Automation of Ultrasonic-Based Product Tracking & Traceability in Supply Chains

A thesis submitted to the Department of Electrical Engineering, UNIVERSITY OF CAPE TOWN, in partial fulfilment of the requirements for the degree of

Master of Engineering

at the

University of Cape Town

By

Anorld Mkombwe

Supervised by:

A/Prof. SIMON WINBERG

2025

Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for the properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module, and I confirm that my supervisor has seen my report. Any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

Signature of Author: .... Anorld Mkombwe

University of Cape Town

Cape Town South Africa

ABSTRACT

In the sweeping tide of digital evolution, the Internet of Things (IoT) is emerging as a significant catalyst, spearheading a colossal upsurge in deploying various sensors as the Industrial 4.0 buzzword continues dominating all platforms where industry captains converge. Modern-day Supply Chain Management (SCM), product tracking, and traceability are paramount for ensuring efficiency, quality control, and regulatory compliance. At the heart of IoT are wireless sensors and various other sensors forming the ecosystem of technologies that interact. This dissertation explores optimizing ultrasonic-based systems as wireless sensors for tracking and traceability in SCM and logistics. While widely used, traditional radio-frequency identification (RFID) tags and barcode systems encounter limitations in environments with interference or when tracking through dense materials. Ultrasonic technology, with its ability to penetrate various media and provide high-resolution data, presents a promising alternative for environments where other tracking technologies underperform. It can thus be used with near-field communication (NFC) technology and real-time GPS tracking, traditionally reserved for tracking goods in transit.

The research investigates the unique attributes of ultrasonic signals for product identification, focusing on frequency modulation, signal processing techniques, and integration with existing digital frameworks. The study utilizes simulation models and field testing to examine the reliability, accuracy, and cost-effectiveness of ultrasonic-based tracking across various supply chain stages. Innovative algorithms and device design modifications address key challenges, such as signal attenuation, environmental noise, and energy consumption, enhancing signal clarity and data retrieval efficiency. This project primarily focused on the costs and benefits of using a simple ultrasonic sensor for product tracing and tracking. A cheap ultrasonic sensor was used and connected to an Arduino device and an affordable buzzer and LED for audible and visual alerts to anyone nearby. These simple, standard, low-cost devices were programmed with open-source code and libraries downloadable in Arduino IDE to achieve comparable results that previously cost an arm and a leg with current technologies such as Blockchain (BCT), global positioning system (GPS), or RFID. The sensor was programmed to sense objects passing through at a certain distance and then increment a count that displayed real- time results locally and on a centralized cloud platform. This enabled the results to be monitored and queried in any part of the world where there is internet connectivity.

These methods in SCM have been quite expensive to set up and maintain, thus prompting the need for an IoT-based system with low-cost input but reliable performance to achieve the purpose. This project also aims to provide a solution for automatically tracking and tracing goods without human involvement before goods are packaged for transportation, where GPS tracking is ineffective.

Results demonstrate that automated ultrasonic tracking can improve product traceability, particularly in complex industrial environments where traditional methods struggle. By incorporating ultrasonic technology, supply chains benefit from enhanced visibility, which supports real-time inventory management, reduces errors, and increases responsiveness to disruptions. This dissertation concludes with recommendations for implementing ultrasonic systems in conjunction with existing technologies such as Blockchain technology, RFID scanning and tagging systems, and other IoT-based infrastructure, offering a heterogeneous approach that maximizes the strengths of each technology to create a robust, scalable solution for modern supply chains.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Associate Professor Simon Winberg, for his unwavering support. His guidance and profound advice from the beginning to the end were invaluable in completing this thesis. I could not have completed my research work without his patience and consistency, even during the module registration season.

I also thank the Department of Electrical Engineering at the Faculty of EBE, University of Cape Town, for providing essential technical support, specifically to the faculty administrator, Nicole Moodley. Her work and unwavering support deserve recognition and appreciation.

Additionally, I would like to extend my gratitude to my wife and children for the emotional support and encouragement that helped me complete this program, which at one time I was tempted to abandon due to financial and health challenges that were ravaging our family.

Lastly, I would like to thank my colleagues, who consistently provided guidance and suggestions that were pivotal to the completion of this Master's thesis.

TABLE OF CONTENTS

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analogue to Digital Converter
AI	Artificial Intelligence
BCT	Blockchain Technology
DIP	Dual In-line Package
EMW	Electro Magnetic Wave
FSPL	Free Space Path Loss
GIS	Geographic Information System
GPS	Global Positioning System
GSM	Global Systems for Mobile Communication
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IIC/I2C	Inter-Integrated Circuit protocol
IoT	Internet of Things
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LoRaWAN	Long Range Wide Area Network
LTE	Long Term Evolution
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
PWM	Pulse Width Modulation
RF	Radio Frequency
RFID	Radio Frequency Identifier
SCM	Supply Chain Management
UART	Universal Asynchronous Receiver Transmitter
Wi-Fi	Wireless Fidelity

NOMENCLATURE

Analog to Digital Converter (ADC): An ADC is an electronic integrated circuit that converts data from its analogue form to a digital form of ones and zeros.

Digital to Analog Converter (DAC): This electronic circuit converts digital data into 1s and 0s into analogue form.

Free Space Path Loss (FSPL): This is the degradation or attenuation of a radio signal between the transmitter and the receiver.

Global Systems for Mobile Communications (GSM): A second-generation (2G) mobile communication standard developed in Europe by the European Telecommunications Standards Institute (ETSI) to govern protocols and standards used.

Hypertext Transfer Protocol (HTTP): A set of rules governing how web browsers and servers communicate online. It is a fundamental part of the internet that allows users to access websites and web applications.

Inter-Integrated Circuit (I2C): protocol is a serial communication method that uses two wires to connect devices on a bus. It's commonly used to connect microcontrollers to sensors and other ICs

Internet of Things (IoT): A network of "things" or physical objects embedded with sensors, various technologies and software to connect and exchange data over the Internet.

Long-term evolution (LTE) is a mobile wireless evolution technology for fourth-generation (4 G) networks that came after 3G but before 5 G. Due to its purely packet-switched communication protocols, LTE is often deemed an all-IP network.

Long-Range Wide Area Network (LoRa WAN): This is a low-power, wide-area networking (LPWAN) protocol for connecting devices to the Internet. It's used to connect battery-powered devices like sensors to networks.

Message Queuing Telemetry Transport (MQTT): a lightweight, publish-subscribe messaging protocol commonly used on the Internet of Things (IoT) to enable efficient communication between devices with limited bandwidth and processing power, like sensors and actuators, by sending data through a central broker based on topic subscription.

Near-field communication (NFC): This technology allows devices to exchange information wirelessly when they're close together. It is embedded in smartphones, enabling users to tap their phones when purchasing.

Universal Asynchronous Receiver Transmitter (UART): This hardware protocol allows devices to exchange data serially using two wires. It is commonly used in microcontrollers, embedded systems, and computers.

Wireless Fidelity (Wi-Fi): A wireless standard developed by IEEE for devices to communicate within a Local Area Network (LAN) using radio waves or wirelessly. It is also regarded as the IEEE802.11x standard.

CHAPTER 1: INTRODUCTION

The proliferation of online shopping applications continues to skyrocket, resulting in a colossal upsurge of the need to quadruple the supply chain efficiency. The Internet of Things (IoT) continues to catapult the exponential growth of demand for goods and services. IoT-based product tracking and tracing systems are becoming increasingly popular. They can automate and digitize supply chain processes to maximize operational efficiencies, saving companies millions of dollars or helping evade operational human errors [2, 3]. The current Supply Chain Management (SCM) systems are slow and prone to interference, especially inside warehouses, which is the main component linking chain partners. This paper aims to improve SCM inefficiencies by introducing IoT sensors, such as the Ultrasonic sensor, to help optimize efficiency in SCM. It also looks at the building blocks that make a simple indoor system that can completely eradicate the need for the human element inside the warehouse. It then discusses the potential application of the research work in emerging markets. The paper focuses more on the project's ability to upload results onto the cloud, allowing stakeholders to monitor goods count remotely in real-time. It concludes by comparing existing SCM-enabling technologies and gives a convincing conclusion as to why companies must begin to shift their focus and consider IoT-based tracking systems as pivotal to the growth of SCM tracking systems, in the context of Industry 4.0.

Trillion-dollar companies like Amazon Services were among the pioneers, competing with other brands like Alibaba, AliExpress, eBay, and Walmart, to mention a few. New players in the industry like Shein, Takealot, Wish, and recently Temu have been among the few in supply chain management (SCM), which has been the buzzword in industry and commerce. Walmart was the first company to use 100% bar codes on all its products back in 1983 and expected its suppliers to adopt the practice [4]. This shows how critically important it is to use enabling technologies such as RFID, Barcode systems, GPS, Blockchain, and, recently, IoT-based systems to enhance product tracking and traceability.

The production of goods and services to satisfy an insatiable appetite from a consumer base of ever-evolving trends has forced most of these suppliers to design systems to meet the customers' needs and enable real-time monitoring and enumeration of goods passing through a monitored point. Huge employment figures have been created as individuals latch onto established brands to get a fair share of their cake. Online advertisements about how individuals can market products on behalf of these giants and retain a profit continue to attract much attention from today's tech-savvy generation. Drop shipping is among the many platforms used to aid SCM and ensure goods are distributed to the final end-user or consumer of products and services. Traditionally, stock take is performed manually by having a group of employees physically count each item in an inventory system or location or by using digital tools, such as inventory management software and barcode scanners [8][9]. The purpose of stock-taking is to ensure perfect accuracy in the organization's recorded stock levels within the inventory system.

With all these challenges continuing to wreak havoc on the SCM ecosystem, IoT-based systems have the potential to resolve them. The IoT architecture consists of three fundamental layers: Perception, Network, and Application.

- **The Perception Layer** is the physical layer that contains sensors or IoT devices used to gather data needed in the IoT chain.

- **The Network Layer** sends data and connects to other devices or servers. Transmission and processing of the data from sensors is also done at the network layer.

- **The Application Layer** provides application-specific services to the users via user interfaces.

This thesis focuses on the Perception Layer of the IoT architecture, utilizing the ultrasonic sensor to help capture data that the above layers will use to track and account for products in the SCM process. All the sensors within the IoT spectrum operate at the Perception Layer, and their job is to acquire information as inputs to the upper layers. Sensors and actuators collect data or execute final commands from the upper layers, interacting with the final user for decision-making.

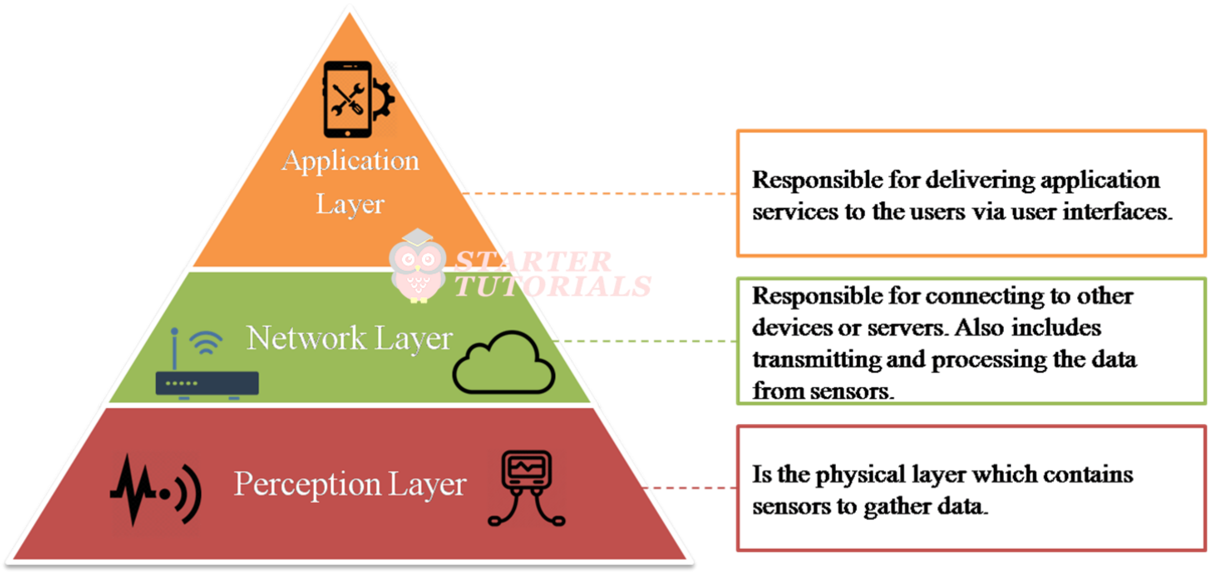


Fig 1. 1: IoT Architecture. (Image Source: https://www.startertutorials.com/blog/iot-architecture-layers.html)

1.1 BACKGROUND

Several ongoing studies in SCM aim to enhance tracking processes and ensure the delivery of quality products to end users or final consumers. One such study that has gained significant traction is the use of Blockchain Technology. Blockchain continues to gain momentum as it involves distributed tamperproof vaults or ledgers that can be accessed by all involved stakeholders simultaneously. The ledgers relate to each one containing an exact copy of the others. This means that a product a distribution company supplies remains unaltered throughout the supply chain until it reaches its destination. Any changes that may be added to the distributed ledgers are agreed upon, signed, and resealed again by all parties involved. The data is, therefore, not so easy to alter as a hacker will be required to access all ledgers within the chain to affect the changes [5, 6, 7]. If the data of one ledger changes, the hashing key and the other blocks within the chain will change. This makes Blockchain technology one of the most secure methods employed in the supply chain industry. RFID, which scans product codes, has been used to check inventory stock and track products along the chain. This method remains popular even though it may offer security loopholes regarding the traceability and trackability of goods. The second most contemporary technology in the supply chain industry is IoT-connected devices, with sensors at the front end collecting all the relevant data for use in various chain stages [9]. Against this backdrop, this project is motivated as it seeks to contribute to the IoT ecosystem by using some sensors to enhance product tracking and traceability. Unlike RFID, which uses a barcode system to capture a product's details from point A to B, an ultrasonic sensor performs physical counting as goods move along the conveyor belt with little or no human intervention. This project is thus premised on the fact that there is not much in the SCM field currently as research continues to roll out to establish the best possible scale from small enterprise companies to large corporate firms. Blockchain technology is expensive for many companies as it involves state-of-the-art encryption algorithms that small startups cannot afford. However, tracking goods using a simple sensor like an ultrasonic/ultrasound sensor or Infrared (IR) sensor can achieve more outstanding results at a low cost. The Arduino was created to simplify electronics in embedded systems, project piloting, and "domotics" (a term derived from Latin word "domus" meaning home; used in home automation when electronics, electrotechnics, and information technology are combined). It has programmable inputs and outputs, allowing control of different components and actuators, including distance sensors, push buttons, DC motors, remote controls, servomotors, photoresistors, etc.

1.2 PROBLEM DESCRIPTION

The research problem aims to automate product traceability and tracking efficiency, thus reducing the inevitable inefficiencies and errors when done manually. The supply chain process is complicated and requires many resources to achieve the bare minimum. Though much research has gone into the area of automation of the SCM processes, a lot still needs to be done to continue improving the systems and guarantee a robust, tamper-proof, and efficient system. Many inefficiencies and loopholes continue to haemorrhage various sections of the chain [10]. Companies and organizations continue to invest a fortune to move raw materials to the production centre and the finished products to the final consumer. By automating some of the phases within the chain and creating a centralized monitoring system, operational costs (OPEX) can be drastically reduced, giving companies an accurate perception of the operating ecosystem.
According to an article by Deloitte, smart sensors convert the physical world into digital insights that generate new value throughout the supply chain. By providing managers with real-time information on inventory, machinery, and materials, these sensors enhance visibility across the supply chain and enable analytics that help predict demand, optimize sourcing, and support high-value manufacturing decisions. The deployment of smart sensors has immensely aided in lowering the cost of production and increasing operational efficiency.

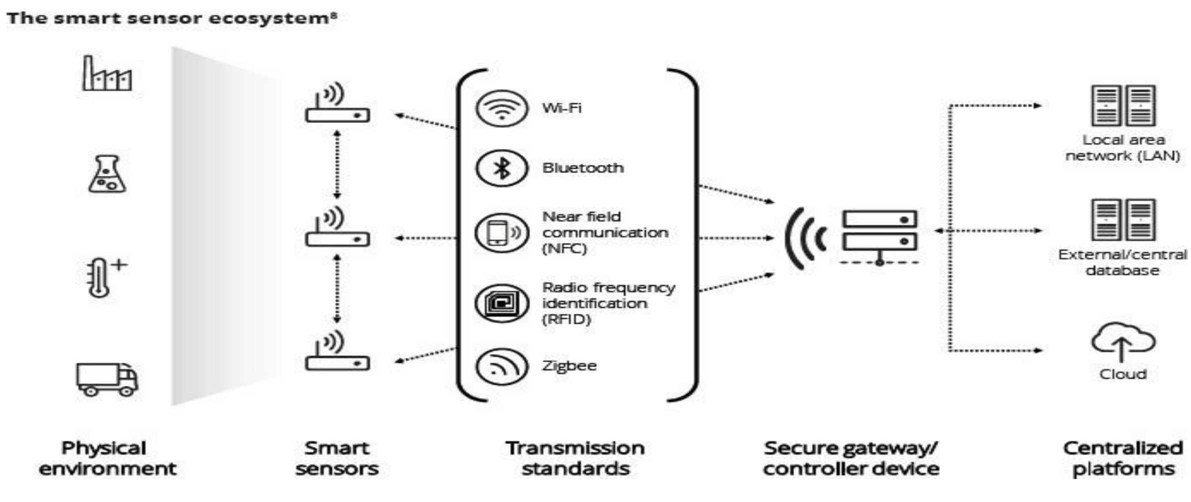


Fig 1. 2: Illustration of a Smart Sensor Ecosystem.(Image Source:[6])

In Fig 1.2 above, the **physical environment** resembles an industrial setup where raw materials arrive from the primary factories. After the raw materials are processed into finished goods and packed in different package sizes, the goods pass through a checkpoint where they must be counted before being loaded into a delivery van or truck on their journey to retail outlets. It is at the counting phase where the ultrasonic **smart sensors**, interlaced with microcontrollers, come into play to help captains of industries check the number of goods manufactured per production cycle. To monitor these figures remotely, a **transmission standard** (ESP32 shield) such as Wi-Fi, RFID, and NFC will be used to send the data over a secure network using protocols such as SSH or any public data encryption technique to a centralized platform accessible by different stakeholders.
Real-time inventory tracking enhanced by smart sensors is a plus in today's business operations. Tesla is one big company that uses smart sensors for remote diagnostics and any required software upgrades for its various services, including cars. This has had a plus in Quality of Experience (QoE) with its ever-growing clientele. The communication company Ericson has also not been left behind in this regard, as they use smart sensors to track the vessel position and cargo containers in real time [11]. As more and more companies come on board to reap the benefits of using smart sensors, more expertise will be demanded in that regard, and more investment will have to be made in the production and design of more accurate miniature sensors that will intertwine all ecosystems with little to no human intervention. This is a lucrative moment for companies to assess their SCM to pilot sensors and immensely reduce the cost of production and operations. New revenue opportunities can be realized by having an eagle's eye on all phases of the supply chain, using smart sensors, and driving efficiencies while reducing costs.

1.3 FOCUS

This project focuses on designing a sensor-based system to optimize the supply chain. The goal is to use an ultrasonic sensor connected to an Arduino board (microcontroller) to help with product traceability and trackability. Traditionally, product traceability and counting have always been carried out with the involvement of people, but can be replaced by an automated system with unmatched accuracy and business performance enhancement.

1.3.1 HYPOTHESES

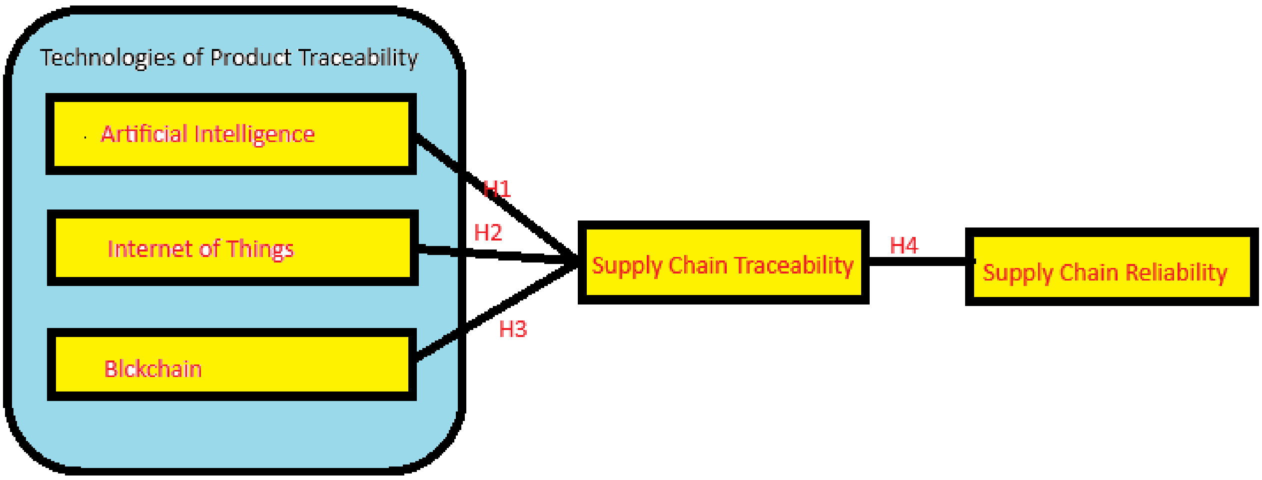


Fig. 3: Hypothesis [3]

H1 - "The integration of ultrasonic sensors in the supply chain process improves the efficiency and accuracy of tracking goods by enhancing real-time inventory management and reducing instances of misplaced or unaccounted-for items." This hypothesis can be tested by comparing the performance of a supply chain system with and without ultrasonic sensors in terms of metrics such as tracking accuracy, inventory discrepancies, and operational efficiency.
The main objective of this project is to design a monitoring system based on the Arduino Uno R3 microcontroller interfaced with an ultrasonic sensor that enables automated counting of goods in the supply chain process. By embedding smart sensors into the supply chain process, companies have a holistic overview and end-to-end visibility into suppliers, distributors, and consumers. Using sensors to update inventory will avoid overstocking and understocking goods. The main goal of this project is to prove that an environment without sensors to automate repetitive processes in the supply chain is likely to fail due to human inefficiencies and limitations. Sensors do twice the work humans do without any room for error, which may cost companies millions of dollars.

