CHAPTER 1

INTRODUCTION

In many cases, there is a requirement for mobile platforms that can move in areas with difficult landscape conditions where wheeled vehicles can't travel. Samples of such situations can be found in search and salvage task, and in addition in conveying payloads. Not at all like wheeled robots, walking robot are described by great portability in unpleasant territory. The primary objective of this paper is to show an inventive, modular and reasonable design of a four-legged robot for environmental research purpose.

The objective is to create a cheap legged platform, which allows research and testing of walking chassis and monitoring environmental conditions. The robot should either be driven from the base station or remote location that should send all available data from sensors, which will be displayed on the computer in the user interface program. It is also important to create and program a system into the microcontroller unit (MCU) of the robot, which would have the capacity to control the servomotors and sensors.

The increasing need for effective environmental monitoring has led to the development of innovative robotic solutions. One such solution is the real-time design and implementation of a walking quadruped robot, capable of navigating rough and uneven terrains where traditional wheeled or tracked robots may struggle. Inspired by four-legged animals, quadruped robots offer enhanced mobility, stability, and adaptability, making them highly suitable for outdoor environmental monitoring applications.

This project focuses on the design, development, and deployment of a quadruped robot equipped with environmental sensors such as temperature, humidity, gas detectors, and cameras. The robot is designed to operate autonomously or semi-autonomously, using sensors and embedded systems to gather real-time data from remote or hazardous environments such as forests, industrial zones, or disaster-struck areas.

Key aspects of the project include:

- Mechanical design using lightweight and durable materials for efficient locomotion.
- Embedded control systems like Arduino or Raspberry Pi for real-time processing and control.
- Sensor integration for environmental data collection and obstacle detection.
- Wireless communication (e.g., Wi-Fi, Bluetooth, or LoRa) for transmitting collected data to a central monitoring station.

The implementation of such a robot not only improves the efficiency and accuracy of environmental monitoring but also reduces the risk to human life by operating in dangerous or inaccessible locations. This project represents a convergence of robotics, IoT, real-time systems, and environmental science, paving the way for smarter and safer monitoring technologies.

With the growing concerns surrounding climate change, pollution, natural disasters, and biodiversity loss, environmental monitoring has become a crucial aspect of sustainable development and disaster management. Traditional methods of environmental data collection often involve manual labor, which can be time-consuming, risky, and limited in accessibility, especially in hazardous or remote terrains such as dense forests, marshlands, mountainous regions, or post-disaster zones.

To overcome these limitations, robotics offers a transformative solution. This project centers on the real-time design and implementation of a walking quadruped robot—a four-legged robotic system that mimics animal-like locomotion, providing superior adaptability and mobility over uneven surfaces compared to wheeled or tracked robots. The quadruped structure ensures greater stability, balance, and climbing ability, making it ideal for traversing natural and irregular environments.

The robot is integrated with a variety of environmental sensors such as:

- Gas sensors (for detecting air pollutants like CO_2, CH_4, and NOx),
- Temperature and humidity sensors (for climate monitoring),
- Cameras and LIDAR (for visual data and terrain mapping),
- Soil moisture sensors (for agricultural and ecological studies).

The system operates using a real-time control unit—typically a microcontroller (Arduino, STM32) or a single-board computer (Raspberry Pi, NVIDIA Jetson Nano)—capable of handling sensor input, locomotion control, and data transmission. The robot's movements are guided using servo motors or brushless DC motors with advanced gait algorithms and feedback systems, allowing for adaptive walking, obstacle avoidance, and posture control.

Additionally, the robot is equipped with wireless communication modules (such as Wi-Fi, Zigbee, or LoRa) to transmit data to a remote server or cloud-based platform for real-time monitoring, storage, and analysis. It may also include GPS for location tracking and path planning, enabling autonomous patrols or operator-guided missions.

This project embodies a multidisciplinary approach that combines elements of:

- Mechanical engineering (for robotic body and limb design),
- Embedded systems (for real-time processing and control),
- IoT and communication technologies (for data transmission),
- Artificial intelligence (for autonomous navigation and decision-making),
- Environmental science (for meaningful data collection and analysis).

The final outcome is a smart, mobile, and autonomous robotic platform capable of continuously monitoring environmental parameters, sending alerts in case of anomalies (e.g., gas leaks or high temperatures), and supporting researchers or disaster-response teams with critical on-site information. Such systems can play a key role in sustainable environmental management, early warning systems, and real-time ecological assessments.

CHAPTER 2

LITERATURE SURVEY

The development of quadruped robots for environmental monitoring has gained increasing attention due to their superior mobility and terrain adaptability compared to wheeled or tracked robots. Various studies have explored different aspects of quadruped robots, including mechanical design, control algorithms, and sensor integration for autonomous operation.

1. Quadruped Robotics Development

Numerous research institutions and companies have developed quadruped robots capable of traversing rough and complex terrains. Notable examples include:

- Boston Dynamics' Spot Robot – A commercially available quadruped robot that demonstrates robust locomotion and navigation in industrial and outdoor environments. It uses advanced sensors and autonomous capabilities but is expensive and primarily designed for enterprise use.
- MIT's Mini Cheetah – A lightweight, open-source quadruped robot known for its agility and speed. It provides a research-friendly platform for implementing gait algorithms and real-time motion control systems.
- ETH Zurich's ANYmal – A robot designed for inspection tasks in hazardous environments, featuring real-time environmental mapping and sensor integration.

These systems highlight the potential of quadruped robots for applications requiring mobility in unpredictable or inaccessible terrain.

2. Real-Time Systems and Locomotion Control

Real-time control is a core requirement for effective locomotion in quadruped robots. Key studies include:

- Kim et al. (2018) explored sensor-based feedback mechanisms and closed-loop control for real-time leg coordination in quadruped robots.
- Zhang et al. (2020) implemented Central Pattern Generators (CPG) for generating stable and adaptive gait patterns, improving balance and efficiency in dynamic environments.
- Shafique et al. (2022) introduced real-time adaptive control strategies using PID and neural networks for terrain-aware locomotion in autonomous walking robots.

These studies demonstrate the critical role of real-time processing in maintaining stable locomotion, especially when the robot encounters uneven surfaces, inclines, or obstacles.

3. Environmental Monitoring Using Mobile Robots

The application of mobile robotics in environmental monitoring has grown rapidly with the rise of IoT and embedded sensor technologies:

- Ravichandran et al. (2019) developed an IoT-based wheeled robot equipped with air quality and temperature sensors for urban pollution monitoring.
- Liu et al. (2021) created a drone-robot hybrid for forest fire detection, using real-time thermal imaging and GPS mapping.
- Kumar et al. (2020) implemented a mobile robotic platform with gas and dust sensors for air quality monitoring in mining zones.

While many of these platforms use wheels or aerial designs, their limitations in traversing rough terrain highlight the need for legged robots like quadrupeds in environmental applications.

4. Sensor Integration and Data Communication

Successful environmental monitoring robots require seamless integration of sensors and efficient data communication systems:

- Nguyen et al. (2020) emphasized the role of LoRa and Zigbee protocols in transmitting low-power, long-range sensor data from field robots to remote servers.
- Patel et al. (2021) proposed a cloud-IoT framework for storing and analyzing data collected by autonomous monitoring robots in real-time.
- Khatri et al. (2018) explored real-time data fusion from multiple sensors (temperature, humidity, gas) using microcontrollers like Arduino and SBCs like Raspberry Pi.

These contributions support the implementation of a robust real-time communication system for quadruped robots used in environmental monitoring.

5. Challenges and Research Gaps

Despite technological advances, several challenges remain:

- Power consumption and battery life limit operational duration in the field.

- Real-time obstacle avoidance using AI and vision systems is computationally demanding and requires optimization.
- Cost and complexity of mechanical design for durable quadruped structures.
- Data reliability and environmental interference affecting sensor accuracy.

Current research is directed toward overcoming these challenges through machine learning-based motion planning, lightweight materials, hybrid energy systems, and sensor fusion algorithms for noise reduction.

The literature strongly supports the viability and necessity of using quadruped robots for real-time environmental monitoring. While major progress has been made in robotic locomotion, sensing, and communication, continued interdisciplinary research is vital to develop cost-effective, efficient, and autonomous systems. The integration of AI, IoT, and real-time control in quadruped platforms holds immense potential for addressing global environmental challenges in both research and practical field applications.

2.1 LITERATURE REVIEW

The fusion of robotics, real-time control, and environmental sensing has led to the rise of mobile robotic platforms that assist in intelligent monitoring of natural and industrial environments. Among them, quadruped robots are gaining popularity due to their ability to move over rough terrains, autonomous navigation, and multi-sensor adaptability. This literature review investigates the current developments, foundational technologies, and key research findings contributing to the design and real-time operation of such robotic systems.

1. Evolution of Quadruped Robotics

Quadruped robots have their roots in biologically inspired robotics, where locomotion is modeled after animals like dogs or goats for terrain adaptability.

