IoT BASED INVENTORY STOCK VERIFICATION

A PROJECT REPORT

Submitted by

THARUNIKA K	710021106037
ANUSRI S	710021106318
SRI HARIHARAN B	710021106305

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS, COIMBATORE

ANNA UNIVERSITY: CHENNAI – 600 025

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Project report titled "**IoT BASED INVENTORY STOCK VERIFICATION**" is the bonafide work of "THARUNIKA K (710021106037), ANUSRI S (710021106318), SRIHARIHARAN B (710021106305)" who carried out the project work under my supervision

Dr. V.R. Vijaykumar M.E, Ph.D.,	**Ms. S. Manju M.E,**
Head of the Department,	Supervisor,
Associate Professor,	Department of ECE,
Department of ECE,	Anna University Regional Campus,
Anna University Regional Campus,	Coimbatore-641 046
Coimbatore – 641046.	

ACKNOWLEDGEMENT

First and foremost, we place this project work on the feet of **GOD ALMIGHTY** who is the power of strength in each step of progress towards the successful completion of the project.

We owe wholehearted sense of reverence and gratitude to our Guide **Ms. S. MANJU M.E.,** and Head of the Department of ECE, for his efficient and excellent guidance, inspiring discussions insightful and timely encouragements for the successful completion of the project.

I thank **Dr. M. SARAVANAKUMAR MBA., Ph.D.,** Dean Regional Campus, Anna University Regional Campus, Coimbatore for his great support with blessings. I also extend my heartfelt thanks to all Faculties and staff of ECE department who have rendered their valuable help in making this project successful.

THARUNIKA K	710021106037
ANUSRI S	710021106318
SRI HARIHARAN B	710021106305

ABSTRACT

Inventory management refers to the calculation of the available stocks. Inventory management is implemented in the various stages of the production line. It is a necessary tool for the ever-growing industries, whose demands for products are growing and delays cause a major problem. This system is not only limited to industries, but are extended to medical and other fields. The presented design is cost- effective and can be implemented to address simpler or complex issues. The complex issues involve the stock management of wheat, barley, and other cereal manufacturing plants which require high efficiency. Our design can also be used for stock management of liquids as well, such as in detergents, soft drinks, and more. In medical applications, they are used in efficient storage and stock updating of drugs. The technology used for this efficient stock management is based on Internet of Things. IOT is basically a physical network of components with embedded software, electronic concepts, and components. It allows various components to be sensed and remotely maintained by a network infrastructure. It also facilitates integration amidst the physical space and computer-like systems.The idea utilizes the ultrasonic sensor to connect with Arduino UNO, to measure the inventory and send a message to the supplier and/or to the company personnel about the available stocks, as well as display the present stock availability details on an excel sheet using RFID.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

1	RFID	RADIO FREQUENCY IDENTIFICATION
2	GSM	GLOBAL SYSTEM FOR MOBILE COMMUNICATION
3	IDE	INTEGRATED DEVELOPMENT ENVIRONMENT

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In our design, we strive to improve the performance, yet maintain the simplicity of the system. Here, we use the power of IOT to simplify and make it efficient by eliminating any unnecessary human interference and automating the entire network responsible for inventory management. The dividing line between the pre-existing design and the proposed design is the use of dedicated hardware for Inventory management. The other ingenious idea incorporated in our design that sets it apart is the effective utilization of the ultrasound transducer to measure the inventory. Since our design has a dedicated hardware, it can run on batteries. This is effective when the system is installed in industries that do not depend mainly on electric power for its operation. In the presented design, an ultrasonic transducer is implemented to measure the stocks available.

1.2 PROBLEM STATEMENT

In traditional industrial settings, stock verification processes are predominantly manual, leading to inefficiencies, inaccuracies, and increased labor costs. The reliance on human intervention for stock monitoring, tracking, and verification poses challenges such as delayed response times, human errors, and the inability to provide real-time insights into inventory status. These limitations can result in stockouts, overstock situations, and ultimately impact the overall operational efficiency of industrial facilities.

Fig.1 STOCK INVENTORY

1.3 OBJECTIVE

- Enable real-time tracking of inventory levels and movements to provide instant visibility into stock status.
- Enhance the accuracy of stock verification by minimizing human errors associated with manual data entry and verification processes.
- Implement automated data collection through IoT sensors and devices to reduce reliance on manual input and ensure data integrity.
- Optimize operational costs by reducing labor expenses related to manual stock verification and minimizing losses from stockouts or overstock situations.
- Improve overall supply chain efficiency by streamlining stock verification processes and enabling quicker decision-making based on real-time data.
- Optimize the energy consumption of IoT devices to ensure sustainable and cost-effective operation of the system.

1.4 CHAPTER ORGANIZATION

Chapter 2: Deals with existing system that some authors have already did for IoT based Inventory Stock Verification System.

Chapter 3: Deals with the existing technology that has been created to Track the Live Location of Buses.

Chapter 4: Deals with proposed methodology that has been created to overcome the existing technology to improve the verification process. Explains the various controller and their types used in the system. Deals the result and discussion about the project and its efficiency.

Chapter 5: Concludes with contribution made in the report.

CHAPTER 2

LITERATURE SURVEY

2.1INTRODUCTION

Inventory management is an essential aspect of supply chain management, and the use of Radio Frequency (RF) technology can significantly improve inventory management systems. RF technology allows for real-time data collection and analysis of inventory levels, which can help optimize inventory management processes. The literature on inventory management systems using RF technology highlights the advantages of using RF devices such as handheld scanners and RFID tags to track inventory levels and movements. RFID tags can be attached to products, and their movements can be monitored as they move through the supply chain, providing real-time data on inventory levels and location.

2.2 PREVIOUS WORK

Apoorva Verma et.al, in “Smart Shelf for Smart Kitchen”, has proposed the Smart Shelf is capable of measuring grocery items. The Smart Shelf consists of two different sections, weight sensing and level sensing. Level sensing section consists of fixed size container having RFID tag defining container size with product description, RFID tag reader and an ultrasonic sensor for measuring the level of the content of the container. Weight sensing consists of the RFID tag with similar container specification and content identification, RFID tag reader and weight sensor measuring all contents on the shelf .

