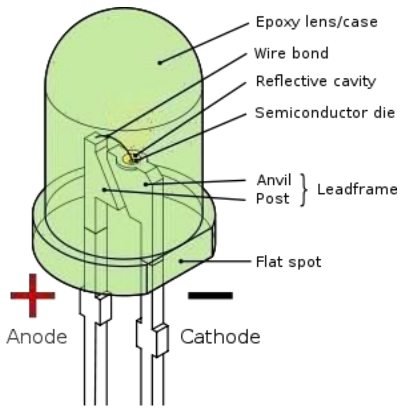


Fig 5.2.5.: Inside a LED

Fig 5.2.6: Parts of a LED

Working:

The structure of the LED light is completely different than that of the light bulb. Amazingly, the LED has a simple and strong structure. The light-emitting semiconductor material is what determines the LED's color. The LED is based on the semiconductor diode. When a diode is forward biased (switched on), electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An LED is usually small in area (less than 1 mm^2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. However, they are relatively expensive and require more precise current and heat management than traditional light sources. Current LED products for general lighting are more expensive to buy than fluorescent lamp sources of comparable output. They also enjoy use in applications as diverse as replacements for traditional light sources in automotive lighting (particularly indicators) and in traffic signals. The electrical symbol and polarities of led are shown in fig: 3.4.3.

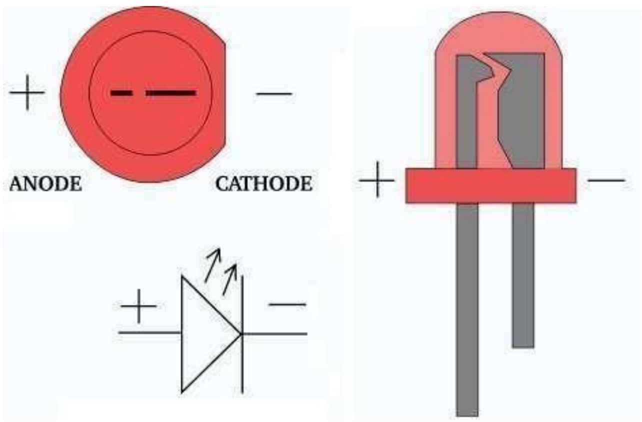


Fig 5.2.7: Electrical Symbol & Polarities of LED

LED lights have a variety of advantages over other light sources:

- High-levels of brightness and intensity
- High-efficiency
- Low-voltage and current requirements

- Low radiated heat
- High reliability (resistant to shock and vibration)
- No UV Rays
- Long source life
- Can be easily controlled and programmed Applications of LED fall into three major
- Visual signal application where the light goes more or less directly from the LED to the human eye, to convey a message or meaning.
- Illumination where LED light is reflected from object to give visual response of these objects.
- Generate light for measuring and interacting with processes that do not involve the human visual system.

5.2.5 DC Motor:

A DC motor uses electrical energy ,very typically through the interaction of magnetic fields and current carrying-conductors. The reverse process, producing electrical energy from mechanical energy, is accomplished by an alternator, generator or dynamo. Many types of electric motors can be run as generators, and vice versa. The input of a DC motor is current/voltage and its output is torque (speed). The efficiency and performance of a DC motor depend on factors like voltage, current, and magnetic field strength.

Fig 5.2.8 DC Motor

The DC motor has two basic parts: the rotating part that is called the armature and the stationary part that includes coils of wire called the field coils. The stationary part is also called the stator. Figure shows a picture of a typical DC motor, Figure shows a picture of a DC armature, and Fig shows a picture of a typical stator.

From the picture you can see the armature is made of coils of wire wrapped around the core, and the core has an extended shaft that rotates on bearings. You should also notice that the ends of each coil of wire on the armature are terminated at one end of the armature. The termination points are called the commutator, and this is where the brushes make electrical contact to bring electrical current from the stationary part to the rotating part of the machine.

When voltage is applied to the motor, current begins to flow through the field coil from the negative terminal to the positive terminal. This sets up a strong magnetic field in the field winding. Current also begins to flow through the brushes into a commutator segment and then through an armature coil. The current continues to flow through the coil back to the brush that is attached to other end of the coil and returns to the DC power source. The current flowing in the armature coil sets up a strong magnetic field in the armature.

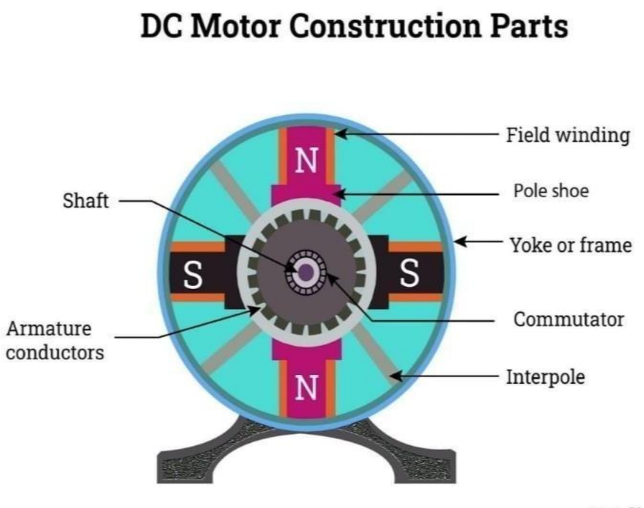


Fig 5.2.9 Simple electrical diagram of DC motor

A Direct Current (DC) motor is an electrical device that transforms electrical energy into mechanical energy through the generation of a magnetic field, powered by direct current. When you energize a DC motor, it creates a magnetic field within its stator. This field interacts with magnets situated on the rotor, prompting it to spin. To maintain the rotor's continuous rotation, the commutator delivers current to the motor’s windings via brushes linked to the power source. The magnetic field in the armature and field coil causes the armature to begin to rotate.