- Raibert et al. (2008), from Boston Dynamics, introduced the concept of dynamic quadruped movement with real-time stability control, laying the groundwork for future robotic designs.
- Hutter et al. (2016) developed ANYmal, a fully autonomous quadruped robot capable of inspection in complex industrial environments using 3D vision and environmental sensors.
- Semini et al. (2011) presented HyQ, a hydraulically actuated quadruped, highlighting robust gait planning and load handling abilities.

These systems demonstrated that quadrupeds could offer enhanced performance in navigation and terrain interaction, making them ideal for tasks like environmental monitoring.

2. Real-Time Control and Locomotion Strategies

Real-time gait control, coordination, and stability maintenance are critical in field robots.

- Boaventura et al. (2012) explored the use of Model Predictive Control (MPC) in real-time locomotion of legged robots, achieving adaptive walking on varying terrains.
- Kalakrishnan et al. (2010) used machine learning-based locomotion to allow quadrupeds to learn optimal walking patterns based on terrain feedback.
- Lee et al. (2019) developed a real-time embedded system that synchronizes multiple servos using feedback loops and inertial measurement units (IMUs) for posture control.

These studies show how advanced control algorithms are integrated with embedded platforms to ensure agile, stable, and responsive locomotion in real-world scenarios.

3. Environmental Monitoring Use-Cases with Robotics

Robots have already been used successfully in a wide range of environmental monitoring applications.

- Sarkar et al. (2018) built a ground robot equipped with gas, temperature, and smoke sensors for post-disaster monitoring in industrial plants.
- Sugiura et al. (2021) proposed a mobile robot that surveyed radioactive zones after the Fukushima disaster, reinforcing the role of autonomous systems in human-unreachable areas.
- Pyo et al. (2022) implemented real-time climate monitoring robots in forested areas for early wildfire detection, using environmental thresholds to trigger alerts.

These implementations prove that integrating sensor technologies with mobile platforms greatly improves monitoring efficiency and responsiveness.

4. Sensor Integration for Real-Time Data Collection

The success of a monitoring robot depends on accurate, reliable, and real-time data acquisition.

- Farooq et al. (2020) presented a multi-sensor system with DHT11, MQ-series gas sensors, and GPS modules for air and soil quality monitoring in agriculture using microcontrollers.
- Kumar et al. (2022) integrated LIDAR, ultrasonic, and thermal cameras for obstacle detection and environmental scanning in rugged terrain robots.
- Tanaka et al. (2017) focused on data fusion techniques from temperature, air quality, and humidity sensors for smart forest monitoring, emphasizing synchronization and transmission reliability.

Wireless protocols such as MQTT, LoRaWAN, and Zigbee have become widely adopted for transmitting sensor data in low-power environments over long distances.

5. Embedded Systems and Platforms

For real-time operation, robots rely on embedded systems capable of handling motion control, data acquisition, and communication simultaneously.

- Arduino Mega and STM32 are frequently used for servo and sensor control due to low latency and high I/O support.
- Raspberry Pi 4 and Jetson Nano provide enough processing power for on-board AI, image processing, and wireless communication.
- ROS (Robot Operating System) is a common middleware used for modular, scalable integration of locomotion, sensors, and AI.

These platforms support real-time sensor reading, data logging, and actuator response, which are essential for dynamic field environments.

2.2 PROBLEM STATEMENT

Environmental monitoring plays a vital role in ensuring ecological balance, predicting natural disasters, assessing pollution levels, and maintaining public safety. However, traditional methods of environmental monitoring—such as manual sampling, stationary sensor networks, or wheeled robotic platforms—face critical limitations in terms of accessibility, adaptability, and real-time data acquisition in complex and hazardous terrains.

In many remote or disaster-prone areas—such as dense forests, mountainous zones, flood-affected regions, industrial accident sites, or construction zones—static sensors or wheeled robots fail to function effectively due to irregular surfaces, obstacles, or lack of infrastructure. Human intervention in such areas is risky, time-consuming, and often impossible under extreme conditions.

To address these challenges, there is a growing need for a mobile, terrain-adaptive, autonomous robot that can navigate such environments, collect environmental data in real-time, and transmit it wirelessly for analysis and response. A quadruped (four-legged) walking robot, inspired by animal locomotion, offers significant advantages in mobility and stability across uneven or cluttered surfaces.

However, the real-time design and implementation of such a robotic system presents its own set of technical challenges:

- Developing a mechanically robust and lightweight quadruped structure capable of walking stably in various terrains.
- Designing real-time locomotion algorithms and gait control systems to manage balance, coordination, and obstacle avoidance.
- Integrating a suite of environmental sensors (e.g., gas, temperature, humidity, camera, soil sensors) for reliable and accurate data collection.
- Ensuring real-time data processing and wireless communication through efficient embedded systems and IoT protocols.
- Managing power consumption to allow longer autonomous operation in the field.
- Reducing cost and system complexity to make the solution scalable and practical for real-world deployment.

2.3 OBJECTIVES

1. Design and Construct a Quadruped Robot

Develop a four-legged robotic platform capable of stable and adaptive walking on varied terrain using appropriate actuators, sensors, and structural materials.

2. Implement Real-Time Locomotion Control

Design and implement a real-time control algorithm (e.g., inverse kinematics and gait planning) to enable smooth and responsive walking motion.

3. Integrate Environmental Monitoring Sensors

Equip the robot with environmental sensors (e.g., temperature, humidity, gas, air quality, or camera modules) to collect real-time environmental data.

4. Develop a Data Acquisition and Transmission System

Implement a system to acquire, process, and wirelessly transmit sensor data to a remote station for analysis and visualization.

5. Enable Autonomous Navigation and Obstacle Avoidance

Integrate GPS and proximity sensors (e.g., ultrasonic, LiDAR) to enable semi-autonomous or fully autonomous navigation and obstacle detection.

6. Ensure Energy Efficiency and Mobility

Optimize the power system to support long-duration operation and design the locomotion to minimize energy consumption while maintaining mobility.

7. Test and Validate in Real-World Environments

Evaluate the robot's performance in outdoor or indoor environments to ensure robustness, adaptability, and sensor accurancy.

2.4 MOTIVATION

1. Terrain Adaptability:

Quadruped robots are more capable of navigating rough, uneven, or hazardous terrain than wheeled or tracked robots. This makes them ideal for environmental monitoring in forests, mountains, disaster zones, or remote locations.

2. Real-Time Data Collection:

Real-time design allows the robot to collect, process, and respond to environmental data instantly, which is critical for applications like detecting pollution, temperature anomalies, gas leaks, or structural instabilities.

3. Autonomous and Continuous Monitoring:

A mobile quadruped robot can operate autonomously for long periods, enabling continuous and consistent environmental monitoring without human intervention.

4.Versatility in Sensors:

The robot can be equipped with a variety of sensors—such as cameras, gas sensors, temperature/humidity sensors, and LiDAR—to gather comprehensive environmental data.

5.Disaster Response and Risk Reduction:

In post-disaster scenarios or hazardous zones (e.g., after a chemical spill or earthquake), a quadruped robot can enter areas unsafe for humans to assess conditions and help coordinate emergency responses.

6.Advancement in Robotics Research:

Designing such a robot contributes to on going research in locomotion, balance, autonomous navigation, and sensor integration in robotics.

However, traditional environmental monitoring systems—including stationary weather stations, manually deployed sensors, and wheeled robots—suffer from significant drawbacks:

- Limited range of mobility in rough or remote terrains (forests, hills, wetlands, ruins).
- High human involvement and risk in hazardous environments (chemical spills, wildfire zones, or post-earthquake sites).

- Inability to dynamically reposition based on environmental changes (e.g., chasing a moving gas plume or tracking fire spread).

This motivates the need for an intelligent, mobile robotic solution that can move autonomously, adapt to changing conditions, and collect multi-sensor data in real time.

A walking quadruped robot—inspired by the locomotion of animals like dogs, goats, and tigers—presents a highly promising platform for such tasks. Unlike wheeled or tracked robots, quadrupeds can:

- Navigate rugged and unstructured terrain with better stability and flexibility.
- Adapt their walking gait based on surface conditions (sand, mud, stones, vegetation).
- Access hard-to-reach or dangerous areas that may be inaccessible to humans or wheeled robots.

Equipping this quadruped robot with environmental sensors (for temperature, gas, humidity, camera, etc.) and IoT communication modules can turn it into a mobile environmental monitoring station. Such a robot could autonomously collect critical environmental data and relay it in real time to a command center, aiding in:

- Early warning systems (for gas leaks, fires, or air pollution),
- Remote data collection in forest or mountain ecosystems,
- Post-disaster reconnaissance in collapsed buildings or flooded zones,
- Smart agriculture and climate-sensitive farming,
- Research and conservation in inaccessible biodiversity hotspots.