H2 - "Will a completely automated tracking system be more efficient than a system managed and controlled by people?" This hypothesis remains to be proved by results to see if the system can efficiently enumerate the goods, thus reducing human error and possible theft of goods during counting. Human errors have always been witnessed across all facets of business processes. Human beings are prone to mistakes, and just as the popular saying goes, "To err is human." Eliminating the human element since the Industrial Revolution, which had roots in Britain in the mid-18th Century before spreading throughout the world, proved this hypothesis.

H3 - "Can IoT-based product tracking and traceability outshine other technologies such as Blockchain technology (BCT), RFID bar codes, GPS, or Artificial Intelligence (AI)?"
BCT has undoubtedly been one of the most reliable product tracing technologies employed in the SCM. It is of critical importance to note that technologies such as BCT, GPS Tracking, and the use of RFID. However, they remain popular and are expensive, and most of them will be replaced by the coming of IoT-based sensor ecosystems. These technologies may not be replaced entirely but will work hand-in-glove with IoT-based tracking systems, complementing each other. The sensor frenzy is, without a doubt, a huge step towards an amalgamated ecosystem that will traverse the supply chain industry, creating a robust system that can do a hundred times the work at a lesser cost.

1.4 OBJECTIVES

This research project aims to prototype an IoT Ultrasonic-based tracking system that is intertwined with an Arduino Uno R3, an ESP32 microcontroller, an LED and a buzzer. This will be used as a reference point to execute a comparative analysis of how the system leverages and performs against traditional SCM technologies. Such technologies include GPS technology, Blockchain Technology (BCT), RFID, and Barcode Scanners. The comparison will be in terms of cost and their ability to perform and obtain the required goals. To achieve this objective, the below sub-objectives will guide this study:
Development of the Ultrasonic sensor-based system:

- Use an Ultrasonic sensor to capture ultrasonic sounds in the spectrum.
- Develop a C++ code to interface an Ultrasonic sensor with an Arduino Uno R3, an LCD, a buzzer and an LED.
- Blueprint and execute an Arduino UNO block for an SCM system that will interface with an Ultrasonic sensor and other peripherals or output devices, including an LCD, a buzzer, and an LED.
- Execute tests of the Ultrasonic sensor-based SCM system's ability to deliver counting objects on a conveyor belt and post the result to a centralized database, in this case, a cloud platform.

Evaluation of IoT-based tracking system against traditional mature technologies:

- Implement object or product tracking for objects passing through a designated point on the conveyor machine.
- Record and evaluate captured results to analyze how the system compares in terms of product tracking and traceability, and the cost involved in developing such a simple but effective system.
- Comparing the results obtained against a sophisticated existing system within the SCM chain and giving recommendations.

1.5 SCOPE AND LIMITATIONS

The limitation of this project was a budgetary issue, as building a project of such a magnitude may entail incorporating many sensors that will require much money. I avoided incorporating many sensors as this would also recalibrate the trajectory and topic of the research, as other sensors do not operate in the ultrasound frequency band of above 20KHz to 1000KHz, for instance. The research topic would also not be retained as it is, since many sensors will be incorporated. It would only make sense to entitle the topic "**Automation of Sensor-Based Tracking and Traceability in the Supply Chain**." The fact that this topic entails me zeroing in more on the ultrasonic sensor as the centre of the research meant that the spectrum of my review became relatively narrower. The time factor also became an issue when trying to unite diverse areas of interest to build a formidable topic and create an industrially relevant solution. Ultrasonic devices operate in the 20 kHz to 1 MHz range, as mentioned above, and it is assumed that, with sufficient intensity, these frequencies may be traumatic and detrimental to hearing. These ethical issues also reduce the extent to which one can experiment with the sensor, as this may come with health hazards to people and animals. These moral issues may be felt in the long run, such as hearing loss, which may occur after some time. Other effects may be immediate, such as nausea, headaches, dizziness, tinnitus, fatigue, etc. Even though ultrasound is inaudible, protracted exposure may cause hearing loss [12].

The use of ESP32 together with Arduino instead of just using the ESP32 shield alone has been limited by the following:

- The ESP Wroom-32 does not come with a 5V pin, making it difficult to power peripheral devices like the ultrasonic sensor and the LCD unit.
- Using Arduino UNO R3 alone will not achieve the intended results of uploading the results for real-time monitoring on the cloud.

The recommendation is that another type of Arduino, such as the Arduino Uno Rev 4 Wi-Fi, Arduino Nano 33 IoT, or Arduino Mega, be used in future research based on this one, since they come equipped with Wi-Fi and Bluetooth capabilities. Alternatively, an ESP NodeMCU can be used since it has 5V power supply pins.

The other main limitation encountered was regarding double-counting of goods or the inability of the designed system to count different goods. It is important to note that under this assumption, the goods to be counted will be packaged in a similar packaging with the same boxing and size. To design a system with the ability to detect different goods and packaging, several IoT-based sensors will have to be incorporated into the project. To evade objects double-counting, adequate spacing should be ensured between the boxes moving along the conveyor belt, and also the back-off time in the code developed will have to be increased. The time it takes for the Cloud database to update will also have to be reduced, as the database may not be able to capture accurate figures in a free-flow environment.

1.6 METHODOLOGY OVERVIEW

This project will use the Rapid Application Development (RAD) methodology. This system has four main building blocks, namely:

- Requirements Planning – project requirements are defined and timelines, goals, budget restraints, features, user expectations
- User Design – through multiple iterations, the user design is built for real-time use and testing.
- Rapid Construction – the model prototype is developed quickly after major concerns and flaws are considered.
- Cutover – the product is launched after the model is completed [13].

The project requirements for the Ultrasonic sensor-based tracking system are defined at the onset using the RAD methodology. The second phase will develop a prototype confined to the agreed-upon resources. To make this prototype, an Arduino UNO R3 microcontroller will relate to an ESP Wroom 32 shield, and the Ultrasonic sensor will connect directly to the Arduino. Other output devices, including an LCD, a buzzer, and an LED, will also be connected to the Arduino. Jumper wires will be required for connections, and current-limiting resistors will also be needed for low-current components like LEDs and buzzers. The prototype will be connected to a computer (PC) by installing Arduino IDE software. The IDE uses C++ code written on a sketch before being uploaded to the Arduino board and the ESP32 board for operations. The IDE runs on the Microsoft Windows platform and is GUI-based. The proper board should be selected in the IDE, and COM ports should also be specified. To login to ESP32, special drivers must be downloaded and installed on the machine to interact with the shield. The relevant libraries, such as the <HTTP.h> After the final design is connected and the sketch is developed and loaded onto the shields, the prototype is ready for testing according to the 4 phases of the RAD cycle, as illustrated in Fig 1.3 below. A detailed overview of the RAD process will ensue in Chapter 3.

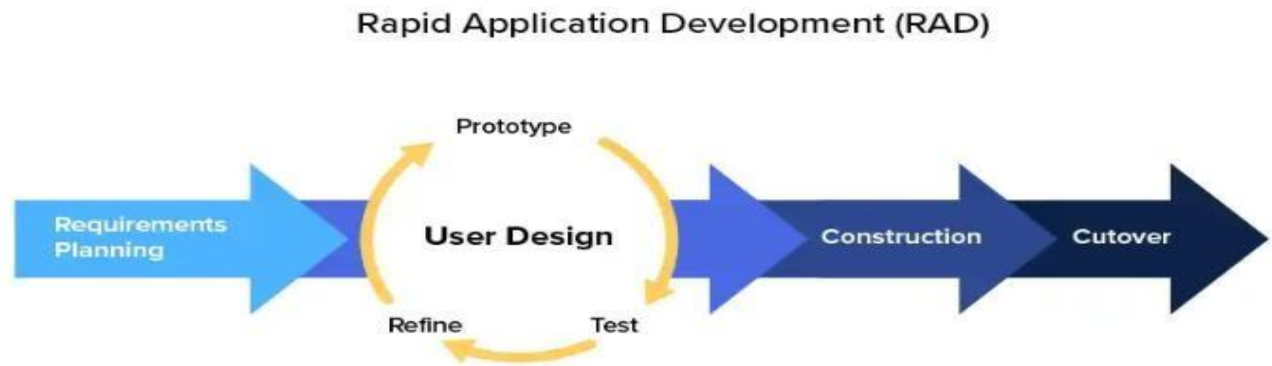


*Fig 1. 4: RAD Methodology Cycle – [*Source: https://watkyn.com/blog/rad-pros-cons/*]*

1.7 THESIS OVERVIEW

This proposed research report will be organized as follows, from the first chapter to the final chapter:
Chapter 1 - Introduces the environmental parameters used as inputs to the Arduino and explores the potential for intelligent monitoring. It also explores the various sub-sections detailing where the project might find relevance, offering practical solutions to the research problem.

Chapter 2 - provides a detailed project literature review, looking at works already done in the research area by other scholars and researchers. In conclusion, the project's future scope and prospects are also outlined, and project gaps are closed.

Chapter 3 - Presents the methodology of how the system will work, including the hardware and software requirements for this project. The details about the methods and materials, including the hardware and software components, and the schematic diagram showing how the sensors will be interlaced with the microcontroller to achieve the project's primary goals. Systems architecture and all schematics are also included in this chapter, and all design parameters are clearly outlined.

Chapter 4—This chapter discusses the step–by–step processes needed to design a working prototype. The design process is highlighted, detailing the intricate details of the project's functionality.

Chapter 5—The prototype of an Ultrasonic-based tracking system design is implemented. It elaborates further on a system operating on an Arduino microcontroller with an ultrasonic sensor as the input and other output devices displaying the results locally and on cloud.

Chapter 6—The results are collected and analyzed. The results will be the counted objects as the prototype is tested each time, and proof that it is economically viable to use the ultrasonic-based sensor to optimize product tracking and traceability in SCM. Results will also include any captured screenshots of recorded real-time objects on the monitored systems.

Chapter 7 - Outlines the conclusion and recommendations as per project expectations.

1.8 SIGNIFICANCE OF PROJECT

Miniaturization of devices and the introduction of micro-electro-mechanical systems (MEMS) have seen massive energy-efficient production of systems that fit in small spaces but achieve more significant results [14, 15, 16]. Traditional wireless networks designed for high bandwidth traffic patterns have greatly impeded smart-sensor communication. The introduction of power energy-efficient low-range wireless technologies such as ZigBee, Wi-Fi, LPWAN, and NFC has enabled the processing of data at the edge, popularly known as "edge computing," thus reducing the amount of data that may need to be sent to a central point for processing. These short-range wireless technologies offer more scalable and tailor-made solutions for different business use cases. This project is significant in that respect as it enables real-time monitoring and tracking of goods in the processes, enhancing efficiency in SCM. By leveraging smart sensors to enhance automated data collection and processing, companies can expand management visibility across the supply chain. This helps companies lower operating costs, boost asset efficiency, and create additional revenue streams.

1.9 PRACTICAL APPLICATIONS OF PROJECT

The anticipated results in this project are a pragmatic use case in industry and SCM. Currently, many companies are employing legacy methods to enumerate goods passing through a monitored point. This has introduced errors and inefficiencies in the process and has come at a high operational cost that threatens to derail companies and reduce revenues significantly. If this system is integrated with a cloud platform or any centralized monitoring system, efficiency in all sections of SCM will improve immensely. The Arduino can then be integrated with other sensors to measure environmental metrics such as temperature, humidity, moisture, or even the colour of things being counted by the ultrasonic sensor. Temperature and humidity will help monitor the goods' operational environment to ensure that no moisture is affecting goods that are moisture sensitive or to ensure there is enough moisture for any of the goods that are affected by the absence of it. A load cell sensor/strain gauge can be used with the ultrasonic sensor to weigh the goods passing through and discard those that do not measure up to the expected weight. Some of the practical applications of the project's expected results include:

- **Storage tank monitoring –** The application of ultrasonic level sensors in monitoring the levels of storage tanks or bins has been a game-changer in the supply chain world. Incidents of traffickers of liquid goods, accused of off-routing to a hiding place and siphoning some of the products, have dramatically tanked because of real-time monitoring using ultrasonic sensors. By transmitting and receiving sound waves above the range of human hearing, ultrasonic sensors have been used to give accurate measurements, leaving no room for any shenanigans during the trafficking of these products. The object being detected reflects some sound waves, and the distance from the sensor is determined by measuring the round-trip time it takes for the sound to return to the sensor.

- **Inventory tracking**—Inventory systems can be tracked in real-time using ultrasonic sensors, helping to reduce shrinkage or loss. Demand planning, therefore, completely avoids overstocking and out-of-stock situations.

- **Cargo monitoring -** Ultrasonic cargo sensors often help inform drivers if a trailer is empty or loaded. These sensors can also obtain information about how long it will take to load and offload a truck.

- **Navigation—**Ultrasonic sensors can be employed on ships to comply with international regulations for weather data collection, and the data obtained can help captains navigate storms.

- **Object detection—**Ultrasonic sensors can detect objects in varying conditions, such as heavy soiling, mist, and dust. They can also detect transparent plastic bowls, blister packages, and more.

- **Conveyor line monitoring—**Ultrasonic sensors can accurately count the number of goods or items passing down a conveyor machine. The items can be sorted and classified while the flow rate is monitored. This project aims to count the items and provide quantities as needed accurately.

- **Self-driving Vehicles**—Ultrasonic sensors interfaced with Arduino UNO/ESP32 can also determine the distance between a vehicle and a potential collision point. With little or no changes to the code, this project finds relevance in self-driving cars using device-to-device (D2D) or machine-to-machine (M2M) communication, as cars detect the proximity of other vehicles on the roads to avoid collisions. These cars also use ultrasonic sensors when self- parking to determine the distances around them nearest to obstacles.

- **Self-opening Mall Doors and People Enumeration** – another typical practical application of expected results is in the doorways at big malls' entrance points. This project will find massive applications there using an ultrasonic identifier that triggers a door to open once a person walking into or out of the mall approaches a particular range. Movement sensors are currently prevalent, but ultrasonic sensors, such as distance sensors, can open these doors automatically without human intervention. The sensors will then be used to enumerate the number of people walking into the mall and check at the end of the day if it matches the number that went out. This will help many property owners take pragmatic statistics of the number of people entering and exiting substantial shopping centers and may use this data to make decisions such as restocking requirements of goods on demand. They can also use the statistics in the event of a disaster to figure out how many people were in the mall and how many have been rescued. This information will help the rescue teams to work with accurate figures, such as authorities, who can tell the exact number of people on board an airplane in the event of a fatal disaster.

CHAPTER 2: LITERATURE REVIEW

The supporting literature upon which this project is established is discussed in detail. The concept of SCM is discussed, and various technologies are considered for the field. Several SCM systems have been developed based on GPS tracking, RFID, BCT systems, and contemporary methods premised on IoT designs.

2.1 OVERVIEW OF IOT-BASED SCM SYSTEM

The IoT ecosystem comprises various enabling technologies such as Blockchain Technology (BCT), Cloud Computing, Near-Field Communications (NFC), Smart Tags, QR Codes, RFID, and Artificial Intelligence, to mention a few. The IoT ecosystem has been revolutionized by the proliferation of these IoT devices, making it possible to monitor goods in real time from the production chain to the end consumer. Ultrasonic sensors are part of a wide range of wireless sensors used to gather data for processing to help oil the supply chain logistical processes. Ultrasonic sensors are not classified as transmission standards like Wi-Fi, Bluetooth, NFC, RFID, Zigbee, Long Range WAN (LoRa WAN), etc., but are part of large communities of smart tags being used as sensors. Smart sensors remain a priority as an emerging technology in the IoT ecosystem and will continue to play a pivotal role in SCM.

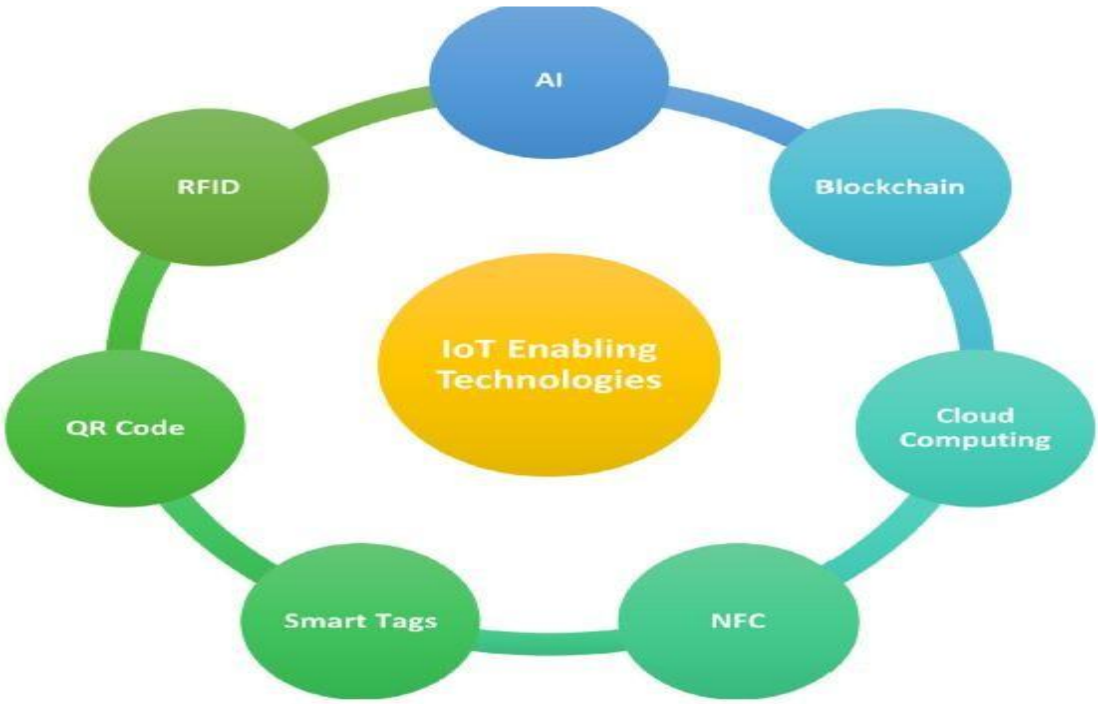


Fig 2. 1: The IoT Enabling Technologies (Image Source:[2])

2.1.1 DEFINITIONS

The Electromagnetic Wave (EMW) Spectrum has been the reference point for all wireless communication systems with frequencies ranging from Very Low Frequencies (VLF) to Extremely High Frequencies (EHF). The human ear can hear sounds in the acoustic frequency range between 20Hz and 20kHz. These sounds are classified as Audio frequencies or Audible frequencies. Frequencies below that, i.e., 0Hz to 20Hz, are classified as Infrasound, whereas frequencies above the Audio Frequencies (AF) are regarded as Ultrasound and range from 20kHz to 200MHz (20MHz). There is no theoretical upper limit for ultrasonic frequencies, though applications of this frequency band usually restrict it to 20MHz [17, 18]. Frequencies between ELF and VHF (3Hz to 300MHz) are called **Radio Frequencies**. Frequencies between UHF and EHF (300MHz to 300GHz) are called **Microwave Frequencies,** and frequencies between 3THz and 750THz are classified as **Infrared frequencies**. Those above 3THz are classified as **Ionizing Radiation** and find application in X-rays and Gamma Rays.

name	ITU[1] number	frequency	wavelength
extremely low frequency (ELF)*	1 (~10^{1} Hz)	3 - 30 Hz	100,000 - 10,000 km
super low frequency (SLF)*	2 (~10^{2} Hz)	30 - 300 Hz	10,000 - 1,000 km
ultra low frequency (ULF)*	3 (~10^{3} Hz)	300 - 3000 Hz	1000 - 100 km
very low frequency (VLF)	4 (~10^{4} Hz)	3 - 30 kHz	100 - 10 km
low frequency (LF)†	5 (~10^{5} Hz)	30 - 300 kHz	10 - 1 km
medium frequency (MF)†	6 (~10^{6} Hz)	300 - 3000 kHz	1000 - 100 m
high frequency (HF)†	7 (~10^{7} Hz)	3 - 30 MHz	100 - 10 m
very high frequency (VHF)	8 (~10^{8} Hz)	30 - 300 MHz	10 - 1 m
ultra high frequency (UHF)	9 (~10^{9} Hz)	300 - 3000 MHz	1000 - 100 mm
super high frequency (SHF)	10 (~10^{10} Hz)	3 - 30 GHz	100 - 10 mm
extremely high frequency (EHF)	11 (~10^{11} Hz)	30 - 300 GHz	10 - 1 mm
tremendously high frequency (THF)‡	12 (~10^{12} Hz)	0.3 - 3 THz	1 - 0.1 mm

Fig 2. 2: The EMW Spectrum (Image Source:[23])

2.1.2 WHAT ARE ULTRASONIC WAVES?

A sound wave is a vibration that travels through a medium like air, metal, or water. Ultrasonic waves are "sounds with frequencies too high for the human ear to hear," typically above 20 kHz. As defined above, the AF range is between 20Hz and 20KHz; within this range, the voice frequency ranges from 300Hz to 3400Hz. Sound waves not meant to be audible are also called ultrasonic waves. Frequencies below 20Hz (Infrasound) are audible to animals and sounds produced by earthquakes. The range 20Hz to 20KHz is the human hearing audio frequency range. The range 20KHz to around 1000KHz (1MHz) is the Ultrasound range and is audible to bats. Bats commonly use it for co-location and in medical diagnostics. Ultrasonic frequencies are very prevalent in medical diagnostics, especially during scanning the baby in its mother's womb from conception to birth.