Xiaojun Jing and Peng Tang, in “Research and Design of the Intelligent Inventory Management System Based on RFID” proposed a system in which RFID is integrated along with ZIGBEE. ZIGBEE focus on sensing indicator in limited area and RFID can identify the character and location of some sensing node [2]. The main disadvantage of RFID is that it has a limited detection range. Also if multiple loads are on the same palette than it is difficult for the RFID reader to read signal precisely [3].

S. Jayanth et.al, in “Inventory Management System using IoT”. In this paper, the Ultrasonic sensor is used to measure the inventory. The Ultrasonic sensor is used to measure the time taken for a pulse to travel from the top of the container to the surface of the filled container and return back. The time is used to determine the distance from the top of the container to the surface of the inventory. If the value is less than the threshold value than email is sent to the supplier [3].

Jyotir Moy Chatterjee et.al, in “Internet of Things based system for Smart Kitchen” described Smart Kitchen using IoT system with multiple sensors, which includes weight sensor, gas sensor and temperature sensor. The gas sensor is used to detect the leakage of gas in the system, the weight sensor is used to detect the weight of the cylinder. A temperature sensor is used to detect the current room temperature. The threshold value is set into the kitchen, when it crosses the values it will send a notification to the user, about the weight of a gas cylinder and also leakage of the gas cylinder. The server can communicate with the user through an android.

2.3 SUMMARY

Literature survey regarding IoT based industrial stock verification has been carried out. In this, we searched 40 papers and we shortlisted to some 5 papers. Next chapter consists of our proposed system.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION

This project aims to explore and implement an IoT-based solution that will revolutionize the current manual stock verification practices in industrial settings. The proposed system will leverage sensors and connected devices to continuously monitor inventory levels, track stock movements, and provide instant updates to the central management system. The integration of IoT will not only enhance the accuracy and speed of stock verification but also enable predictive analytics for demand forecasting and preventive maintenance of equipment.

3.2 EXISTING SYSTEM

One of the existing systems for IoT-based smart inventory management is the RFID (Radio Frequency Identification) system. In this system, RFID tags are attached to the inventory items, and RFID readers are used to detect the presence and level of these items. The data is then transmitted wirelessly to a central server, where it is analyzed and used to manage inventory levels. If any product is taken without scanning, this system will not be able to update inventory. So our proposed system developed various sensors like ultrasonic for detect Presence of Products.

3.3 BLOCK DIAGRAM

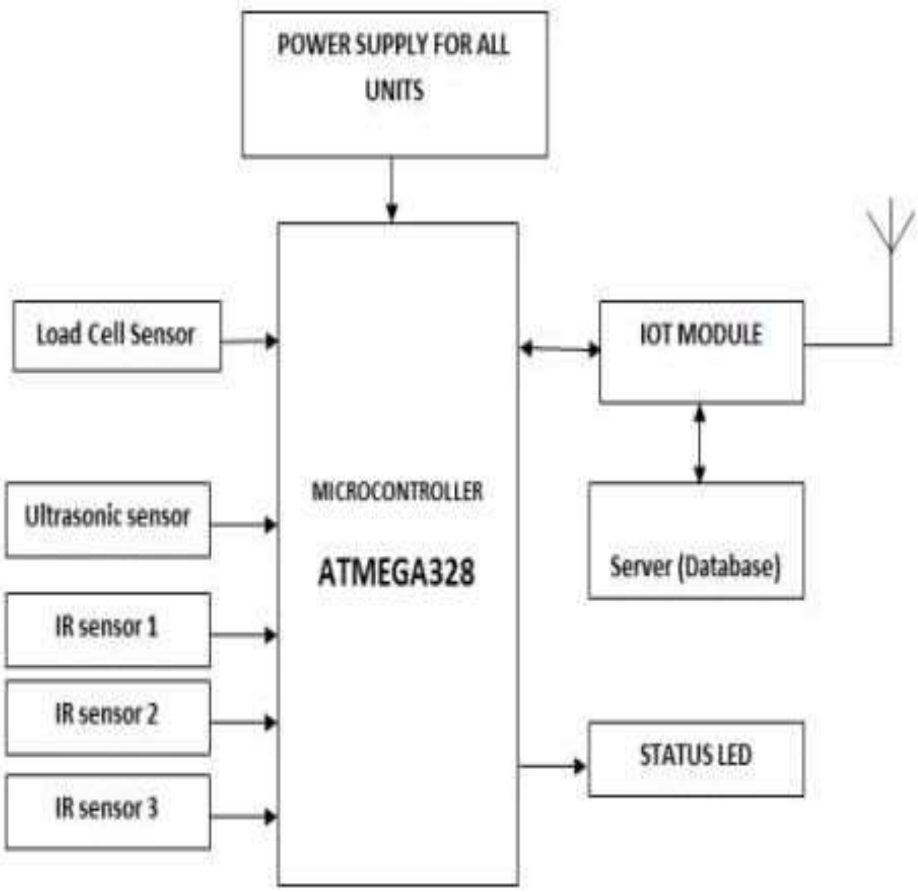


Fig.3.3. BLOCK DIAGRAM OF PROPOSED SYSTEM

The given block diagram shows a microcontroller-based system with sensors and a server. The power supply provides power to all the units in the system. The load cell sensor measures the weight of objects, the ultrasonic sensor measures the distance to objects, and the IR sensors detect the presence of objects. The microcontroller (ATMEGA328) collects data from the sensors and processes it. The IoT module sends the processed data to the server, where it is stored and analyzed. The status LED indicates the status of the system.Power supply: Provides power to all units in the system. Load cell sensor: Measures the weight of objects. Ultrasonic sensor: Measures the distance to objects. IR sensors: Detect the presence of objects, Microcontroller (ATMEGA328): Collects and processes data from sensors, IoT module: Sends processed data to server, Server: Stores and analyzes data, Status LED: Indicates the status of the system.