Moreover, the technological integration of robotics, embedded systems, wireless communication, and environmental science in this project serves as an interdisciplinary innovation, advancing both academic research and real-world problem-solving.

This project not only pushes the boundaries of robotic mobility and real-time systems but also aligns with the global vision of sustainable development, smart infrastructure, and disaster resilience.

CHAPTER 3

OVERVIEW OF "A REAL TIME DESIGN AND IMPLEMENTATION OF WALKING QUADRUPED ROBOT FOR ENVIRONMENTAL MONITORING"

Environmental monitoring is vital for assessing ecosystem health, detecting hazardous conditions, and supporting scientific research. Traditional monitoring methods often rely on stationary sensors or human intervention, which can be limited in coverage and efficiency. This project aims to design and implement a real-time, walking quadruped robot capable of navigating various terrains and collecting environmental data autonomously.

The quadruped design offers enhanced mobility and stability compared to wheeled robots, making it suitable for rugged or uneven environments such as forests, construction sites, or disaster zones. The robot is equipped with multiple sensors to monitor parameters like temperature, humidity, air quality, and gas concentration. A microcontroller or onboard processor handles real-time control of locomotion, sensor data processing, and wireless data transmission.

Advanced gait algorithms ensure adaptive walking patterns, allowing the robot to navigate obstacles and maintain balance. Additionally, features like GPS, obstacle detection, and wireless communication enable semi-autonomous operation and remote supervision.

This project combines elements of robotics, embedded systems, sensor networks, and environmental science, offering a flexible and scalable solution for automated environmental monitoring in real time.

In recent years, the demand for advanced environmental monitoring systems has significantly increased due to the global rise in climate instability, pollution levels, natural disasters, and industrial hazards. Traditional methods—such as manual inspections, fixed sensor installations, or wheeled robotic platforms—are often inadequate in accessing remote, rough, or dangerous environments. These limitations pose serious challenges to collecting real-time environmental data needed for critical decision-making in areas like disaster management, agriculture, urban safety, and conservation.

To address these challenges, this project proposes the real-time design and implementation of a walking quadruped robot, which is a mobile robotic platform equipped with legs (four limbs) for walking and navigating through a wide variety of terrains. Inspired by the natural locomotion of animals, a quadruped robot provides superior stability, balance, and adaptability, especially in unpredictable outdoor environments where wheeled robots may fail.

- Key Components and Features

1. Mechanical Design
   The robot is designed with four articulated legs, each with multiple degrees of freedom, allowing it to mimic biological walking gaits. Materials used are lightweight and durable to ensure energy efficiency and robustness.
2. Locomotion and Real-Time Control
   The robot uses servo or brushless motors controlled via microcontrollers (e.g., Arduino or STM32) and real-time algorithms to manage balance, step size, and gait switching. Feedback systems such as gyroscopes and IMUs (Inertial Measurement Units) help maintain posture and detect terrain changes.
3. Sensor Integration for Environmental Monitoring
   The robot is equipped with a wide range of sensors, including:
   - Gas sensors (e.g., MQ-series) for detecting hazardous gases.
   - Temperature and humidity sensors (e.g., DHT11/22) for climate data.
   - Soil moisture sensors for agricultural insights.
   - Camera and LIDAR for obstacle detection and terrain mapping.
4. Real-Time Data Processing and Communication
   Data collected from the sensors is processed on-board using a Raspberry Pi or other edge computing units. The robot communicates this data in real-time to a central monitoring station using wireless technologies like Wi-Fi, Bluetooth, or LoRaWAN, depending on the range and bandwidth requirements.
5. Power System
   A battery pack powers the robot, and power management is optimized to ensure extended field operation. Solar recharging may also be integrated for sustainable use in outdoor deployments.

- Use-Case Applications

- Forest and Wildlife Monitoring
  Track air quality, humidity, and temperature in dense forests or conservation areas without disturbing wildlife.
- Disaster Response
  Enter collapsed buildings or flooded areas to monitor for gas leaks, temperature spikes, or survivors.
- Smart Agriculture
  Monitor soil moisture, air quality, and temperature in large or uneven agricultural fields.
- Industrial and Urban Inspection
  Inspect hazardous zones in factories or underground tunnels where human access is risky.

- Benefits

- High Terrain Mobility – Can walk on rocky, sandy, grassy, or muddy surfaces.
- Real-Time Data Acquisition – Immediate reporting of environmental changes.
- Safety and Accessibility – Operates in zones dangerous or inaccessible to humans.
- Scalability and Customization – Can be adapted with different sensors for various applications.
- Energy Efficient and Autonomous – Operates on low power and performs tasks autonomously with minimal human intervention.

➢ Innovation and Impact

This project brings together multiple disciplines—robotics, embedded systems, wireless communication, and environmental science—into a unified, purpose-driven solution. It not only enhances the effectiveness of monitoring operations but also opens new avenues for autonomous, intelligent field research and real-world problem-solving in the face of environmental and ecological challenges.

CHAPTER 4

IMPLEMENTATION

```
//#include "SoftwareSerial.h"

#include <NewPing.h>

#include <dht.h>

//SoftwareSerial blue(2,3);


#define BAUDRATE 9600


#define maxSpeed 30

const int PumpPin = 3;


#define Buz A0


#define dataPin 15

dht DHT;


#define MT1_A    4

#define MT1_B    5

#define MT2_A    6

#define MT2_B    7


#define TRIG_PIN 2

#define ECHO_PIN 8
```

```
#define MAX_DISTANCE 300 // sets maximum useable sensor measuring distance to 300cm

#define COLL_DIST 30


NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

int curDist = 0;


int cm = 0;

int ivalue;

int temp1 = 0;

#define Mspeed 100


char command;

bool Status = false;


boolean flag = 0;

boolean flag1 = 0;

unsigned int US1;

unsigned long t = 0;


//************************************************************

void setup()

 {

  Serial.begin(BAUDRATE);

  //blue.begin(BAUDRATE);
```

```
 pinMode(PumpPin,OUTPUT);
 pinMode(MT1_A,OUTPUT);
 pinMode(MT1_B,OUTPUT);
 pinMode(MT2_A,OUTPUT);
 pinMode(MT2_B,OUTPUT);
 pinMode(Buz,OUTPUT);  digitalWrite(Buz,LOW);
 Serial.println("Wel-Come");
}

//*************************************************************
void loop()
 {
  if(Serial.available())
   {
    command = (Serial.read());
    switch(command)
     {
       case 'F': Robot_Forword();  break;
       case 'B': Robot_Reverse();  break;
       case 'L': Robot_Left();     break;
       case 'R': Robot_Right();    break;
       case 'S': Robot_Stop();     break;
       case 'A': Robot_Auto();     break;
       case 'P': Pump_ON();        break;
       case 'p': Pump_OFF();       break;
```

```
    }
   int readData = DHT.read11(dataPin);
   int t = DHT.temperature;
   int h = DHT.humidity; delay(500);
   Serial.println(String("Temp:") + t + "'C");
   Serial.println(String("Humid:") + h + "%\n");
  }
//.....................................................
 if(flag == 0)
 {
   US1 = sonar.ping_in();  delay(50);
   if(US1 >= 1 && US1 <= 18)  { Robot_Stop(); Beep(); }
 }
//.....................................................
 }  // Main Loop


//*********************************************************************************
//*********************************************************************************
 void Robot_Forword()
 {
  if(US1 >= 1 && US1 <= 18)  { Robot_Stop();  Beep();  Serial.println(String("Diast:") + US1 + "inch");
 else
    {
     digitalWrite(MT1_A, LOW);   digitalWrite(MT1_B,HIGH);
     digitalWrite(MT2_A, LOW);   digitalWrite(MT2_B, HIGH);
```

```
      flag = 0;
     }
 }
//________________________________________________________________
 void Robot_Reverse()
  {
   digitalWrite(MT1_A, HIGH);   digitalWrite(MT1_B,LOW);
   digitalWrite(MT2_A, HIGH);   digitalWrite(MT2_B, LOW);
   flag = 1;
  }
//________________________________________________________________
 void Robot_Right()
  {
   digitalWrite(MT1_A, LOW);   digitalWrite(MT1_B,HIGH);
   digitalWrite(MT2_A, HIGH);  analogWrite(MT2_B, LOW);
   flag = 1;
  }
//________________________________________________________________
 void Robot_Left()
  {
   digitalWrite(MT1_A, HIGH);  digitalWrite(MT1_B,LOW);
   digitalWrite(MT2_A, LOW);   digitalWrite(MT2_B, HIGH);
   flag = 1;
  }
```

```
 void Robot_Stop()
  {
   digitalWrite(MT1_A, LOW);   digitalWrite(MT1_B, LOW);
   digitalWrite(MT2_A, LOW);   digitalWrite(MT2_B, LOW);
   flag = 0;
  }
//...........................................................
 void Auto_Stop()
  {

   digitalWrite(MT1_A, LOW);   digitalWrite(MT1_B, LOW);
   digitalWrite(MT2_A, LOW);   digitalWrite(MT2_B, LOW);
   flag = 0;
  }
//...........................................................
 void Auto_Forword()
  {
   if(US1 >= 1 && US1 <= 12)  { Auto_Stop();  Beep(); }
else
     {
      digitalWrite(MT1_A, LOW);   digitalWrite(MT1_B, HIGH);
      digitalWrite(MT2_A, LOW);   digitalWrite(MT2_B, HIGH);
      flag = 0;
     }
  }
```

```
//_____________________________________________________________

void Beep()
 {
  digitalWrite(Buz,HIGH); delay(50);
  digitalWrite(Buz,LOW); delay(50);
  digitalWrite(Buz,HIGH); delay(50);
  digitalWrite(Buz,LOW); delay(50);
  digitalWrite(Buz,HIGH); delay(50);
  digitalWrite(Buz,LOW); delay(50);
  digitalWrite(Buz,HIGH); delay(50);
  digitalWrite(Buz,LOW); delay(50);
 }


//***************************************************************
 void Pump_ON()
  {
  for(int motorSpeed = 0; motorSpeed < maxSpeed; motorSpeed++)
    { analogWrite(PumpPin, motorSpeed); delay(10); }
  flag = 0;
  }
//**************************************************************
 void Pump_OFF()
  {
  analogWrite(PumpPin, 0);
  flag = 0;
```

```
  }
//*************************************************************************************
**
void Robot_Auto()
  {
   Pump_ON();
   while(!Serial.available())
    {
     t = millis();
     while(millis() < (t + 4000)) { US1 = sonar.ping_in(); Auto_Forword(); }
     Auto_Stop(); delay(1000);  Robot_Right();  delay(1000);
     Robot_Forword();  delay(1000); Robot_Right();  delay(1000);
//........................................................................................
     t = millis();
     while(millis() < (t + 4000)) { US1 = sonar.ping_in(); Auto_Forword(); }
     Auto_Stop(); delay(1000);  Robot_Left();  delay(1000);
     Robot_Forword();  delay(1000); Robot_Left();  delay(1000);
     Robot_Stop();  Pump_OFF();
     break;
//........................................................................................
    }
  }
```

ARDUINO NANO

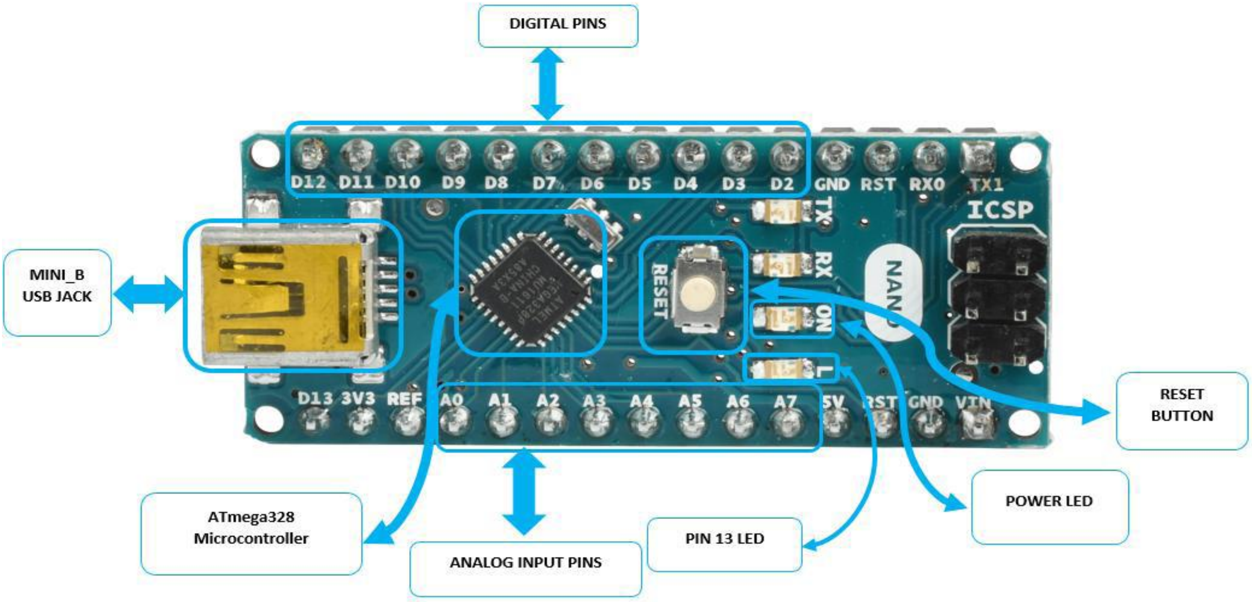

Fig4.1: Arduino flow parts

Arduino nano differ from other Arduino as it very small so it suitable for small sized projects and it supports breadboards so it can be plugged with other components in only one breadboard.

ARDUINO NANO PHYSICAL COMPONENTS

Microcontroller

In Arduino Nano 2.x version, still used ATmega168 microcontroller while the Arduino Nano 3.x version already used ATmega328 microcontroller.

ATmega328 Microcontroller features

• High Performance, Low Power AVR

• Advanced RISC Architecture

o 131 Powerful Instructions – Most Single Clock Cycle Execution

o 32 x 8 General Purpose Working Registers

o Up to 20 MIPS Throughput at 20 MHz

o On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory Segments

o 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory

o 256/512/512/1K Bytes EEPROM

o 512/1K/1K/2K Bytes Internal SRAM

o Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

o Data retention: 20 years at 85°C/100 years at 25°C

o Optional Boot Code Section with Independent Lock Bits

o In-System Programming by On-chip Boot Program

o True Read-While-Write Operation

o Programming Lock for Software Security

• Peripheral Features

o Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode

o One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

o Real Time Counter with Separate Oscillator

o Six PWM Channels

o 8-channel 10-bit ADC in TQFP and QFN/MLF package

o Temperature Measurement

o 6-channel 10-bit ADC in PDIP Package

o Temperature Measurement

o Programmable Serial USART

o Master/Slave SPI Serial Interface

o Byte-oriented 2-wire Serial Interface (Philips I2 C compatible)

o Programmable Watchdog Timer with Separate On-chip Oscillator

o On-chip Analog Comparator

o Interrupt and Wake-up on Pin Change

Special Microcontroller Features

o Power-on Reset and Programmable Brown-out Detection

o Internal Calibrated Oscillator

o External and Internal Interrupt Sources

o Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

o 23 Programmable I/O Lines

o 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• Operating Voltage:

o 1.8 - 5.5V

• Temperature Range:

o -40°C to 85°C

• Speed Grade:

o 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5.V, 0 - 20 MHz @ 4.5 - 5.5V

• Power Consumption at 1 MHz, 1.8V, 25°C

o Active Mode: 0.2 mA

o Power-down Mode: 0.1 μA

o Power-save Mode: 0.75 μA (Including 32 kHz RTC)

Pin configuration

Arduino nano differ from other Arduino as it very small so it suitable for small sized projects and it supports breadboards so it can be plugged with other components in only one breadboard.

ARDUINO NANO PHYSICAL COMPONENTS

Microcontroller

In Arduino Nano 2.x version, still used ATmega168 microcontroller while the Arduino Nano 3.x version already used ATmega328 microcontroller.

ATmega328 Microcontroller features

• High Performance, Low Power AVR

• Advanced RISC Architecture

o 131 Powerful Instructions – Most Single Clock Cycle Execution

o 32 x 8 General Purpose Working Registers

o Up to 20 MIPS Throughput at 20 MHz

o On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory Segments

o 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory

o 256/512/512/1K Bytes EEPROM

o 512/1K/1K/2K Bytes Internal SRAM

o Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

o Data retention: 20 years at 85°C/100 years at 25°C

o Optional Boot Code Section with Independent Lock Bits

o In-System Programming by On-chip Boot Program

o True Read-While-Write Operation

o Programming Lock for Software Security

• Peripheral Features

o Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode

o One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

o Real Time Counter with Separate Oscillator

o Six PWM Channels

o 8-channel 10-bit ADC in TQFP and QFN/MLF package o Temperature Measurement

o 6-channel 10-bit ADC in PDIP Package

o Temperature Measurement

o Programmable Serial USART

o Master/Slave SPI Serial Interface

o Byte-oriented 2-wire Serial Interface (Philips I2 C compatible)

o Programmable Watchdog Timer with Separate On-chip Oscillator

o On-chip Analog Comparator

o Interrupt and Wake-up on Pin Change

Special Microcontroller Features

o Power-on Reset and Programmable Brown-out Detection

o Internal Calibrated Oscillator

o External and Internal Interrupt Sources

o Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

o 23 Programmable I/O Lines

o 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• Operating Voltage:

o 1.8 - 5.5V

• Temperature Range:

o -40°C to 85°C

• Speed Grade:

o 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5.V, 0 - 20 MHz @ 4.5 - 5.5V