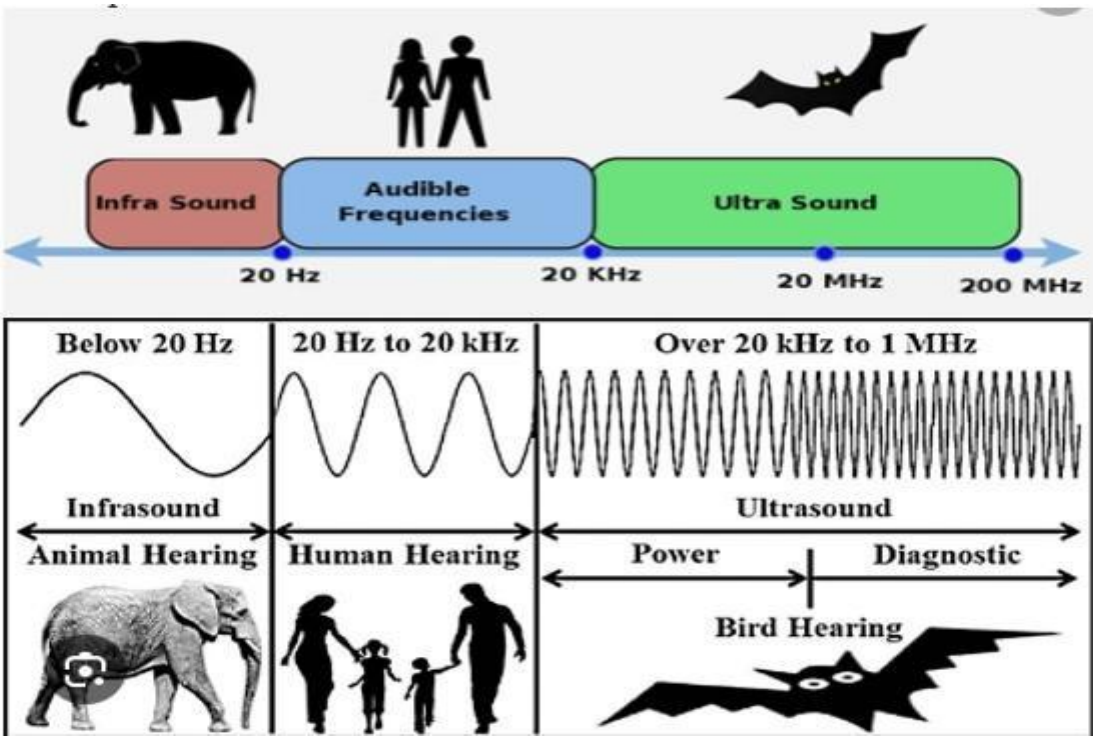
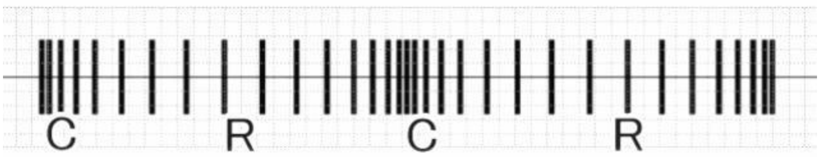


Fig 2. 3: Ultrasonic Sound Frequency Band (Image Source:[22])

2.1.3 ULTRASONIC WAVE TYPES

Three types of ultrasonic waves are available: Longitudinal, Transverse, and Surface waves. Two types of elastic waves coexist simultaneously in a solid. The Longitudinal or density wave has displacements in the same direction as the wave's propagation. In contrast, the Transverse or shear wave has displacements perpendicular to the wave's direction of travel, as illustrated below. The longitudinal wave is used in ultrasonic processing machines.

(a) Longitudinal wave



R： Rarefaction C： Compression

(b) Transverse wave

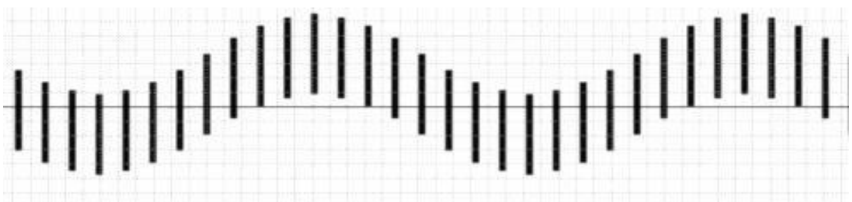


Fig 2. 4:Longitudinal and Transverse Waves https://www.sonotec.com/en/column/ultrasonic.html

2.1.4 WORKING PRINCIPLE OF ULTRASONIC SENSOR

Ultrasound works on the same principle as a wireless or microwave link. They are transducers that can generate or sense ultrasound energy and convert it into an electrical signal or ultrasound, depending on the category. They can be divided into three categories which are:

- **Transmitters** – convert electrical energy into ultrasound energy for transmission.
- **Receivers** – they receive ultrasound energy from space and convert it to electrical energy.
- **Transceivers** – they work as both transmitters and receivers.

Ultrasound sensors work on the principle of the **Piezo-electric effect** or **Capacitive effect**. The piezoelectric is used in speakers and microphones. When a charge is introduced to a quartz crystal substance, the particles rearrange, causing a membrane or diaphragm to vibrate quickly and producing an electromagnetic field (emf) in sympathy with the vibrations. In ultrasonic sensors, the vibrations produce an ultrasonic sound (Transmitter) or respond to the ultrasonic waves incident on the transducer (Receiver). These vibrations are then transduced into an equivalent electric signal interpreted as an analogue or digital signal.

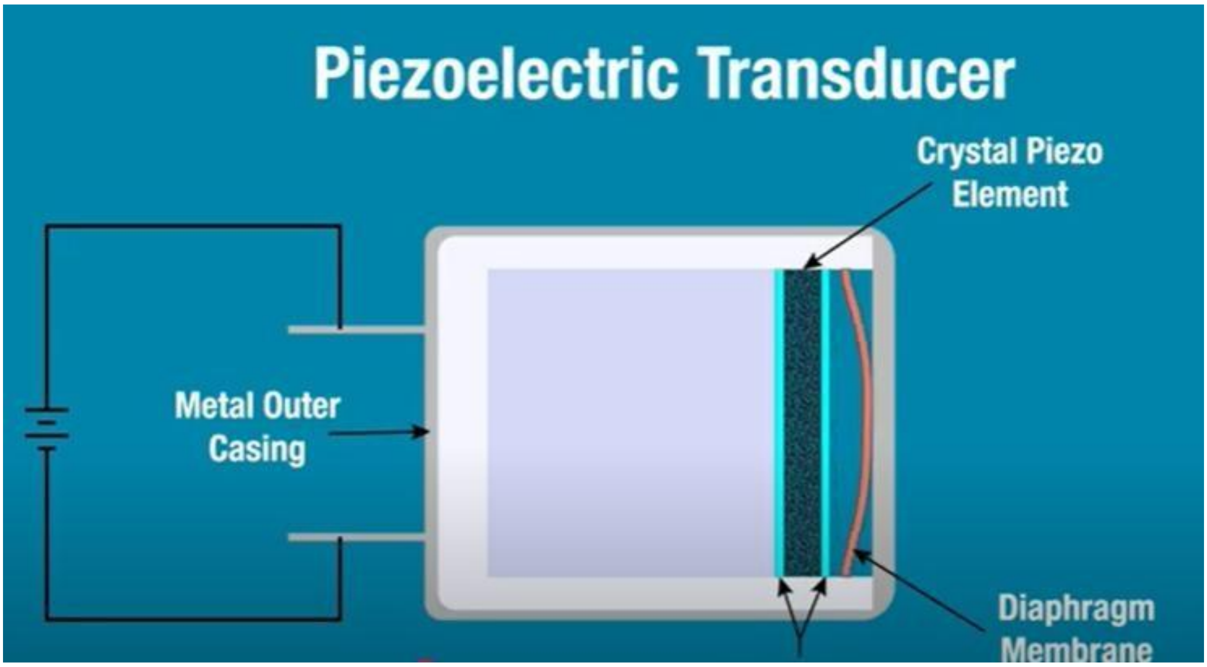


Fig 2. 5: Piezoelectric Transducer (Image Source: www.youtube.com/watch?v=2ojWO1QNprw)

The Capacitive transducer – also known as Capacitive Micromachined Ultrasonic Transducer (CMUT), uses capacitors with two conductive plates separated by a dielectric material. When connected to a power source, one plate gains electrons, becoming negatively charged, whilst the other loses electrons and thus becomes positively charged. In a capacitive transducer, one electrode (plate) sits on a dielectric membrane suspended above the cavity of the silicon substrate. The substrate itself will act as a second electrode. These electrodes become charged when a current is applied to the transducer, and this causes the upper plate to be attracted to or repelled from the lower plate. This moves the membrane like the Piezoelectric one, and alternating the current causes faster movements. These movements cause ultrasonic signals to be produced (Transmitter) as the membrane moves, acting as a diaphragm. The bouncing signal also causes the membrane to vibrate (Receiver); thus, ultrasonic sounds are converted to an equivalent electronic signal.

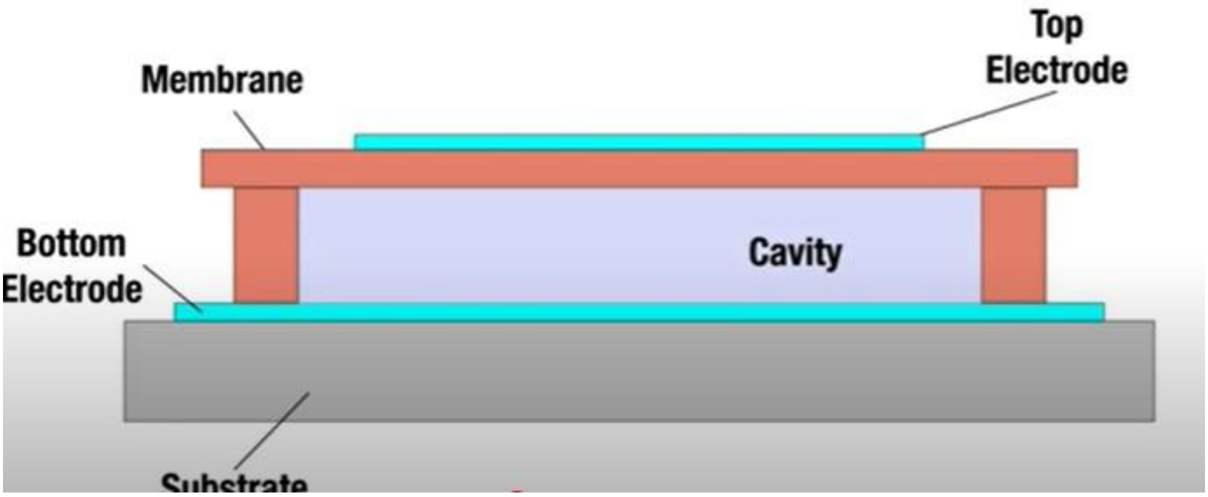


Fig 2. 6: Capacitive Transducer (Image Source: www.youtube.com/watch?v=2ojWO1QNprw)

A bat is a typically nocturnal animal that uses ultrasound during its movements. A dolphin uses the same principle of echolocation and travels in water to determine obstacles, danger, or prey around the deep waters. Echolocation uses sound waves to determine the position or location of objects. It uses sonar for echolocation to fly perfectly in dark caves and navigate without bouncing on obstacles. It sends a signal that bounces against the walls and ricochets an echo that helps the bat calculate the exact distance to an obstacle and then navigate smoothly past it. Like a bat, an ultrasonic sensor emits a wave or ultrasound via the transmitter, which travels in a pointed direction and bounces back upon encountering an obstacle or object. The signal bouncing from the object, as shown in the diagram below, will then be intercepted by the receiver side of the ultrasonic sensor. This sound is thus transduced into an electric signal, producing an analogue input signal to the microcontroller.

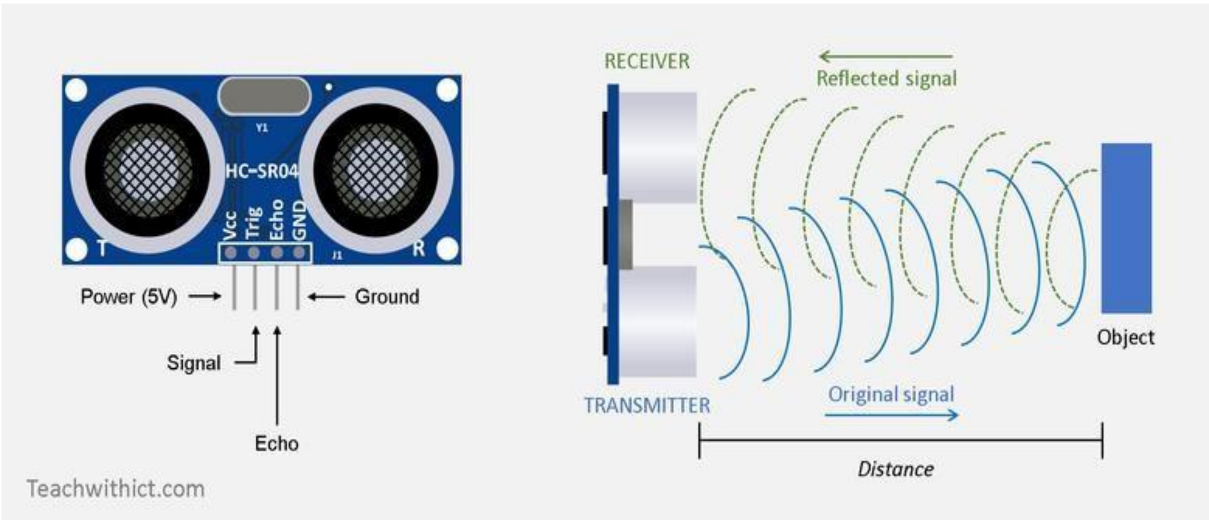


Fig 2. 7: Ultrasonic Sensor Principle (Image Source: teachwithict.com)

By this principle, an ultrasonic sensor works as a microwave transceiver that relies on a clear line-of-sight (LOS) for effective signal communication. An ultrasonic sensor is mainly used to measure the distance to an obstacle and, therefore, finds much relevance in self-driving cars and when parking vehicles to determine if a car is nearing obstacles from all sides. It sends an ultrasonic sound wave and calculates the time it takes to receive a reflection. This principle is known as the **time-of-flight principle**. The transmitter will signal behind a car and calculate the exact distance before the car's rear or sides bump against an obstacle. The formula for distance calculation is:

L = ½ x (T) x (C)……….. (i)

where L is the distance, T is the time it takes for an ultrasound to travel to the object and back to a sensor

and C is sonic speed or speed of sound = **1235km/h** = **343m/s** at sea level or **5331km/h** in water.
½ is for distance to and from the object; hence, we calculate distance once.

In this project, the ultrasonic sensor works on the same principle as microwave transceivers that lose communication when an obstacle comes in the way of the signal. The sensor will send a signal continuously until an impediment caused by goods moving along the conveyor belt comes in the way. This signal intercepting the signal at a certain distance will cause an interruption in signal propagation, which will then be interpreted as goods pass, resulting in increments on the counter. It is important to note that just as in a microwave link, a transmitted signal will not bounce back if the bouncing surface is placed at a significant distance from the sensor due to free space path loss (FSPL). This is important as the objects passing closer to the sensor will only cause a transmitted signal to bounce back at the same ultrasonic frequency and be received by the receiver side of the sensor. The range of an ultrasonic sensor is thus determined by the frequency of the transducer, which ranges from 5-15 feet (approx. 150 to 450 cm). The lower the frequency, the further it travels, and the higher it is, the shorter the range. The time delay it takes for a transmitted signal to bounce back and be received by the receiver side will be used to calculate the distance. Since the time is to and from, the result is divided by two (multiplied by half) to get the actual distance. The longer the delay, the longer the distance, and the shorter the delay, the shorter the distance to an object.
The HC-SR04 Ultrasonic sensor is equipped with four pins as follows:

- **V_{cc}** – 5V power connection.
- **TRIG –** Trigger pin (input).
- **ECHO –** Echo pin (output).
- **GND –** Ground

TRIG – Trigger pin sends a 5V 10us pulse.
HC-SR04 – Transmits 8 ultrasonic (40kHz) pulses.
ECHO – The echo pin outputs a pulse of 150us to 25ms.

