# Fun-Filled MP3 Player for Endless Entertainment

A course project report submitted in partial fulfilment of the requirement of

## SMART SYSTEM DESIGN

by

JILLA.KIRTHAN (2203A52151)

Under the guidance of

**Mr.B.Girirajan**

Asst. Professor, Department of ECE

**Mr.Y.Srikanth**

Asst. Prof., Department of ECE

# ABSTRACT

The use of an Arduino MP3 player with a distance sensor allows for the creation of interactive projects that respond to the presence or absence of physical objects. These projects typically involve wiring a distance sensor and an MP3 player to an Arduino board, and programming the board to trigger specific audio files based on the distance detected by the sensor. This can be used in a variety of applications, from interactive exhibits to installations and art projects. Some examples of similar projects include using ultrasonic distance sensors to create musical instruments or smart glasses for the visually impaired, and using proximity sensors to trigger MP3 playback in response to motion or objects.

# CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 ABOUT THE PROJECT

An Arduino MP3 player with a distance sensor is an interesting project that combines audio playback and proximity sensing capabilities. The basic idea is to create a device that can play audio files (such as music or recorded sounds) using an Arduino microcontroller and also incorporate a distance sensor to trigger certain actions based on the proximity of objects or people.Certainly! An Arduino MP3 player with a distance sensor, IoT-based project adds the capability to remotely control and monitor the MP3 player using the Internet of Things (IoT) technology. This allows you to interact with the MP3 player. Remember to consult the datasheets, documentation, and example code for the components you're using, as they will provide specific details on their functionalities and how to interface with them using the Arduino. Additionally, always follow safety precautions when working with electronic components and ensure proper power supply and connections.

## 1.2 OBJECTIVES

1.Build an Arduino-based MP3 player: Develop a system that can play audio files from an SD card or a storage device using an Arduino microcontroller and an MP3 player module.

2.Integrate a distance sensor: Incorporate a distance sensor, such as an ultrasonic sensor or an infrared sensor, to detect the proximity of objects or people.

3.Trigger audio playback based on distance: Implement a mechanism that starts playing audio files when an object or person comes within a specified distance range of the sensor.

4.Real-time notifications: Set up notifications to alert the user when an object or person is detected within a certain proximity range. This can be achieved through email alerts, SMS notifications, or push notifications on a mobile app.

5.Interactive user interface: Design a user-friendly interface that allows users to interact with the MP3 player and distance sensor, providing options to select audio files, adjust volume, and view sensor data.

# CHAPTER 2

# PROJECT DESCRIPTION

## 2.3 HARDWARE DESCRIPTION

### 2.3.1 Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit. Atmega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

| | |
|---|---|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |

| | |
|---|---|
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328) EEPROM 1 KB (ATmega328) |

Fig 2.3 Ardino UNO

## Applications:

- Xoscillo, an open-source oscilloscope
- Arduinome, a MIDI controller device that mimics the Monome
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars • Gameduino, an Arduino shield to create retro 2D video games
- ArduinoPhone, a do-it-yourself cellphone
- Water quality testing platform
- Automatic titration system based on Arduino and stepper motor
- Low cost data glove for virtual reality applications
- Impedance sensor system to detect bovine milk adulteration
- Homemade CNC using Arduino and DC motors with close loop control by Homofaciens,DC motor control using Arduino and H-Bridge.

### 2.3.2 MP3 SERIAL PLAYER:


Fig.2.3.2 Mp3 Serial Player

An MP3 serial player is a module that allows you to play audio files in the MP3 format using a serial communication interface. It simplifies the process of audio playback by handling the decoding and output of audio signals, making it easy to integrate into various electronic projects. Here's some information about the MP3 serial player and its applications.An MP3 serial player typically consists of a microcontroller with built-in MP3 decoding capabilities, a storage interface (such as an SD card slot or onboard flash memory), and audio output ports (usually analog or digital). The module communicates with a host device, such as an Arduino or other microcontrollers, through a serial protocol (commonly UART or SPI).DIY audio projects: MP3 serial players are widely used in do-it-yourself audio projects, such as building custom sound systems, home automation voice prompts, interactive installations, or musical instruments that require audio playback.Robotics and animatronics: MP3 serial players find applications in robotics and animatronics projects where pre-recorded audio cues or voice responses are needed. They can be used to give robots or animatronic characters the ability to play specific sound effects or provide audio feedback.IoT devices: With the integration of IoT technologies, MP3 serial players can be incorporated into smart home systems or voice-enabled devices. For example, an IoT-based doorbell system can use an MP3 player to play different melodies or personalized messages based on user preferences.Educational projects: MP3 serial players can be utilized in educational projects to create interactive learning experiences. For instance, they can be used in museum exhibits, interactive books, or educational toys to provide audio instructions or narrations.Advertising and marketing: MP3 serial players can be used in promotional displays or marketing campaigns to deliver targeted audio messages. They can be integrated into point-of-sale displays, kiosks, or interactive advertisements to attract attention and engage customers.

Key Features:

1.File format support

2.Storage capacity

3.Control options.

4.Audio output

### 2.3.3 DISTANCE SENSOR


Fig 2.3.3 Pulse Sensor

## Distance Sensor Specifications :

A distance sensor is an electronic device designed to measure the distance between the sensor and an object or surface. It uses various technologies, such as ultrasonic waves, infrared light, or laser beams, to calculate the distance based on the time taken for the signal to travel and return.There are several types of distance sensors available, including ultrasonic sensors, infrared (IR) sensors, laser distance sensors, and time-of-flight (ToF) sensors. Each type has its own advantages and is suitable for different applications.

Here are some common applications of distance sensors:

1.Proximity detection: Distance sensors are commonly used for proximity detection in various systems. For example, in robotics, they enable robots to navigate and avoid obstacles by detecting objects in their path. Proximity sensors are also used in automated doors, elevators, and parking systems to detect the presence of individuals or vehicles.

2.Object detection and counting: Distance sensors can be used to detect and count objects passing through a certain area. This is useful in applications such as automated assembly lines, traffic monitoring systems, and people counting in public places like shopping malls or museums.

3,Level measurement: Distance sensors are utilized in industrial settings to measure the level of liquids, powders, or bulk materials in tanks or containers. They provide accurate and reliable measurements without the need for direct contact with the substance being measured.

3.Distance monitoring and control: Distance sensors can be employed to monitor the distance between two objects or surfaces and trigger actions based on specific distance thresholds. This can be used in applications like automated parking systems, security systems, and collision avoidance systems in vehicles.

4.Robotic navigation and mapping: Distance sensors, particularly laser distance sensors and ToF sensors, are extensively used in robotics for mapping and navigation purposes. They help robots understand their environment, create maps, and plan their movements accordingly.

5.Augmented reality and virtual reality: Distance sensors are utilized in augmented reality (AR) and virtual reality (VR) systems to provide depth perception and positional tracking. They enable more immersive and interactive experiences by accurately sensing distances in the surrounding environment.

## 2.4 SOFTWARE DESCRIPTION

The software used here is ARDUINO SOFTWARE: The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches: Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ino.

The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port.

The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

13

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the

extension pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the ino extension on save.

Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor. Additional commands are found within the five menus: File, Edit, Sketch, Tools,and help. Programming on arduinouno

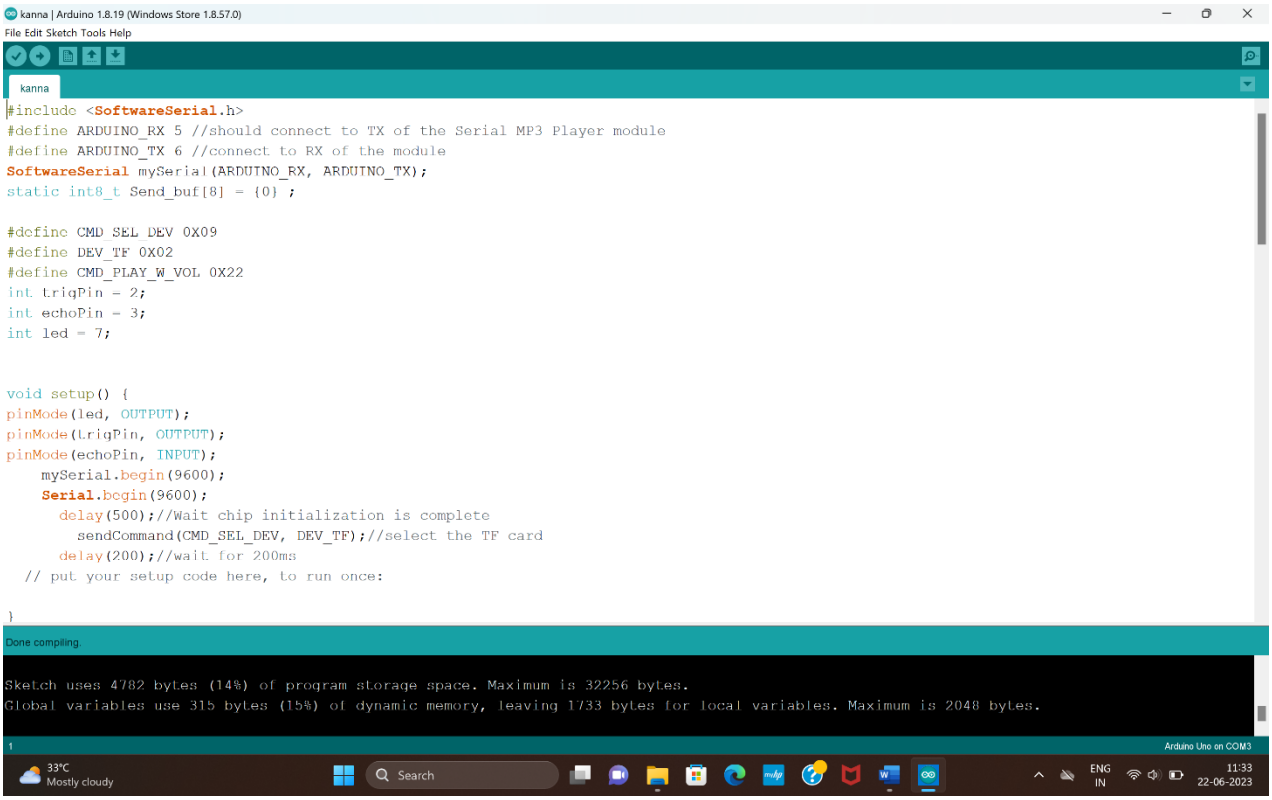


Fig.2.4 Software IDE

In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno. Arduino Company provides user friendly software which allows writing any code for any function wanted to be performed by the Arduino-Uno and upload it to the board.Refer to appendix A for the full source code of the Arduino-Uno board

# CHAPTER 3
# IMPLEMENTATION

## 3.1 CIRCUIT DIAGRAM & CONNECTIONS

For designing IoT Based Building an Arduino Mp3 Player with Distance Sensor, assemble the circuit as shown in the figure below.

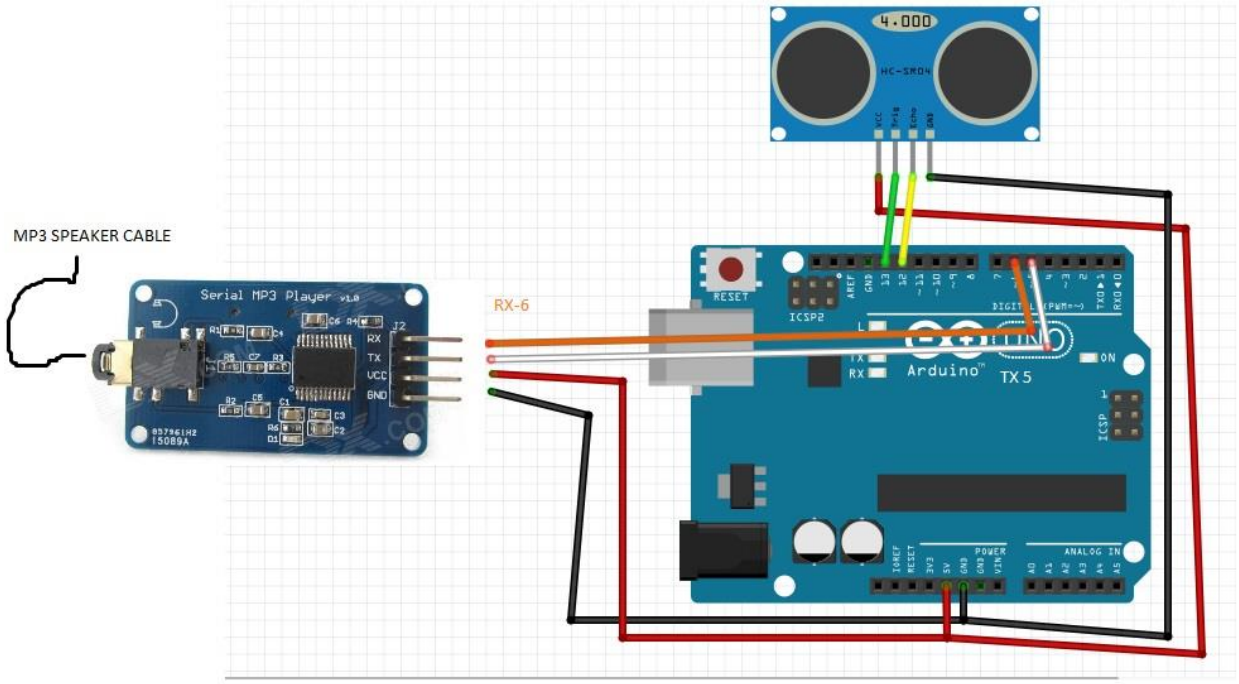


*Fig.3.1 Circuit Diagram & Connections*

1.Connect (trigPin,echoPin) of distance sensor to (13,12) of Arduino board.

2.Connect (RX,TX) of Mp3 Serial Player to (5,6) of Arduino board.

3.Connect distance sensor and mp3 serial player Vcc to 5v .

4.Connect distance sensor and mp3 serial player Gnd to Gnd of Arduino board.

## 3.2 RESULTS:

The experimental result is as shown in below fig.3.2

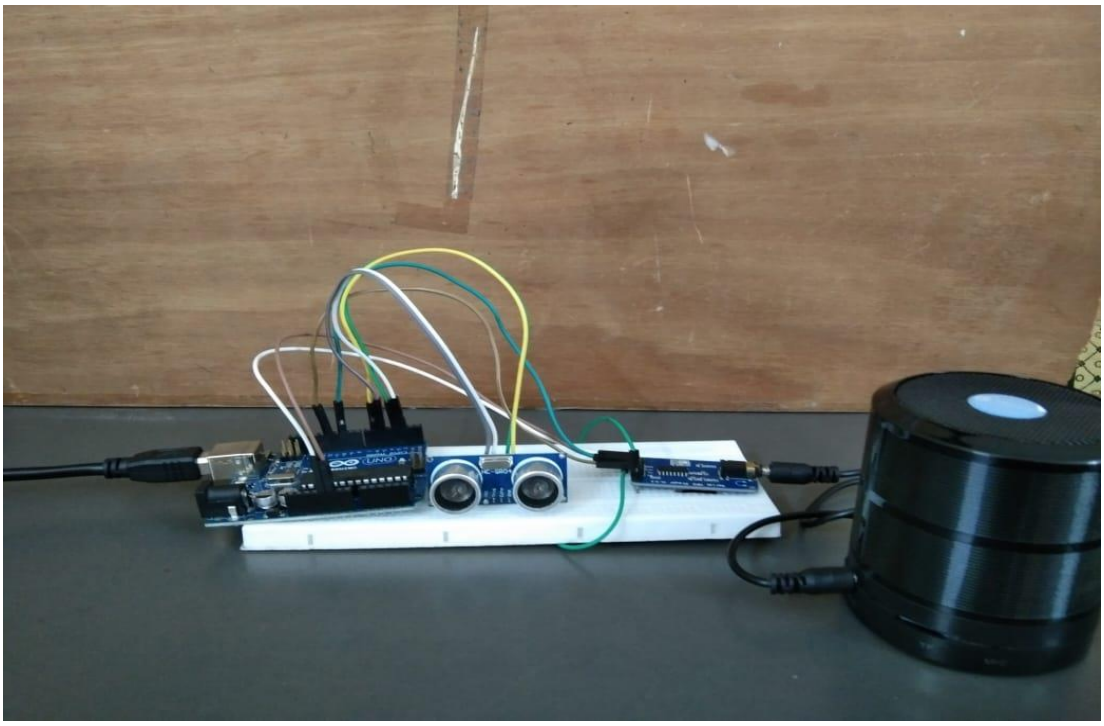

*Fig .3.2 Hardware Implementation of Health Monitoring System*

Managed to successfully apply the IOT based Building an mp3 player with distance sensor using aurdino and it was user friendly and cost effective. User friendly as in anyone can use this wherever they are by simply opening thingspeaks account on an android screen and everything works .And it is cost effective as in it will cost exactly as the project requires(optimum price)

## 3.3 ADVANTAGES

1.Compact and Portable: The Arduino-based MP3 player with a distance sensor can be designed to be small and portable, making it easy to carry and use in various settings.

2.Cost-effective: Arduino boards and components are relatively affordable, making this project a cost-effective solution for creating an MP3 player with distance sensing capabilities.

3.Customizability: Arduino offers a wide range of libraries, modules, and sensors that can be easily integrated and customized according to specific requirements. This allows for flexibility in designing the functionality and features of the MP3 player.

4.Easy Programming: Arduino uses a simple and beginner-friendly programming language, making it accessible to individuals with varying levels of programming experience.

5.Integration with IoT: By incorporating IoT capabilities, the MP3 player can be connected to the internet, enabling remote monitoring, control, and data analysis.

6.Proximity-Based Interaction: The distance sensor allows for proximity-based interaction, triggering actions such as audio playback based on the proximity of objects or people. This can provide an intuitive and interactive user experience.

7.Versatile Applications: The combination of an MP3 player and a distance sensor opens up possibilities for various applications, including interactive installations, smart home automation, personalized audio experiences, and more.

## 3.4 DISADVANTAGES

1.Limited Processing Power: Arduino boards have limited processing power compared to more advanced microcontrollers or single-board computers. This can restrict the complexity and performance of certain features and functionalities.

2.Limited Storage Capacity: Arduino boards typically have limited onboard storage, which can impact the number and duration of audio files that can be stored and played.

3.Lack of Audio Quality: The audio output quality of Arduino-based MP3 players may not match dedicated audio devices or professional audio equipment. The sound quality may be limited due to the hardware capabilities of the Arduino and the MP3 player module used.

4.Dependency on External Power: Arduino boards typically require an external power source, which may limit the mobility and convenience of the MP3 player.

5.Programming and Technical Skills: Building an Arduino MP3 player with a distance sensor requires programming and technical skills. This project may not be suitable for individuals without prior knowledge or experience in Arduino programming and electronics.

6.Reliability and Robustness: The reliability and robustness of the system may depend on the quality of components used, wiring connections, and overall design. Ensuring proper connections and adequate protection against electrical issues is crucial for a stable and durable project.

# CHAPTER 4

# CONCLUSION

In conclusion, the Arduino MP3 player with a distance sensor project combines the capabilities of an MP3 player and a distance sensor, creating a versatile and interactive device. By integrating these components with Arduino and leveraging IoT technology, the project offers a range of exciting possibilities.The combination of an MP3 player and a distance sensor allows for unique functionalities. The project enables automatic audio playback based on proximity, providing an immersive and interactive experience. Users can enjoy personalized soundscapes triggered by objects or people coming within a specific distance range. Additionally, the integration of IoT capabilities allows for remote monitoring and control, adding convenience and flexibility to the system.The Arduino MP3 player with a distance sensor project finds applications in various domains. It can be utilized in smart homes, where it can trigger specific audio responses based on the proximity of occupants or objects. In public spaces, the device can provide interactive audio displays or proximity-based announcements. In robotics, it can enhance navigation and obstacle avoidance capabilities.Through this project, makers and enthusiasts can explore the integration of audio playback, proximity sensing, and IoT technologies. It promotes creativity, learning, and experimentation with Arduino and its ecosystem of sensors, modules, and libraries. The project also encourages documentation and sharing, contributing to the knowledge and inspiration of the Arduino community.Overall, the Arduino MP3 player with a distance sensor project offers an engaging and interactive experience, combining the joy of music with the practicality of proximity sensing. It demonstrates the versatility of Arduino as a platform for creating innovative projects that blend different technologies to address real-world challenges and enhance user experiences.

## FUTURE SCOPE

The future scope of an Arduino MP3 player with a distance sensor is promising, as advancements in technology and the growing popularity of IoT open up new possibilities. Here are some potential future developments and applications for this project:

Smart home integration: With the increasing adoption of smart home technology, an Arduino MP3 player with a distance sensor can be integrated into smart home ecosystems. It could communicate with voice assistants like Amazon Alexa or Google Assistant, allowing users to control audio playback and trigger actions based on proximity using voice commands.

Context-aware audio: By incorporating additional sensors, such as light sensors or motion sensors, the MP3 player can dynamically adjust audio playback based on the

environment. For example, it can increase the volume in a noisy environment or pause playback when no one is present in the room.

Health and wellness applications: The distance sensor can be utilized in health and wellness applications, such as monitoring the distance between a user and their exercise equipment during workouts or providing audio cues for maintaining a safe distance during physical therapy exercises.

Personalized experiences: By leveraging user data and machine learning algorithms, the MP3 player can learn user preferences and adapt the audio playback experience accordingly. It can suggest and play music based on the user's proximity to certain locations or time of day.

Interactive installations and exhibitions: An Arduino MP3 player with a distance sensor can be used in interactive installations or exhibitions, where audio cues or background music can change based on the proximity of visitors. This can enhance the overall experience and engagement of the audience.

Retail and marketing applications: In retail settings, the distance sensor can be employed to trigger audio advertisements or promotions when a customer approaches a particular product or display. This can provide a more interactive and personalized shopping experience.

Educational tools: The Arduino MP3 player with a distance sensor can be utilized as an educational tool for teaching concepts like sound, distance measurement, and IoT. It can be integrated into STEM curriculum or used in maker spaces to engage students in hands-on learning.

Wearable devices: With the miniaturization of components, it may be possible to develop wearable versions of the MP3 player with a distance sensor. This can enable applications like fitness trackers with audio feedback or personalized audio experiences based on proximity.

Accessibility aids: The MP3 player with a distance sensor can be modified to serve as an assistive device for individuals with visual impairments. It can provide audio cues about nearby objects or obstacles, enhancing mobility and navigation.

Collaboration with other IoT devices: As IoT devices continue to proliferate, an Arduino MP3 player with a distance sensor can collaborate and exchange data with other devices, such as smart lighting systems or security cameras. This can enable coordinated actions and enhance overall automation and control.

# BIBLIOGRAPHY


1) Gulraiz J. Joyia, Rao M. Liaqat, Aftab Farooq, and Saad Rehman, Internet of Medical Things (IOMT): Applications, Benefits and Future Challenges in Healthcare Domain, Journal of Communications Vol. 12, No. 4, April 2017.

2) Shubham Banka, Isha Madan and S.S. Saranya, Smart Healthcare Monitoring using IoT. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15, pp. 11984-11989, 2018.

3) K. Perumal, M. Manohar, A Survey on Internet of Things: Case Studies, Applications, and Future Directions, In Internet of Things: Novel Advances and Envisioned Applications, Springer International Publishing, (2017) 281-297.

4) S.M. Riazulislam, Daehankwak, M.H.K.M.H., Kwak, K.S.: The Internet of Things for Health Care: A Comprehensive Survey. In: IEEE Access (2015).

5) P. Chavan, P. More, N. Thorat, S. Yewale, and P. Dhade, "ECG - Remote patient monitoring using cloud computing," Imperial Journal of Interdisciplinary Research, vol. 2, no. 2,2016.

6) Rajeev Piyare (2013), "Internet of Things: Ubiquitous Home Control and Monitoring System Using Android Based Smart Phone", International Journal of Internet of Things.

# APPENDIX:

```
#include <SoftwareSerial.h>
#define ARDUINO_RX 5 //should connect to TX of the Serial MP3 Player module
#define ARDUINO_TX 6 //connect to RX of the module
SoftwareSerial mySerial(ARDUINO_RX, ARDUINO_TX);
static int8_t Send_buf[8] = {0} ;

#define CMD_SEL_DEV 0X09
#define DEV_TF 0X02
#define CMD_PLAY_W_VOL 0X22
int trigPin = 2;
int echoPin = 3;
int led = 7;


void setup() {
pinMode(led, OUTPUT);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
    mySerial.begin(9600);
    Serial.begin(9600);
      delay(500);//Wait chip initialization is complete
        sendCommand(CMD_SEL_DEV, DEV_TF);//select the TF card
      delay(200);//wait for 200ms
  // put your setup code here, to run once:

}

void loop() {
  long duration, distance;
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW);
  duration=pulseIn(echoPin, HIGH);
  distance =(duration/2)/29.1;
  Serial.print(distance);
  Serial.println("CM");
  delay(10);

if(distance >=5 && distance <=10)
 {
    sendCommand(CMD_PLAY_W_VOL, 0X1E01);//play the second song with volume
30 class (step up)(3.7-4.7ft)(115-145cm)
```

```
   delay(50);
    digitalWrite(led, HIGH); // activate LED
    delay(2000);            // pause before continuing in the program
    digitalWrite(led, LOW);  // turn-off LED
    delay(2000);
  }
else if(distance >=13 && distance <=20)
  {
     sendCommand(CMD_PLAY_W_VOL, 0X1E02);//play the second song with volume
30 class (step down)(6ft-8ft)(145-400)
   delay(50);
    digitalWrite(led, HIGH); // activate LED
    delay(2000);            // pause before continuing in the program
    digitalWrite(led, LOW);  // turn-off LED
    delay(2000);            // pause before continuing in the program
  }
else if(distance >=21 && distance <=26)
  {
     sendCommand(CMD_PLAY_W_VOL, 0X1E03);//play the second song with volume
30 class (step up)(3.7-4.7ft)(115-145cm)
   delay(50);
    digitalWrite(led, HIGH); // activate LED
    delay(2000);            // pause before continuing in the program
    digitalWrite(led, LOW);  // turn-off LED
    delay(2000);
  }
else if(distance >=27 && distance <=33)
  {
     sendCommand(CMD_PLAY_W_VOL, 0X1E04);//play the second song with volume
30 class (step up)(3.7-4.7ft)(115-145cm)
   delay(50);
    digitalWrite(led, HIGH); // activate LED
    delay(2000);            // pause before continuing in the program
    digitalWrite(led, LOW);  // turn-off LED
    delay(2000);
  }
else
  {
     digitalWrite(led, LOW); // turn-off LED
     delay(500);            // for display purposes of read-out
  }
} // end of if((distance <=120))

void sendCommand(int8_t command, int16_t dat)
```

```
{
  delay(20);
  Send_buf[0] = 0x7e; //starting byte
  Send_buf[1] = 0xff; //version
  Send_buf[2] = 0x06; //the number of bytes of the command without starting byte and
ending byte
  Send_buf[3] = command; //
  Send_buf[4] = 0x00;//0x00 = no feedback, 0x01 = feedback
  Send_buf[5] = (int8_t)(dat >> 8);//datah
  Send_buf[6] = (int8_t)(dat); //datal
  Send_buf[7] = 0xef; //ending byte
  for(uint8_t i=0; i<8; i++)//
  {
    mySerial.write(Send_buf[i]) ;
  }
}
```