# CHAPTER 1

# INTRODUCTION

The Smart Cattle System represents a significant advancement in livestock management, offering comprehensive monitoring and protection capabilities. With fire detection sensors, the system can pre-emptively identify and notify farmers of potential fire hazards in grazing areas or barns, enabling rapid response measures to safeguard both cattle and farm infrastructure. Moreover, water level sensors ensure continuous access to adequate hydration for cattle by monitoring water levels in troughs or tanks and alerting farmers of any deficiencies, thereby preventing dehydration related health issues. Additionally, GPS tracking technology provides real time insights into cattle location and movement patterns, empowering farmers to monitor their livestock's whereabouts and detect any deviations from expected behaviour. This enables farmers to take proactive measures to address issues such as cattle straying or theft, enhancing overall security and asset protection. Through the seamless integration of these technologies and wireless communication protocols, the Smart Cattle System delivers actionable data and insights to farmers, facilitating informed decision making and optimizing operational efficiency. By prioritizing animal welfare, safety, and productivity, the Smart Cattle System exemplifies the transformative potential of IoT technology in modern agriculture, driving sustainable practices and ensuring the long term viability of livestock operations. This system will not only locate the exact depth and GPS location of miner but it will continuously be updating the pulse rate of miner. This system in this way help the rescue team to dig at right places with exact depth to take out miners back to ground.

# PROBLEM STATEMENT

The Smart Cattle System aims to address several critical challenges faced by livestock farmers, including the risk of fire outbreaks, inadequate access to water, and difficulties in monitoring cattle location and security. Currently, farmers lack effective tools for early detection of fires in grazing areas or barns, leading to potential loss of livestock and property damage. Additionally, there is a pressing need for improved water management solutions to ensure consistent access to clean water for cattle, as well as enhanced methods for monitoring and securing cattle against theft or straying. The absence of comprehensive monitoring and protection mechanisms contributes to increased operational risks, reduced productivity, and compromised animal welfare. Therefore, there is a clear need for an integrated solution that leverages advanced technologies such as fire detection sensors, water level sensors, and GPS tracking to mitigate these challenges and enhance the overall management and safety of livestock operations.

# OBJECTIVES

* Enhance safety and security of livestock through advanced technologies.
* Improve monitoring of cattle health and well being.
* Optimize resource management, particularly water utilization.
* Enhance tracking and management capabilities for livestock.
* Provide user customization and communication features.
* Ensure efficient utilization of resources and promote animal welfare.
* Minimize risks associated with fire outbreaks, theft, and straying.
* Enable real time monitoring of cattle location and movement patterns.
* Facilitate prompt identification and addressing of health issues.

**CHAPTER 2**

# LITERATURE SURVEY

## LITERATURE REVIEW

# OUTCOME OF LITERATURE SURVEY

.

**CHAPTER 3**

**PROPOSED METHODOLOGY**

# METHODOLOGY

**1. Identification of Monitoring Parameters:**

Identify key parameters critical for monitoring cattle health, safety, and location, including fire detection, water availability, and GPS tracking.

**2. Sensor Selection and Integration:**

Select appropriate sensors for each parameter, such as flame sensors for fire detection, water level sensors for monitoring water availability, and GPS modules for tracking cattle location. Integrate selected sensors into a wearable device, such as a smart collar or tag, ensuring proper placement and connectivity.

**3. Microcontroller Setup and Communication:**

Utilize microcontrollers, such as Arduino or ESP8266, to manage sensor data collection and processing within the wearable device.

Configure microcontrollers to communicate with sensors and extract relevant data.

**4. Fire Detection Module:**

* Install flame sensors within the wearable device to detect fire outbreaks in the surrounding environment.
* Program microcontrollers to process sensor data and trigger alerts in case of fire detection.
* Implement wireless communication protocols, such as Bluetooth or Wi Fi, to transmit fire alerts to the user's mobile device.