3.4CONCLUSION

This economical system employs ultrasonic sensors to gauge both solid and liquid supplies. Positioned atop containers, these sensors streamline installation. The system's automated SMS alerts to vendors reduce human error. With a threshold that guarantees operational stocks until restocking, it is self-sufficient and forestalls delays. Perfectly suited for hospitals and industries, its low cost, straightforward implementation, and efficient design make it adaptable.

CHAPTER 4

PROPOSED SYSTEM

4.1 INTRODUCTION

IoT based industrial stock verification management system is a modern approach to managing inventory that leverages the power of the internet of things (IoT) technology. The system uses a network of interconnected devices, sensors, and software to track inventory levels, monitor product movement, and optimize inventory levels in real time. This IoT based system works by collecting and analyzing data from ultrasonic sensors and RFID system, including point-of-sale systems, and other connected devices. This data is then processed and analyzed to provide real-time insights verification and management.

4.2 PROPOSED SYSTEM

The proposed system for IoT-based smart inventory management using Arduino Uno, Ultrasonic sensors, is an advanced solution for businesses looking to manage their inventory efficiently. The proposed system has been designed to tackle the challenges faced by businesses in managing their inventory efficiently. The system comprises of several components that work together to ensure accurate, real-time inventory management. These sensors are placed at strategic locations throughout the storage facility, and they trigger when an item is added or removed from the inventory.

4.3 METHODOLOGY

- In this proposed method we implement sensor technology in inventory to monitor the available stocks and its count
- Ultrasonic detects the number of stocks available in a particular inventory.
- The GSM Module sends alert messages time to time if it is either overstocked or understocked.
- The data collected from RFID are collected automatically in an Excel sheet.