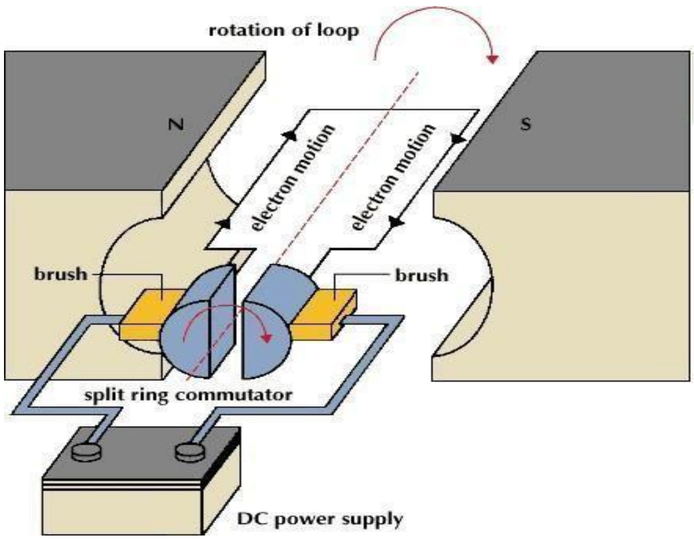


Fig 5.2.10 Operation of a DC Motor

This field interacts with magnets situated on the rotor, prompting it to spin. To maintain the rotor's continuous rotation, the commutator delivers current to the motor's windings via brushes linked to the power source. The magnetic field in the armature and field coil causes the armature to begin to rotate. This occurs by the unlike magnetic poles attracting each other and the like magnetic poles repelling each other. As the armature begins to rotate, the commutator segments will also begin to move under the brushes. As an individual commutator segment moves under the brush connected to positive voltage, it will become positive, and when it moves under a brush connected to negative voltage it will become negative.

In this way, the commutator segments continually change polarity from positive to negative. Since the commutator segments are connected to the ends of the wires that make up the field winding in the armature, it causes the magnetic field in the armature to change polarity continually from north pole to south pole. The commutator segments and brushes are aligned in such a way that the switch in polarity of the armature coincides with the location of the armature's magnetic field and the field winding's magnetic field. The switching action is timed so that the armature will not lock up magnetically with the field. Instead the magnetic fields tend to build on each other and provide. When the voltage is de-energized to the motor, the magnetic fields in the armature and the field winding will quickly diminish and the armature shaft's speed will begin to drop to zero. If voltage is applied to the motor again, the magnetic fields will strengthen and the armature will begin to rotate again.

DC MOTOR DRIVER:

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs.

When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high- speed output clamp diodes should be used for inductive transient suppression. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293and L293D are characterized for operation from 0°C to 70°C.

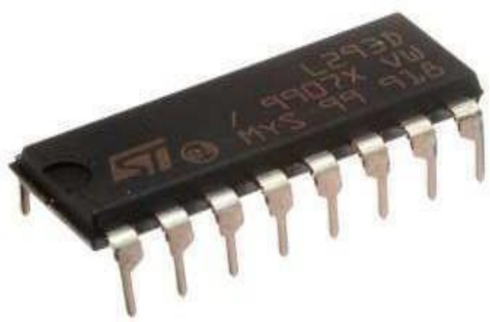


Fig 5.2.11 L293D IC

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Microcontrollers like Arduino, PIC, ARM .

Pin Diagram of L293D motor driver IC:

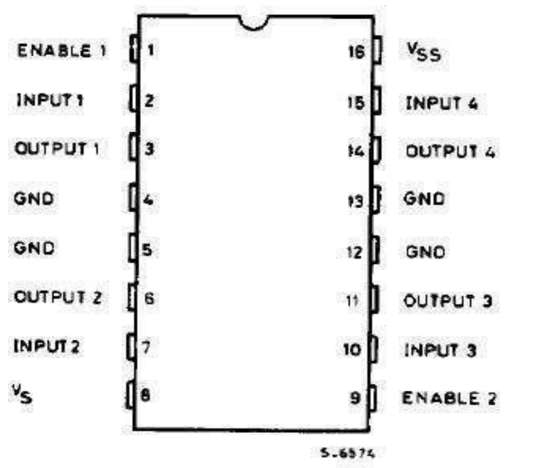


Fig 5.2.12 L293D Pin Diagram

Features of L293D:

- 600mA Output current capability per channel
- 1.2A Peak output current (non repetitive) per channel
- Enable facility
- Over temperature protection
- Logical “0”input voltage up to 1.5 v
- High noise immunity
- Internal clamp diodes

The L293D is a dual H-bridge motor driver IC designed to control the direction and speed of DC motors efficiently. It features an output current capability of up to 600mA per channel, allowing it to drive moderate-power motors with ease. Additionally, it supports a peak output current of 1.2A per channel for short durations, making it suitable for handling transient load spikes. The IC includes an enable facility, which provides precise control over motor operation by enabling or disabling specific channels as needed. To ensure reliability, the L293D is equipped with over-temperature protection, preventing damage due to excessive heat during extended operation.

5.2.6 Smoke sensor:

This semiconductor gas sensor detects the presence of combustible gas and smoke at concentrations from 300 to 10,000 ppm. The sensor's simple analog voltage interface requires only one analog input pin from your microcontroller.This flammable gas and smoke sensor detects the concentrations of combustible gas in the air and outputs its reading as an analog voltage.

Feature

- Power requirements: 5 VDC @ ~160mA
- Interface Type: Resistive
- Dimensions: 0.75" diameter x 0.65" tall excluding leads (19.1mm diameter x 16.55mm tall.

Operating temp range: -4 to +122 °F (-20 to +50 °C)

Connections connecting five volts across the heating (H) pins keeps the sensor hot enough to function correctly. Connecting five volts at either the A or B pins causes the sensor to emit an analog voltage on the other pins. A resistive load between the output pins and ground sets the sensitivity of the detector. Please note that the picture in the datasheet for the top configuration is wrong.

Both configurations have the same pinout consistent with the bottom configuration. The resistive load should be calibrated for your particular application using the equations in the datasheet, but a good starting value for the resistor is 10 kΩ.

Fig 5.2.13 MQ02 smoke

5.2.7 Metal sensor:

A88 Metal detector non-contact metal induction detection module as a metal detector. When it approaches any metal, it makes a sound. This is a module specifically designed to detect metal. The module operates by inducing currents in metal objects and responding when it occurs.

Fig 5.2.14 Metal

Sensor

A nice onboard buzzer signals when it detects something and an onboard potentiometer allow adjustment of sensitivity.

The power cables of the Metal detector non-contact metal induction detection module will need soldering on for the module to function, positive to the outside of the module and negative between the potentiometer and an electrolytic capacitor.

Features of a88 metal detector non-contact metal induction detection module :

- V+ Connect to power positive
- V- connect to power negative
- Adjust the potentiometer, let the modules work normally.
- Small and easy to use module.
- It comes with a Buzzer for metal detection indication.