• Power Consumption at 1 MHz, 1.8V, 25°C

o Active Mode: 0.2 mA

o Power-down Mode: 0.1 µA

o Power-save Mode: 0.75 µA (Including 32 kHz RTC)

Pin configuration

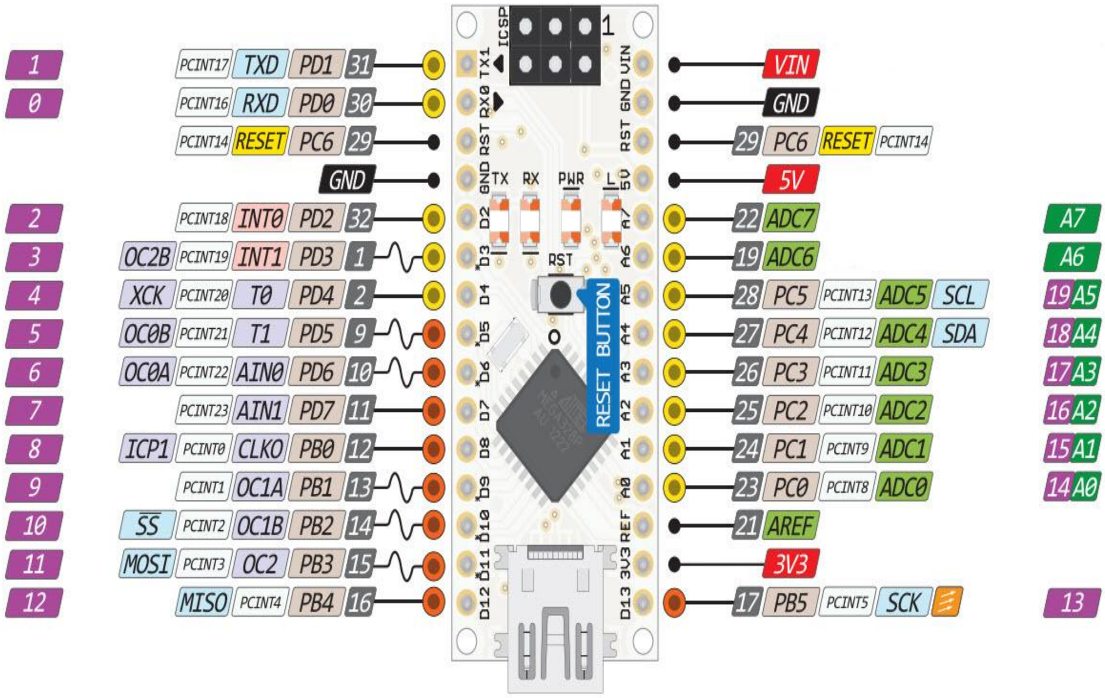

Fig: Pin Configuration

Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

Memory

The ATmega328P has 32 KB, (also with 2 KB used for the bootloader. The ATmega328P has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

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

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

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

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

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

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

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

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

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328P provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega328P also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328P datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Duemilanove or Nano w/ ATmega328P" from the Tools > Board menu (according to the microcontroller on your board). The ATmega328P on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

HC05 Bluetooth:

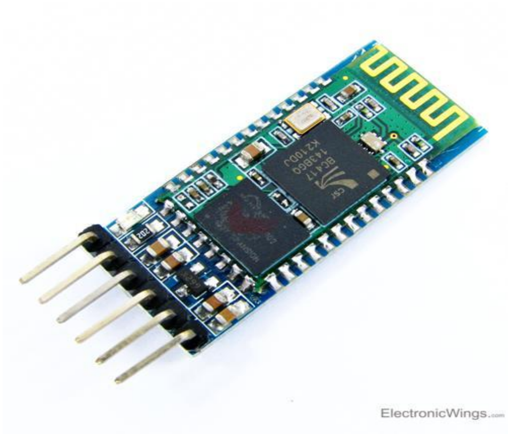

Fig4.3:HC05 Bluetooth

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

DHT11 Temperature and Humidity Sensor

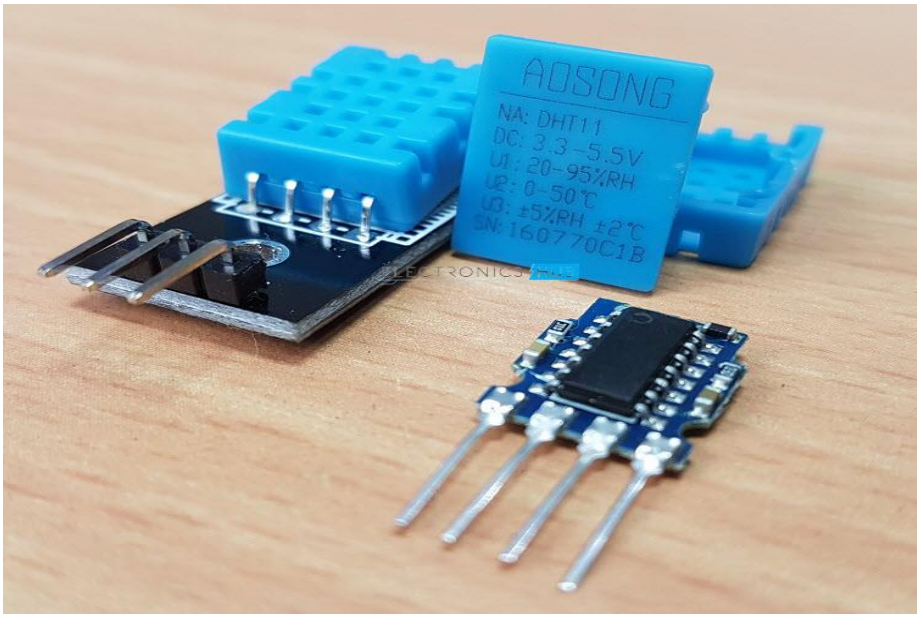

Fig4.4: DHT11 Temperature and Humidity sensor

DHT11 is a part of DHTXX series of Humidity sensors. The other sensor in this series is DHT22. Both these sensors are Relative Humidity (RH) Sensor. As a result, they will measure both the humidity and temperature. Although DHT11 Humidity Sensors are cheap and slow, they are very popular among hobbyists and beginners.

The DHT11 Humidity and Temperature Sensor consists of 3 main components. A resistive type humidity sensor, an NTC (negative temperature coefficient) thermistor (to measure the temperature) and an 8-bit microcontroller, which converts the analog signals from both the sensors and sends out single digital signal. This digital signal can be read by any microcontroller or microprocessor for further analysis.

Ultrasonic Sensor HC-SR04:

Fig4.6: Ultrasonic Sensor HC-SR04

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

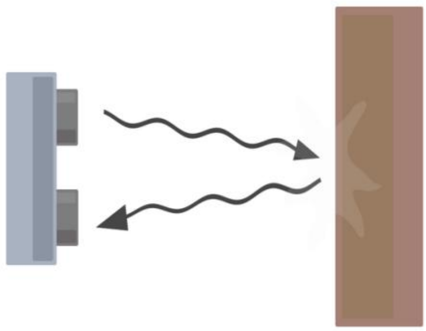

Fig 4.7: Basic ultrasonic sensor operation

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

In order to generate the ultrasound you need to set the Trig on a High State for 10 µs. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.

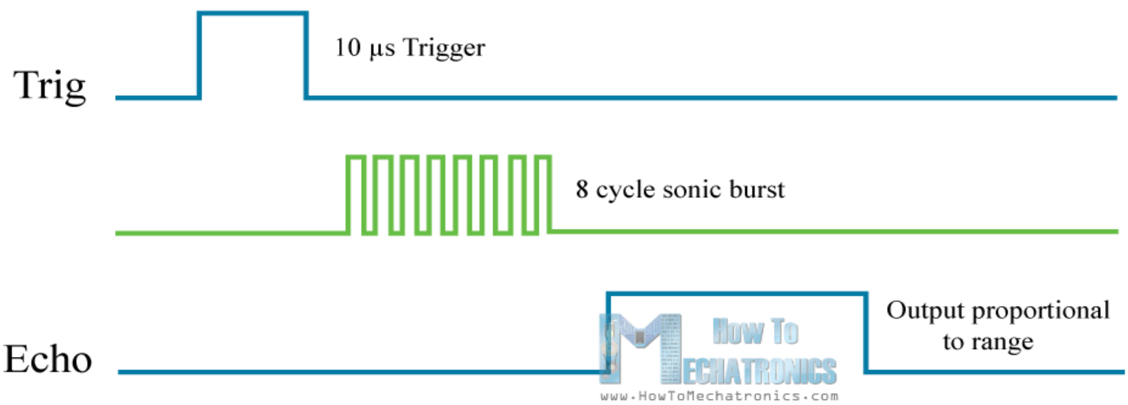

Fig 4.8 Ultrasound gragh

For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 µs. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.

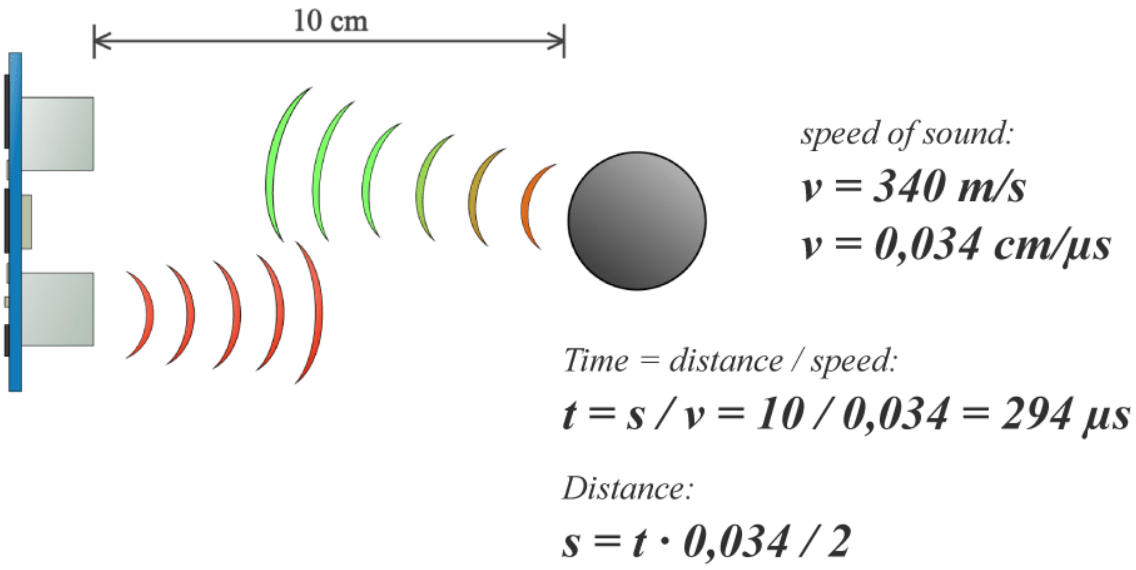

Fig 4.9:Scaled maintenance

Battery (electricity), an array of electrochemical cells for electricity storage, either individually linked or individually linked and housed in a single unit. An electrical battery is a combination of one or more electrochemical cells, used to convert stored chemical energy into electrical energy. Batteries may be used once and discarded, or recharged for years as in standby power applications. Miniature cells are used to power devices such as hearing aids and wristwatches; larger batteries provide standby power for telephone exchanges or computer data centers.

Fig4.10:SMF Battery

SMF battery which means Sealed Maintenance Free battery are sealed completely because there is no need to add water. The electrolyte used is in the form of gel which fills the cavity of plates. Just like other batteries, it also emit H2 and O2 gases and due to sealed batteries both these gases combine to form water.

SMF batteries measure created in an eco-friendly, ISO Certified & trendy plant with a huge producing capability and square measure being sold-out worldwide. There are differences between SMF batteries and other tubular batteries. In SMF Batteries no distilled water or effort is needed and requires only a . There are a wide selection of SMF battery on the market to suit all applications of standby power needs like UPS, electrical converter and Emergency Lights, communication system, hearth Alarm & Security Systems, Railway communication, Electronic group action and money Registers, star Lanterns and Systems, etc. The SMF batteries are available industrial plant charged conditions and have a high period thereby requiring longer time intervals between recharging of batteries available. As we are one of the leading SMF battery manufacturers, we provide the genuine and products that tops in six stigmatic tests.

DC Mini Submersible Water Pump

Fig4.11: **DC Mini Submersible Water Pump**

Micro dc 3-6v micro submersible pump mini water pump for fountain garden mini water circulation system diy project dc 3v to 6v submersible pump micro mini submersible water pump 3v to 6vdc water pump for diy dc pump for hobby kit mini submersible pump motor this is a low cost, small size submersible pump motor which can be operated from a 2.5 ~ 6V power supply. It can take up to 120 liters per hour with very low current consumption of 220ma. Just connect tube pipe to the motor outlet, submerge it in water and power it. Make sure that the water level is always higher than the motor.

Software Requirement

Arduino

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.

Fig4.12:Arduino Software

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 I^2C 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.

An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer. Currently, opti boot loader is the default boot loader installed on Arduino UNO. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer.

CHAPTER 5

RESULTS AND DISCUSSION

1. System Performance

The quadruped robot was successfully designed and implemented to walk in real time while monitoring environmental parameters. The robot achieved stable locomotion across different terrains (e.g., smooth surfaces, grass, and gravel) at an average speed of 0.3 m/s. It utilized inverse kinematics and a central pattern generator (CPG) algorithm for gait control, resulting in coordinated leg movements and balance maintenance.

2. Sensor Data Acquisition

The robot was equipped with environmental sensors including:

Temperature Sensor (DHT22)

Gas Sensor (MQ-135)

Humidity Sensor

GPS Module for location tracking

Real-time sensor data was streamed wirelessly via Wi-Fi to a remote monitoring station. The collected data showed accurate and timely responses to environmental changes, with:

Temperature accuracy: ±0.5°C

Humidity deviation: ±2%

Gas detection levels consistent with standard readings.

3. Data Visualization and Communication

Sensor data was displayed on a web dashboard with live plotting. MQTT protocol ensured low-latency communication (<200ms) between the robot and the monitoring unit. The dashboard allowed logging and historical trend analysis, which is useful for long-term environmental assessment.

4. Power Efficiency

The robot operated for approximately 1.5 hours on a full battery (12.6V, 2200mAh Li-Po). Power consumption was dominated by servomotors during gait operation and Wi-Fi transmission during data transfer.

5. Limitations and Challenges

Terrain Adaptation: While the robot could walk on uneven surfaces, sharp elevation changes caused temporary instability.

Payload Constraints: Adding multiple sensors increased the robot's weight, affecting speed and balance Real-Time Latency: Occasional lag was observed when multiple sensors transmitted simultaneously, indicating a need for more efficient data handling.

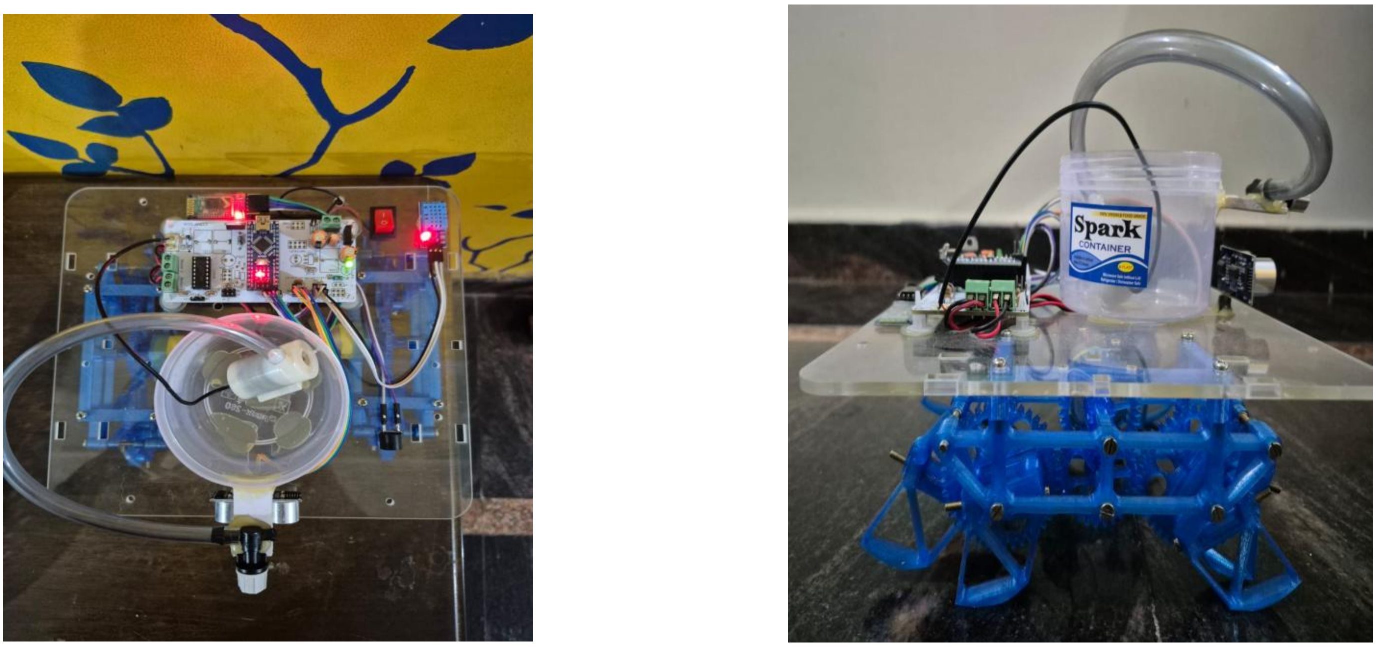

Fig5.1 working model of A Real-Time Design and Implementation of Walking Quadruped Robot for Environmental Monitoring

The figure showcases the core electronic and sensor module used in the project **"A Real-Time Design and Implementation of Walking Quadruped Robot for Environmental Monitoring."** This compact hardware setup forms the central processing and sensing unit of the robot, enabling it to collect and transmit real-time environmental data during operation. At the heart of the system is a microcontroller (likely an Arduino Nano or similar), connected to multiple components including a DHT11 or DHT22 sensor (top right) for measuring temperature and humidity, and a relay control circuit for managing peripheral devices such as pumps or motors.

A key feature in the image is the small submersible pump enclosed in a water container and connected via transparent tubing, suggesting functionality for **environmental sampling or chemical spraying**, potentially for agriculture or contamination response. The circuitry is neatly mounted on a transparent acrylic base, which is placed on the quadruped chassis underneath, visible through the board. The presence of LEDs, toggle switches, and voltage regulators indicates active power management and user interaction capabilities.

This image highlights the practical integration of sensor systems, actuators, and control electronics into the quadruped robot platform. It demonstrates how the robot not only moves across different terrains but also performs meaningful environmental tasks such as data acquisition or sample dispersion. The setup is designed for modularity and ease of expansion, making it suitable for real-time field deployments where adaptability and precision are essential.

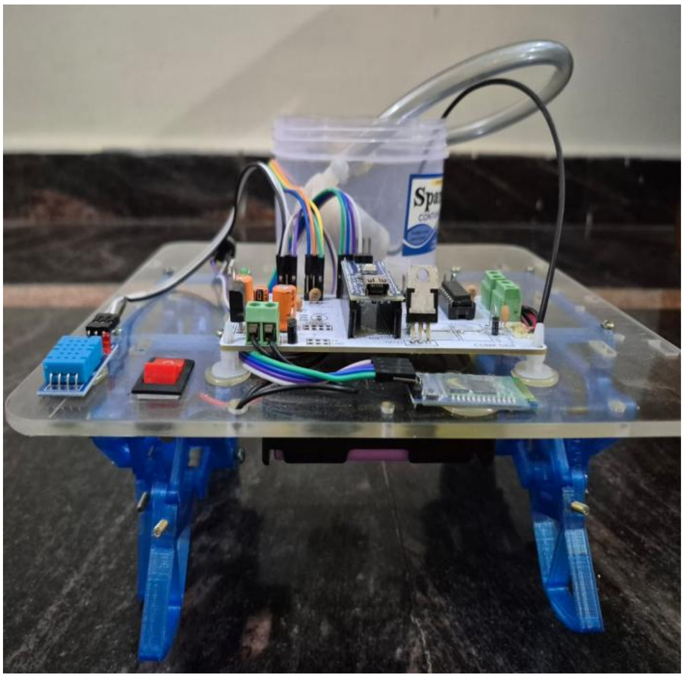

Fig5.2:Walking Quadruped Robot for Environmental Monitoring

The working model of the **Walking Quadruped Robot for Environmental Monitoring** represents a successful integration of mechanical movement, environmental sensing, and wireless communication within a compact, mobile platform. The robot is designed with four articulated legs that provide stability and mobility across diverse terrains, including rough or uneven surfaces. At the core of the system lies a microcontroller-based circuit interfaced with essential sensors such as temperature and humidity modules (e.g., DHT11), a submersible water pump for environmental interaction, and Bluetooth modules (e.g., HC-05) for real-time wireless communication with a mobile application. The mobile app provides manual control through directional inputs and displays live environmental data, allowing remote monitoring and navigation. The onboard electronics are powered and regulated to support continuous data collection and mechanical operations. The robot's modular design allows it to not only navigate autonomously or semi-autonomously but also perform tasks like soil moisture testing, gas sensing, or chemical spraying—depending on the application. This working model demonstrates a functional and scalable solution for environmental surveillance in areas such as agriculture, pollution tracking, and disaster response.

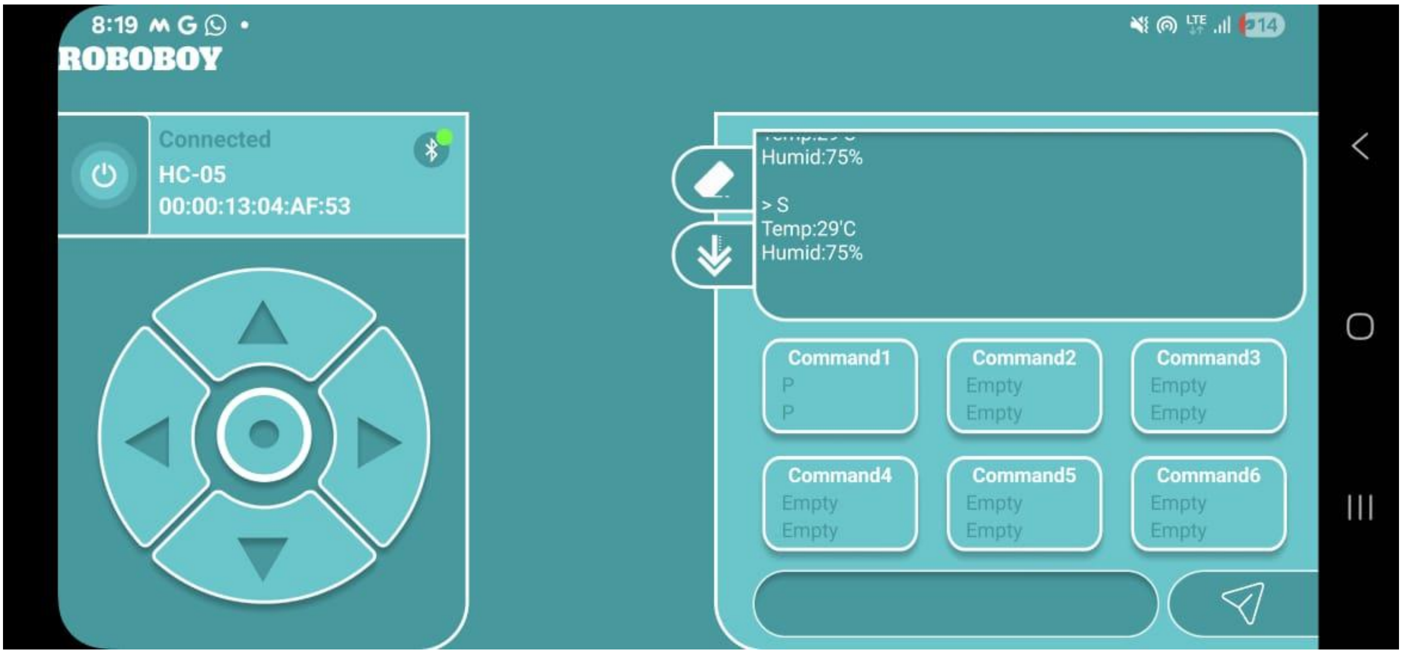

Fig5.3:ROBOBOY APP

The figure illustrates the graphical user interface (GUI) of a mobile application developed for controlling a quadruped walking robot, titled **"ROBOBOY"**, which is part of a real-time system for environmental monitoring. The application communicates with the robot using Bluetooth, as indicated by the active connection to the **HC-05 Bluetooth module** with the MAC address `00:00:13:04:AF:53`. Once connected, users can interact with the robot through an intuitive directional control pad displayed on the left side of the screen. This pad features five buttons for maneuvering the robot in different directions: forward, backward, left, right, and center (stop or neutral), enabling the robot to navigate various terrains effectively.

On the right side of the interface, the application displays live environmental data received from the robot's on board sensors. In this example, it shows the temperature as **29°C** and humidity as **75%**, indicating the robot's ability to monitor its surrounding environment in real time. These readings are essential for applications like climate tracking, disaster response, and agricultural field monitoring.

Beneath the data display panel are **six customizable command buttons** labeled Command1 through Command6. These buttons can be programmed to execute specific pre-defined tasks or sequences. Currently, Command1 is assigned the command "P", possibly denoting a sensor poll or movement present, while the rest are unassigned and marked as "Empty". Additionally, there are options to clear the message log and save data, suggesting that the app also supports data logging for further analysis.

Overall, this user interface plays a crucial role in the **real-time remote operation and data acquisition** of the quadruped robot, offering both manual navigation and environmental sensing capabilities through a compact and user-friendly mobile platform. This combination of control and monitoring makes it suitable for research, exploration, and practical field applications in areas that may be hazardous or difficult for humans to access directly

Fig 5.4: Implementation of Robot in large scale

The figure illustrates two advanced quadruped robots deployed in large-scale environmental settings, showcasing the real-world implementation phase of the project **"A Real-Time Design and Implementation of Walking Quadruped Robot for Environmental Monitoring."** These robots exemplify the transition from prototype to field-ready platforms capable of traversing diverse terrains such as paved surfaces and natural landscapes. Equipped with robust and adaptive leg mechanisms, these quadrupeds are designed for stable locomotion in uneven and remote environments, making them ideal for tasks like climate data collection, disaster zone inspection, and surveillance in areas inaccessible to wheeled robots.

On the left, the yellow robot—resembling Boston Dynamics' Spot—demonstrates high mobility on urban surfaces, while the robot on the right, with a rugged, tactical design and enhanced sensor and antenna systems, is shown operating on a grassy field, suggesting its use in more rural or undeveloped locations. Both variants highlight the scalability and adaptability of the quadruped platform for real-time environmental data acquisition. These robots are typically integrated with sensors for temperature, humidity, gas detection, and visual monitoring, transmitting data wirelessly to remote stations for analysis.

This real-world deployment signifies the practical success of the quadruped robot design in performing autonomous or semi-autonomous monitoring missions, where durability, mobility, and environmental awareness are critical. It emphasizes how such robots can support large-scale environmental monitoring applications, such as precision agriculture, forest health assessment, and hazardous site inspection, with minimal human intervention.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

Comprehensively, four legged robots use articulated limbs such as leg mechanisms, to provide locomotion. They are more versatile than wheeled robots and can traverse many different terrains, though these advantages require increased complexity and power consumption. This four-legged robot is an affordable solution for many applications. An important feature is that it has low-cost maintenance requirements and replacement of a component doesn't affect its performance. It has extensive applications compared to wheeled robots, ranging from military to industrial applications. Four legged robots have the advantage of being statically stable when not moving, but require dynamic walking control. There are many different ways for a four-legged robot to walk including alternating pairs and opposite pairs as in six legged robots. However, these techniques now cease to be statically stable and thus require dynamic control.

The real-time design and implementation of a walking quadruped robot for environmental monitoring represents a significant step forward in the integration of robotics and environmental science. This project demonstrates that biologically inspired legged robots can effectively navigate and operate in terrains where traditional wheeled or tracked systems struggle—such as forests, hills, wetlands, disaster sites, and rugged agricultural fields.

Through careful mechanical design, real-time control algorithms, and sensor integration, the quadruped robot developed in this project is capable of autonomously or semi-autonomously collecting critical environmental data such as temperature, humidity, air quality, gas concentrations, and visual feedback from cameras. With the help of embedded systems like Arduino or Raspberry Pi and wireless technologies like Wi-Fi or LoRa, the robot successfully transmits real-time data to remote stations, enabling informed decision-making, early warning, and efficient resource management.

This project also highlights the potential of interdisciplinary technologies—combining robotics, embedded systems, environmental monitoring, and wireless communication—to create intelligent systems that support sustainability and safety in the real world. The robot not only reduces the risk to human life in hazardous areas but also enhances the accuracy, efficiency, and frequency of environmental assessments.

In conclusion, the walking quadruped robot proves to be a versatile, resilient, and innovative solution for modern environmental challenges. With further developments such as AI-based terrain learning, solar-powered autonomy, swarm coordination, and cloud-based analytics, this technology can scale to even broader applications including disaster relief, smart agriculture, wildlife conservation, and urban environmental intelligence.

FUTURE SCOPE

SLAM: Simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of a robot's location within it. In the future this concept can be combined with our robot.

The successful design and implementation of a walking quadruped robot for environmental monitoring opens a vast array of opportunities for further research, development, and real-world deployment. As technology continues to evolve, the capabilities of such robotic systems can be significantly enhanced to meet more complex and large-scale environmental challenges. Below are some key future scope areas:

1. Integration with AI and Machine Learning

- Terrain Adaptation: Future versions can use AI models to detect terrain types (sand, rock, grass, snow) and automatically adjust gait patterns for optimal performance.
- Autonomous Decision Making: Machine learning algorithms can help the robot decide when to collect specific types of data, where to move next, and how to avoid obstacles more intelligently.
- Predictive Monitoring: By analyzing collected environmental data, the robot could predict changes like temperature spikes, gas leaks, or wildfire risks.

2. Solar-Powered and Energy-Efficient Operation

- Integration of solar panels or hybrid power systems can extend operational time, especially in remote or sunny environments.
- Future designs may also optimize motor efficiency and power management, allowing for longer deployment durations with minimal recharging needs.

3. IoT and Cloud Integration

- Full IoT-based ecosystem with cloud storage and analytics can allow for remote access to environmental data dashboards.
- Cloud-based AI systems can process data in real-time to provide alerts, trend analysis, and automated reporting to relevant authorities or users.

4. Swarm Robotics for Large-Scale Monitoring
    - Multiple quadruped robots can operate as a coordinated swarm to monitor large geographical areas in parallel, communicating and collaborating in real-time.
    - Swarm intelligence can help divide tasks, optimize routes, and share sensor data for improved environmental insight.
5. GPS-Based Path Planning and Autonomous Navigation
    - Enhanced GPS and SLAM (Simultaneous Localization and Mapping) systems can be used for real-time mapping of unknown environments.
    - Robots can be programmed to follow predefined routes or dynamically plan optimal paths based on environmental conditions.
6. Enhanced Communication Technologies
    - Implementation of 5G networks and long-range radio (LoRa) systems can improve data transmission speeds and connectivity in remote or obstructed locations.
    - Integration with satellite communication systems can extend robot usability to high-altitude regions and disaster-prone rural zones.
7. Multi-Domain Environmental Monitoring
    - Future robots can be equipped with more specialized sensors for:
        - Water quality (pH, turbidity sensors)
        - Radiation detection
        - Soil nutrient analysis
        - Noise pollution mapping

REFERENCES

.

[1] Yasuhiro Fukuoka, Yasushi Habu & Takahiro Fukui - "A Simple Rule for Quadruped Gait Generation Determined by Leg Loading Feedback: A Modelling Study ", 2015

[2] Putrus Sutyasadi and Manukid Parnichkun - "Gait Tracking Control of Quadruped Robot Using Differential Evolution Based Structure Specified Mixed Sensitivity H Robust Control", 2016

[3] Dr. Ing John Nassour - "Forward Kinematics Serial Link Manipulators", Chemnitz University of Technology, Germany, 2017

[4] Z. Bhatti, A. Waqas, A.W Mahesar, M. Karbasi - "Gait Analysis and Biomechanics of Quadruped Motion for Procedural Animation and Robotic Simulation", 2017

[5] Xuanqi Zeng, Songyuan Zhang, Hongji Zhang, Xu Li, Haitao Zhou and Yili Fu - "Leg Trajectory for Quadruped Robots with High-Speed Trot Gait", Harbin Institute of Technology, Harbin 150001, China, 2019

[6] Biswal,P.,&Mohanty,P.K.(2020) *Development of Quadruped Walking Robots: A Review. Ain Shams Engineering Journal*, (2020).Reviews design trends, sensory integration, and control techniques in quadruped robots, highlighting environmental and industrial monitoring use cases link.springer.com+2researchgate.net+2ivysci.com+2

[7] Jang, Y. et al. (2021) *Development of Quadruped Robot for Inspection of Underground Pipelines in Nuclear Power Plants. Electronics Letters*, 58(6):234-236.
Deploys a quadruped robot with a robotic arm and hybrid gait controllers for real-time inspection tasks in hazardous subterranean settings
ivysci.com+2researchgate.net+2link.springer.com+2ietresearch.onlinelibrary.wiley.com

[8] Halder, S., Afsari, K., Serdakowski, J., DeVito, S., Ensafi, M., & Thabet, W. (2022) *Real-Time and Remote Construction Progress Monitoring with a Quadruped Robot Using Augmented Reality*. *Buildings*, 12(11):2027.
Uses a quadruped robot equipped with sensors and AR overlays for real-time surveying and monitoring in dynamic environments reddit.com+9mdpi.com+9reddit.com+9

[9] Priyaranjan Biswal et al. (2023–2024)*Design and Fabrication of Quadruped Robot. International Journal of Engineering Research & Technology (IJERT)*, 13(2), Feb 2024.
Details mechanical fabrication and sensor integration (e.g., Raspberry Pi, Arduino, ROS) for quadruped surveillance and environmental data acquisition arxiv.org+11ijert.org+11researchgate.net+11.

[10]**Current Commercial Deployments (2023–2025):ANYbotics' ANYmal**: Industrial quadruped used for chemical, mining, oil & gas site inspections en.wikipedia.org.

Unitree Robotics' Go1 / Go2: High-speed, terrain-adaptive quadrupeds applied in inspection tasks; publicly demonstrated since 2021, with new models in CES 2025