The pulse width of the Echo pulse is used to calculate distance. If no object is detected, the Echo pulse will time out after 38ms (**Fig. 2.8**). When an object is detected, a 500ms pulse reply is sent from the object's surface to the receiver of the HC-SR04 (**Fig. 2.9**). To improve the distance of the Ultrasonic sensor, the <NewPing> library is added to the Arduino IDE.

```
#include <NewPing.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

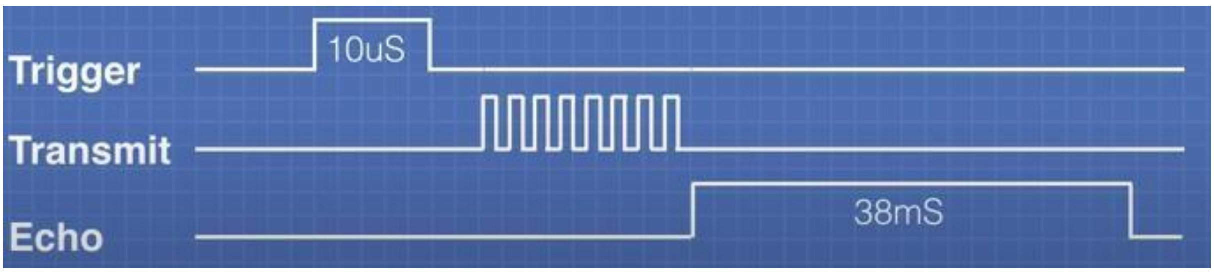


Fig 2. 8: No object detected.

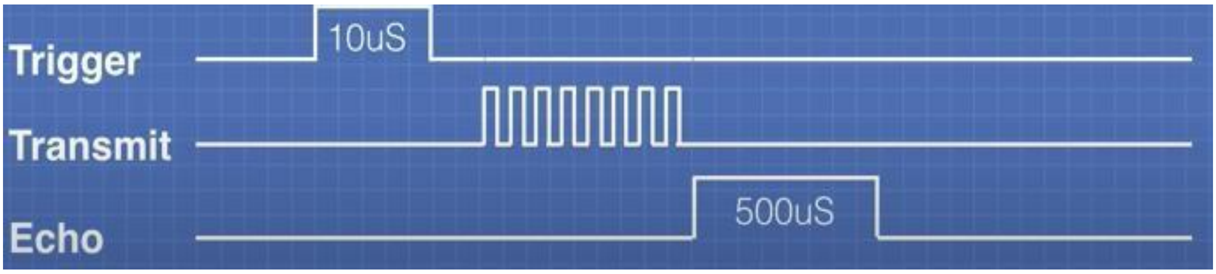


Fig 2. 9: Object detected.

The beam shape of the signal will differ depending on the model used. It is critical to note that an Ultrasonic sensor uses the same principle as a transceiver microwave dish antenna, which captures signals more efficiently in a specific direction. The -3dB points or half-power beamwidth (HPBW), where the signal pattern falls by 50% or 0.707 of maximum power, can be observed in the beamwidth below (Fig. 2.10). These different models of Ultrasonic sensors will determine the distance to which the signal will be. It is imperative to note that not only is the distance or coverage range specified by the frequency as highlighted before, but also by the sensor model, as high-end sensors are designed with precise accuracy and capabilities that enable them to cover longer distances at the same frequency. These different models are shown in Fig. 2.11 and will work as a guideline for selecting sensors suitable for each unique use case.

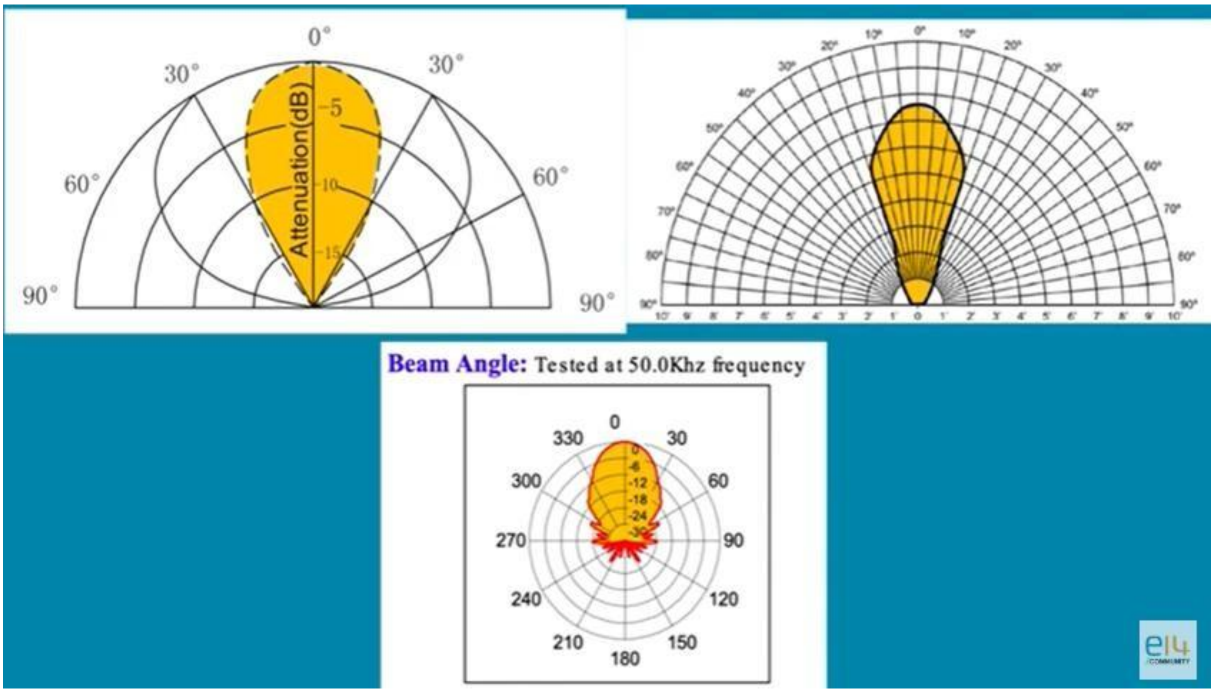


Fig 2. 10: Ultrasonic Antenna Radiation (Source: www.youtube.com/watch?v=2ojWO1QNprw)

The Typical Examples of Ultrasonic Sensors data sheet below shows the various beam patterns of different models. This information will help users accurately choose the appropriate type of sensor for a particular use case.

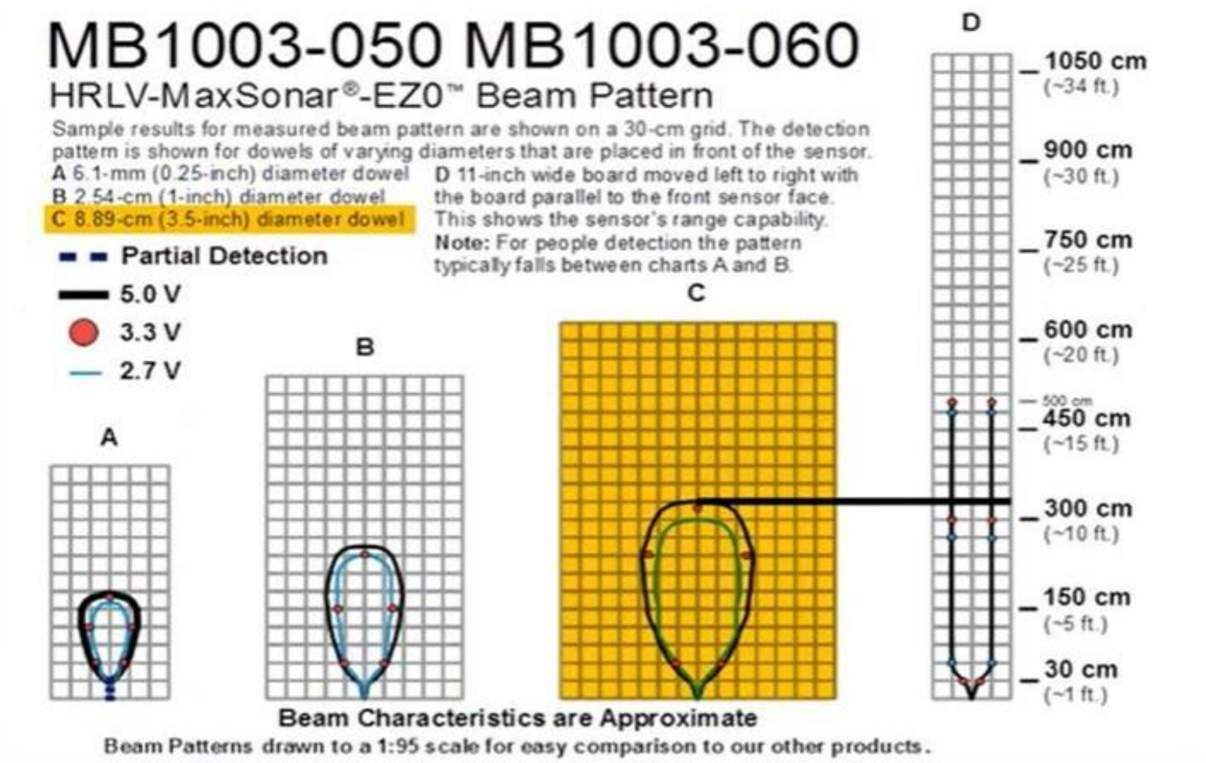


Fig 2. 11: Sensor Approx. Distances (Image Source: www.youtube.com/watch?v=2ojWO1QNprw)

The object to be detected must be close enough and large but not too close as this may result in it being in the blind zone the same way as in Microwave transceivers.

2.1.5 WORKING PRINCIPLE OF ARDUINO UNO REV 3:

The Arduino UNO acts as the project's central control unit, handling arithmetic, logic processing and task execution. It is a handy tool for controlling electronics for simple to complex tasks. All input and output peripherals are connected to the Arduino, centralizing system control. The Arduino board is equipped with an ATmega328 processor that executes commands input to the board and executes commands to the actuators, performing needed tasks.

Arduino PIN Layout and Description

The Arduino board is built on an open-source platform that allows anybody to replicate the board and improve on it. All the schematic files, design files, or software are available to the public. There have been different derivatives or clones of the hardware after the original hardware design by Arduino engineers in Italy, who wanted to create a microcontroller chip that enables high-level machine coding like C, C++, and Python to be used to program a microprocessor, as opposed to the original assembly language. The Dual-In-Line Package (DIP) version, not the surface mount, will be used for this project. The DIP is based on the ATmega328 Microcontroller, as shown below. Microcontrollers listen to sensors and then talk to actuators [12]. In this project, the **ATmega328** microcontroller will take inputs (listen) from the **HC-SR04** Ultrasonic sensor and instruct LEDs, buzzers, or LCDs to execute or display the information.

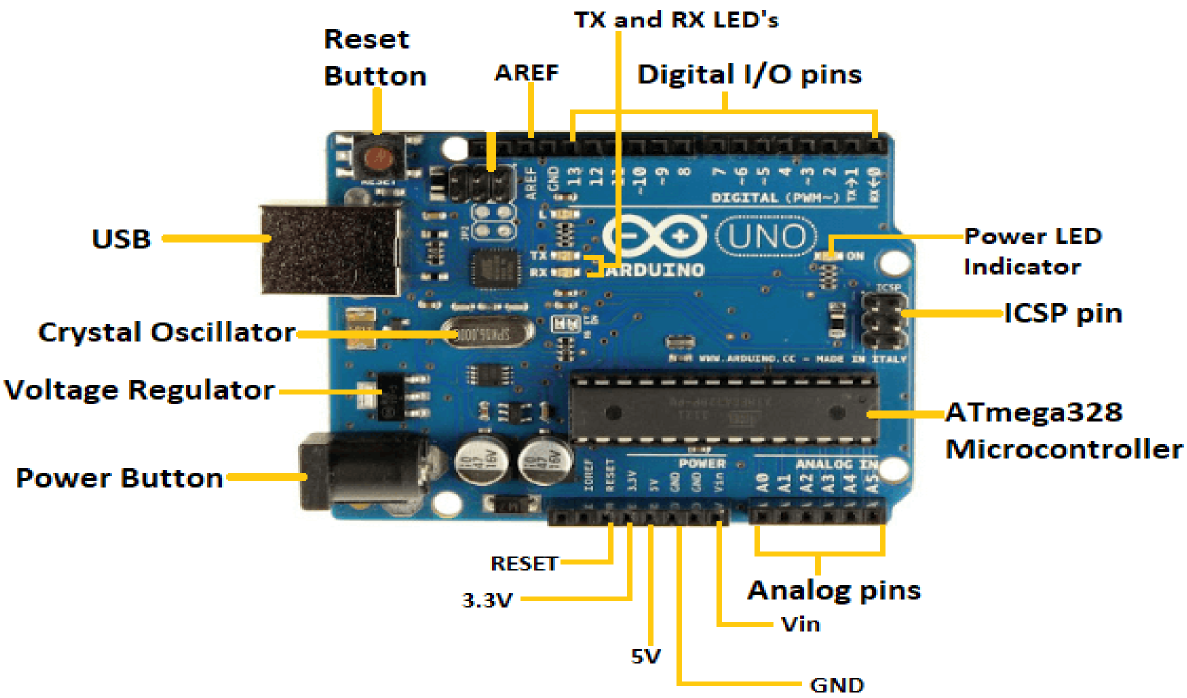


Fig 2. 12: Arduino Uno PIN Layout [https://www.javatpoint.com/arduino-uno-pinout]

Digital Pins on the Arduino can be INPUT or OUTPUT (I/O) pins and are pin 0 to pin 13 as shown. These pins give access to the microcontroller ATmega328. Also included as part of the digital pins are the Pulse Width Modulation (PWM) pins shown with a squiggly tilde symbol (~), i.e., pins 3, 5, 6, 9, 10, and 11. Pins 0 and 1 are receive (Rx) and transmit (Tx) pins, respectively, and are used for talking to the PC or any other devices. The onboard Tx/Rx LEDs indicate whether the transaction with external devices works well.
Analogue pins A0 to A5 connect to the microcontroller's analogue-to-digital converter (ADC). These pins are for analogue inputs or outputs with continuous, non-discrete values, such as temperatures read from a temperature sensor or resistance from a variable resistor meant to vary the luminance of an LED.
Next to the analogue pins are the power pins, which include the 3.3V and 5V source and the ground (GND) pins. Next to the power pins is the V_{in} pin for input voltage to the Arduino board. The board is also equipped with a built-in external power jack (Power Button) for external power, such as a PP3 9V battery, which can substitute USB power from the computer supplied with the USB data cable. The Reset button is also an essential trigger to reset everything and repeat processes. This works like the reset point next to the power pins: resetting the microcontroller by sending a low on the headers.

2.1.6 THE 1602 LCD MODULE:

The Liquid Crystal Display (LCD) is called the 16/2 because it has 16 columns and two rows for displaying alphanumeric values and characters as output information. The 1602 LCD has 16 characters per line and two such lines (2 rows) for 32 alphanumeric characters that can be displayed.

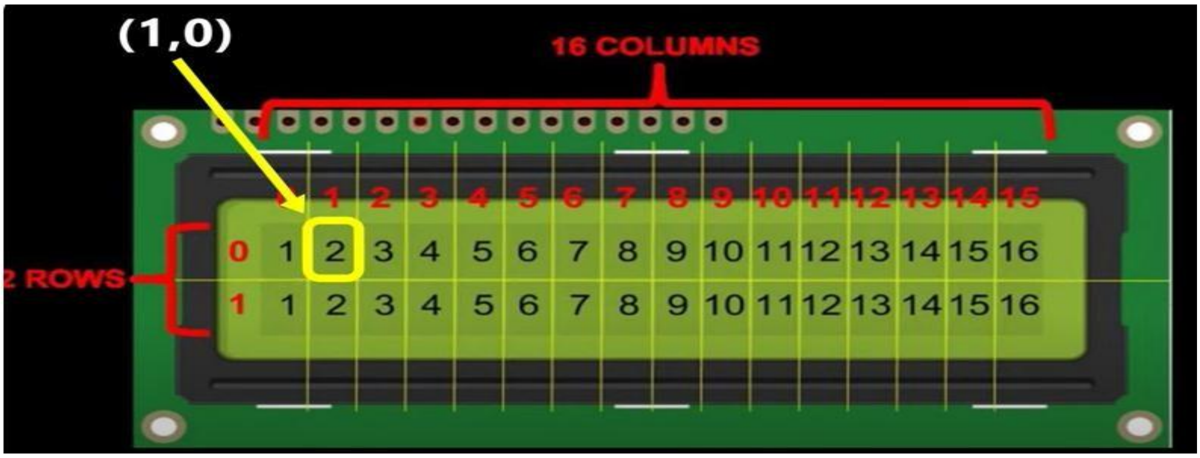
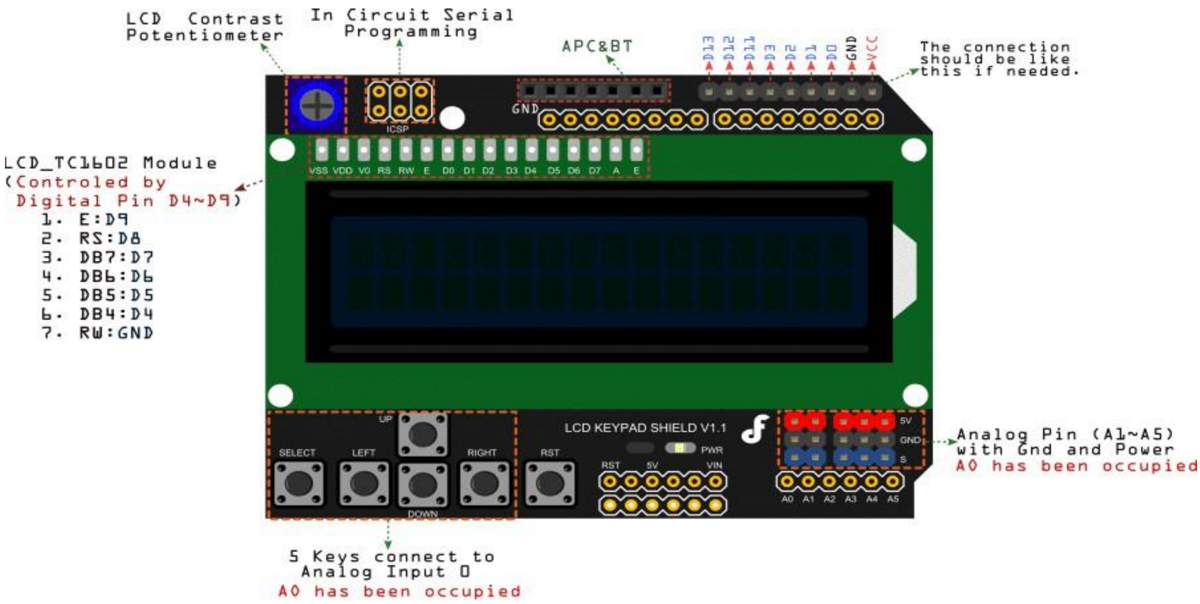


Fig 2. 13: The 1602 LCD Module

A location must be specified on the 16/2 LCD monitor to print or display information on the LCD. For instance, position 2 of the 16 columns in the figure above would be written in code as C/C++ Code:

LCD.setCursor(1, 0); // prints information on upper 16 characters of LCD where: 1 – will be the column position (columns are 0 to 15 = 16 columns)
0 - will be the row position (rows are 0 to 1 = 2 rows)

LCD Pin Description:



Pin	Label	AKA
1	VSS	GND
2	VDD	+5 Volts
3	V0	LCD Contrast 0-5V
4	RS	Register Select 1:Data 0:Command
5	RW	Read/Write – 1:Read 0:Write
6	E	Enable/clock enable – falling edge triggered
7	D0	Bit 7 – not used in 4-bit
8	D1	Bit 6 – not used in 4-bit
9	D2	Bit 5 – not used in 4-bit
10	D3	Bit 4 – not used in 4-bit
11	D4	Bit 3
12	D5	Bit 2
13	D6	Bit 1
14	D7	Bit 0
15	A	Anode + LED backlight
16	K	Cathode – LED backlight to GND

Fig 2. 14: The 1602 LCD Module Pin Layout
[https://wiki.dfrobot.com/LCD_KeyPad_Shield_For_Arduino_SKU__DFR0009]

It's much easier to connect the LCD to the one already mounted on the serial module to reduce the number of pins needed to connect the Arduino and the LCD. Without the serial module, the LCD alone would require about 12 connections. Equipped with the module, the number of pins reduces to only four, as shown in **Table 1** below.
The four pins of the Inter-Integrated Serial Bus are:

- **V_{cc}** = 3.3V or 5V
- **GND** = Ground
- **SDA** = Serial Data
- **SCL** = Serial Clock

LCD I2C	Arduino Uno, Nano	Arduino Mega
Vin	5V	5V
GND	GND	GND
SDA	A4	20
SCL	A5	21

Table 1 – Comparison of the 4-pins on different Arduino types

To have this LCD work, we will need to install IIC/I2C libraries (<Wire.h>) for the serial module and the LCD libraries (<LiquidCrystal_I2C.h>) onto our Arduino IDE.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Once the libraries are installed in the IDE, the physical connections follow: Vin =5V from the Arduino power pin is connected via a jumper to Vin on the LCD. The ground (GND) is also connected before other jumpers connect the Arduino's SDA and SCL pins to the LCD's respective pins.
The Pins on the serial module will determine the address to use when communicating with the LCD.

Table 2 illustrates the pin statuses and the corresponding address to be used. To avoid using many pins, pins A0, A1, and A2 are available at the back of the serial module interfacing the LCD. Their binary state will determine the address in the sketch to communicate with the LCD.

```
LiquidCrystal_I2C lcd(0x27);
```

A0	A1	A2	Address
0	0	0	0x27
0	0	1	0x26
0	1	0	0x25
0	1	1	0x24
1	0	0	0x23
1	0	1	0x22
1	1	0	0x21
1	1	1	0x20

Table 2 – Pin statuses for different addresses on the I2C adapter

This project will be using only one I2C serial module, and therefore, there is no need to change the address on the A0, A1, and A2 pins, which will remain as [0 0 0], i.e., 0x27. If another module was going to be connected to the Arduino, there was going to be a need to change the address of the I2C module. By changing the address on the LCD serial module pins marked A0, A1, and A2, a corresponding address will have to be selected in the code. For instance, if the second serial module connects to another device, such as another LCD or a compatible device, it must have a different address [19, 20].

Therefore, by toggling one of the respective pins to one, for example, 0 1 0, the address in the code will have to be changed to 0x25, as shown in **Table 2** above.

The simple code snippet to test the Arduino and LCD is as below, and the screenshot is the test result. We start by including the serial module connected to the LCD to reduce the pins needed (include <Wire.h>). The module to control the IIC LCD is also included (#include <LiquidCrystal_I2C.h>), and under the void setup (), we initialize the LCD to clear any displays on it with lcd.begin () statement. The line lcd.print () enables content inside the brackets to be displayed on the screen.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
lcd.begin();
lcd.backlight();
lcd.print("UCT MEng EEE5104Z");
}

void loop() {

}
```

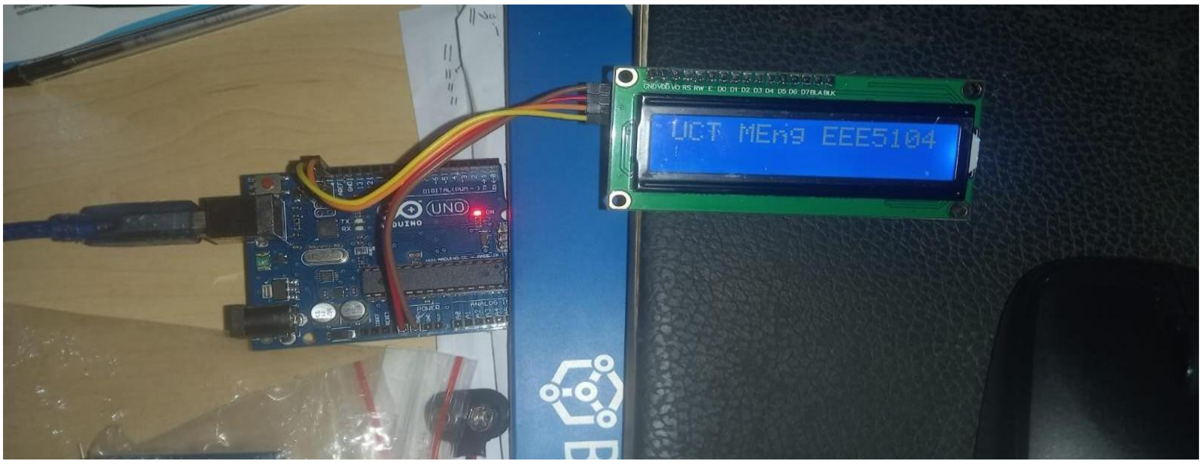


Fig 2. 15: Testing LCD Module

2.1.7 BUZZERS AND LEDS:

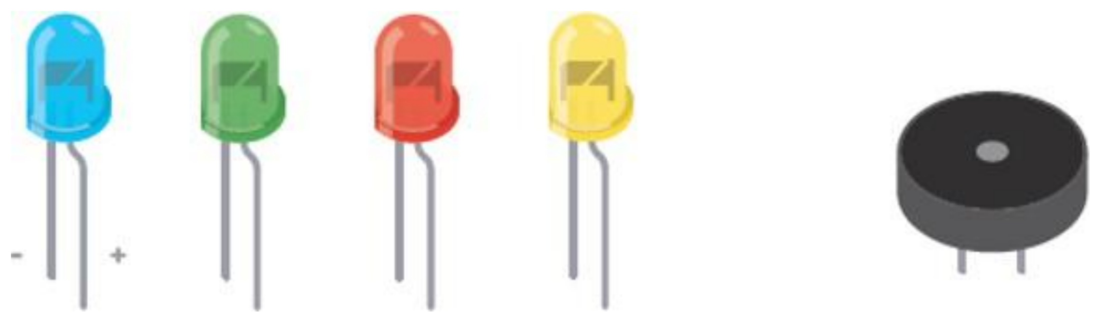


Fig 2. 16: Testing LEDs and Buzzers (Image Source: [11])

LEDs and a buzzer are other components or output actuators used in this project. LEDs comprise an anode terminal (+ve) and a cathode terminal (-ve). The positive (+ve) terminal is always longer than the negative (-ve) terminal. LEDs are a special diode illuminating when a current passes from the +ve side to the -ve. It is critical to know that they only conduct in one direction, and therefore, polarity should be noted when connecting them in a circuit. The project will use an LED to show visual alarms each time an object passes through the ultrasonic sensor monitoring point.
A buzzer uses the piezo-electric principle already discussed, the same principle employed in operating audio speakers and microphones. A piezo is an electric component that creates noise when vibrations are detected. Once the ultrasonic sensor detects an object, it converts the ultrasonic sounds into an electric signal, and these electric signals drive a buzzer, causing it to vibrate and create an audible noise.