5.2.8 Buzzer:

Basically, the sound source of a piezoelectric sound component is a piezoelectric diaphragm. A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate (brass or stainless steel, etc.). A piezoelectric ceramic plate is attached to a metal plate with adhesives. Applying D.C. voltage between electrodes of a piezoelectric diaphragm causes mechanical distortion due to the piezoelectric effect. For a misshaped piezoelectric element, the distortion of the piezoelectric element expands in a radial direction. And the piezoelectric diaphragm bends toward the direction. The metal plate bonded to the piezoelectric element does not expand. Conversely, when the piezoelectric element shrinks, the piezoelectric diaphragm bends in the direction Thus, when AC voltage is applied across electrodes, the bending is repeated, producing sound waves in the air.To interface a buzzer the standard transistor interfacing circuit is used. Note that if a different power supply is used for the buzzer, the 0V rails of each power supply must be connected to provide a common reference.

If a battery is used as the power supply, it is worth remembering that piezo sounders draw much less current than buzzers. Buzzers also just have one 'tone', whereas a piezo sounder is able to create sounds of many different tones. To switch on buzzer -high 1 To switch off buzzer -low 1

Fig 5.2.15 Buzzer

Notice (Handling) In Using Self Drive Method

- When the piezoelectric buzzer is set to produce intermittent sounds, sound may be heard continuously even when the self drive circuit is turned ON / OFF at the "X" point shown in Fig.9.

- This is because of the failure of turning off the feedback voltage.
- Build a circuit of the piezoelectric sounder exactly as per the recommended circuit shown in the catalog. Hfe of the transistor and circuit constants are designed to ensure stable oscillation of the piezoelectric sounder.
- Design switching which ensures direct power switching.
- The self drive circuit is already contained in the piezoelectric buzzer. So there is no need to prepare another circuit to drive the piezoelectric buzzer.
- Rated voltage (3.0 to 20Vdc) must be maintained. Products which can operate with voltage higher than 20Vdc are also available.
- Do not place resistors in series with the power source, as this may cause abnormal oscillation. If a resistor is essential to adjust sound pressure, place a capacitor (about 1μF) in parallel with the piezo buzzer.
- Do not close the sound emitting hole on the front side of casing.
- Carefully install the piezo buzzer so that no obstacle is placed within 15mm from the sound release hole on the front side of the casing.

5.2.9 Gas sensor:

Introduction:

The MQ6 is a simple-to-use liquefied petroleum gas (LPG) sensor. It can be used in gas leakage detecting equipment in consumer and industry applications, this sensor is suitable for detecting LPG, iso-butane, propane, LNG. Avoid the noise of alcohol, cooking fumes and cigarette smoke.

The sensitivity can be adjusted by the potentiometer. These gas sensor modules use gas sensors from Hanwei Electronics. When their internal heating elements are activated, these gas sensors respond to their specific gas by reducing their resistance in proportion to the amount of that gas present in the air exposed to the internal element. The MQ6 can detect gas concentrations anywhere from 200 to 10000ppm. This sensor has a high sensitivity and fast response time. The sensor's output is an analog resistance. The drive circuit is very simple; all you need to do is power the heater coil with 5V, add a load resistance, and connect the output to an ADC.

Fig 5.2.16 **MQ-6 GAS Sensor**

LPG Gas sensor MQ06:This is a simple-to-use liquefied petroleum gas (LPG) sensor, suitable for sensing LPG (composed of mostly propane and butane) concentrations in the air. The MQ6 can detect gas concentrations anywhere from 200 to 10000ppm. This sensor has a high sensitivity and fast response time. The sensor's output is an analog resistance. The drive circuit is very simple; all you need to do is power the heater coil with 5V, add a load resistance, and connect the output to an ADC.

Pin Definition:

1. Signal Output
2. GND
3. Power

Connecting and Testing

The gas sensor canister plugs into the socket on the front of the module. The gas sensors are essentially resistive devices and are not polarized, so there is no need to be concerned about plugging it in “backwards.” It will work in either orientation. The 4-pin SIP header on the Gas Sensor Module makes it easy to connect to a breadboard or SIP socket. The four connections are defined in the table below. The sensitivity can be adjusted by the potentiometer. These gas sensor modules use gas sensors from Hanwei Electronics. When their internal heating elements are activated, these gas sensors respond to their specific gas by reducing their resistance in proportion to the amount of that gas present in the air exposed to the internal element.

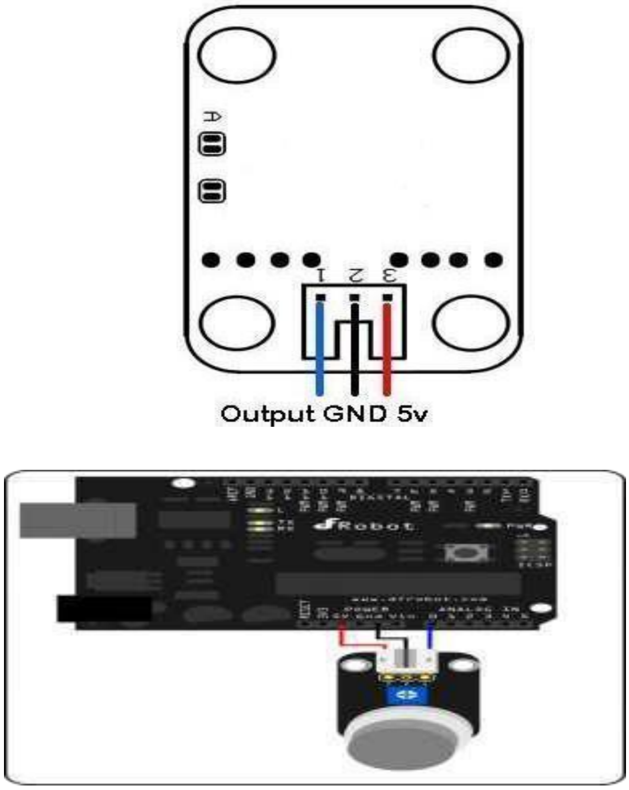


Fig 5.2.17 Connection of Gas

Sensor

Connection to a 5V microcontroller, such as the BASIC Stamp® module, would be pretty straight forward and require two I/O pins; one input for detecting the alarm signal and the other an output for controlling the internal heater. For the Propeller to control the heater switch input (HSW) you would need an NPN switching transistor, such as a 2N3904 and a 1 kΩ resistor. The schematic for this connection is shown below.

These gas sensor modules use gas sensors from Hanwei Electronics. When their internal heating elements are activated, these gas sensors respond to their specific gas by reducing their resistance in proportion to the amount of that gas present in the air exposed to the internal element. The gas sensor canister plugs into the socket on the front of the module. The gas sensors are essentially resistive devices and are not polarized, so there is no need to be concerned about plugging it in "backwards." On the gas sensor modules this is part of a voltage divider formed by the internal element of each gas sensor and potentiometer R3 (Set Point). This change in resistance alters the output voltage of the sensor module, which can then be measured and processed by a microcontroller or an analog-to-digital converter.