**5. Water Level Monitoring Module:**

Incorporate water level sensors into the wearable device to monitor water availability in watering troughs or tanks.

* Configure microcontrollers to retrieve data from water level sensors and assess water levels.
* Set threshold values to trigger alerts in case of low water levels, enabling timely intervention to prevent dehydration in cattle.
* Establish wireless communication for transmitting water level alerts to the user's mobile device.

**6. GPS Tracking Module:**

Integrate GPS modules into the wearable device to enable real time tracking of cattle location.

* Program microcontrollers to retrieve GPS data and communicate location information to the user's mobile device.
* Implement features for tracking cattle movement and identifying potential straying or loss incidents.
* Ensure seamless wireless communication for transmitting GPS data and location updates to the user's mobile device.

# CHAPTER 4

# HARDWARE AND SOFTWARE REQUIREMENTS

## HARDWARE REQUIREMENTS

“IOT based smart Cattle System” is a project that requires both hardware and software so let’s see the hardware that this project needs The following hardware components are the requirements for making of the device. Each component and their description are given below:

|  |  |  |  |
| --- | --- | --- | --- |
| **SL.NO** | **COMPONENT NAME** | **SPECIFICATION** | **NO. OF**  **COMPONENT** |
| 01 | Arduino Board | At mega 2560 | 1 |
| 02 | Node MCU | ESP32 | 1 |
| 03 | GPS | Neo 6 | 1 |
| 04 | Fire Sensor | - | 1 |
| 05 | Water Level Sensor | - | 1 |
| 06 | Buzzer | - | 2 |
| 07 | Jumpers | M-F, F-F | 10 |

Fig.4.1

# ARDUINO BOARD

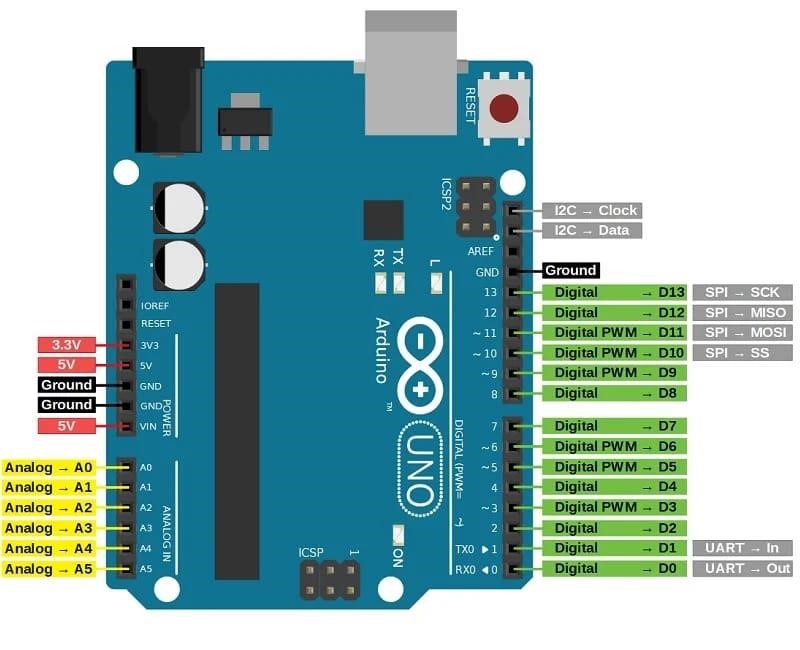


Fig 4.1.1

Arduino is a open source hardware and software platform used to design and build electronic devices. The Arduino board consists of sets of analog and digital I/O (Input / Output) pins, which are further interfaced to breadboard, expansion boards, and other circuits. Such boards feature the model, Universal Serial Bus (USB), and serial communication interfaces, which are used for loading programs from the computers.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in circuit system programming (ICSP) header, and a reset button.

It also provides an IDE (Integrated Development Environment) project, which is based on the Processing Language to upload the code to the physical board. The projects are authorized under the GPL and LGPL. The GPL is named as GNU (General Public License). The licensed LGPL is named as GNU (Lesser General Public License). It allows the use of Arduino boards, it's software distribution, and can be manufactured by anyone.

## General Pin Functions

* **LED**: There is a built in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
* **VIN**: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 20V), the USB connector (5V), or the VIN pin of the board (7 20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3.3V**: A 3.3 volt supply generated by the on board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields that block the one on the board

## Special pin functions

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pin Mode(), digital Write(), and digital Read() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull up resistor (disconnected by default) of 20 50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analog Reference() function.

**Serial / UART**: Pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.

* **External interrupts:** pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* **PWM (pulse-width modulation):** pins 3, 5, 6, 9, 10, and 11. Can provide 8-bitPWM output with the analog Write () function.
* **SPI (Serial Peripheral Interface):** pins 10 (SS), 11 (MOSI), 12 (MISO), and 13(SCK). These pins support SPI communication using the SPI library**.**
* **TWI (two-wire interface) / I²C:** pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
* **AREF (analog reference):** Reference voltage for the analog inputs. AREF(analog reference): Reference voltage for the analog inputs.

## Specifications of Arduino

* The operating voltage is 5V.
* The recommended input voltage will range from 7v to 12V.
* The input voltage ranges from 6v to 20V.
* Digital input/output pins are 14.
* Analog i/p pins are 6.
* DC Current for each input/output pin is 40 mA.
* DC Current for 3.3V Pin is 50 mA.
* Flash Memory is 32 KB.

## BUZZER

Buzzer is a mechanical, electromechanical or piezoelectric audio 27ignalling device. A piezo electric buzzer can be driven by an oscillating electronic circuit or other audio signal source. A piezoelectric diaphragm consists of a piezoelectric ceramic plate which has electrodes on both sides and a metal plate. 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.

To interface a buzzer the standard transistor interfacing circuit is used. If adifferent power supply is used for the buzzer, the 0V rails of each power supply mustbe connected to provide a common reference.



Fig. 4.1.2

## NODE MCU ESP32

The NodeMCU ESP32 is a versatile and powerful development board based on the ESP32 microcontroller. While it shares many features with the ESP32 CAM module, such as Wi-Fi and Bluetooth capabilities, it does not come with an integrated camera like the OV2640. However, it can be used for a wide range of IoT applications including wireless sensor networks, home automation, and IoT projects requiring connectivity and processing capabilities. Below is an overview of the NodeMCU ESP32 specifications:

* Microcontroller: ESP32-D0WDQ6 V3
* Processor: Dual-core Tensilica LX6 microprocessor
* Built-in Flash: 4MB
* RAM: Internal 520KB + External PSRAM support
* Antenna: Onboard PCB antenna
* WiFi Protocol: IEEE 802.11 b/g/n
* Bluetooth: Bluetooth 4.2 BR/EDR and BLE
* Wi-Fi Mode: Station / SoftAP / SoftAP+Station
* Security: WPA/WPA2/WPA2 Enterprise/WPS
* Peripheral Interface: UART/SPI/I2C/PWM
* Digital I/O Pins: 36
* Analog Input Pins: 18 (ADC)
* UART Baudrate Rate: Default 115200bps
* Power Supply: 5V via micro USB or Vin pin
* Transmitting Power:
* 802.11b: 17 ±2dBm (@11Mbps)
* 802.11g: 14 ±2dBm (@54Mbps)
* 802.11n: 13 ±2dBm (@HT20, MCS7)
* Receiving Sensitivity:
* CCK, 1Mbps: -90 dBm
* CCK, 11Mbps: -85 dBm
* 6Mbps (1/2 BPSK): -88 dBm
* 54Mbps (3/4 64 QAM): -70 dBm
* HT20, MCS7 (65Mbps, 72.2Mbps): -67 dBm
* Power Consumption:
* Active Mode: 160-260mA
* Deep Sleep: < 10μA
* Light Sleep: < 20mA



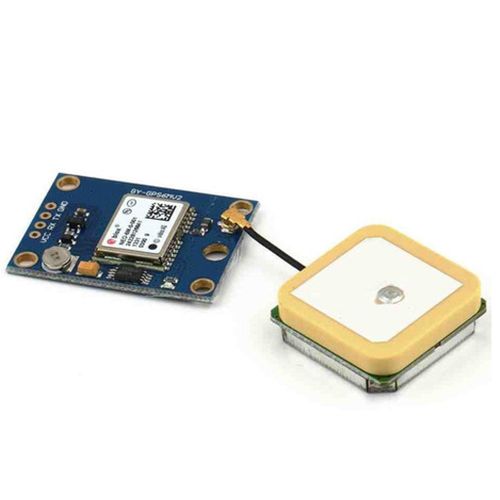
Fig 4.1.3

## GPS Module

The NEO-6 GPS module is a compact and reliable GPS receiver module commonly used in various navigation and tracking applications. The NEO-6 GPS module provides accurate positioning information with low power consumption, making it suitable for battery-operated and portable applications. It communicates via UART interface and outputs standard NMEA sentences, allowing easy integration with microcontrollers like Arduino or ESP32 for location-based projects. With its fast time to first fix and reliable performance, the NEO-6 GPS module is widely used in IoT devices, vehicle tracking systems, drones, and other applications requiring precise positioning and navigation capabilities.

Here's an overview of the specifications of the NEO-6 GPS module:

* Receiver Type: u-blox NEO-6M GPS receiver
* Positioning Accuracy:
* Horizontal: < 2.5 meters (autonomous)
* SBAS (WAAS, EGNOS, MSAS): < 2.0 meters
* Time to First Fix (TTFF):
* Cold Start: < 27 seconds
* Warm Start: < 27 seconds
* Hot Start: < 1 second
* Sensitivity:
* Tracking & Navigation: -161 dBm
* Reacquisition: -160 dBm
* Cold Start: -147 dBm
* Update Rate: Up to 5Hz
* Protocol: NMEA 0183
* Communication Interface: UART (TTL)
* Baud Rate: Default 9600 bps (configurable)
* Operating Voltage: 3.3V to 5.0V
* Power Consumption:
* Acquisition: 45 mA
* Tracking: 25 mA
* Operating Temperature: -40°C to +85°C
* Dimensions: Approximately 25mm x 25mm x 10mm (excluding connector)

Fig.4.1.4

## FLAME SENSOR

This tiny Flame sensor infrared receiver module ignition source detection module is Arduino compatible can use to detect flame or wavelength of the light source within 760nm~1100nm also useful for Lighter flame detect at the distance 80cm.

Greater the flame, farther the test distance. It has the Detect angle of 60 and very sensitive to flame spectrum.

It produces the one channel output signal at the D0 terminal for further processing like an alarm system or any switching system. The sensitivity is adjustable with the help of blue potentiometer given on the board.

**Features :**

* Indicator light: a green one for the switch, a red one for power.
* Built in a potentiometer for sensitivity control.
* Onboard signal output indication, output effective signal is high, at the same time the indicator light up, the output signal can directly connect to microcontroller IO.
* Can detect fire or wavelength in 760 ~ 1100 nm nano within the scope of the light source.
* Detection angle about 60 degrees, the flame spectrum especially sensitive.
* The flame of the most sensitive sensors flame, the regular light is also a response, generally used for fire alarm purposes.