2.1.8 ESP WROOM-32:

This shield will act as a modem for connection between the Arduino UNO R3 and the cloud. Several boards in the Arduino family have Wi-Fi or Bluetooth capabilities, but UNO 3 is not one of them. We need a board with internet capabilities to send our data to the cloud. The ESP Wroom-32 has been selected for this project because of its ability to connect to the internet via Wi-Fi.

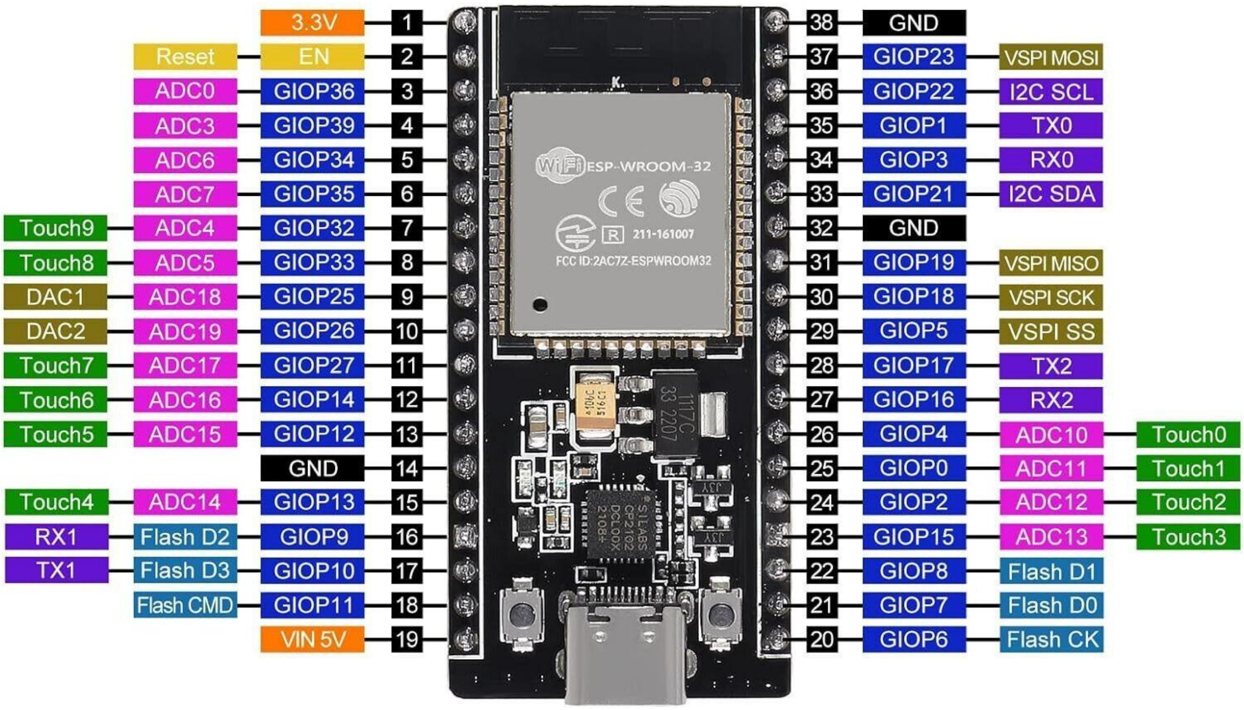


Fig 2. 17: ESP Wroom-32 Pin Layout [Facebook-IoT Group]

The shield operates in three modes: the **Station Mode**, the **Access Mode** and the **Duo Mode**.

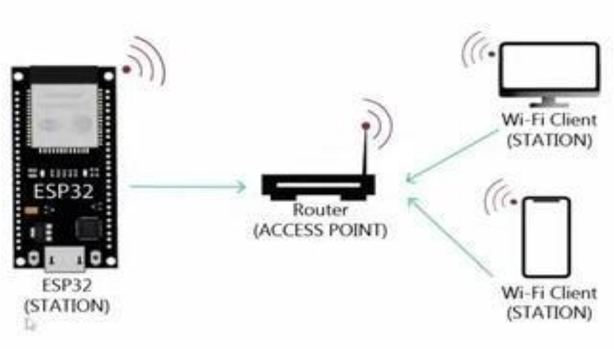


Fig 2. 18: ESP Wroom-32 – Station Mode

In Station Mode (Fig 2.18), the shield is programmed to connect to the Internet as an independent end device. It obtains an IP address from an access point, such as a network router.

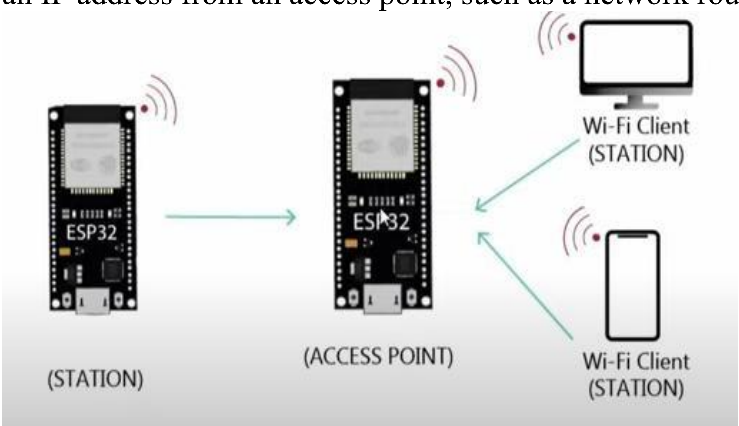


Fig 2. 19: ESP Wroom-32 – Access Mode

In Access Mode (Fig. 2.19), the shield acts as a router, assigning IP addresses to devices connecting to the Internet. It forms the hub of the network, where all devices connect. In Duo mode, the ESP shield is both an access point and a station itself.

2.1.9 ARDUINO IDE:

Arduino IDE is the platform used to create code and program shields. The Arduino Integrated Development Environment is a software application available for download that is used to program the Arduino board to perform any intended tasks. The code is written or copied into the IDE before it can be loaded onto the board. Arduino Version: 2.3.4 Copyright © 2024 Arduino SA will be used for this project. The beautiful part about using the Arduino IDE is that it can also be used to program the ESP32 shield, which is not a device for Arduino. The interface for the IDE looks as follows:

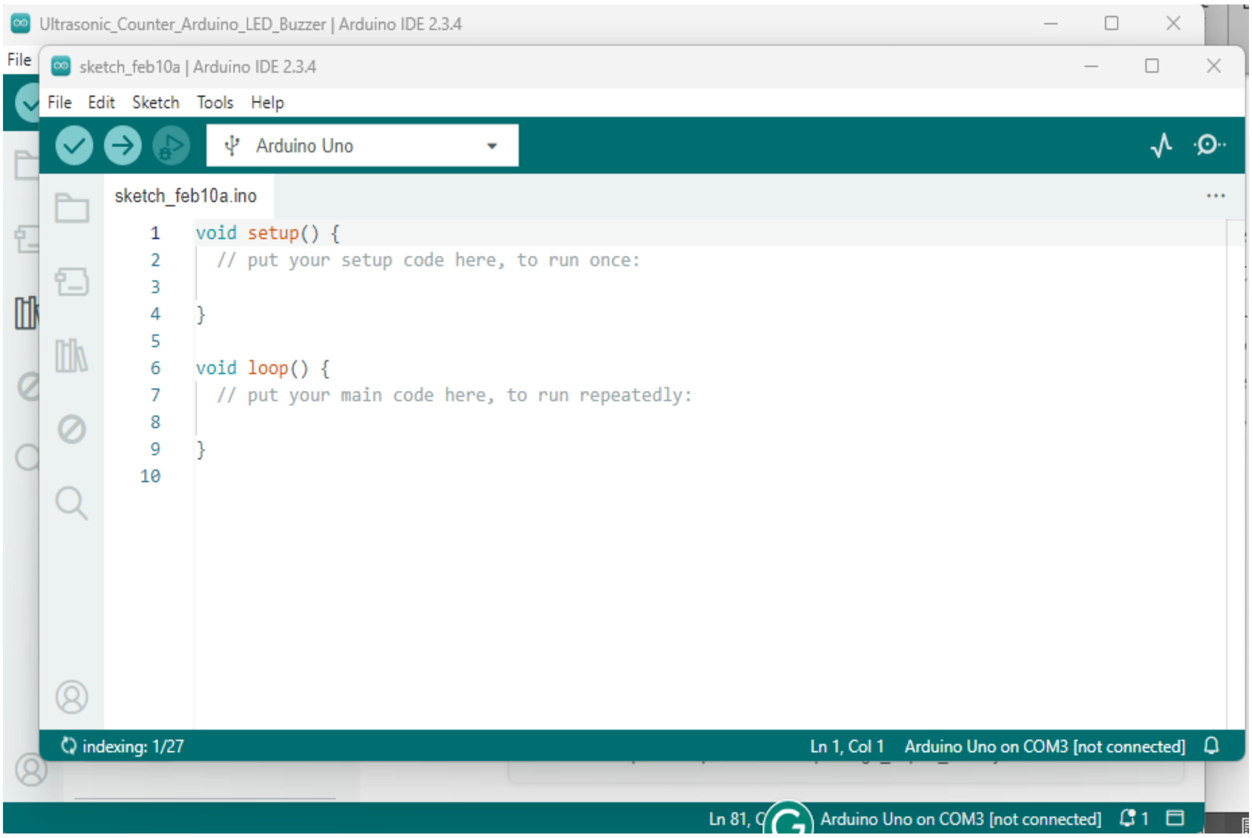


Fig. 2. 21: Arduino IDE Interface

The IDE is available for download and can be installed on any machine. As seen in the top left corner, the version is 2.3.4, and the bottom right shows that the COM port is not connected. Once a board is connected and the correct port is selected, the board type should appear in the field with "Arduino Uno" currently. This also applies when configuring an ESP32 shield. When the IDE is launched, and in cases where a particular example code has not been loaded, the basic code syntax with void setup () and void loop () will appear.

CHAPTER 3: METHODOLOGY

This chapter focuses on the design of an ultrasonic sensor-based tracking prototype connected to the Arduino Uno and the ESP32 shields. It will examine how the devices are interlinked and process ultrasonic sounds reflected by a passing object. This chapter will also consider the design restraints and project requirements to get the prototype to work.

3.1 METHODOLOGY

Several design considerations will take centre stage in the design process of the ultrasonic sensor-based tracking system. Intricate details providing guidelines for system development will be of primary focus. Some of these design parameters include:

- The hardware design parameters
- The software design parameters
- The design restraints and requirements
- The system setup and system testing and evaluation

The hardware capabilities are paramount parameters to consider when designing a system, considering the processing efficiency and limitations of the hardware. A well-written software program will never outperform the limits of the hardware it is running on. Hardware capabilities should not be ignored when building a robust system that can deliver the purpose for which it was designed. A capable piece of hardware may be under-utilized if operated by a poor software design. Therefore, Hardware and software complement each other to build a fault-resistant system that can deliver and be optimized to full functionality.

3.2 USER REQUIREMENTS

In an article penciled by Deloitte, one of the global accounting firms in their article [6], in which they spoke about driving the supply chain innovation using smart sensors, they listed the following [Fig 3.1] as some of the sensors that are finding much relevance in this drive [2]. This research paper focuses on a sensor that measures motion using ultrasonic sounds. The ultrasonic sensor will, therefore, be used to measure static and dynamic objects to determine the amount or number of goods.

Several other IoT sensor devices will have to be incorporated to improve the accuracy of the product tracking and tracing device. However, for this project, only an ultrasonic sensor was used, which also had a limitation in terms of its ability to pick a product according to the colour of the box or other parameters such as shape and size. The extract below [23,25] outlines a list of IoT sensors that must be incorporated to enhance product traceability with sheer accuracy. It can be observed that the shift from traditional means such as GPS, use of RFID, and Blockchain Technology is being catapulted by IoT inventions and technological advancements. Smart sensors that consume less power and can be miniaturized with nanotechnology are quickly replacing sophisticated methods that need a lot of resources and investments.

The focus, therefore in this chapter, is to show how an ultrasonic sensor as small part of a large community of smart sensors in the IoT ecosystem, can be used in SCM tracking. The methodology then becomes part of the thesis that illustrates how this can be achieved using the sensor of choice for the problem at hand.

Type	Definition	Example
Acoustic	Recognize audio vibration or frequency to determine activity, location, intensity	Piezo microphones, electret microphones, condenser microphones
Chemical	Measure fluid composition and concentration of biological/chemical compounds	MEMS technology, fuel cell
Electrical	Identify and examine changes or disruptions in electrical or magnetic signals based upon environmental inputs or conditions	Voltage, current, power
Environmental	Monitor and assess deviations in physical state, conditions, or surroundings	Temperature, humidity, color, moisture, light, pressure, liquid flow, air flow, heat, surface temperature
Image	Convert light waves into electrical signals to constitute a digital, optical form for visible condition monitoring	Infrared, ultraviolet (UV), visible spectrum camera
Motion and force	Measure static and dynamic objects to determine the amount, type, and rate of change to physical properties	Proximity (ultrasonic/acoustic, infrared), strain/weight, vibration, accelerometers, shock accelerometers, gyroscopic, position, motion, magnetic field, rotational
Touch	Detect body capacitance during physical contact between objects	Capacitive touch, resistive touch

Fig 3. 1: Illustration Deloitte Table of Typical Sensors used in SCM [23]

User requirements parameters to get a system to work must detail transparent processes to be adhered to, guaranteeing system functionality. Such processes may include, but are not limited to: Setting up the correct hardware and ensuring pin connections have continuity and that the polarity of components is adhered to. Incorrect voltages to the hardware components may result in permanent damage to peripherals, which may become very difficult to troubleshoot and localize if a fault or a system is not coming up.

- In this project, the distance at which the objects are to be detected by the ultrasonic sensor and the choice of the sensor to use are critical. Knowing the sensor's frequency range will help users understand the system's capabilities and anticipated behaviours.
- Compatibility between diverse devices and components must be considered. The knowledge that Arduino Uno is not equipped with Wi-Fi capabilities will prompt a system designer to incorporate a complementary device to enable an Internet connection. Again, connecting components such as LEDs and buzzers via a current-limiting resistor will help safeguard devices from unforeseen damage.
- Interconnecting the hardware and the software also becomes critical to enabling the proper and accurate design of optimized code that executes correctly on the hardware. Users' results must be correctly analyzed to give precise feedback on system performance.

The Arduino Uno must be connected to a PC via the USB port using a cable for a printer, and a Windows-based GUI system will be used to upload to the boards.

3.2.1 DESIGN RESTRAINTS

Some design constraints that I ran into while designing this system include:

- A level shifter is needed to help convert the 5V from the Arduino Tx/Rx pins to 3.3V, which is required on the Tx/Rx pins 16 and 17 on the ESP. They pose a risk of voltage that may be supplied to the ESP32 shield, thus limiting its lifespan. A voltage divider that requires resistor values of 1 kilo-ohm and two kilo-ohm could also be used, but these were unavailable when connecting the devices.
- Another design constraint I encountered involved using the ESP32 shield alone in the project, connecting all peripheral devices and connecting it directly to the internet to upload results to the cloud. Unfortunately, the ESP Wroom-32 board only comes with a 3.3V voltage output, which is insufficient for power devices like the Ultrasonic sensor and the LCD.

Again, the **ThingSpeak Cloud** platform requires much time to process the following input from the Arduino to update its database accurately with a new entry. Approximately 15 seconds of waiting time is required, which means that the debounce time in the code will also need to be adjusted to accommodate this delay. This will limit entries prompted by a detected object from overwriting each other, resulting in some entries being captured and recorded accurately. However, this can be solved using paid cloud platforms like Arduino IoT Cloud.

3.3 SYSTEM DESIGN SPECIFICATIONS

The system design follows a UML guideline, which directs the design of the system software and gives a brief overview of how the system will be structured. This UML diagram offers a structural overview of how components in an ultrasonic-based tracking system interact within a supply chain to enhance traceability. The ultrasonic sensor is the most essential part of the whole, on which this thesis is premised. The sensor receives ultrasonic waves, which are very critical for the function of the prototype, which are passes to the microcontroller for processing, before outputs can be displayed on the connected peripherals such as the LCD, LED, and the buzzer. That output signal is also extended through the microcontroller (ESP32) to display the results on the cloud.

UML Diagrams to Include:

1. **Use Case Diagram** – Describes user/system interactions.
2. **Class Diagram** – Shows the software structure (useful for code planning).
3. **Sequence Diagram** – Illustrates how components interact over time.
4. **Deployment Diagram** – Depicts hardware and network layout.

✅ **1. Use Case Diagram**

Actors:

- Operator (or System Admin)
- Arduino UNO
- ESP32
- ThingSpeak Server (or other IoT cloud service)

Use Cases:

- Start Conveyor Belt
- Count Passing Goods
- Activate LED and Buzzer
- Send Count to ESP32
- Upload Data to Cloud
- Monitor Data on Dashboard

✅ **2. Simplified Class Diagram**

plaintext Copy Edit

```
+------------------+       +---------------+      +--------------------+
| UltrasonicSensor |<----->|   ArduinoUNO   |<----->|     ESP32Module    |
+------------------+       +---------------+      +--------------------+
| - triggerPin     |       | - count        |      | - wifiSSID         |
| - echoPin        |       | - sensor       |      | - wifiPassword     |
| +getDistance()   |       | - ledPin       |      | - sendToServer()   |
+------------------+       | - buzzerPin    |      +--------------------+
                           | +detectObject()|
                           | +updateCount() |
                           +---------------+
```

3. Sequence Diagram

plaintext Copy Edit

```
Operator      ArduinoUNO      UltrasonicSensor      ESP32      ThingSpeak Server
   |              |                  |                |              |
   | Start System |                  |                |              |
   |------------->|                  |                |              |
   |              | Trigger Sensor   |                |              |
   |              |----------------->|                |              |
   |              |<--Distance--<----|                |              |
   |              | Count Object, Activate LED/Buzzer               |
   |              |--------------------------------->|               |
   |              |            Send Count             |              |
   |              |----------------------------------------------->|
   |              |                              Post Data         |
```

4. Deployment Diagram

plaintext Copy Edit

```
+--------------------------+           WiFi Network           +-------------------------+
|       Arduino UNO        |--------------------------------->|      ESP32 Module       |
| - Ultrasonic Sensor      |                                  | - WiFi Communication    |
| - Buzzer / LED           |                                  | - Sends Data to Cloud   |
+--------------------------+                                  +-------------------------+
                                                                        |
                                                                        |
                                                               +--------------------+
                                                               | ThingSpeak Server  |
                                                               | - Data Dashboard   |
                                                               +--------------------+
```

Fig. 3. 2: UML Diagrams

The UML shows that the ultrasonic sensor captures the object's attributes, which sends ultrasonic waves from its transmitter. When these waves are intercepted by the passing object with a range of 20cm, the waves are reflected to the receiver side of the sensor, where they are captured, and the sound waves are transduced to an electric signal for the microcontroller. The input to Arduino will then process the information received from the sensor and cause outputs to be displayed on the LED, the LCD, and the buzzer. Part of that output information is also sent to ESP32 to be relayed to the cloud (ThingSpeak Server). The cloud serves as a centralized point that resembles a shared ledger in BCT, where all stakeholders have everyday access to indelible records.

3.4 HARDWARE DESIGN

The following is a list of components and devices required to set up and build a system to accomplish the intended results. The alternative Arduino R4 Wi-Fi shield is the most lucrative device for this accomplishment, as it can connect to the internet without looking for a third-party device.

1. **Arduino Hardware—**The Arduino Uno Rev3 will be used for this project as the microcontroller to bring functionality to the whole project. It will interface between input sensors (such as an ultrasonic sensor) and output actuators (LCD, buzzer, and LED).
2. **Arduino IDE—**The Arduino Integrated Development Environment is a software application available for download that is used to program the Arduino board to perform any intended tasks. The code is written or copied into the IDE before it can be loaded onto the board. Arduino Version: 2.3.3 Copyright © 2024 Arduino SA will be used.
3. **Arduino Code** (sketch)-This program executes the needed functions. The code for the Arduino is essential C or C++, although some unique boards can take a program written in Python.
4. **Ultrasonic Sensor (HC-S04)** – the ultrasound input interface
5. **II2/I2C 1602 LCD** – for displaying the number of objects captured
6. **Active Buzzer** – to give an audible sound each time a different object passes on the conveyor belt.
7. **Thin carbon-film resistors** – current limiting resistors 220-ohm or 330-ohm will do.
8. **Breadboard 830-pin** for connecting all the devices and components.
9. **Male/female 10cm jumper wires** will connect microcontroller pins, sensors, and actuators.
10. **Plastic/Metal casing** – provides housing and protection to the prototype.
11. **The ESP32 board will be required to interface with** Arduino and the cloud, as Arduino UNO Rev3 lacks Wi-Fi capabilities to connect to the internet.