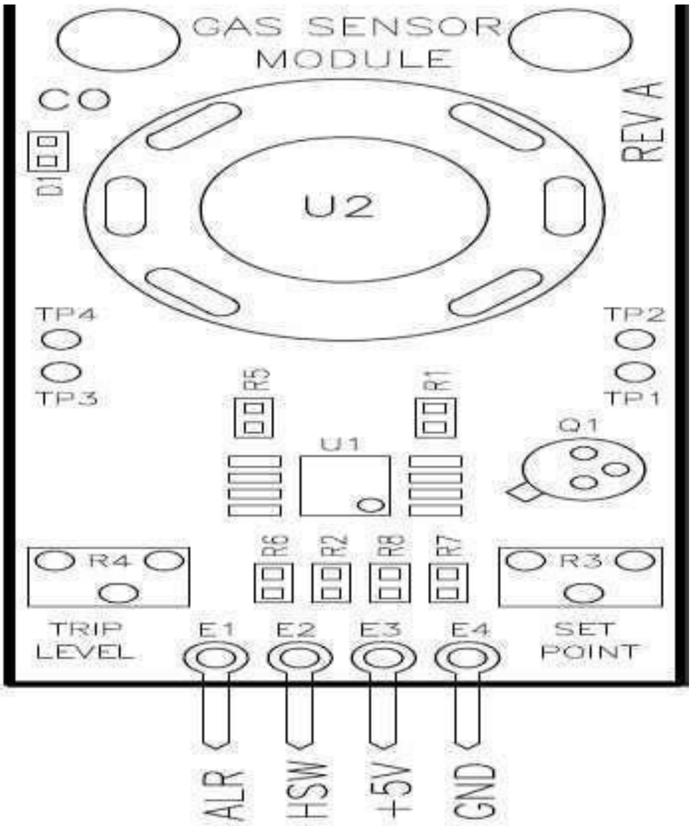
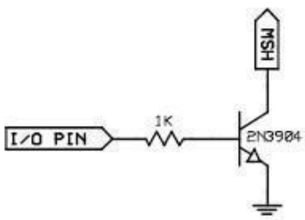


Fig.5.2.18 Layout of Gas Sensor Module

The sensitivity of the sensor can be adjusted using the potentiometer (R3), allowing calibration based on specific detection requirements. As the concentration of the target gas increases, the resistance decreases, leading to a proportional increase in the sensor's output voltage. These modules are commonly used in air quality monitoring, industrial safety systems, and gas leak detection applications.



Switching circuit for the Propeller chip. Note: HIGH signal is required to turn on heater due to inversion of signal by the transistor.

Pin	Name	Function	Level
E1	ALR	Alarm output to host microcontroller	0V / 5V
E2	HSW	Heat switch input from microcontroller, active low	0V / Floating*
E3	+5V	+5 VDC power	5V
E4	GND	Ground, connects to common ground	0V
TP1	Test Point 1+	Buffered output of sensors (voltage divider)	0V – 5V
TP2	Test Point 2-	Ground, connects to common ground	0V
TP3	Test Point 3+	Trip Level voltage set by potentiometer (R4)	0V – 5V
TP4	Test Point 4-	Ground, connects to common ground	0V

Table 5.1Pin Definitions and Ratings:

Note:

• The HSW line is internally pulled up to 5V via a 10 kΩ resistor.

• Sensor may rattle if shaken – this is normal.

Theory of Operation:

The output of this voltage divider is fed into the non-inverting inputs of the two op-amps on the LT1013 dual op-amp IC. Op-amp A is configured as a buffer with unity gain and is used to provide a non-loaded test point for the signal voltage at TP1 (+) and TP2 (-). The signal voltage is also being fed into op-amp B which is configured as a comparator that gets its reference voltage at the inverting input from potentiometer R4 (Trip Level) and is also available at TP3 (+) and TP4 (-). The output of op-amp B goes out to the ALR pin through a 1 kΩ resistor providing a TTL-compatible signal to a microcontroller.

This output also connects to a red LED on the gas sensor modules. The zero gas span adjustment is set via potentiometer R3. As mentioned above R3 allows you to change the span/range of the voltage divider formed by the gas sensor and R3 which is the bottom leg of the divider, electrically speaking. Adjusting R3 to lower values will make the gas sensor less sensitive but more stable. Avoid setting R3 below 200 ohms as at this point you will be close to shorting the output to ground. Setting R3 to higher values will make the gas sensor more responsive, but without a minimum load it will become unstable after a certain point.

The trip level adjustment is set via potentiometer R4. This is just a simple voltage divider that lets you set the voltage from 0V to 5V. This voltage is compared to the voltage coming from the gas sensor/R3 divider. When the voltage from the gas sensor is higher than the voltage set by potentiometer R4 the red LED will light and the ALR output will be high (5 V). The section below describes how to configure these gas sensor modules to detect gas with minimal calibration.

Calibration :

The procedure for setting these potentiometers is explained below. Please note that turning the potentiometer clockwise decreases voltage, while turning the potentiometer counter-clockwise increases it. This can be compared to a water valve.

- For the CO sensor, please read the CO Sensor Specifics on page 5 before attempting calibration.
- Place the Gas Sensor Module in a clean air environment and supply power to the module. The heater should be active during this time. Allow at least 10 minutes before making adjustments.
- Adjust potentiometer R4 until the voltage across TP3 (+) and TP4 (-) reads approximately 0.80 V.
- Adjust potentiometer R3 until the voltage across TP1 (+) and TP2 (-) reads approximately 0.80 V.
- At this point adjusting R3 up/down should make the LED toggle on/off. Adjust R3 so the LED just goes off.
- Apply your gas source to the gas sensor. The LED should light up.
- Remove the gas source and allow the sensor to settle. The LED should go back out.
- If the LED does not go out within 60 seconds, adjust R3 until the LED goes out and repeat the two previous steps.

The gas sensor module’s ALR pin should only be checked when the heater is on and the readings have stabilized.