4.4 WORKING

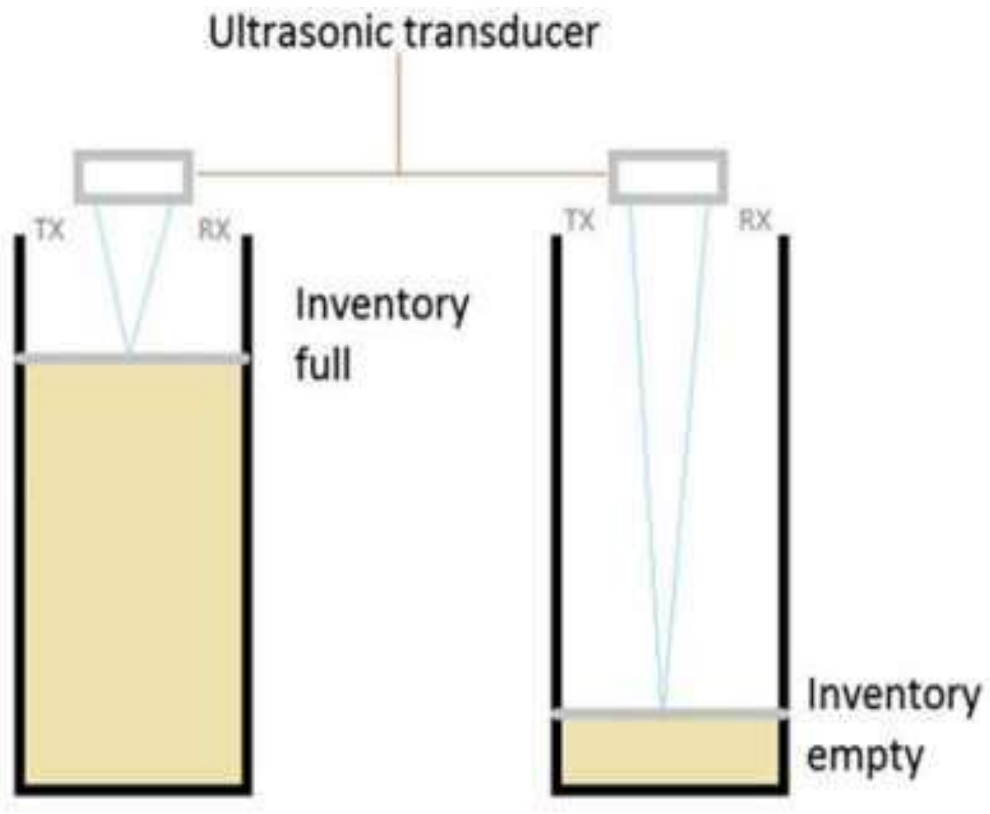


Fig.4. ULTRASONIC TRANSDUCE

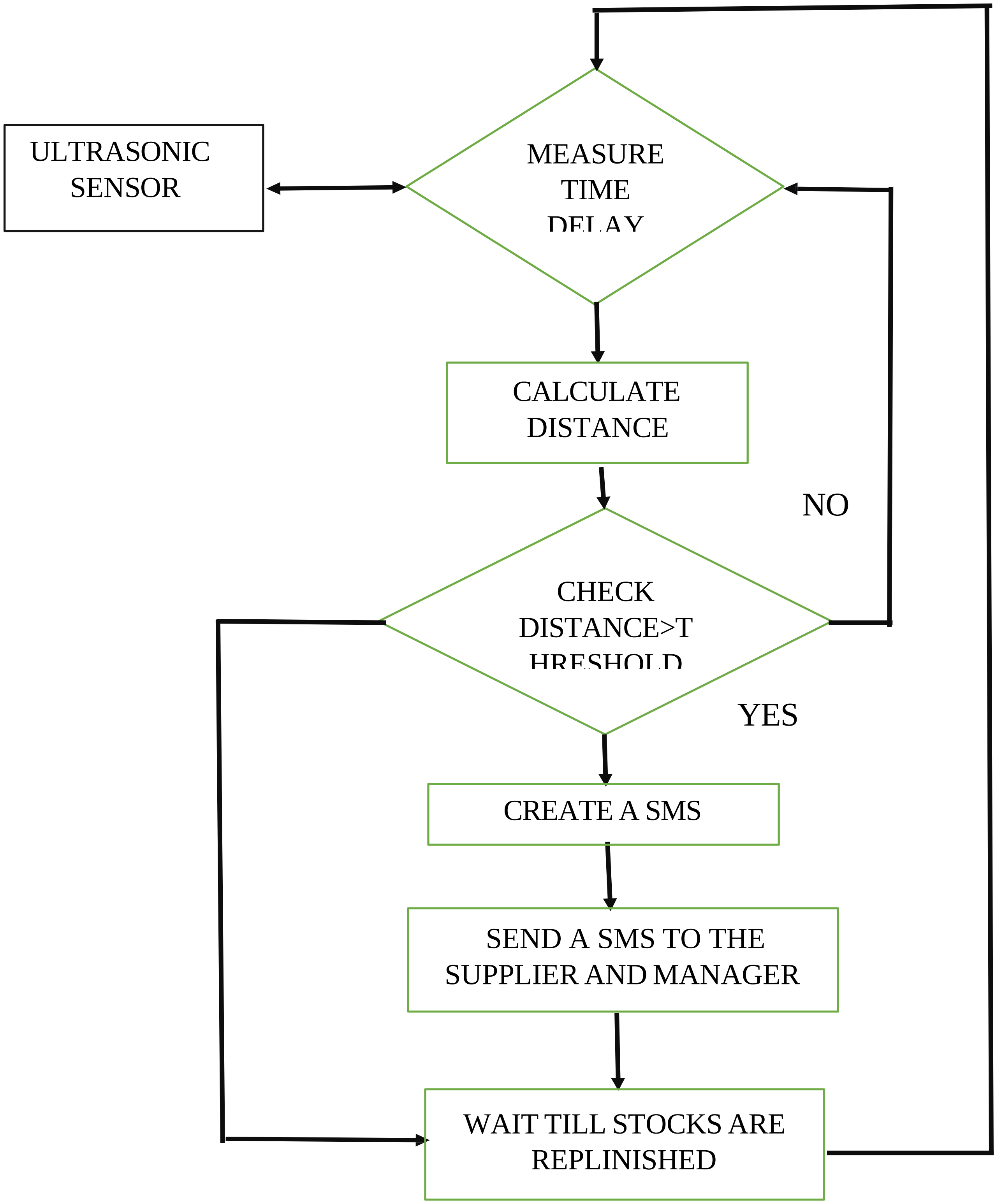


Fig.4.2FLOW CHART

4.5 WORK FLOW DESCRIPTION

The sensor is used to measure the time taken for a pulse to travel from the top of the container to the surface of the filled container and return back. This time is used to determine the distance from the top of the container to the surface of the inventory. By assuming two values, i.e., max and threshold, where max is the distance for full inventory and the threshold is the distance for acceptable minimum inventory. It is clear that max < threshold, since the distance increases with decrease in stocks. The threshold value to be so chosen that the industry should be capable of functioning till the new goods arrive. The stock measurement occurs. The heart of the system is Arduino UNO, which is used for interfacing to the ultrasonic sensor to determine the time.

Speed = Distance/Time.

where speed refers to the speed of sound in air Speed = 330 m/s Hence, the

above equation becomes

34000 = Distance/(Time/2)

17000 = Distance/Time Distance =

Time * 17000

4.6 CONCLUSION

This cost-effective system utilizes ultrasonic sensors for measuring both solid and liquid stocks. Placed on top of containers, these sensors simplify mounting. The system's automated SMS notifications to suppliers minimize human errors. With a threshold ensuring operational stocks until replenishment, it's self-sustaining and prevents delays. Ideal for hospitals and industries, its low cost, easy implementation, and efficient design make it versatile.

CHAPTER 5
HARDWARE REQUIREMENTS

5.1 ULTRASONIC SENSORS

The ultrasonic sensor is a 4-pin multi-use sensor. It can be used as an obstacle avoider and distance calculator. Ultrasonic sensors **measure distance by using ultrasonic waves**. It emits an ultrasonic wave and receives the wave reflected from the target. The distance can be calculated with the following formula:

Distance $D = 1/2 \times T \times C$ Where T=

Time and C= Speed

The idea is to use the ultrasonic sensor to measure the distance between
the sensor and the boxes that are stack. Based on this value of distance we can predict how much box is currently stacked in a row if we know the width of one box

Fig 5.1 ULTRASONIC SENSOR

5.2 ARDUINO UNO

Arduino Uno REV 3 is an Arduino board based on the microcontroller ATmega328P. It carries 14 digital I/O pins out of which 6 can be used as PWM outputs. Moreover, 6 analog input pins are available on the board and the clock frequency is 16MHz. Arduino UNO is one of the most used boards from the Arduino family. The robust and clean design helps you shape your ideas into reality. Know that Arduino UNO REV3 is an advanced version of Arduino UNO. The new version includes four solder pads JP2 attached with the pins PB4 to PB7 of the USB AT mega. Uno stands for one in Italian and this name was picked for the release of Arduino Software (IDE) 1.0. The version 1.0 of Arduino Software (IDE) and Uno board both are considered as the reference versions of Arduino, which evolved with time with new features. The UNO board is the first USB board from the Arduino family. Arduino is an open- source platform which means you can get a hold of Arduino boards and software and edit and modify them as per your requirements. Arduino IDE software is free to use for anyone, moreover, as you join this platform you can get help from the Arduino community