3.4.1 BASIC WORKING PRINCIPLE OF PROTOTYPE:

How It Works:

1. The **Ultrasonic Sensor** continuously measures the distance in front of its Line of Sight (LoS), detecting the presence of an object falling within the specified range.
2. If an object comes within the detection threshold (20 cm), it is detected, and the counter increments.
3. **The buzzer** and **LED** are briefly activated to indicate detection.
4. The **LCD** updates with the latest object count.
5. The **Debounce mechanism** prevents multiple detections for a single object pass.
6. The information count is updated on a server in the cloud via an ESP microcontroller that connects to the Internet (the ThingSpeak.mathworks.com platform has been chosen for this project).

7. The inventory continues to be updated as the number of objects increases, thus enabling remote monitoring automated goods count on a conveyor machine.

3.4.2 HARDWARE CONNECTIONS

The devices and components listed above were connected, as shown in Fig. 3.3.

The **Ultrasonic Sensor** pins were connected as follows:

- V_{cc} connected to the 5V feed on the breadboard.
- The GND was also connected to the ground rail on the breadboard.
- Ultrasonic sensor Echo pin to pin 13 of the Arduino.
- Ultrasonic sensor: Trigger pin to pin 10 of the Arduino.

The **II2 LCD** pins were connected as follows:

- V_{cc} connected to the 5V feed on the breadboard.
- The GND was also connected to the ground rail on the breadboard.
- SCL pin to SCL/A5 of the Arduino.
- SDA pin to SDA/A4 of the Arduino

The **Buzzer** pins were connected as follows:

- Positive was connected to pin 4 of the Arduino breadboard through a 220/330-ohm resistor.
- The GND was also connected to the ground rail on the breadboard.

The **LED** pins were connected as follows:

- Positive was connected to pin 2 of the Arduino breadboard through a 220/330-ohm resistor.
- The GND was also connected to the ground rail on the breadboard.

Once the components have been connected through the breadboard or directly onto the Arduino board, a USB cable is inserted into the board's USB connector to load the program that will operate the hardware.

The Arduino IDE is launched, and the proper Arduino type is selected under **Tools>Boards Manager> Select Board**. Also, of critical importance here is choosing the correct communication port (COM port) that will enable the program or sketch to be uploaded from the IDE to the physical Arduino hardware. The sketch is created in C/C++ code syntax according to the Arduino board's physical wiring and peripherals. Once a bug-free program has been written and checked for IDE errors, it can be loaded onto the board, and the hardware will be tested for proper execution.

The UML below is a guideline for writing the code to execute on the shield and perform the expected tasks.

Fig 3.3 below shows all devices connected to the Arduino Uno Rev 3, which serves as the central control unit of the system. However, this setup, when loaded with the sketch, will only display the object count to be viewed locally as the object counted will be displayed on the LCD screen display. A step further is needed, and this is how unique this thesis is from other research done in this regard. One of the critical aspects of the thesis is to automate all stages and leave little to no human element in it. This set may require an individual within the warehouse if objects pile up on a conveyor system.

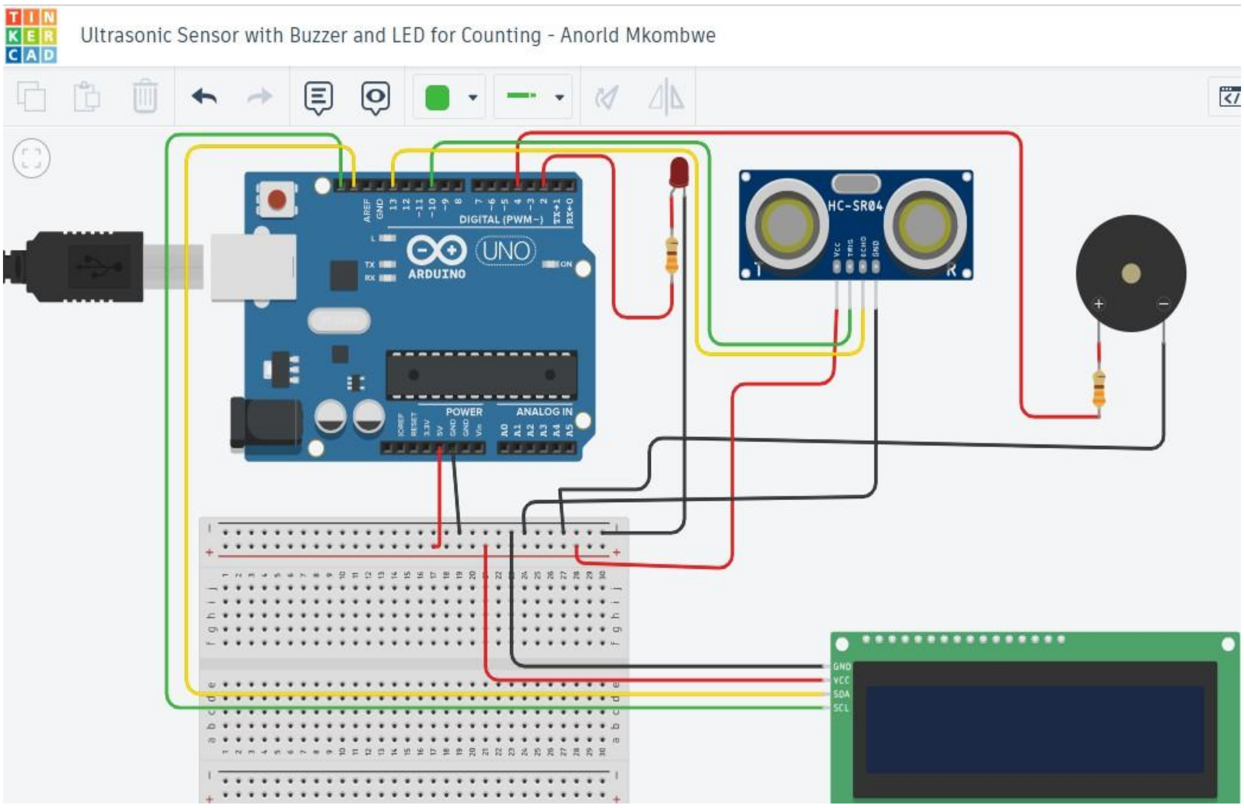


Fig 3. 3: Arduino connections to Ultrasonic Sensor, LCD, LED and Buzzer

Hardware Connections Between Arduino Uno and ESP32:

1. Arduino TX (Pin 1) → ESP32 RX (GPIO16)
2. Arduino RX (Pin 0) → ESP32 TX (GPIO17)
3. Common Ground (GND → GND)

Since the Arduino Uno operates at 5V logic and the ESP32 at 3.3V, we use a **logic level shifter** or a **voltage divider** (Fig. 3.2) on the TX line from the Arduino to the ESP32

The Arduino voltage pins have 5V and 3.3V, while the ESP32 power pin emits 3.3V only. If the transmit and receive pins on both are to be connected, a device such as a **Logic Level Shifter** must be connected (Fig. 3.4). This device converts the 5V from Arduino to 3.3V, consumed by the ESP32 pins and vice versa. A voltage divider can be incorporated without a Level Shifter to accommodate the voltage requirements between the two shields. The ESP Wroom-32 alone could have been used in this project since it has enough pins to connect to all peripheral I/O devices and save on hardware cost requirements. However, the shield does not have a 5V output voltage that will be required to power up peripherals like the LCD and the Ultrasonic sensor, which require 5V [27].

Fig. 3.4 complements the work done in Fig. 3.3. The output information from the Arduino Uno will have to be passed to ESP32 via the Tx and Rx ports of Arduino to the Rx and Tx ports, respectively. The ground pins of each board will have to be connected in common. After the hardware setup, the only thing that remains will be to load the code or sketches to each respective shield via the Arduino IDE.

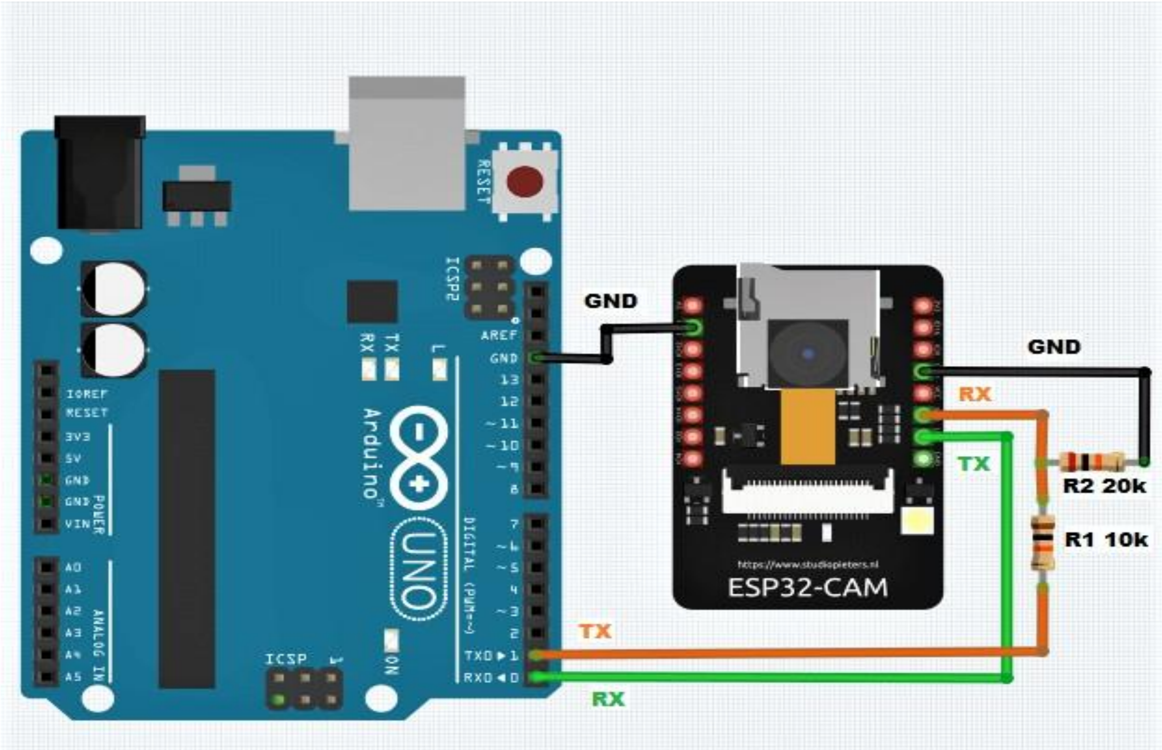


Fig 3. 4: Arduino Uno to ESP32 Connections

3.4.3 PROJECT SETUP

Setting up the hardware for this project involves noting the pin connections and following the step-by-step instructions on connecting the ultrasonic sensor to the Arduino board and all the other peripherals to the Arduino. The Arduino is also connected to the ESP32 board, which has Wi-Fi capabilities, thus enabling the data to be sent over the Internet to a centralized monitoring system.

The hardware connections are as described in 3.4.2 and involve a non-complicated step-by-step adherence to instructions. The measured distance could not be very accurate due to the speed of sound in different media, temperatures, and other factors like humidity and air pressure. However, by including libraries for the Ultrasonic sensor, such as the <NewPing.h>, the distance accuracy can be improved as this library considers factors affecting the speed of sound in different media.

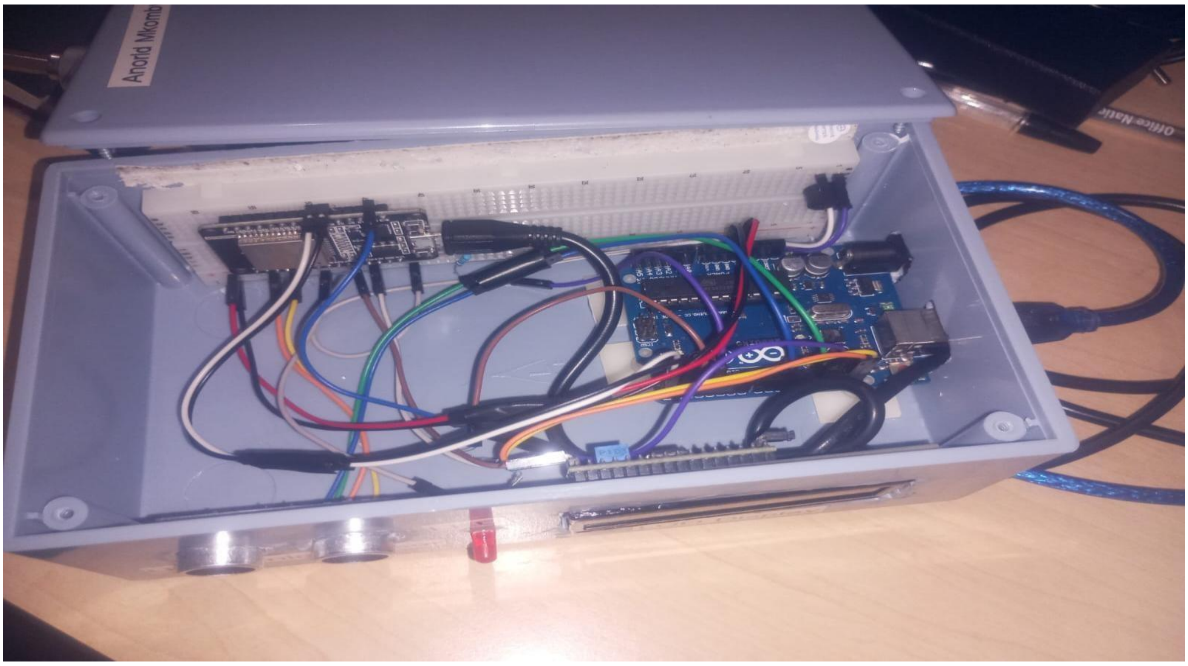


Fig 3.4: Ultrasonic Sensor-based Prototype – Top-view

Fig 3.5: Ultrasonic Sensor-based Prototype – Front-view

CHAPTER 4: DESIGN

This chapter focuses on the system's setup and how the prototype will work effectively to achieve the intended results. The system will be run in a Microsoft Windows GUI-based environment. The maximum distance for the sensor to detect moving or passing objects is of critical importance. This helps counter any results captured after objects not intended to be counted pass in the direction the ultrasonic sensor faces. The sensor can send an ultrasonic sound wave up to a range of 450cm, as described in Chapter 2 of the Literature Review. However, only the objects passing within the intended range will be captured by setting the range at which the sensor must trigger an increment on the LCD, sound a buzzer, and light up the LED. An Arduino IDE is also needed to upload code to the Arduino Uno device and the ESP32 shields.

4.1 SYSTEMS SETUP

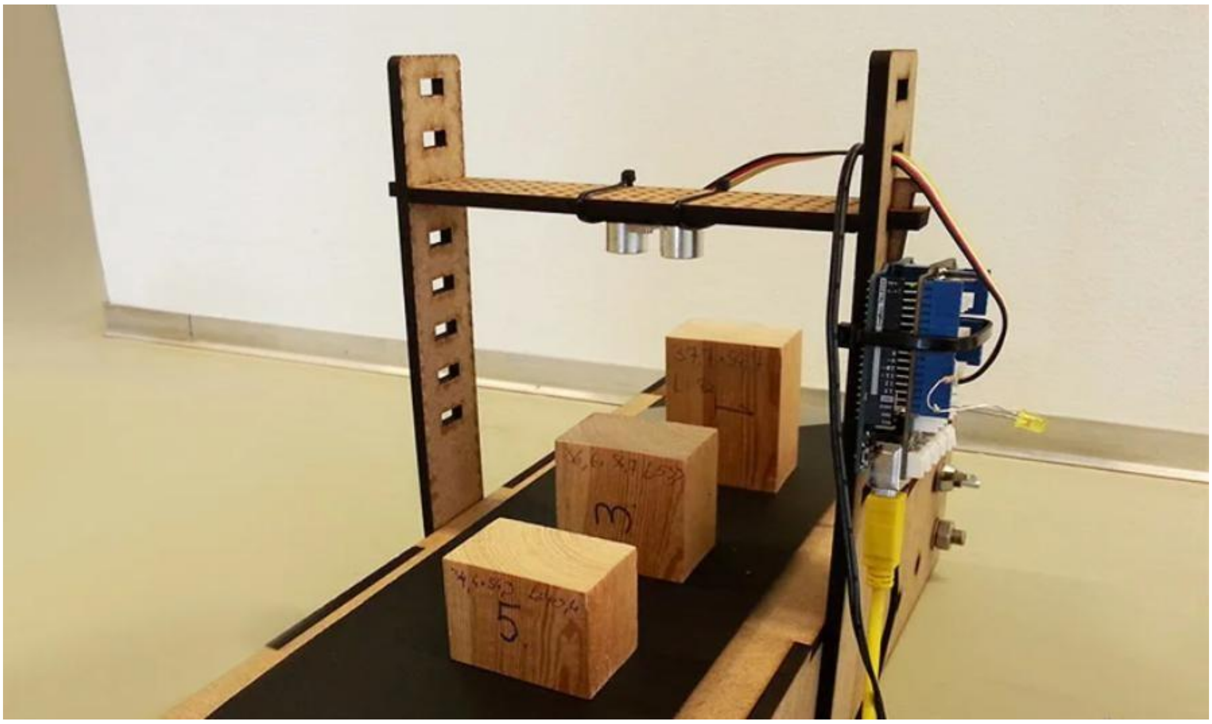


Fig 4. 1: Miniature model of envisaged Arduino Ultrasonic Sensor

The equipment needed for the system setup has already been listed. It will include an Ultrasonic sensor, an LCD, an LED, a buzzer, resistors, jumper wires, Arduino Uno and ESP32, as mentioned in Chapter 2. A casing for these devices will also be needed to keep them together for a proper prototype to be built. Once the system is set up, as in Fig. 4.1, it is time to execute the function by understanding how it works.

To achieve this, an Arduino Uno R3 will be interfaced with an Ultrasonic sensor strategically positioned to monitor the passing zone or point for the goods in the supply chain that need to be observed for quality purposes or counted. The sensor must be mounted fixed with the conveyor belt 10cm away. This object monitoring distance can be increased to as much as 450cm (4.5m), depending on the quality of the sensor used and the frequency selected. A higher frequency in the ultrasonic band (20kHz to 1000kHz) will mean the distance will be shorter. The lower frequencies, closer to 20kHz, will guarantee a greater distance for the coverage zone.

This ultrasonic sensor will continuously measure distance; anything passing beyond the predetermined 20cm will not be detected or counted. In fact, at a distance beyond 20cm, movements will not trigger a count that will be displayed on the LCD, nor will they trigger a buzzer to ring while lighting the LED. The moment anything falls in the expected range, this will trigger the ultrasonic sensor that will then increment the sensor, thus counting every object passing in that 20cm range. This distance can, however, be changed depending on the sensitivity required and how close the conveyor belt is to the sensing point. This trigger will also cause the LED mounted on the Arduino circuit to flicker, indicating visual alerts that can help in product enumeration. A buzzer and LED provide additional feedback when the object is detected. This sound buzzer accompanying the LED creates a combined tracking technique that is difficult to miss and can assure product traceability and traceability in the SCM. If a buzzer or LED is not triggered, it may indicate that the object passing through the monitored zone has not been counted. This will raise the alarm or probably trigger the conveyor belt to stop, prompting a human figure to investigate. However, to avoid human intervention, the goods passing on a conveyor machine can be counted in two stages, and the statistics from each object count will be compared for accuracy. The results of the incremented counter will be displayed on the LCD. They can thus be transmitted to a remote monitoring hub when the Arduino is interfaced with a Wi-Fi capability shield such as an ESP32 board.

Other product attributes, such as weight, size, colour, and dimensions, could be enhanced and captured correctly when additional sensors are used, such as a strain gauge for weight or colour sensors to differentiate products according to different colours. However, this project uses an ultrasonic or ultrasound sensor to enhance product traceability in supply chain management. Combining all these IoT devices will help improve accuracy and create a robust, error-free system that is difficult to circumvent due to unscrupulous human intent.

The delay set in the code must also tally with the delay needed in the ThingSpeak Cloud platform of 15 seconds to capture the objects correctly without overwhelming or updating the system in retrospect. This is for demonstration purposes only. In a live production environment, faster platforms such as AWS Cloud or Arduino IoT Cloud can be used on a paid plan to receive and update objects quickly in real time.

4.2 BASIC CONNECTIVITY BLOCK DIAGRAM:

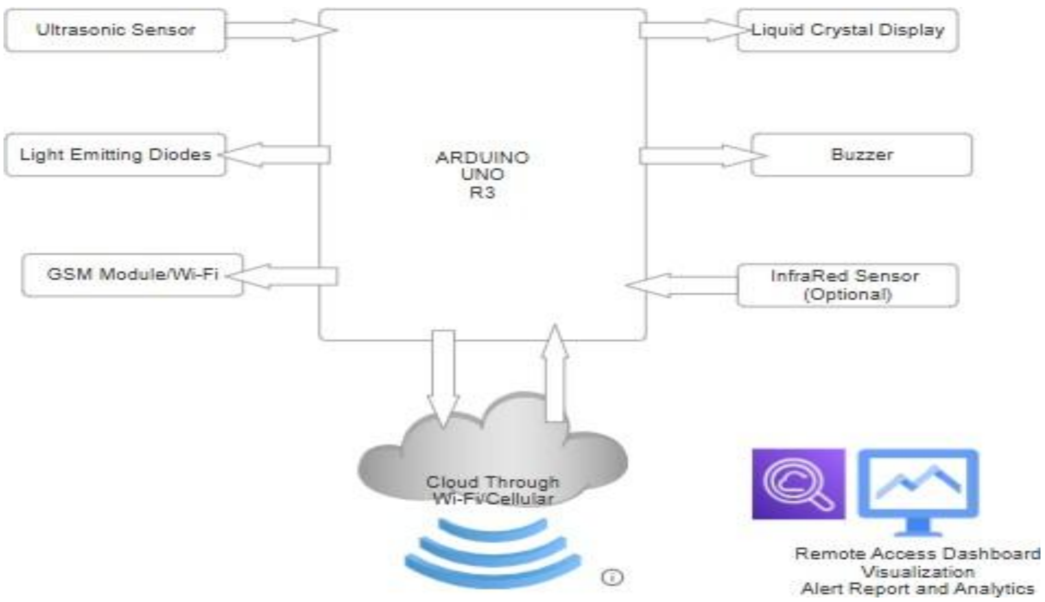


Fig. 4. 2: Block diagram of the proposed system [32].