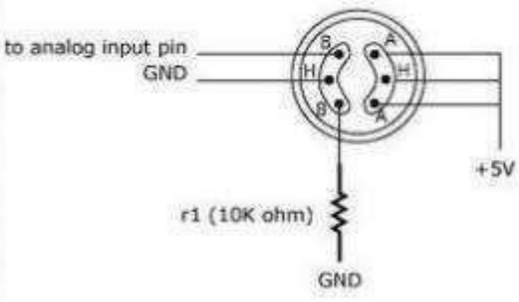
Note: Temperature and humidity are factors that could affect the sensor, making calibration difficult. If this should happen try adjusting the voltages used in the calibration up or down as necessary to find a more stable range.

SENSITVITY ADJUSTMENT:

Resistance value of MQ-6 is difference to various kinds and various concentration gases. So, when using this component, sensitivity adjustment is very necessary. We recommend that you calibrate the detector for1000ppm of LPG concentration in air and use value of Load resistance (RL) about 20KΩ (10KΩ to 47KΩ). When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

Usage:

VCC - 5V GND - GND S - Analog pin0



FEATURES:

- High sensitivity to LPG, ion-butane, propane.
- Small sensitivity to alcohol, smoke.
- Fast response.
- Stable and long life.

• Simple drive circuit.

APPLICATION:

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, iso-butane, propane, LNG, avoid the noise of alcohol and cooking fumes and cigarette smoke. These sensors provide reliable and fast detection, making them essential for preventing hazardous gas leaks in homes, factories, and commercial spaces. They are designed to specifically detect combustible gases like LPG, iso-butane, propane, and LNG while minimizing false alarms caused by alcohol vapors, cooking fumes, and cigarette smoke.

5.2.10 GPS:

The National Marine Electronics Association (NMEA) has developed a specification that defines the interface between various pieces of marine electronic equipment.GPS receiver communication is defined within this specification. Most computer programs that provide real time position information understand and expect data to be in NMEA format. This data includes the complete PVT (position, velocity, time) solution computed by the GPS receiver. The idea of NMEA is to send a line of data called a sentence that is totally self contained and independent from other sentences. There are standard sentences for each device category and there is also the ability to define proprietary sentences for use by the individual company. All of the standard sentences have a two letter prefix that defines the device that uses that sentence type. (For gps receivers the prefix is GP.) which is followed by a three letter sequence that defines the sentence contents. In addition NMEA permits hardware manufactures to define their own proprietary sentences for whatever purpose they see fit.

All proprietary sentences begin with the letter P and are followed with 3 letters that identifies the manufacturer controlling that sentence. For example a Garmin sentence would start with PGRM and Magellan would begin with PMGN.

RMC - NMEA has its own version of essential gps pvt (position, velocity, time) data. All of the standard sentences have a two letter prefix that defines the device that uses that sentence type. (For gps receivers the prefix is GP.) which is followed by a three letter sequence that defines the sentence contents.It is called RMC, The Recommended Minimum, which will look similar to:

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

Where:
RMC                    Recommended  Minimum  sentence  C
123519                 Fix taken at 12:35:19 UTC

A                      Status   A=active   or   V=Void.
4807.038,N             Latitude 48 deg 07.038' N

01131.000,E  Longitude 11 deg 31.000' E

022.4                  Speed over the ground in knots

084.4                  Track angle in degrees True

230394                 Date - 23rd of March 1994

03.1  ,W               Magnetic Variation

*6A                    The checksum data, always begins
with *
```

Neo-6 AGPS Receiver

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer rous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix0(TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly.

The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments. Additionally, the NEO-6 series supports assisted GPS (A-GPS) technology, which enhances positioning speed and accuracy by using data from external sources. With support for both UART and SPI communication interfaces, these GPS modules can seamlessly integrate with a wide range of microcontrollers and embedded systems.

Their robust design ensures stable performance in various environmental conditions, making them ideal for use in automotive navigation, fleet tracking, drones, wearable devices, and other GPS-based applications. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints. . Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

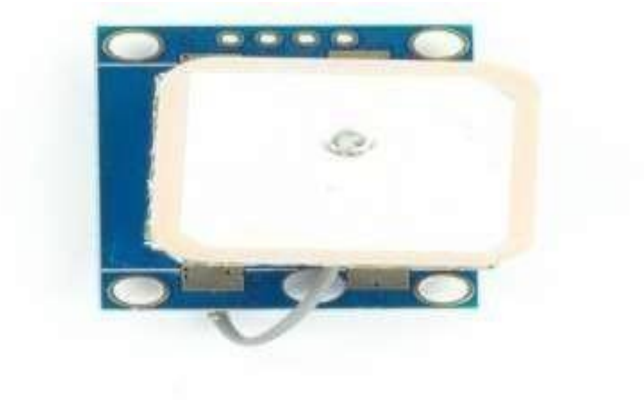


Fig 5.2.19 Neo-6 GPS Receiver

Specifications:

Supply Voltage: 2.7 to 3.6V **Supply current**: 67 mA **Antenna gain**: 50 dB

Operating temperature: -40 to 85°C **Antenna Type:** Passive and active antenna **Interfaces :** UART,USB,SPI,DDC

Sensitivity-

- **Tracking & Navigation:-** 160 dBm
- **Reacquisition:** -160 dBm
- **Cold Start (Autonomous):** -146 dBm

SPECIFICATIONS:

- **Vcc**-Supply Voltage
- **Gnd**-Ground pin
- **TX and RX**-These 2 pins acts as an UART interface for communication

CHAPTER 6
RESULT ANALYSIS AND REPORT

6.1 RESULTS:

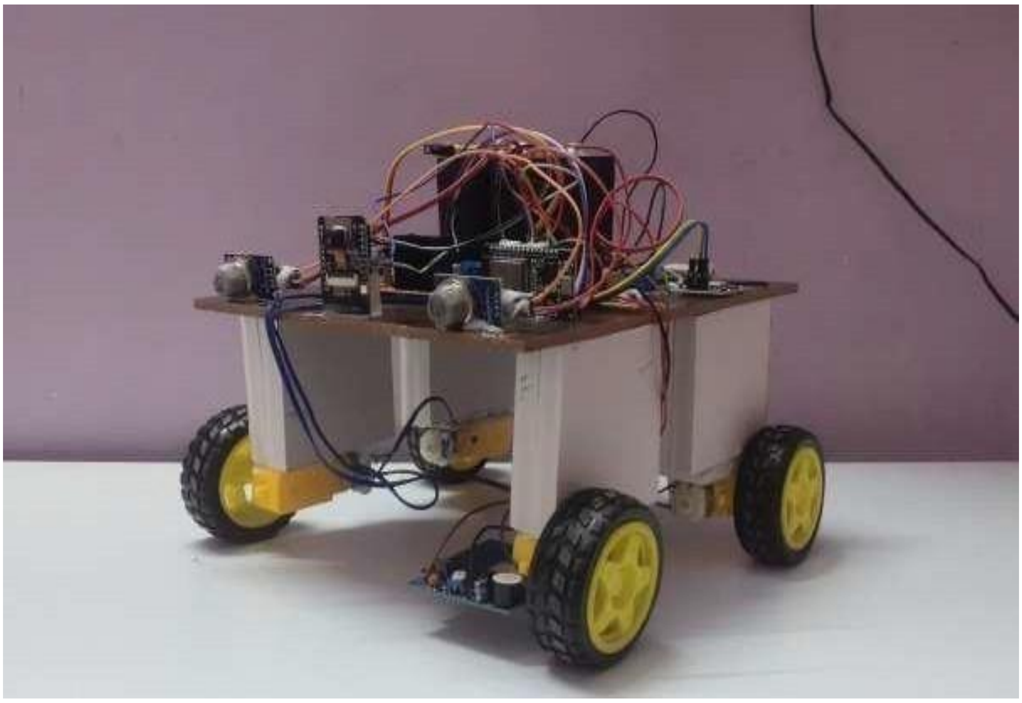


Fig 6.1 Multi Threat Surveillance Robot

The Multi-Threat Surveillance Vehicle is designed to autonomously detect environmental hazards such as gas leaks, smoke, and metal objects, providing real-time alerts to operators. Equipped with an ESP32-CAM module, the vehicle streams live video footage, enabling remote monitoring. Its Arduino Nano microcontroller controls a four-wheel robot platform, ensuring mobility across various terrains.

The vehicle's GPS module updates its location on a web interface, allowing operators to track its position in real-time. Powered by a 12V 2A battery, the system ensures extended operational duration in the field. This integrated approach enhances situational awareness and response capabilities in security and surveillance operations.

Fig 6.2 Interface Of Device For Operating

The above fig shows the interface of operating Vehicle, display with the left, right, front, back bottons. Along with this if the sensors reads the data then the sensor reading changes to 1, else display 0. Also the system shows the location above the sensors reading, so that it will be easy to identify the information by the user also easy to opeate the vehicle.

6.2 ADVANTAGES :

- **Real-Time Monitoring:** The integration of an ESP32-CAM module allows for live video streaming, providing operators with immediate visual information to assess situations accurately.

- **Comprehensive Threat Detection:** Equipped with sensors for gas, smoke, and metal detection, the vehicle can identify a wide range of potential hazards, enabling prompt responses to various threats.

- **Enhanced Mobility:** Built on a four-wheel robot platform, the vehicle can navigate diverse terrains, ensuring accessibility to areas that may be challenging for traditional surveillance methods.
- **Precise Location Tracking:** With GPS integration, the vehicle's position is continuously updated on a web interface, facilitating effective tracking and coordination during operations.
- **Extended Operational Duration:** Powered by a 12V 2A battery, the system is designed for prolonged use, reducing the need for frequent recharging and ensuring reliability during extended missions.
- **Dual Alert System:** Features immediate local alarms (buzzer) plus remote notifications through IoT connectivity, ensuring threats are never missed.
- **Remote Operation Capability:** Wi-Fi enabled web interface allows control from safe distances, ideal for hazardous environments like chemical spills or conflict zones.
- **Autonomous Safety Response:** Automatically stops and alerts upon threat detection, preventing human exposure to dangers.

6.3 APPLICATIONS:

- **Border Surveillance:** Deployed along borders to monitor and detect unauthorized crossings, ensuring national security.
- **Coastal Surveillance:** Patrols coastal areas to detect illegal activities such as smuggling or unauthorized fishing, safeguarding maritime resources.
- **Critical Infrastructure Protection:** Monitors critical infrastructure sites like power plants, water treatment facilities,communication towers to prevent sabotage or unauthorized access.
- **Disaster Response:** Assists in disaster-stricken areas by providing real-time surveillance, aiding in search and rescue operations, and assessing damage.
- **Military Reconnaissance:** Conducts surveillance in conflict zones, gathering intelligence and monitoring enemy movements to support military operations.
- **Law Enforcement Support:** Assists law enforcement agencies in monitoring public events, tracking suspects, and gathering evidence for investigations.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

The Multi-Threat Surveillance Vehicle represents a significant advancement in security and monitoring technologies. By integrating gas, smoke, and metal detection sensors with real-time video streaming, GPS tracking, and autonomous navigation, this system offers a comprehensive solution for various applications, including border surveillance, critical infrastructure protection, and disaster response. Its ability to operate across diverse terrains and provide realtime data enhances situational awareness and enables timely interventions. As unmanned surveillance vehicles continue to evolve, they are expected to play an increasingly vital role in enhancing safety and security across multiple sectors. So by this vehicle we can able to know the happening threat so that we can save of the person life.

The Multi-Threat Surveillance Vehicle represents a groundbreaking advancement in security and monitoring technologies, integrating multiple state-of-the-art features to provide real-time threat detection and response. Designed for a variety of high-risk applications, this vehicle is equipped with gas, smoke, and metal detection sensors, ensuring the rapid identification of hazardous substances and potential threats in its vicinity. Additionally, it features real-time video streaming, GPS tracking, and autonomous navigation, making it an ideal solution for border surveillance, critical infrastructure protection, and disaster response.

One of the most crucial aspects of this surveillance vehicle is its integration of gas detection sensors, which can identify harmful gases such as carbon monoxide, methane, and other toxic substances. This capability is particularly useful in industrial zones, underground tunnels, and areas prone to chemical leaks, where human intervention may be hazardous. Smoke detection technology enhances fire prevention and early warning systems, reducing the risk of widespread damage. Metal detection sensors further contribute to security efforts by identifying concealed weapons, explosives, or other metallic threats, making the vehicle highly effective in high-security zones such as airports, government buildings, and military bases. Real-time video streaming and GPS tracking allow for continuous monitoring of designated areas, enabling operators to assess threats remotely and deploy response teams when necessary.

High-definition cameras equipped with night vision and thermal imaging provide visibility even in low-light or adverse weather conditions, ensuring uninterrupted surveillance. The integration of GPS tracking enables precise location monitoring, making the vehicle an invaluable asset for border patrol units, law enforcement agencies, and security personnel working in remote or high-risk areas. With autonomous navigation, the vehicle can patrol predefined routes without human intervention, reducing manpower requirements while increasing efficiency.

The vehicle’s versatility extends beyond security applications. In disaster response scenarios, it plays a crucial role in search and rescue operations by providing live video feeds of affected areas and detecting hazardous conditions such as gas leaks, structural instability, and fire hazards. Emergency responders can use the data gathered by the vehicle to make informed decisions about evacuation procedures, resource allocation, and medical assistance. Its robust design allows it to operate in environments where traditional vehicles may struggle, such as flood zones, debris-covered landscapes, and earthquake-affected areas.