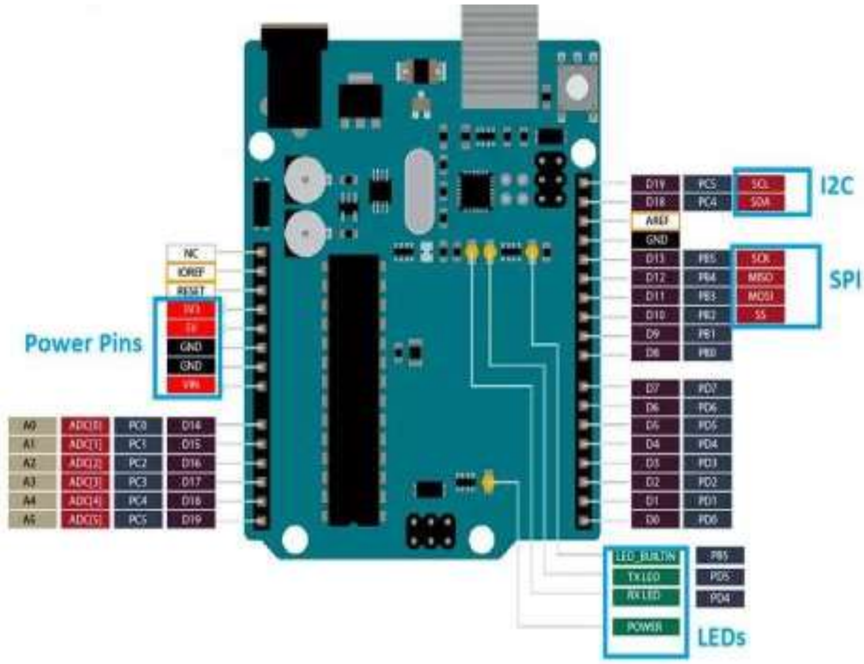


Fig 5.2 PIN CONFIGURATION OF ARDUINO UNO REV

5.3 RFID SYSTEM

The inventory stock verification management system is one of the applications of RFID technology, because RFID can uniquely identify products or goods attached with tags. The RFID reader em-18 used to read 125 kHz tags. The RFID reader operates at a voltage of 5 V. The Reader has two fundamental objectives: initially reader to transmit a carrier signal, and the subsequent is to receive a response from any tags in the proximity of the reader. The RFID reader emits a shortrange radio signal which is picked by a RFID tag and the tag is triggered on. The RFID reader em-18 is attached at the beginning of the stockroom read the tags. The RFID em-18 works on UART. The em-18 RFID reader is interfaced to Arduino UNO and is programmed by using Arduino IDE. The passive tags are adhered to the products or goods to uniquely identify them. As soon as the RFID reader scans the tag, then the tag transmits back a short string of data. The passive tags 125 kHz are more beneficiary over active and semi-passive tags because of its low cost, low power consumption, no interference effect. The unique ID of the tag contains 12 bytes of data 0D0021A8D256. The last two digits indicate the checksum which is the result of XOR operation of first ten bytes. The data scanned by the reader is used to decode the encoded data in the tag's IC and the data is collected in excel sheet.

Fig 5.3 RFID READER AND TAGS

5.4 GSM MODULE 800L

The SIM800L GSM/GPRS module is a miniature GSM modem that can be used in a variety of IoT projects. You can use this module to do almost anything a normal cell phone can do, such as sending SMS messages, making phone calls, connecting to the Internet via GPRS, and much more. To top it all off, the module supports quad-band GSM/GPRS networks, which means it will work almost anywhere in the world. At the heart of the module is a SIM800L GSM cellular chip from simcom. The operating voltage of the chip ranges from 3.4V to 4.4V, making it an ideal candidate for direct LiPo battery supply. This makes it an excellent choice for embedding in projects with limited space.

In this project, we are using this module to send alert messages regarding the stock is either full or empty to the individual person who is verifying the stock by interfacing with Arduino UNO and programmed using Arduino IDE.

Fig 5.4 GSM MODULE 800L

CHAPTER 6
SOFTWARE DESCRIPTION

6.1 INTRODUCTION

This chapter deals with the description of the software used for hardware working. Here the system uses Arduino IDE as programming software. Arduino is an integrated development environment for Embedded C that is designed for beginners. It supports different ways of stepping through the code, step- by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap

6.2ARDUINO IDE

Arduino IDE (Integrated Development Environment) is a software tool used for programming and development of Arduino boards. It is an open-source platform, available for free, and is compatible with multiple operating systems including Windows, Mac OS, and Linux. The main features of the Arduino IDE include:

- **Code Editor:** The code editor is the main interface of the Arduino IDE, where you can write, edit and upload code to the Arduino board. It includes features such as syntax highlighting, auto-completion, and code snippets to make programming easier.

- **Sketches:** Arduino programs are referred to as "sketches" and can be easily created and saved within the IDE. The sketch contains two main functions: the setup () function, which is called once at the start of the program, and the loop () function, which is called repeatedly as long as the program is running.

- **Library Manager:** The Library Manager allows users to easily install and manage libraries for their Arduino projects. It includes a collection of pre- built libraries that can be used to add functionality to your projects. Users can also create their own libraries and add them to the IDE.

- **Serial Monitor:** The Serial Monitor allows users to communicate with the Arduino board and monitor the data being sent and received through the serial port. This is particularly useful for debugging and troubleshooting.

- **Board Manager:** The Board Manager lets users to select the type of Arduino board they are using, configure settings, and install the required drivers. This is important because different Arduino boards may have different specifications and require different drivers.

- **Upload:** The Upload feature allow users to upload their sketches to the Arduino board and begin executing the program. Users can select the correct board and serial port before uploading the sketch.

- **Tools:** The Tools menu includes a range of options for configuring and customizing the IDE. This includes options for setting the board type, serial port, programmer, and other settings.

6.3 CONCLUSION

Overall, the Arduino IDE is a user-friendly software tool that simplifies the programming process for beginners and experienced users alike. It is compatible with a wide range of Arduino boards and shields, making it a versatile tool for a variety of applications. With its many features and community support, the Arduino IDE is an essential tool for anyone interested in electronics and programming.

CHAPTER 7
EXPERIMENTAL ANALYSIS AND RESULT

7.1 HARDWARE SETUP

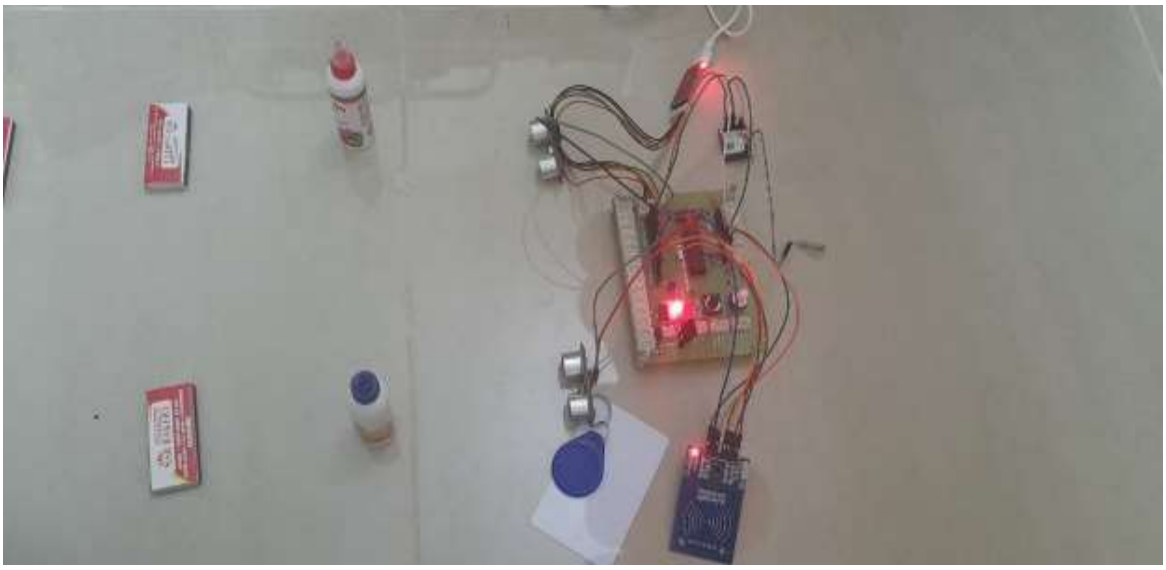
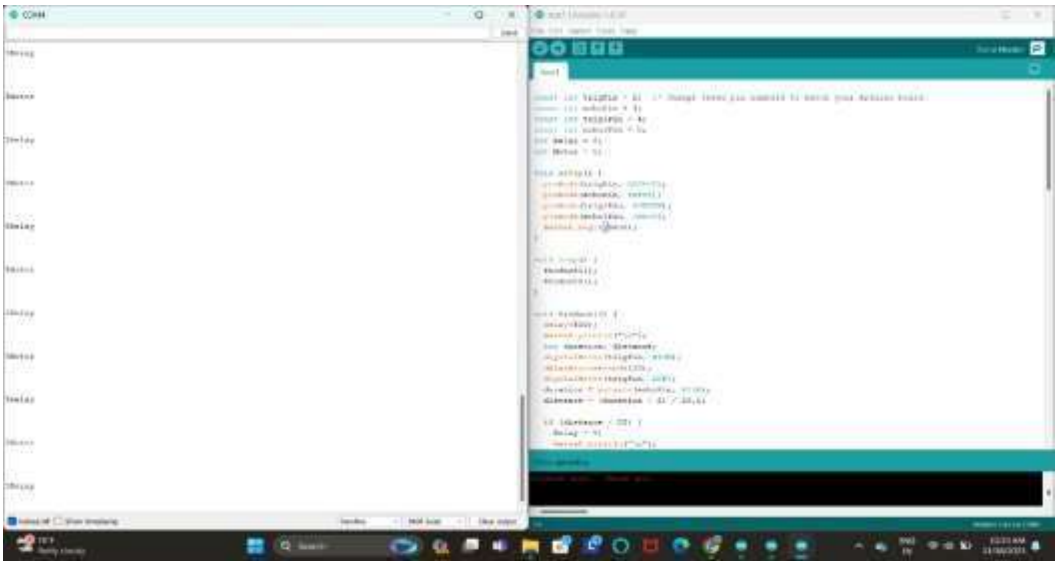


Fig7.1.1HARDWARE SETUP

7.2 SOFTWARE RESULTS



7.2.1Fig SOFTWARE RESULTS

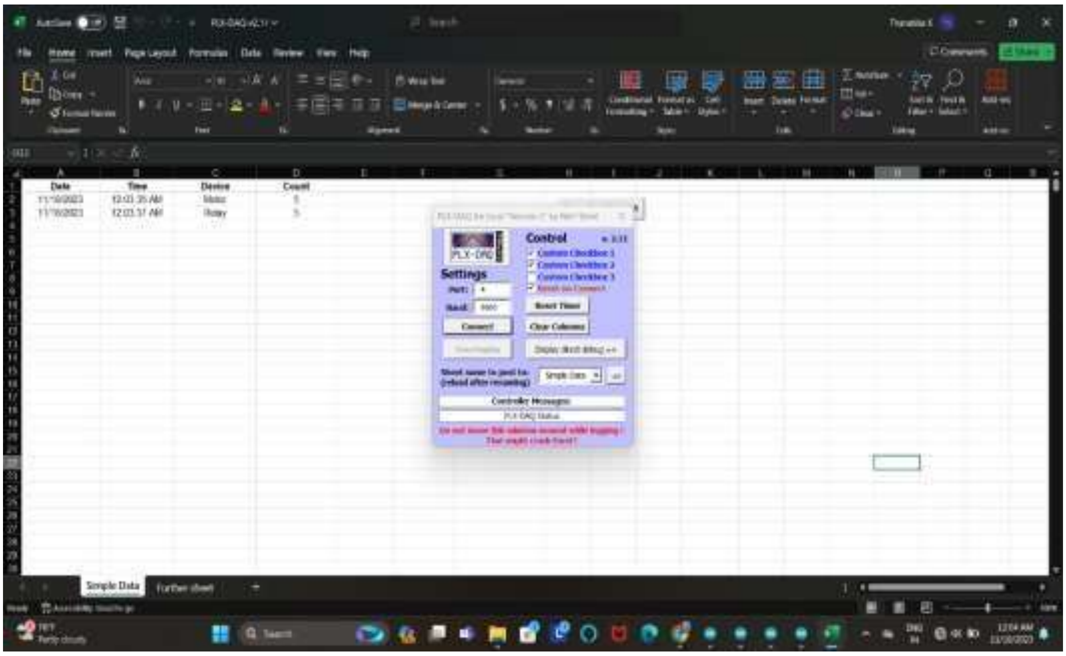


Fig 7.2.2 UPDATED STOCK DATA IN EXCEL

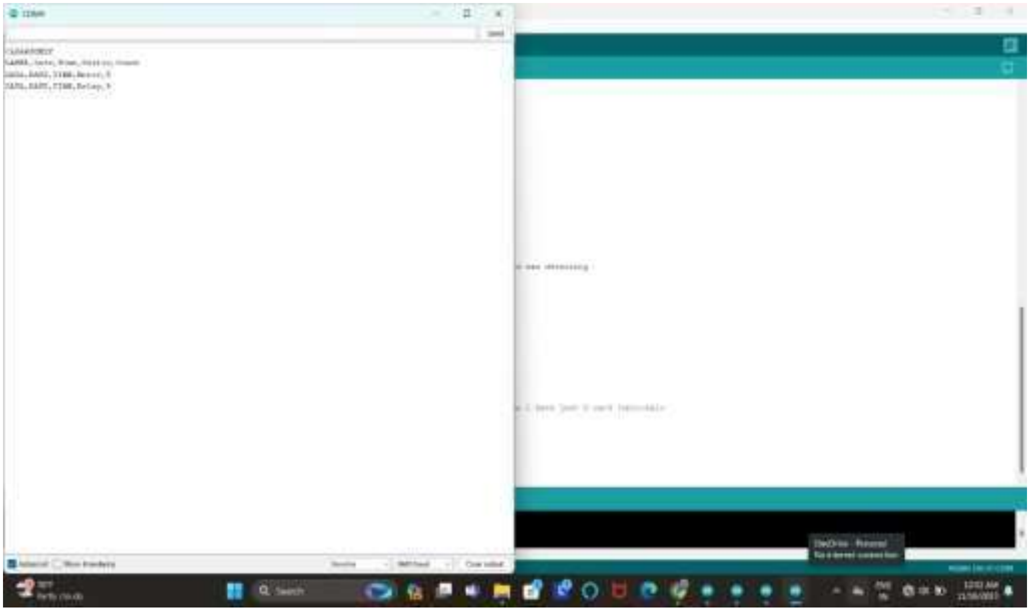


Fig   7.2.3. STOCK COUNT

CHAPTER 8

CONCLUSION & FUTURE WORK

8.1 CONCLUSION

It is evident that this system is cost-effective. Using ultrasonic sensors to measure the stocks, we have simplified the system which can be used for both solid and liquid stocks. Since the ultrasonic sensor needs to be placed on the top of the container, it is simpler and easier to mount. As we can clearly see, the system directly sends SMS to the supplier, thereby reducing human errors. Since the threshold value is so chosen that the stocks are sufficient for operation till the new stocks arrive, the system is self-sustained and there is no delay caused due to insufficient inventory. This will simplify the stock verification process and makes it easier. Because of the low cost, easy implementation, and efficient design, it can be implemented in hospitals, small-scale industries, and large-scale industries, where the limitations are our imagination.

8.2FUTURE WORK

This design can be installed in inventories and we can update the stocks in real time. This system can be further improved by implementing cloud-based solutions allows for centralized data storage, analysis, and accessibility. This enables real-time monitoring of stock levels, facilitates remote management, and provides historical data for more advanced analytics. We can also combine data from multiple sensors, such as incorporating additional types of sensors (e.g., temperature or humidity sensors), to provide a more comprehensive understanding of the inventory environment. This can enhance the accuracy of stock monitoring and contribute to better decision-making.

REFERENCES

1. K. Stravoskoufos and colleagues presented a study that addresses the challenges in the design and implementation of IoT systems by using IoT-A and FIWARE technologies in the proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER) in 2016, The paper is available in the conference proceedings and can be found on pages 146-153.

2. Wartha and V. Londhethat proposes a context aware method for improving the security and privacy of RFID technology in the International Journal of Engineering and Computer Science published a study in 2015 by N. The article is available in volume 4 and includes page numbers 178-88.
3. S.M. Huynh, D. Parry, A. Fong, and J. Tang presented a home localization system designed to help locate misplaced objects. The system is described in their paper, which can be found in the conference proceedings on pages 462-463.

4. M. Bruccoleri, S. Cannella, and G. La Porta that investigates the impact of workers' behavior on inventory record inaccuracies within supply chains in the International Journal of Physical Distribution and Logistics Management published a research article in 2014.The article can be found in volume 44, issue 10, and includes page numbers.

5. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami that outlines the architecture and future prospects of the Internet of Things (IoT) in The Future Generation Computer Systems journal featured an article in September 2013 the article can be identified by its volume and issue number, 29 (7), and includes page numbers 1645-1660.

6. A. Ramaa, K.N. Subramanya, and T.M. Rangaswamy that explores the effects of a warehouse inventory management system on supply chain in The International Journal of Computer Applications published a study in 2012.

APPENDIX

```
#include <GSM.h>
#include <MFRC522.h>
#include <SPI.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance.
const int trigPin = 2;  // Change these pin numbers to match your Arduino board.
const int echoPin = 3;
const int trig1Pin = 4;
const int echo1Pin = 5;
int Motor = 0;
int Relay = 0;
byte card_ID[4];
byte Device1[] = {0xB8, 0x85, 0xE5, 0x12};
byte Device2[] = {0x62, 0x18, 0x12, 0x51};
int NumbCard[2];
int j=0;
String Device;//user name
long Count;//user number
int n ;
GSM gsm;  // Create a GSM object
GSM_SMS sms;  // Create an SMS object

const char* phone_number = "+91 xxxxxxxxxx";

void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(trig1Pin, OUTPUT);
  pinMode(echo1Pin, INPUT);

  Serial.println("CLEARDATA");
  Serial.println("LABEL,Date,Time,Device,Count,RFID UID");
  delay(1000);

  Serial.println("Scan PICC to see UID...");
  Serial.println("");
}
```

```
// -------------------------------------------------------------------
void loop() {
  rfid();
  delay
  Product1();
  Product2();
}
void Product1() {
  delay(500);
  Serial.println("\n");
  int duration, distance;
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;

  if (distance < 23) {
    Relay = 5;
    Serial.println("\n");
    Serial.print(Relay);
    Serial.print("Relay");
  }
  if (distance > 23 && distance < 35) {
    Relay = 4;
    Serial.println("\n");
    Serial.print(Relay);
    Serial.print("Relay");
  }
  if (distance > 35 && distance < 47) {
    Relay = 3;
    Serial.println("\n");
    Serial.print(Relay);
    Serial.print("Relay");
  }
  if (distance > 47 && distance < 60) {
    Relay = 2;
    Serial.println("\n");
    Serial.print(Relay);
    Serial.print("Relay");
  }
  if (distance > 60 && distance < 70) {
    Relay = 1;
    Serial.println("\n");
```

```
    Serial.print(Relay);
    Serial.print("Relay");
  }
  if(distance>70){
    Relay = 0;
    Serial.println("\n");
    Serial.print(Relay);
    Serial.print("Relay");
  }
  if (Relay < 2) {  // Example threshold for low stock
    sendSMS("Low stock alert: Relay is running low!");
  }
  else if (Relay == 5) {  // Example threshold for low stock
    sendSMS("stock alert: Relay stock is full!");
  }
  else if (Relay == 0) {  // Example threshold for low stock
    sendSMS("Empty stock alert: Relay stock is empty!");
  }
}

void Product2() {
  delay(500);
  Serial.println("\n");
  int duration, distance;
  digitalWrite(trig1Pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig1Pin, LOW);
  duration = pulseIn(echo1Pin, HIGH);
  distance = (duration / 2) / 29.1;

  if (distance < 23) {
    Motor = 5;
    Serial.println("\n");
    Serial.print(Motor);
    Serial.print("Motor");
  }
  if (distance > 23 && distance < 35) {
    Motor = 4;
    Serial.println("\n");
    Serial.print(Motor);
    Serial.print("Motor");
  }
  if (distance > 35 && distance < 47) {
    Motor = 3;
```

```
    Serial.println("\n");
    Serial.print(Motor);
    Serial.print("Motor");
  }
  if (distance > 47 && distance < 60) {
    Motor = 2;
    Serial.println("\n");
    Serial.print(Motor);
    Serial.print("Motor");
  }
  if (distance > 60 && distance < 70) {
    Motor = 1;
    Serial.println("\n");
    Serial.print(Motor);
    Serial.print("Motor");
  }
  if(distance>70){
    Motor = 0;
    Serial.print(Motor);
    Serial.print("Motor");
  }
  if (Motor < 2) {  // Example threshold for low stock
    sendSMS("Low stock alert: Motor is running low!");
  }
  else if (Motor == 5) {  // Example threshold for low stock
    sendSMS("stock alert: Motor stock is full!");
  }
  else if (Motor == 0) {  // Example threshold for low stock
    sendSMS("Empty stock alert: Motor stock is empty!");
  }
}
void rfid(){
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
  return;//got to start of loop if there is no card present
 }
 // Select one of the cards
 if ( ! mfrc522.PICC_ReadCardSerial()) {
  return;//if read card serial(0) returns 1, the uid struct contians the ID of the read card.
 }

 for (byte i = 0; i < mfrc522.uid.size; i++) {
     card_ID[i]=mfrc522.uid.uidByte[i];

       if(card_ID[i]==Device1[i]){
```

```
       Device="Relay";
       Count=5;
       j=0;//first number in the NumbCard array : NumbCard[j]
      }
      else if(card_ID[i]==Device2[i]){
       Device="Motor";//user name
       Count=5;//user number
       j=1;//Second number in the NumbCard array : NumbCard[j]
      }
      else{
          Serial.println("Wrong info");
     }
}
      if(NumbCard[j] == 1){//to check if the card already detect
      //if you want to use LCD
      //Serial.println("Already Exist");
      }
      else{
      NumbCard[j] = 1;//put 1 in the NumbCard array : NumbCard[j]={1,1} to let the
arduino know if the card was detecting
      n++;//(optional)
      Serial.print("DATA,DATE,TIME," + Device);//send the Name to excel
      Serial.print(",");
      Serial.println(Count); //send the Number to excel
      }
      delay(1000);
}
void sendSMS(const char* message) {
  // Send an SMS using the GSM module
  Serial.println("Sending SMS: " + String(message));
  if (sms.beginSMS(phone_number)) {
    sms.print(message);
    sms.endSMS();
    Serial.println("SMS sent!");
  } else {
    Serial.println("SMS sending failed.");
  }
}
// ------------------------------------------------------------------
void array_to_string(byte array[], unsigned int len, char buffer[])
{
    for (unsigned int i = 0; i < len; i++)
    {
        byte nib1 = (array[i] >> 4) & 0x0F;
```

```
        byte nib2 = (array[i] >> 0) & 0x0F;
        buffer[i*2+0] = nib1  < 0xA ? '0' + nib1  : 'A' + nib1  - 0xA;
        buffer[i*2+1] = nib2  < 0xA ? '0' + nib2  : 'A' + nib2  - 0xA;
    }
    buffer[len*2] = '\0';
}
```