The figure above (4.2) shows how the Arduino Uno R3 will be interfaced with various sensors and detectors to enhance the expected research project results. The ultrasonic sensor will be used for product tracking and traceability and complemented by other output detectors to display the number of products being monitored as they pass through a conveyor point. The information may then be displayed and transferred to a cloud-based monitoring system or a central monitoring point for analysis or to update an online inventory system.

The device connection's block diagram (Fig. 4.2) shows the Ultrasonic sensor as an input device to the Arduino Uno microcontroller. It is connected to the circuit to detect ultrasonic waves and feeds the information into the microcontroller. The microcontroller processes the data according to the C++ program loaded on the chip and executes functions according to the code. The output devices in the block diagram include a **buzzer** that will give audible alerts every time an object passes within the specified range is detected. An **LED** is also part of the output devices connected to the microcontroller and will give visual alerts by lighting up each time an object is detected. The information or object count will be displayed on the **LCD** screen each time there is an increment. The final output device is a GSM/Wi-Fi module that allows the information to be relayed to a cloud environment for monitoring by stakeholders from anywhere in the world where there is internet. In this thesis, an **ESP32 module** has been used as the interface to connect to a cloud platform, ThingSpeak.Mathworks.com. This cloud platform provides the Remote Access Dashboard for visualization of alert reports and analytics. Additional parameters can be added for monitoring depending on the number of sensors connected to the microcontroller. Sensors such as temperature and humidity sensors, weight checking sensors (strain gauge), IR sensors, accelerometers [32, 33, 35], etc., can be included to enhance accuracy in product traceability and tracking in the SCM project.

CHAPTER 5: IMPLEMENTATION

This chapter covers the final steps needed to activate the system or prototype to work after all the hardware is connected and set, as illustrated in Chapter 3. The only remaining thing will be to connect the Arduino Uno and the ESP32 shields via their respective USB cables to the PC's COM ports. This chapter majors on the implementation steps needed to achieve the prototype results. Care must be taken to ensure no hardware is damaged by pins touching each other, creating a short circuit or overvoltage supplied to shields.

5.1 IMPLEMENTATION OF THE SCM TRACKING SYSTEM ON ARDUINO UNO

After the Arduino IDE is launched and the two windows to load the sketch to the shields are opened, the proper boards and COM ports should be selected for each IDE window. A typical empty code with basic reveals as follows:

Typical Arduino Sketch Syntax: C/C++

```
#include // Libraries
int, float, char // declare variables

void setup () {
//put your setup code here to run once:
}

void loop () {
//put your main code here to run repeatedly:
}
```

The typical code above shows sections where libraries can be called using the statement #include <Name of the Library>. Libraries' names end with the dot h letter, as illustrated in the libraries, to control an LCD display coupled with an I2C serial module. The syntax will be as follows:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
lcd.begin();
lcd.backlight();
lcd.print("UCT MEng EEE5104Z");
}

void loop() {

}
```

The statement **#include <LiquidCrystal_I2C.h>** will consist of all the parameters to operate the LCD. It is important to note that this statement cannot be typed in the included section without first downloading the libraries from the Arduino IDE library section or zipped files on other platforms. The code or sketch can be developed within the Arduino Uno IDE or copied onto the sketch space from a Word document or Notepad.

5.1.1 ACTUAL CODE FOR ARDUINO UNO:

The following is the code designed to operate the Arduino Uno board with the Ultrasonic sensor defined as an INPUT device and the LCD, LED, and Buzzer defined as OUTPUT devices:

//Include Libraries:

#include <Wire.h> //Library for Serial module I2C
#include <LiquidCrystal_I2C.h> //Library for 16/2 LCD connected with serial module

//Declaring Variables:

LiquidCrystal_I2C lcd(0x27, 16, 2); //Defining LCD and indicating the address 0x27 to display const
int trigPin = 10; //ultrasonic Trig pin on Arduino
const int echoPin = 13; //ultrasonic Echo pin on Arduino
const int buzzer = 4; //Buzzer positive pin on Arduino
const int led = 2; //LED positive pin on Arduino
long duration; //defining the time it takes for the sensor to send a sound
signal int distance; //defining the **distance** to be calculated by speed of **sound** x
duration int objectCount = 0; //defining objects to be counted

//Setting pin modes:

void setup () { pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); pinMode(buzzer, OUTPUT);
pinMode(led, OUTPUT);
Serial.begin(115200); // Ensure the same baud rate on ESP32

lcd.begin(); //Initializing the LCD lcd.backlight();
lcd.setCursor(0, 0); lcd.print("Objects: ");
}

//Code section to run over and over:

void loop () { digitalWrite(trigPin, LOW); delayMicroseconds(2); digitalWrite(trigPin, HIGH);
delayMicroseconds(10); digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance = duration * 0.0343 / 2; // Convert to cm using the speed of sound formular
if (distance > 0 && distance <= 20) { // Object detected within prescribed distance objectCount++;
digitalWrite(led, HIGH); digitalWrite(buzzer, HIGH); delay(100); digitalWrite(led, LOW);
digitalWrite(buzzer, LOW);

// Update LCD lcd.setCursor(9, 0); lcd.print(objectCount);
// Send data to ESP32 Serial.println(objectCount);
delay(500); // Debounce delay to avoid multiple counts

}

}

5.1.2 ACTUAL CODE FOR ESP32:

The ESP32 module must be configured with the below to receive the out from the Arduino shield's serial ports using the UART protocol before sending the information onto the cloud using the MQTT broker or the HTTP protocol. I have used the HTTP protocol in this project because of its flexibility and functionality.

//Include Libraries:

```
#include <WiFi.h>                // the Wi-Fi libraries should be included #include <HTTPClient.h>
                                 the HTTP client libraries included
```

//Declaring Variables:

```
const char* ssid = "Network SSID used";          //SSID for the Wi-Fi network to connect to
const char* password = "Password to the network"; //Password for the Wi-Fi network to connect to
String apiKey = " V1SI15XPO0Q21YF6 "; //Write API key as obtained under API keys

int receivedCount = 0;
```

//Setting pin modes:

```
void setup () { Serial.begin(115200);
  Serial2.begin(115200, SERIAL_8N1, 16, 17);  // RX=16, TX=17

  WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) { delay(1000);
  Serial.println("Connecting to WiFi...");
  }
  Serial.println("WiFi Connected.");
  }
```

//Code section to run over and over:

```
  void loop () {
if (Serial2.available()) { receivedCount = Serial2.parseInt(); Serial.print("Received Count: ");
  Serial.println(receivedCount);

  sendToThingSpeak(receivedCount);
  }
  }
  void sendToThingSpeak(int count) {
if (WiFi.status() = = WL_CONNECTED) { HTTPClient http;
  String url = "https://api.thingspeak.com/update?api_key=V1SI15XPO0Q21YF6&field1=0"
  + String(count);

  http.begin(url);
  int httpCode = http.GET();

  if (httpCode > 0) {
  Serial.println("Data sent to ThingSpeak.");
  } else {
  Serial.println("Error sending data.");
```