By gathering intelligence and tracking enemy movements in real-time, it enhances situational awareness and allows defense forces to make strategic decisions with minimal risk to personnel. The integration of advanced AI-driven analytics can further improve threat assessment by identifying suspicious behavior, unauthorized access, and anomalies in monitored areas. In combat zones, the vehicle can assist in surveillance missions, perimeter defense, and remote threat detection, reducing the need for direct human involvement in potentially dangerous situations.

Another key advantage of this vehicle is its adaptability to various terrains and operational conditions. Whether deployed in urban environments, forests, deserts, or mountainous regions, it remains fully functional and efficient. Its rugged design ensures durability, allowing it to withstand extreme temperatures, rough terrains, and harsh weather conditions. The vehicle can be further enhanced with additional modules, such as drone integration for aerial surveillance, robotic arms for object handling, and non-lethal deterrents like sound cannons and flashing lights to disorient intruders.

Looking toward future advancements, the Multi-Threat Surveillance Vehicle is expected to incorporate cutting-edge technologies such as 5G connectivity, artificial intelligence, and cloudbased data storage. 5G integration will enable ultra-low-latency data transmission, allowing for faster communication between operators and the vehicle. AIpowered threat detection systems will enhance the vehicle's ability to identify potential risks autonomously, reducing the need for constant human monitoring. Cloud integration will allow for the storage and analysis of large volumes of surveillance data, facilitating predictive analytics and automated threat response.

Moreover, energy efficiency will be a primary focus in future iterations of the vehicle. The incorporation of solar panels and intelligent power management systems will extend operational time, making the vehicle more sustainable and reducing its reliance on conventional fuel sources. This advancement is particularly beneficial for long-term surveillance missions and remote deployments where frequent refueling or battery replacement may not be feasible.

As unmanned surveillance vehicles continue to evolve, their role in enhancing safety and security across multiple sectors will become even more significant. These vehicles offer a proactive approach to threat detection and response, ensuring the safety of civilians, law enforcement officers, and military personnel alike. By continuously monitoring their surroundings and providing real-time data, they enable quick and effective interventions that can prevent potential disasters, protect valuable assets, and save lives.

In conclusion, the Multi-Threat Surveillance Vehicle is a game-changer in modern security and monitoring operations. Its advanced sensors, autonomous capabilities, and realtime data transmission make it an essential tool for border security, disaster response, military reconnaissance, and critical infrastructure protection. With continuous technological advancements, it is set to become an indispensable asset in the fight against emerging threats, ensuring a safer and more secure future for societies worldwide.

7.2 FUTURE SCOPE:

Enhanced Autonomy with AI:

- Integration of AI-powered image recognition for detecting humans, vehicles, or potential threats in real-time.
- Implementation of machine learning algorithms for pattern recognition in surveillance data.

Advanced Sensor Integration:

- Addition of infrared and thermal cameras for night vision capabilities.
- Inclusion of LiDAR or radar sensors for improved obstacle detection and terrain mapping.

Swarm Robotics for Coordinated Surveillance:

- Deployment of multiple autonomous units working together to cover larger areas.
- Communication between multiple units for collaborative threat detection and response.

5G and IoT Connectivity:

- Incorporation of 5G modules for ultra-low-latency data transmission and cloud connectivity.
- IoT-based data analytics for predictive maintenance and real-time status monitoring.

Improved Power Efficiency:

- Use of solar panels or energy harvesting technologies to extend operational time.
- Implementation of smart power management systems for optimized energy usage.

Remote Weapon System Integration (Military Applications):

- Potential integration with non-lethal deterrents such as sound cannons or flashing lights for security applications.
- Controlled deployment of robotic arms for handling hazardous materials.

Multi-Platform Compatibility:

- Expansion of the system for marine and aerial surveillance applications.
- Compatibility with drones for extended aerial reconnaissance and tracking.

Autonomous Decision-Making:

- Development of self-learning algorithms for threat assessment and risk prioritization.

Edge AI for Real-Time Threat Classification:

- On-device deep learning models (e.g., TinyML) for instant weapon detection, facial recognition, or behavioral analysis without cloud dependency.

REFERENCES

[1] Muhammad Hamza, M Atique-ur-Rehman, Hamza Shafqat, Subhan Bin Khalid, “CGI SCRIPT AND MJPG VIDEO STREAMER BASED SURVEILLANCE ROBOT USING RASPBERRY PI” in 16th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 8th – 12th January, 2019, IEEE 2019.

[2] Juan G. Parada-Salado, Luis E. Ortega-García, Luis F. Ayala-Ramírez, Francisco J. Pérez Pinal, “A Low-Cost Land Wheeled Autonomous MiniRobot For In-Door Surveillance” In Ieee Latin America Transactions, Vol. 16, No. 5, May 2018.

[3] S M Ashish, Madhurya Manjunath, Ravindra L, Mohammed Nadeem, Neelaja K, “Automated Hybrid Surveillance Robot” In International Journal Of Innovations In Engineering Research And Technology [Ijiert], Issn: 2394-3696; Volume 5, Issue 5, May-2018.

[4] T. Saravanakumar, D. Keerthana, D. Santhiya, J. Sneka, D. Sowmiya, “Surveillance Robot Using Raspberry Pi-IoT” in International Journal of Electronics, Electrical and Computational System(IJEECS);ISSN 2348-117X; Volume 7, Issue 3, March 2018.

[5] Harshitha R, Muhammad Hameem Safwat Hussain, “SURVEILLANCE ROBOT USING RASPBERRY PI AND IOT” in 2018 IEEE - DOI 10.1109/ICDI3C.2018.00018; International Conference on Design Innovations for 3Cs Compute Communicate Control.

[6] Gaurav S Bagul, Vikram C Udawant, Kalpana V Kapade and Jayesh M Zope, “IOT BASED SURVEILLANCE ROBOT” in International Journal of Trend in Research and Development, Mar -Apr 2018; Volume 5(2), ISSN: 2394-9333.

[7] Rutuja. R. Bachhav, Priyanka. S. Bhavsar, Devyani. K. Dambhare, “Spy Robot For Video Surveillance And Metal Detection” In International Journal For Research In Applied Science & Engineering Technology (Ijraset); Issn: 2321-9653; Volume 6 Issue V, May 2018.

[8] Neha Joisher, Sonaljain, Tejas Malviya, Vaishali Gaikwad (Mohite), “Surveillance System Using Robotic Car” In International Conference On Control,

Automation.systems,

[9] Fatma Salih, Mysoon S.A. Omer, “Raspberry Pi as A Video Server” In the International Conference on Computer, Control, Electrical, and Electronics Engineering (Icccee) 2018.

[10] Sweeta Deshmukh, Priyadarshini, Mamta, Madhura Deshmukh, Dr.Md. Bakhar, “IoT Based Surveillance Robot” In International Journal Of Innovative Research In Computer And Communication Engineering; Vol.5, Special Issue 4, June 2017

[11] Ghanem Osman Elhaj Abdalla, T. Veeramanikandasamy, “Implementation Of Spy Robot For A Surveillance System Using Internet Protocol Of Raspberry Pi” In 2nd Ieee International Conf On Recent Trends In Electronics Information & Communication Technology, May 19-20, 2017.

[12] Souvik Saha, Arko Singh, Palash Bera, Murshed Nawaz Kamal, Soumya Dutta, Uttam Gorian, Sayak Pramanik, Angshuman Khan, Surajit Sur, “Gps Based Smart Spy Surveillance Robotic System Using Raspberry Pi For Security Application And Remote Sensing” In Ieee , 978-1-5386- 3371-7/17,2017.

APPENDIX

Appendix-1: Gather Components

Before beginning the project, ensure you have all necessary components:

1. ESP32 Microcontroller
2. ESP32 Camera
3. GPS Module
4. Metal sensor: For detecting landmine or metal related threats.
5. Smoke and gas Sensors: For detecting harmful gases and smoke.
6. Power Supply: (5VDC or backup battery for the system).
7. Jumper Wires, Breadboard, Resistors: For prototyping.
8. Smartphone/Device: For monitoring alerts and notifications.

Appendix-2: Circuit Design & Wiring

2.1: ESP32 Pin Connections

1. GPS Module: Connect to the ESP32 using UART (TX/RX).
2. GSM Module: Connect to the ESP32 for serial communication.
3. Night Vision Camera: Connect via I2C/SPI, depending on the camera model
4. Motion Sensors: Connect digital pin(s) of the ESP32 to the motion/vibration sensor.
5. Power Supply: Use the vehicle's 12.2 Amps battery, and ensure the ESP32 & other modules get the appropriate voltage .

2.2: Power Distribution

- Ensure the system gets stable power from the vehicle's 12.2Amps battery.

Appendix-3: Setting Up the ESP32 Environment.

Appendix-4: Writing the Firmware for ESP32

Appendix-5: Testing the System

Appendix-6: Install the System in Vehicle

Appendix-7: Final Testing and Optimization in Mobile app

Appendix-8: Monitor and Maintain

CODE:

```
#include <TinyGPS++.h>  // For
GPS TinyGPSPlus gps;  // GPS
object char c; float l,m; #define
gas 39  #define metal 36
#define smoke 18
#define buz 23
 #define m1a 25//robot motors
#define m1b
26 #define
m2a 27
#define m2b
14 int
s1,s2,s3,s4;
void forward
()
{
  //Serial.println("Moving forward");
digitalWrite(m1a, HIGH);
digitalWrite(m1b, LOW);
digitalWrite(m2a, HIGH);
digitalWrite(m2b, LOW);   delay(5);
} void backward ()
{
  //Serial.println("Moving backward");
digitalWrite(m1a, LOW);
digitalWrite(m1b, HIGH);
digitalWrite(m2a, LOW);
digitalWrite(m2b, HIGH);   delay(5); }
void left ()
{
```

```
  //Serial.println("Moving left");
digitalWrite(m1a, HIGH);
digitalWrite(m1b, LOW);
digitalWrite(m2a, LOW);
digitalWrite(m2b, HIGH);
delay(5); } void right ()
{
  //Serial.println("Moving        right");
digitalWrite(m1a, LOW);//right side
digitalWrite(m1b,       HIGH);
digitalWrite(m2a, HIGH);//left side
digitalWrite(m2b, LOW);  delay(5);
} void stop_motor () {
//Serial.println("stop");
digitalWrite(m1a, LOW);
digitalWrite(m1b, LOW);
digitalWrite(m2a, LOW);
digitalWrite(m2b, LOW);
delay(5); } void setup() {
  // put your setup code here, to run once:
Serial.begin(9600); // Initialize Serial
Serial2.begin(9600); // Initialize Serial
pinMode(gas, INPUT);
pinMode(smoke,INPUT);
pinMode(buz,OUTPUT);
digitalWrite(m1a, LOW);
digitalWrite(m1b, LOW);
digitalWrite(m2a, LOW);
digitalWrite(m2b, LOW);
digitalWrite(buz,1);  delay(700);
digitalWrite(buz,0);  delay(700);
digitalWrite(buz,1);  delay(700);
digitalWrite(buz,0);
```

```
}
        void

loop()
{

 // put your main code here, to run repeatedly:  int s = digitalRead(smoke);   int m =
analogRead(metal);  int g = digitalRead(gas);  // Get GPS coordinates and display   while
(Serial2.available())
 {    char c =
Serial2.read();
//Serial.write(c);    if
(gps.encode(c))
   {
     displayInfo();
   }   }
if(Serial.available()>0
)  {    c =
Serial.read();
if(c=='f')    {
forward();    }    else
if(c=='b')
   {


backward();
}    else
if(c=='l')    {
left();    }
else
if(c=='r')    {
right();    }
else
if(c=='s')
   {
```

```
stop_motor();
    }
  }

if(g == 0)
{     s2=1;
  }
else
{
s2=0;
}
if(s=
=0)
{
s3=1;
}
else
{
s3=0;
  }
  if(m>3000)
  {
s4=1;
}   else
{
s4=0
;   }

  if((s2 == 1) || (s3 == 1)|| (s4 == 1))
  {

digitalWrite(buz,1);
}   else
```

```
 {
   digitalWrite(buz,0);
 }
  Serial.print(gps.location.lat(), 6);   Serial.print(",");
 Serial.print(gps.location.lng(), 6);
 Serial.print(",");
 Serial.print(s2);
Serial.print(",");
Serial.print(s3);
Serial.print(",");
Serial.println(s4);
delay(500); }   void displayInfo() {
if
(gps.location.isValid())
 {    l=gps.location.lat(),
6;    m=gps.location.lng(),
6;


}
}
```