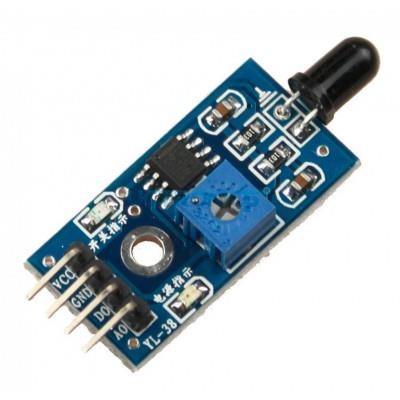


Fig 4.1.5

**Water Level Sensor:**

The water level sensor is a versatile and reliable component designed to detect the level of water in various applications. It provides accurate measurements to ensure efficient water management and monitoring in systems such as tanks, reservoirs, and irrigation systems.

Key Features:

* Precise detection of water levels for effective monitoring and control.
* Compatibility with different types of liquids, including water, chemicals, and fuels.
* Simple installation and integration into existing systems.
* Durable construction for long-term reliability in diverse environments.
* Wide operating temperature range to accommodate various applications.
* Low power consumption for energy-efficient operation.
* Analog or digital output options for seamless integration with monitoring systems.

**Specifications:**

* Detection Method: Capacitive, resistive, ultrasonic, or optical sensing technology.
* Operating Voltage: Typically ranging from 3.3V to 5V DC.
* Output Signal: Analog voltage, digital signal (e.g., HIGH/LOW), or serial communication (e.g., UART, I2C).
* Detection Range: Varies depending on the sensor type and application, ranging from a few millimetres to several meters.
* Accuracy: High accuracy in measuring water levels, typically within a specified tolerance range.
* Response Time: Fast response time for real-time monitoring and control.
* Operating Temperature: Wide temperature range to ensure reliable performance in different environmental conditions.
* Material: Corrosion-resistant materials such as stainless steel, PVC, or polypropylene for longevity and durability.
* Protection Rating: IP (Ingress Protection) rating indicating resistance to dust and water ingress.
* Mounting Options: Various mounting options available, including threaded, flanged, or adhesive mounting.
* Cable Length: Standard cable length for easy connection to monitoring systems.
* Certifications: Compliance with relevant standards and certifications for quality assurance and safety.

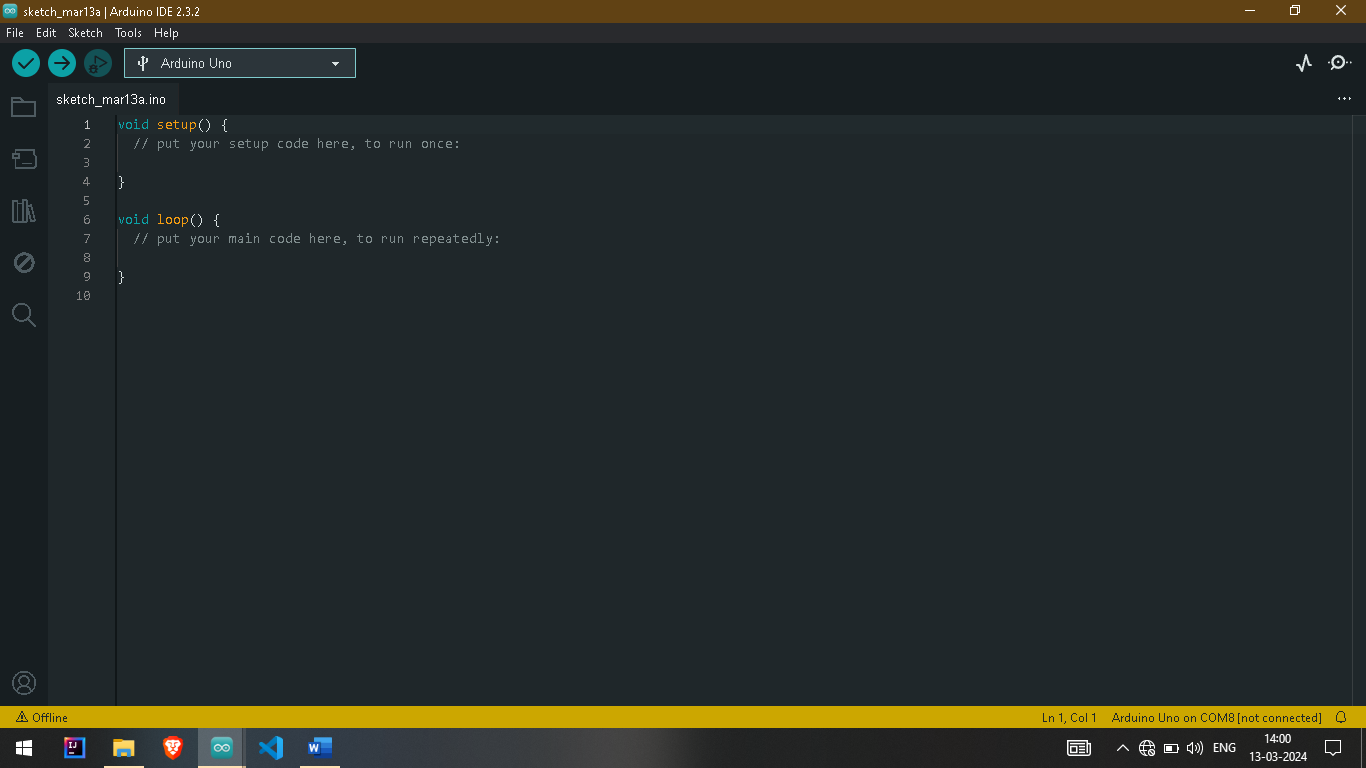


Fig.4.1.6

SOFTWARE REQUIREMENTS

## ARDUINO IDE

Arduino is a type of computer software and hardware company that offers open source environment for user project and user community that intends and fabricates microcontroller based inventions for construction digital devices and interactive objects that can sense and manage the physical world. For programming the microcontrollers, the Arduino proposal provides an software application or IDE based on the Processing project, which includes C, C++ and Java programming software. It also support for embedded C, C++ and Java programming software.



### Fig 4.1.7

Arduino is an open source computer hardware and software company, project and user community that designs and manufactures microcontroller based kits for building digital devices and interactive objects that can sense and control the physical world. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino platform provides an integrated development environment (IDE) based on the Processing project, which includes support for C, C++ and Java programming languages. An Arduino board consists of an Atmel 8, 16 or 32 bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits.

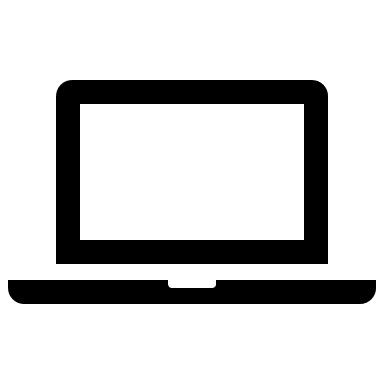
An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add on modules known as shields . Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus so many shields can be stacked and used in parallel. Official Arduinos have used the mega AVR series of chips, specifically the ATmega8 , ATmega168.

**CHAPTER 5**

## IMPLEMENTATION

The Smart Cattle System integrates various sensors and connectivity modules to ensure the well-being and safety of livestock. Powered by Arduino and utilizing components like Peltier modules, GPS, GSM, and sensors, this system provides real-time monitoring and immediate response capabilities. At the core of this system is the Arduino board, which is powered by a battery and recharged by the Peltier module. This setup enables continuous operation even in remote locations. The Arduino communicates with a range of sensors embedded within the smart cattle jackets. These sensors include internal and external sensors designed to monitor vital parameters such as temperature, humidity, and water levels. Using Bluetooth connectivity, the sensor data is transmitted to a gateway device, such as a user's mobile phone, for further processing. From there, the data is relayed to central servers for monitoring and analysis. The use of GPS allows for precise tracking of the cattle's location, ensuring their safety and security.

In the event of a fire, flame sensors integrated into the jackets can quickly detect the presence of flames, triggering an alert to the central servers. Similarly, smoke sensors monitor air quality, ensuring that the environment is safe for the cattle to breathe. Vital signs of the cattle, such as heart rate, are monitored using BPM sensors. This information provides insights into the health and well-being of the animals, allowing for timely intervention if any abnormalities are detected. For water level monitoring, Arduino boards are connected to sensors that detect water levels in troughs or other watering sources. This information helps ensure that the cattle always have access to an adequate water supply. The integration of GPS with NodeMCU allows for precise location tracking of the cattle, enhancing security, and enabling efficient management of grazing areas.



ARDUINO UNO

NODE MCU

ESP32

FIRE SENSOR

WATER LEVEL

GPS [NEO 6]

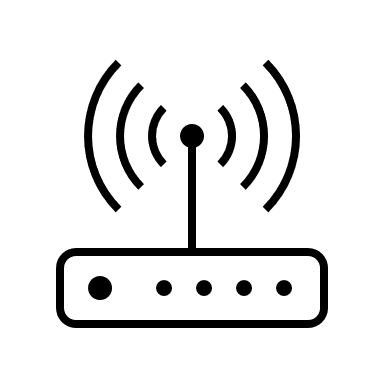


Fig 5.1

**CHAPTER-6**

# RESULTS

We have conducted some experiments to know the quality of performance of all the modules. It has been tested on more than one way. To evaluate the performance of the system, several trials were performed on the project. We have evaluated different functions that are to be performed by the project.



Fig 6.1

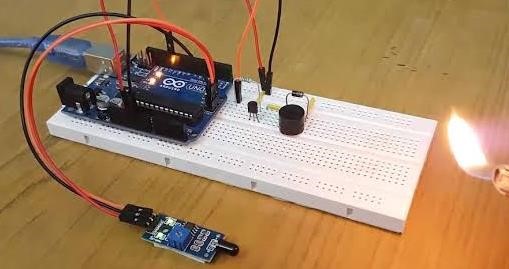


Fig 6.2

**CHAPTER-7**

## CONCLUSION AND FUTURE WORK

In conclusion, the integration of smart technologies such as fire and water level monitoring systems, along with GPS trackers, presents a transformative solution for cattle management. By leveraging these advancements, farmers can enhance their operational efficiency, ensure the well-being of their livestock, and mitigate potential risks effectively.

The utilization of fire monitoring systems allows for proactive detection and prevention of wildfires, safeguarding both the cattle and the surrounding environment from devastation. Simultaneously, water level monitoring ensures adequate hydration for the cattle, promoting their health and productivity. Furthermore, GPS trackers offer real-time location tracking, enabling farmers to monitor their livestock's movements and promptly address any deviations or emergencies.

In the future, enhancing the smart cattle system with automated fire suppression mechanisms and emergency response capabilities could revolutionize livestock management practices. By integrating actuators into the fire monitoring system, such as sprinkler systems or fire retardant dispensers, the system could autonomously respond to fire outbreaks, mitigating potential damage to both cattle and surrounding vegetation. Additionally, incorporating predictive analytics and machine learning algorithms could enable the system to anticipate fire risks based on historical data and environmental factors, allowing farmers to take proactive measures to minimize the threat of wildfires. Moreover, integrating the system with weather monitoring stations would provide real-time updates on weather conditions, facilitating pre-emptive actions to reduce fire risks. Furthermore, enabling remote monitoring and control would empower farmers to manage the system effectively from anywhere, ensuring rapid response to emergencies even in their absence. By implementing these advancements, the smart cattle system could significantly enhance safety, efficiency, and resilience in livestock management, offering substantial benefits to farmers and the broader agricultural community.

## References