```
}
http.end();
} else {
Serial.println("WiFi Disconnected!");
}
delay(15000);  // ThingSpeak requires a 15-second delay between updates
}
```

5.2 FINAL SYSTEM DESIGN

The above steps in 5.1.1 and 5.1.2 will finalize the prototype's design. The next step is to log in to www.ThingSpeak.Mathworks.com or use the link provided for published accounts so that individuals can monitor results displayed on the Cloud platform. Proper connection checks should always accompany the hardware setup. The following check will be a list of physical checks on the prototype before it can be connected to a power source:

- Physically inspect all the jumper wires connecting the devices for loose connections. The wires can easily become loose as they are not permanently soldered to the microcontroller pins.
- Power up the prototype and check if the microcontroller responds to power and the peripherals, such as the LCD screen and ESP32 module.
- Load preliminary code via the Arduino IDE to the Arduino Uno Rev 3 and ESP Wroom-32 microcontrollers.
- Performing preliminary tests by swiping an object before the Ultrasonic sensor and observing the results. The LED and buzzer should go off after an object is detected, as observed in simulations on Tinkercard or Wokwi [20][21].
- Observe the object count on the LCD screen and check if the results are being sent to the cloud platform.

It is important to note that the code may require further debugging in terms of adjusting the object capture distance or the delay timer before the next object can appear on a conveyor belt for counting. A short debouncing time may result in an object being counted more than once, and a very long debouncing time may result in some objects being skipped or not being counted as they move at a constant speed along the conveyor machine.

5.2.1 ULTRASONIC SENSOR-BASED TRACKING SYSTEM

After the codes for both the Arduino shield and the ESP32 shields are loaded on the devices and the devices are connected to the peripheral devices, the units are powered from a computer USB port or a PP3 9-volt battery, which may be used to power the Arduino. The ESP32 is cascaded through a level shifter to get its power from the Arduino. It is important to note that the sketches for both shields are downloaded onto the devices from a standard Arduino IDE after selecting the respective boards and COM ports on the IDE.
To test the project's functionality, an object is passed in front of the Ultrasonic sensor within the 20cm range. The buzzer should beep, and the LED should flash, indicating that an object has been detected.

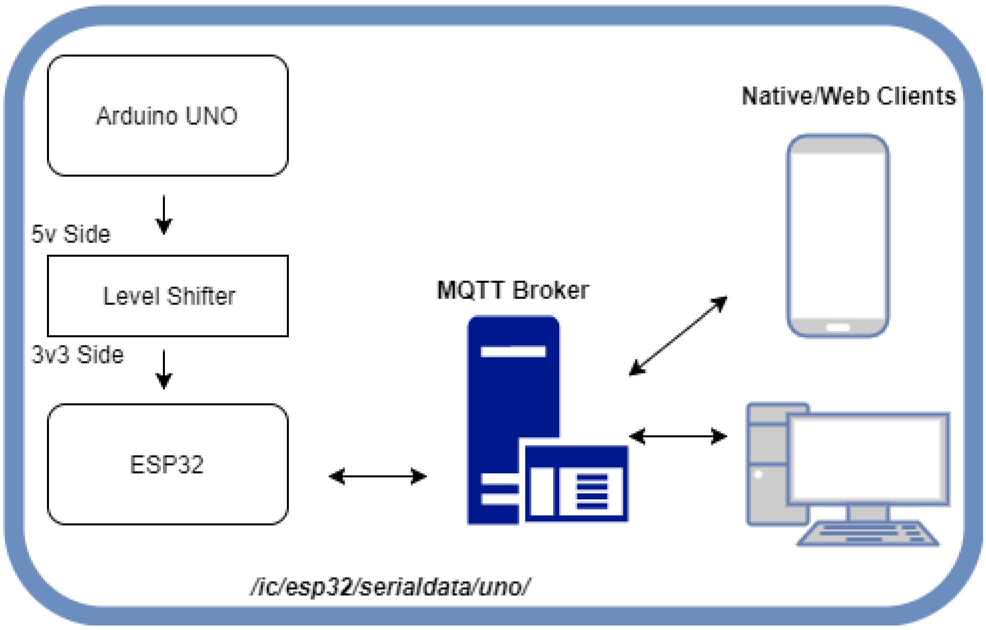


Fig 5. 1: Block Diagram of Arduino Interfaced with ESP32 for Cloud Upload []

Figure 5.1 above illustrates how the Arduino Uno and ESP32 hardware will be interfaced with each other. A level shifter will change voltage requirements between the shields, while an MTTQ broker or HTTP protocol will communicate data to the cloud environment.

CHAPTER 6: RESULTS AND ANALYSIS

Analysis of project results is a critical section of the entire project design as it helps to determine whether results have been obtained and, if not, draw inferences to establish the cause of failure. Without the results, it isn't easy to establish whether set goals and hypothetical assumptions have been answered. Observing and analyzing results concludes, creating a fundamental footing for future research. To test the hypothetical questions raised in Chapter 3, weighing in on the results to see if the project answered the assumptions about whether IoT technology may replace the contemporary tracking means in SCM is crucial.

6.1 METHODOLOGY EVALUATION

It is critical to look at the methods employed during system development and prove their efficiency. It is essential to check that the ultrasonic device works and can send ultrasonic signals and accept the ones reflected by an object within the range of the sensor. The Arduino Uno must also be tested with the ESP32 to see if it works. This also applies to the buzzer and LED; everything must be tested before being connected to circuitry and the code loaded for execution.

6.1.1 DEVICE TESTING AND EVALUATION

The simulations were first done using two free online platforms, Wokwi and Tinkercad, to test the system's functionality. Once these simulations had produced the desired results and the code was loaded onto a physical prototype, it was necessary to do the testing. The devices were connected, as shown in the pictures below, and the code was loaded into two shields, the Arduino Uno and the ESP32 microcontrollers. However, the simulators did not allow the two shields to be connected or used in one simulation. Once the code was loaded in the two shields, the objects were passed within the specified range (20cm) in the code, and it could be observed that the Objects being counted are incrementing on the LCD, and the buzzer is going off while the LED lights up. The object count was also recorded on the ThingSpeak cloud platform, fulfilling the purpose of designing and building a system capable of product tracking and tracing within the SCM.

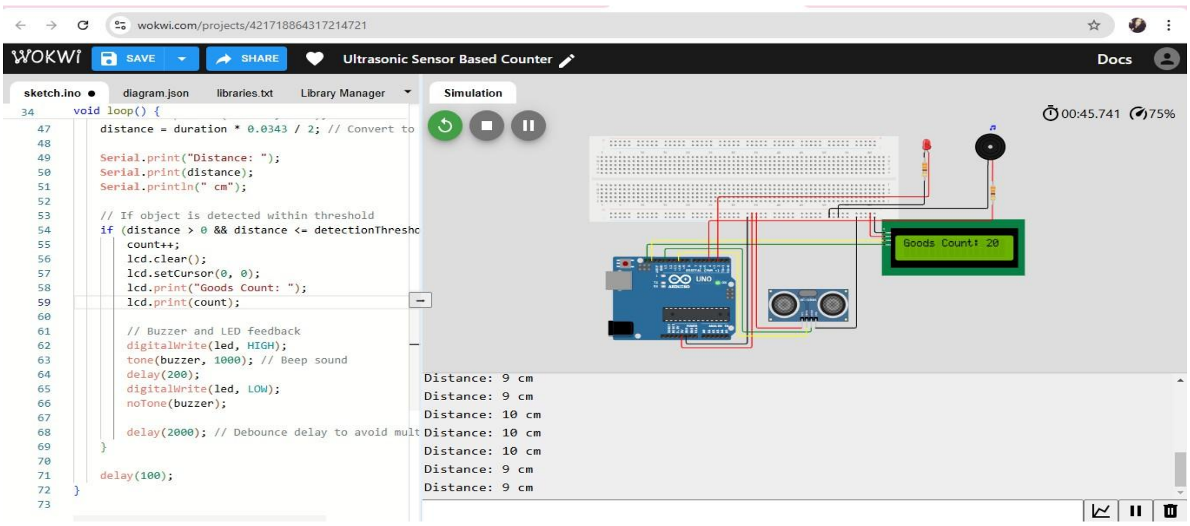


Fig 6. 1: Arduino Only Simulation on Wokwi.com

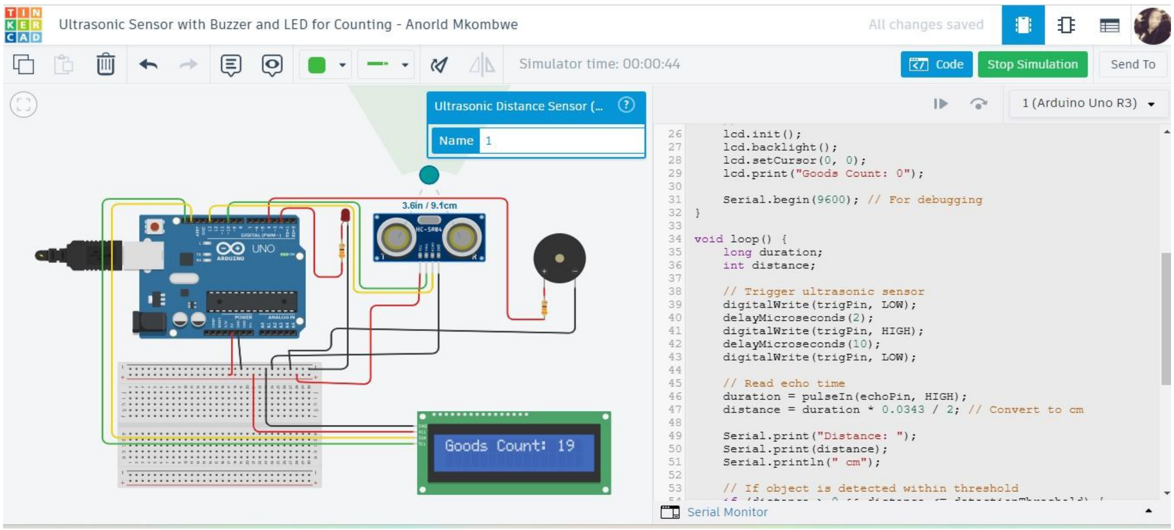


Fig 6. 2: Arduino Only Simulation on Tinkercad.com

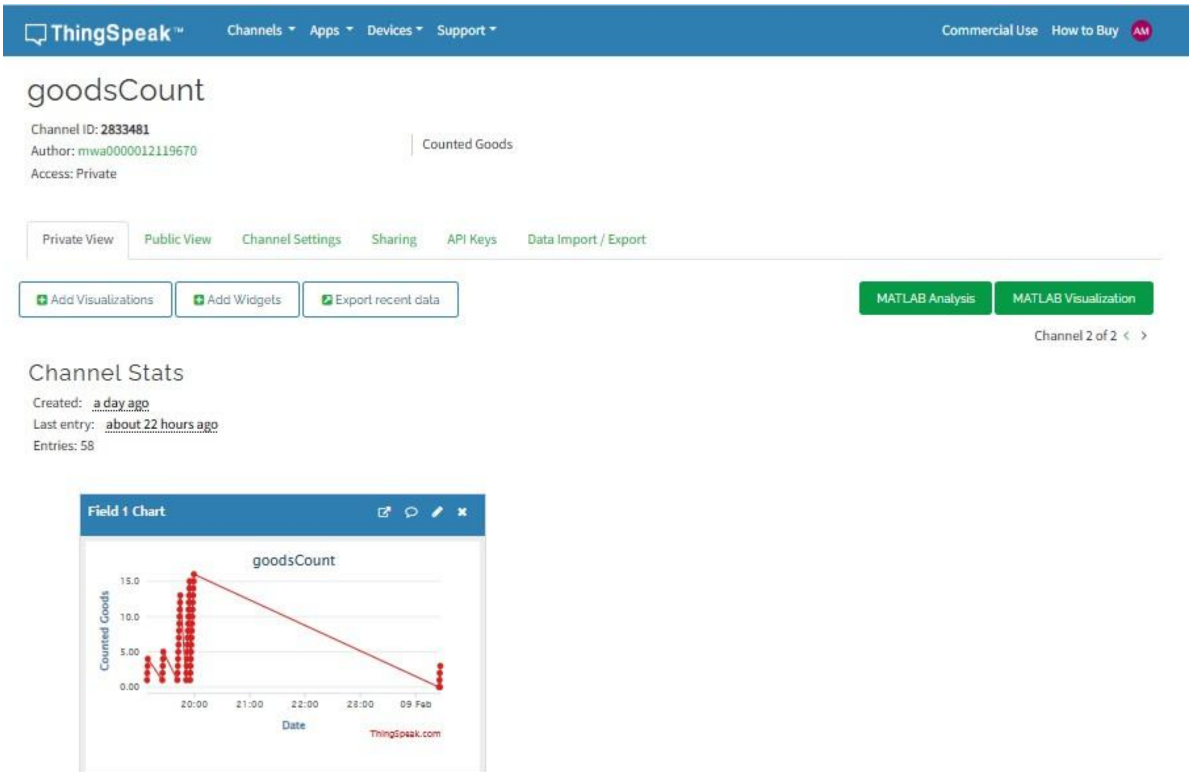


Fig 6. 3: Goods Count recorded on ThingSpeak.Mathworks.com

6.2 COMPARISON BETWEEN TECHNOLOGIES IN SCM:

The comparison between SCM tracking and tracing technologies, such as BCT, GPS, RFID, and IoT, shows that IoT can be integrated with other technologies available. This makes IoT usage for product tracing more efficient and cost-effective. IoT uses smart sensors such as Ultrasonic, Temperature and Humidity, Load cells, and all the sensors mentioned by Deloitte. This thus makes deploying IoT and tracking methods in SCM more active in terms of pricing.

Technology	Function in Supply Chain	Advantages	Limitations
Blockchain	Provides a **secure, decentralized ledger** to record transactions and movements of goods. Ensures **transparency and immutability** of data.	- Prevents fraud and tampering - Enhances trust between stakeholders - Automates contracts via **smart contracts**	- High **computational costs** - Requires industry-wide adoption - Slower transaction speed compared to traditional databases
GPS Tracking	Uses **satellite signals** to provide **real-time location tracking** of shipments. Often used in fleet management and long-distance transport.	- Provides **accurate location tracking** - Enables **route optimization** - Enhances **security against theft**	- **Limited indoors** - **Signal loss** in tunnels or remote areas - Requires **continuous power**
RFID (Radio-Frequency Identification)	Uses **tags and readers** to track goods at warehouses, distribution centres, and retail stores. It can be **active (battery-powered)** or **passive (no battery powered by a reader)**.	**- Fast, automated scanning** - Reduces **manual data entry errors** - Works **without direct line of sight**	- Limited **range for passive RFID** - Can be **interfered with by metals/liquids** - Higher **initial setup cost**
IoT (Internet of Things)	Uses **smart sensors** (temperature, humidity, shock, etc.) and connected devices to provide **real-time data** on goods. Can integrate with Blockchain, GPS, and RFID.	- Provides **real-time condition monitoring** - Enables **predictive analytics** - Enhances **automation and efficiency**	**- Security vulnerabilities - High data processing requirements** - Requires **robust connectivity**

Table 4 – Comparison of existing SCM tracking technologies

Best Use Cases in Supply Chain

- **Blockchain** → Ideal for **securing transaction records, verifying authenticity**, and tracking high-value items.
- **GPS Tracking** → Best for **real-time tracking of shipments** over long distances.
- **RFID** → Perfect for **warehouse management, inventory tracking**, and **automated scanning**.
- **IoT** → Most effective for **monitoring goods' conditions**, such as temperature-sensitive pharmaceuticals or perishable food.

CHAPTER 7: CONCLUSION AND RECOMMENDATIONS

This final chapter wraps up the thesis by reflecting on the initial objectives outlined in the first chapter and the methodology, design, implementation, and findings discussed throughout the study. It will also revisit the project's original goals and compare them with the actual results presented in the results and analysis chapter. Additionally, recommendations will be provided for future work on the SCM adoption of IoT as a reliable technology to optimize product traceability and trackability. Potential improvements to this project will also be mentioned, as it served only as a window to illustrate how effective IoT smart devices can be used to achieve goals previously in the domain of more expensive technologies like BCT and GPS usage.

7.1 CONCLUSIONS

The results show that a smart sensor can be used for product tracking and tracing. The aims or initial objectives highlighted at the beginning of the thesis have been successfully confirmed. A system has been designed and formulated based on IoT smart sensors, particularly the ultrasonic sensor, to help count objects and potentially identify their shape by employing a few ultrasonic sensors. The thesis fulfilled the objectives of:

a). The development of a product tracking system:

- Using an Arduino Uno R3 microcontroller to capture ultrasonic sound and convert it to usable information on product count.
- Development of C++ code to interface an Ultrasonic sensor with an Arduino Uno and output results to peripherals like an LCD for local monitoring, a buzzer for audible alerts, and an LED for visual alerts. The C/C++ code was also developed to view these results locally and, on the cloud, using a microcontroller such as an ESP32 shield.
- Testing the efficiency of an IoT-based tracking system that can easily integrate with traditional tracking technologies such as BCT, GPS, and RFID and create a superstructure that is less costly, effective, and comparable to existing systems.

b). Evaluation of results obtained to justify the wide usage of IoT-based tracking systems:

- Prove beyond a reasonable doubt that the human element can be eliminated, thus creating a robust ecosystem of sensors to perform tasks faster and cheaper.
- Compare the results obtained and conclude that IoT smart sensors are a perfect deployment strategy for improving product traceability for goods in transit and those inside warehouses moving on conveyor machines.

Using IoT-based product tracking and tracing in supply chain management (SCM) offers several advantages, such as improving efficiency, transparency, and overall performance. The following are some key justifications:

1. **Real-time Visibility and Monitoring**

- IoT sensors provide real-time tracking of products at every stage of the supply chain.
- Companies can monitor sensitive goods' location, temperature, humidity, and shock levels.

- Reduces delays and ensures better coordination of the SCM process.

2. **Improved Inventory Management**
    - Smart inventory tracking prevents overstocking or stockouts.
    - Automated updates help optimize storage space and replenishment schedules.
    - Reduces holding costs and improves warehouse efficiency.

3. **Enhanced Supply Chain Transparency**
    - End-to-end tracking builds trust between manufacturers, suppliers, and customers.
    - Reduces fraud, counterfeiting, and theft risks by verifying authenticity.
    - Ensures compliance with regulations (e.g., pharmaceuticals, food safety).

4. **Faster Issue Detection and Response**
    - IoT sensors detect anomalies like temperature fluctuations or route deviations.
    - Alerts enable quick intervention, preventing spoilage or damage.
    - Reduces financial losses and improves product quality.

5. **Increased Efficiency and Cost Reduction**
    - Automation minimizes manual tracking efforts and human errors.
    - Optimized routes reduce fuel consumption and transportation costs.
    - Predictive analytics prevent delays and optimize delivery schedules.

6. **Better Customer Experience**
    - Customers get accurate, real-time delivery updates.
    - Ensures on-time deliveries and reduces customer complaints.
    - Builds brand reputation and customer satisfaction and trust.

7. **Sustainability and Waste Reduction**
    - Optimized logistics reduce fuel emissions and environmental impact.
    - Monitoring storage conditions prevents product wastage (e.g., perishable goods).
    - Supports green supply chain initiatives.

8. **Data-Driven Decision Making**
    - IoT devices collect valuable data on supply chain performance.

- AI-driven analytics help optimize processes and forecast demand.
- Enhance strategic planning and business growth.

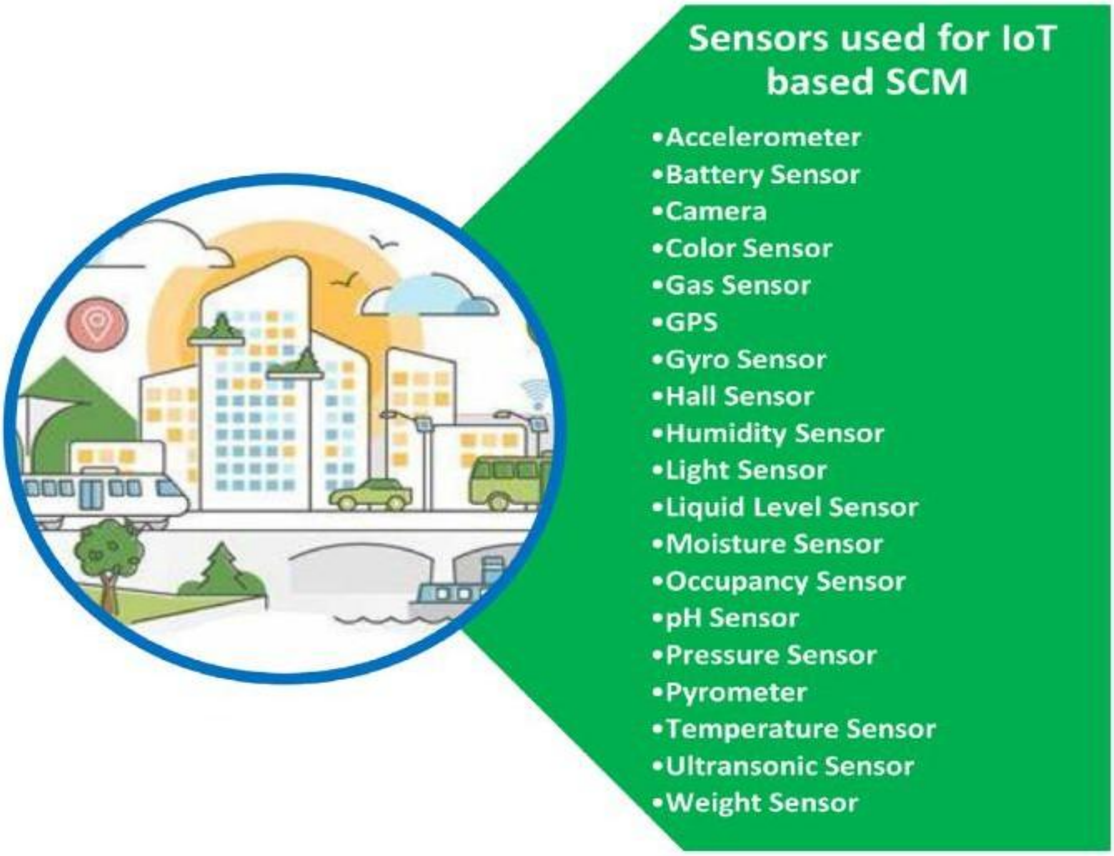


Fig 7. 1: Sensors used for IoT-based SCM (Image Source:[2])

The ever-growing list of IoT-based sensors (Fig. 7.1 above and Fig. 3.1) shows the growing interest in investing in sensors to enhance the shift from traditional GIS, GPS, and RFID and the possible integration of such with the powerful BCT as an SCM method.

The prototype has been successfully designed and developed, and performs precisely how it is expected to behave. This thesis has the potential for sensors to become full throttle in terms of usage in SCM. Sensors are cheaper and are manufactured in bulk, lowering the cost of installing and maintaining them. The RF ultrasound waveband has proven to be a band that can achieve missions by using devices operating in that band with little fear of interference, as with the 2.4GHz or 5GHz Wi- Fi band. The radio spectrum remains scarce, with many operators shifting into the E-band and Terahertz to avoid congested bands.

7.2 RECOMMENDATIONS

The aim of the design of this system, as defined in the RAD methodology of system design, has been achieved. The project will need further work as this serves as a building block for further research and development in using IoT-based sensors to complement and replace some of the costly processes used today in product tracking and traceability.

Other smart sensors (Fig. 7.1) are also recommended to enhance traceability. This will help physically count the objects moving on a conveyor belt and detect their shapes, colours, weight, and condition, especially for perishables. This designed system can be used not only for counting objects but also to trigger an alarm once a threshold has been passed for the number of goods that can be loaded to a carrier before a conveyor system can be triggered to stop.

Once several sensors are used, goods can eventually be traced within the warehouse and in transit. An Arduino microcontroller, for instance, can be twined with a GSM/LTE or other cellular technologies to trace the goods in transit from point A to point B. This will eliminate the need to use or rely on GPS only for goods outside the storehouse. The use of sensors is becoming very prevalent. Therefore, companies willing to migrate to IoT-based tracking systems can reap the benefits of such an emerging technology that is very promising and can potentially disrupt the industry.

In the evolving landscape of Supply Chain Management (SCM), efficient, cost-effective, and real-time tracking of goods is essential for maintaining transparency, minimizing losses, and optimizing logistics. While technologies such as blockchain, GPS, RFID, and barcode readers have dominated the SCM domain, I strongly recommend adopting an **ultrasonic sensor-based system integrated with Arduino Uno** for a targeted, simplified, and economically viable tracking solution, especially in localized, warehouse, or production environments.

Why Choose Ultrasonic + Arduino Uno?

1. Cost-Effective and Accessible

Ultrasonic sensors (like the HC-SR04) combined with the Arduino Uno microcontroller present an extremely low-cost alternative compared to GPS modules, RFID infrastructure, or blockchain-based systems that require high computational and operational overhead. For small- to medium-sized enterprises, this is a significant advantage.

2. Precision in Short-Range Tracking

Ultrasonic sensors are particularly well-suited for short-range detection and object counting, making them ideal for environments such as loading bays, storage racks, and production lines. They can accurately detect the presence and movement of goods without requiring line-of-sight or expensive tags.

3. Real-Time Monitoring and Integration

Using Arduino Uno, sensor data can be processed in real-time and easily integrated with cloud services like ThingSpeak or MQTT brokers for live data visualization and reporting. This enables managers to get actionable insights without the complexity of managing a full GPS or blockchain ecosystem.

4. Low Maintenance and High Reliability

Unlike RFID tags, which can be damaged or detached, or barcode readers, which can fail due to smudges or misalignment, ultrasonic sensors are contactless, robust, and require minimal maintenance. They are immune to lighting conditions and can function effectively in dusty or noisy environments.

5. Scalability with Modularity

The Arduino ecosystem is inherently modular. A network of ultrasonic sensors can be easily deployed across warehouse sections or conveyor belts, providing granular visibility into goods movement. This modularity allows for easy system expansion as the operation grows.

6. Simplicity Over Complexity

While blockchain offers traceability and data integrity, its implementation is technically demanding, costly, and often overkill for local goods tracking. Similarly, GPS is ineffective indoors and RFID requires significant infrastructure. Ultrasonic sensors, on the other hand, provide just enough functionality where it's needed—no more, no less.

Conclusion

For companies seeking a practical, low-cost, and efficient way to track goods—especially in localized SCM environments—an ultrasonic sensor-based system powered by Arduino Uno offers an optimal balance of functionality, affordability, and ease of deployment. While traditional technologies remain valuable in broader or more complex systems, this solution stands out for its simplicity, accuracy, and adaptability in real-world warehousing and inventory scenarios.

REFERENCES

[1] K. Kraisintu and T. Zhang, *The Role of Traceability in Sustainable Supply Chain Management*, M.S. thesis, Dept. of Tech. Mgmt. and Economics, Div. of Logistics and Transportation, Chalmers Univ. of Technol., Gothenburg, Sweden, 2012.

[2] S. Taj *et al.*, "IoT-based supply chain management: A systematic literature review," *Internet of Things*, vol. 23, 2023. [Online]. Available: www.elsevier.com/locate/iot

[3] M. Khan *et al.*, "The impact of technologies of traceability and transparency in supply chains," *Sustainability*, vol. 14, no. 24, 2022. [Online]. Available: https://doi.org/10.3390/su142416336

[4] T. L. Narayana *et al.*, "Advances in real-time intelligent monitoring of environmental parameters using IoT and sensors," *Heliyon*, vol. 10, 2024, Art. no. e28195.

[5] K. Singh, "Smart inventory management using electronic sensor-based computational intelligence," in *Computational Intelligence Techniques for Smart Energy Systems*, 2021, pp. 1–14. [Online]. Available: https://www.researchgate.net/publication/344964639

[6] N. Kholis, F. Baskoro, and Nurhayati, "Automatic packaging conveyor tracking system based on Arduino Uno using photodiodes and SRF04 ultrasonic sensors," *Indonesian Journal of Electronics Engineering*, vol. 4, no. 2, 2021.

[7] C. G. Juarizo *et al.*, "IoT-based end-to-end monitoring of logistics and tracking of truck vehicles using Arduino microcontroller," 2019.

[8] KEAprojects, "Count Objects Using Arduino and Ultrasonic Sensor on a Production Conveyor," *Instructables*. [Online]. Available: https://www.instructables.com/Arduino-using-Ultrasonic-sensor-to-count-log-objec/

[9] S. Fitzgerald and M. Shiloh, *Arduino Projects Book*, Arduino, 2015.

[10] V. Kayalvizhy, A. Professor, KNGAC, Thanjavur, *Internet of Things: A Hands-on Approach*, by A. Bahga and V. Madisetti. Subject Code: 18KP2CS07, 2014.

[11] R. Handayani *et al.*, "Leveraging shifting ultrasonic sensors for three-dimensional cargo volume determination," *International Journal of Technology, Design and Innovation*, vol. 8, no. 2, 2024. [Online]. Available: http://iieta.org/journals/ijtdi

[12] J. Kros, Y. Liao, and J. F. Kirchoff, "Traceability in the supply chain," *International Journal of Applied Logistics*, vol. 9, no. 1, 2019.

[13] M. M. Gabriel and P. K. Kamweru, "Arduino Uno, ultrasonic sensor HC-SR04 motion detector with display of distance in the LCD," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, 2020. ISSN: 2278-0181.

[14] L. A. Ajao, J. Agajo, O. M. Olaniyi *et al.*, "A secure tracking automobile system for oil and gas distribution using telematics and blockchain techniques," *Balkan Journal of Electrical and Computer Engineering*, vol. 7, no. 3, pp. 257–268, 2019. [Online]. Available: https://doi.org/10.17694/bajece.520979

[15] S. Anandhi, R. Anitha, and V. Sureshkumar, "IoT enabled RFID authentication and secure object tracking system for smart logistics," 2019.

[16] M. G. Kibria, H. S. Kim, and I. Chong, "Tracking moving objects for intelligent IoT service provisioning in web objects enabled IoT environment," in *Proc. Int. Conf. Information and Communication Technology Convergence (ICTC)*, 2016, pp. 561–563.

[17] T. Fan, F. Tao, S. Deng, and S. Li, "Impact of RFID technology on supply chain decisions with inventory inaccuracies," *International Journal of Production Economics*, vol. 161, pp. 66–77, 2015. [Online]. Available: https://doi.org/10.1016/j.ijpe.2014.10.004

[18] B. Parameswaran *et al.*, "IoT-based food and grain condition traceability and controlling system in warehouses," *IEEE Xplore*, Aug. 2024.

[19] G. Elert, "The Electromagnetic Spectrum," *The Physics Hypertextbook*. [Online]. Available: https://physics.info/em-spectrum. Archived and retrieved Jan. 21, 2022.

[20] "Tinkercad," [Online]. Available: https://www.tinkercad.com/login

[21] "Wokwi," [Online]. Available: https://wokwi.com/

[22] R. Bhuvana, L. M. Madhushree, and P. S. Aithal, "Comparative study on RFID-based tracking and blockchain-based tracking of material transactions," *Int. J. Appl. Eng. and Management Letters*, vol. 4, no. 2, pp. 22–30, 2020. doi: http://doi.org/10.5281/zenodo.3977270
[23] A. S. Balobaid, S. Shamsudheen, and A. Anoop, "IoT enabled blockchain-based secure delivery tracking system: A performance check," in *Proc. 1st Int. Conf. on Logistics (ICL)*, Jeddah, Saudi Arabia, 2024. doi: 10.1109/ICL62932.2024.10788605
[24] L. A. Ajao, J. Agajo, O. M. Olaniyi *et al*., "A secure tracking automobile system for oil and gas distribution using telematics and blockchain techniques," *Balkan J. Elect. Comput. Eng.*, vol. 7, no. 3, pp. 257–268, 2019. doi: 10.17694/bajece.520979
[25] L. A. Ajao, M. O. Olaniyi, J. Agajo, M. A. Olutoye, and A. O. Ajao, "An enhanced telematics-based automobile tracking and volume monitoring system for the supply chain sustainability in the petroleum industry," 2023.
[26] O. Arsenyeva, T. Romanova, M. Sukhonos, I. Biletskyi, and Y. Tsegelnyk, Eds., *Smart Technologies in Urban Engineering. STUE 2023*, Lecture Notes in Networks and Systems, vol. 807, Springer, Cham, 2023. doi: 10.1007/978-3-031-46874-2_40
[27] M. Sangeetha, "Smart supply chain management using Internet of Things," *Int. J. of Systems, Control and Communications*, vol. 9, no. 2, 2018.
[28] O. Arsenyev *et al*., *Smart Technologies in Urban Engineering: Proceedings of STUE-2023, Volume 1*, Springer, 2023.
[29] M. El *et al*., "Logistics tracking system based on decentralized IoT and blockchain platform," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 23, no. 1, pp. 421–430, 2021. doi: 10.11591/ijeecs.v23.i1.pp421-430
[30] S. K. Nanda *et al*., "Medical supply chain integrated with blockchain and IoT to track the logistics of medical products," *Multimedia Tools and Applications*, 2023. doi: 10.1007/s11042-023-14846-8
[31] A. Cammarano *et al*., "Blockchain as enabling factor for implementing RFID and IoT technologies in VMI: a simulation on the Parmigiano Reggiano supply chain," *Operations Management Research*, 2023. doi: 10.1007/s12063-022-00324-1
[32] M. El *et al*., "Logistics tracking system based on decentralized IoT and blockchain platform," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 23, no. 1, pp. 421–430, 2021. doi: 10.11591/ijeecs.v23.i1.pp421-430
[33] V. Varriale, A. Cammarano, F. Michelino, and M. Caputo, "Sustainable supply chains with blockchain, IoT and RFID: A simulation on order management," *Sustainability*, vol. 13, no. 11, p. 6372, 2021. doi: 10.3390/su13116372
[34] V. Varriale *et al*., "Integrating blockchain, RFID and IoT within a cheese supply chain: A cost analysis," *Journal of Innovation & Knowledge*, vol. 8, no. 1, p. 100486, 2023. doi: 10.1016/j.jik.2023.100486
[35] L. A. Ajao *et al*., "A secure tracking automobile system for oil and gas distribution using telematics and blockchain techniques," *Balkan Journal of Electrical & Computer Engineering*, vol. 7, no. 3, pp. 257–268, July 2019.
[36] M. Hussain *et al*., "Blockchain-based IoT devices in supply chain management: A systematic literature review," *Sustainability*, vol. 13, no. 24, p. 13646, 2021. doi: 10.3390/su132413646
[37] Y. Khan *et al*., "Application of Internet of Things (IoT) in sustainable supply chain management," *Sustainability*, vol. 15, no. 1, p. 694, 2023. doi: 10.3390/su15010694
[38] V. Vadi *et al*., "Smart logistics: A blockchain-IoT-based solution for supply chain transparency," *Journal of Physics: Conference Series*, vol. 2007, p. 012072, 2021.

APPENDIX A

Links to Wokwi and Tinkercad for Ultrasonic sensor-based simulations.

1. https://wokwi.com/projects/421718864317214721
2. https://www.tinkercad.com/things/6GuHVMSwUdL-tremendous-trug-elzing/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard