**ADVANCED EMERGENCY VEHICLE ALERT SYSTEM IN URBAN ENVIRONMENT USING LORA SHIELD**

**A PROJECT REPORT**

*Submitted by,*

|  |  |
| --- | --- |
| **ASHOK.T** | **(923320106004)** |
| **REXSHEN.S.V** | **(923320106037)** |
| **SANJAY.A** | **(923320106038)** |
| **SANJAY KUMAR.M** | **(923320106039)** |

*In partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF ENGINEERING**

**IN**

ELECTRONICS AND COMMUNICATION ENGINEERING

**GOVERNMENT COLLEGE OF ENGINEERING, BODINAYAKKANUR**

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2024**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**ADVANCED EMERGENCY VEHICLE ALERT SYSTEM IN URBAN ENVIRONMENT USING LORA SHIELD”** is the bonafide work of **“ASHOK.T (923320106004), REXSHEN.S.V(923320106037), SANJAY.A(923320106038),SANJAY KUMAR.M(923320106039),”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.D.MARY SUGANTHARATHNAM M.E,Ph.D., Mr.M. RAJAMADASAMY M.Tech**

**HEAD OF THE DEPARTMENT SUPERVISOR**

Electronics and Communication Engineering Electronics and Communication Engineer

Government College of Engineering, Government college of Engineering,

Bodinayakkanur-625582 Bodinayakkanur-625582

**Submitted for the ANNA UNIVERSITY examination held at the center Government college of Engineering, Bodinayakkanur on ………………………………...2024**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DECLARATION**

We, hereby jointly declared that the project work titled “**ADVANCED EMERGENCY VEHICLE ALERT SYSTEM IN URBAN ENVIRONMENT USING LORA SHIELD”** submitted to the anna university report viva voce – MAY 2024 to “ **BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING” ,** is the report of original project work done by us under the guidance of **Mr.M.RAJAMADASAMY M.Tech.,** Department of Electronics and Communication Engineering , Government College of Engineering , Bodinayakanur .

**NAME SIGNATURE**

|  |  |
| --- | --- |
| **ASHOK.T** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **REXSHEN.S.V** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **SANJAY.A** |  |

**SANJAY KUMAR.M \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**I certify that the declaration made by the above candidates is true.**

**SIGNATURE**

**Project guide,**

**Mr. M. RAJAMADASAMY M.Tech.,**

Assistant Professor,

Department of ECE,

Government College of Engineering,

Bodinayakanur – 625 582.

## ACKNOWLEDGEMENT

We take this to acknowledge with great pleasure deep satisfaction and convey our gratitude the contribution of many individuals in the successful completion of this project.

We express our profound gratitude to **Prof. Dr.C.VASANTHANAYAKI, M.E., Ph.D., Principal, Government College of Engineering, Bodinayakkanur,** for all the support and encouragement given to us throughout our project.

### We thank Dr. D. MARY SUGANTHARATHNAM M.E., Ph.D.

**Head of the Department Electronics and Communication Engineering,** Government College of Engineering, Bodinayakkanur, for his valuable suggestions throughout our project.

We express our gratitude to our guide **Prof. M. RAJAMADASAMY M.Tech,(Ph.D), Professor of Electronics and Communication Engineering,** Government College of Engineering, Bodinayakkanur, for his guidance and help in doing this project.

Finally, we thank our parents for their financial and moral support and also our friends who helped us in completing our project work successfully.

**ABSTRACT**

Modern vehicles are constructed with interiors that are noticeably quieter than in the past. The quiet cars are to blame for the absence of outside sounds, including sirens. Since cities are congested and noisy, it is challenging for drivers to hear and differentiate the siren over the loud background noise, which makes it challenging for them to drive safely. Furthermore, the presence of large, tall buildings in cities obstructs siren sound transmission. The sound of the siren often goes along the street. The siren sound travels ineffectively around corners. Sound waves do, however, prefer to keep moving along the previously unobstructed route. Sound waves may bounce off of buildings and, to a certain extent, can travel around corners. The Emergency Vehicle Alert System (EVAS) is successful in transmitting a signal beyond of a siren's audible range. The signal might be sent using one of the several easily available communication frequencies. Communication frequencies have a wider transmission range than siren sounds, which can only transmit information over short distances. Amplifiers may be used to reach the busiest downtown areas where tall buildings could interfere with transmission frequencies. A light, audio recording, alphanumeric display, or any well-known notification method might be used to transmit the notice warning. One or more notice warnings may be used in combination for the driver's quicker perception. As sound isolation improves, the emergency vehicle warning system's effectiveness declines. The sensitivity of emergency vehicles is reduced as a result of the longer emergency response times and driving risks. This concept describes a novel system that would enhance the present alarm systems of emergency vehicles and address the acoustic isolation problem. An alarm system transmits signals from one or more emergency vehicles to the nearby commuter cars.

## TABLE OF CONTENTS

### CHAPTER NO TITLE PAGE NO

**ABSTRACT** 5

[LIST OF FIGURES](#_TOC_250007)  10

ABBREVIATIONS 11

[LIST OF TABLES](#_TOC_250006)  12

1. INTRODUCTION
2. Overview of the project 13
3. Objective 14
4. Block Diagram 15
5. Description of the block diagram 16
6. Flow Chart 19
7. LITERATURE SURVEY 20
8. ARDUINO
9. Overview 22
10. Specifications 24
11. Power 25
12. Pinout Configuration 26
13. Communication 26
14. Programming 27
15. Automatic Reset 28
16. Physical Characteristics and

Shield Compatability 28

4 LORA SHIELD

1. Overview 32
2. Specifications 33
3. Features 34
4. Power Consumption 35
5. Usage Notice 36
6. Order Information 36

5 RADIOHEAD LIBRARY WITH

ARDUINO BOARDS

1. Source File 37
2. Compile Notice 37

6 LORA ANTENNA

1. Overview 41
2. Working 42
3. Specifications 42
4. Mount and Installations 43

7 USB CABLE

1. Overview 44
2. Specification 45
3. Features 45

8 SOFTWARE REQUIRED 46

RESULT 50

9 CONCLUSION 51

APPENDIX 52

REFERENCE 54

LIST OF FIGURES

FIG. NO TITLE PAGE NO

1.31 Transmitter Block Diagram 15

1.32 Receiver Block Diagram 16

1.33 Flowchart of AEVAS 19

3.11 ArduinoUNO 22

3.12 Pin Configuration of Arduino UNO 23

4.11 LoRa Shield 33

5.21 RadioHead Example 38

5.22 Lora Shields and Arduino boards 39

5.23 Connections 39

5.24 Choose the Example 40

5.25 Configure the IDE settings of UNO 40

6.11 LoRa Antenna 41

7.11 USB Cable 45

8.1 Screenshot of Arduino IDE 47

8.2 Transmitter simulation in Arduino IDE 48

8.3 Receiver simulation in Arduino IDE 49

8.4 Result of the simulation 50

ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| S.NO | ABBREVIATION | DESCRIPTION |
| 1 | AEVAS | Advanced Emergency Vehicle Alert System |
| 2 | LORA | Long Range |
| 3 | WIFI | Wireless Fidelity |
| 4 | LORAWAN | Long Range Wide Area Network |
| 5 | MAC | Media Access Control |
| 6 | SDR | Software Defined Radio |
| 7 | IDE | Integrated Development Environment |
| 8 | USB | Universal Serial Bus |
| 9 | SPI | Serial Peripheral Interface |
| 10 | MOSI | Master Out Slave In |
| 11 | MISO | Master In Slave Out |
| 12 | SCK | Serial Clock |
| 13 | AREF | Analog Reference |
| 14 | SDA | Serial Data |
| 15 | SCA | Software Communication Architecture |

LIST OF TABLES

TABLE NO TITLE PAGE NO

2.1 Literature survey 20

3.21 Specifications of Arduino UNO 24

3.31 Power Supply of Arduino UNO 25

3.41 Pinout Configuration of Arduino UNO 26

4.41 Power Consumption of LoRa Shield 35

CHAPTER 1

INTRODUCTION

* 1. OVERVIEW OF THE PROJECT

Traffic congestion has become a significant issue in cities due to the rise in vehicle population over the last ten years. In India, there are more road fatalities and financial losses as a result of the ambulance and fire truck services' difficulties with heavy traffic congestion and the length of time it takes them to get there. Additionally, poor traffic management contributes to a rise in fatalities and accidents. An emergency vehicle alert system, often known as EVAS, is suggested as a solution to this issue. Both life and property are saved by this technique.

Present-day emergency vehicles have the option of using loudspeakers, sirens, or, in certain situations, external assistance from transit personnel to notify other motorists of their presence. When an emergency vehicle approaches with screaming sirens and flashing blue lights, everyone on the road must obediently move to the side of the road to allow it safe passage. Unfortunately, since their windows are shut and they are listening to loud music or radios, automobiles in the route of emergency vehicles often disregard sound alarms. Traffic is slowed down as a result of this condition, and in certain situations, emergency vehicles' response times are significantly lengthened.

Several strategies have been put out to reduce accidents involving emergency vehicles. Installing an emergency vehicle preemption system or a louder siren, for instance (EVP). A louder siren may not be the best approach, even when newer automobiles provide outstanding noise suppression. The EVP system's drawback is that because of signal congestion brought on by high traffic hours, emergency vehicles must halt at each junction and wait for drivers to clear. Additionally, this EVP system's installation costs are considerable; as a result, the system is not a great one. As a result, this initiative puts forward a novel alerting concept.

In this system, the ambulance must be equipped with a path-guided mechanism based on RF communication in order to direct the ambulance's course as well as the movement of the cars in front of it. The majority of the time, those in front of the ambulance get disoriented and struggle to make room for it. This approach may assist to alleviate this issue.

**1.2 OBJECTIVE**

The primary goal of the suggested work is to create an emergency vehicle alert system with the following objectives:

• By using the emergency vehicle to broadcast a signal that will be picked up by automobiles in front, alerting drivers.

• To inform nearby motorists of the emergency vehicle's location, and to create the system with the appropriate inputs.

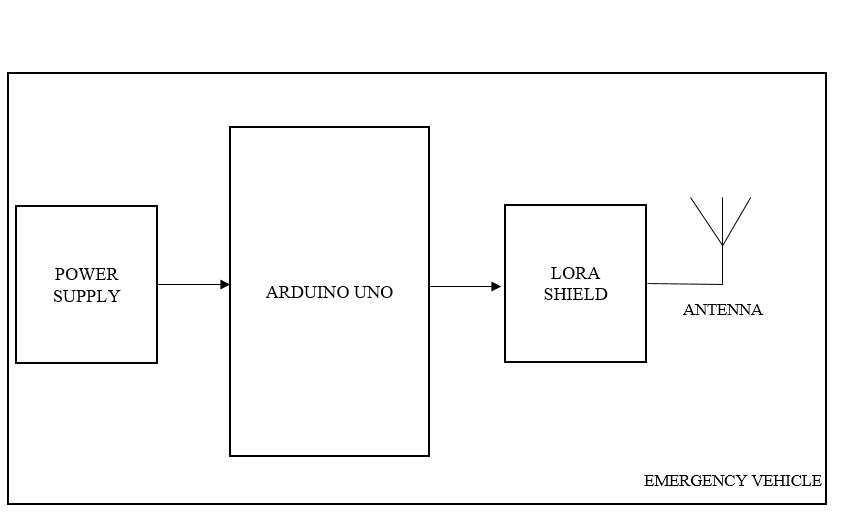
• To combine the required software with the requisite physical components.

• To create a user-friendly, readily accessible system.

This encouraged the development of the initiative that assists in overcoming these challenges by sending emergency vehicles' on-road warning signals to passing motorists. to create a communication system that warns other drivers when an emergency vehicle approaches. The system works by detecting their signals and signalling space so the emergency vehicle may drive more quickly. It lowers financial losses and traffic fatalities.

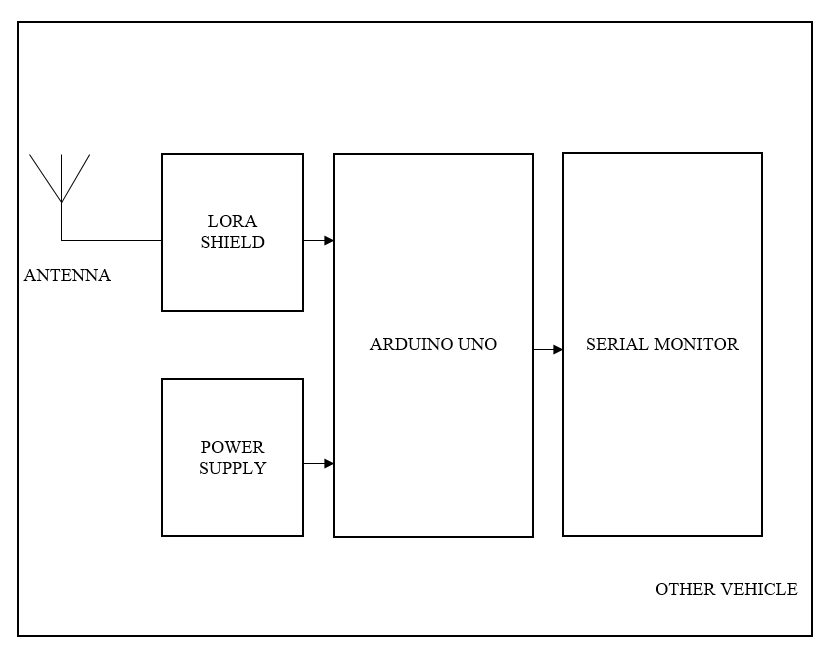
1.3 BLOCK DIAGRAM

**TRANSMITTER**

****

**FIGURE 1.31. Transmitter Block Diagram**

**RECEIVER**



**FIGURE 1.32. Receiver Block Diagram**

**1.4 DESCRIPTION OF THE BLOCK DIAGRAM**

The system's operation, components employed, and fundamental input and output flow are all shown in the block diagram. There are a transmitter unit and a receiver unit in this system. As shown in Fig.1.31, both units include an Arduino UNO serving as a processing unit and a Long Range (LoRa) Shield for communication.

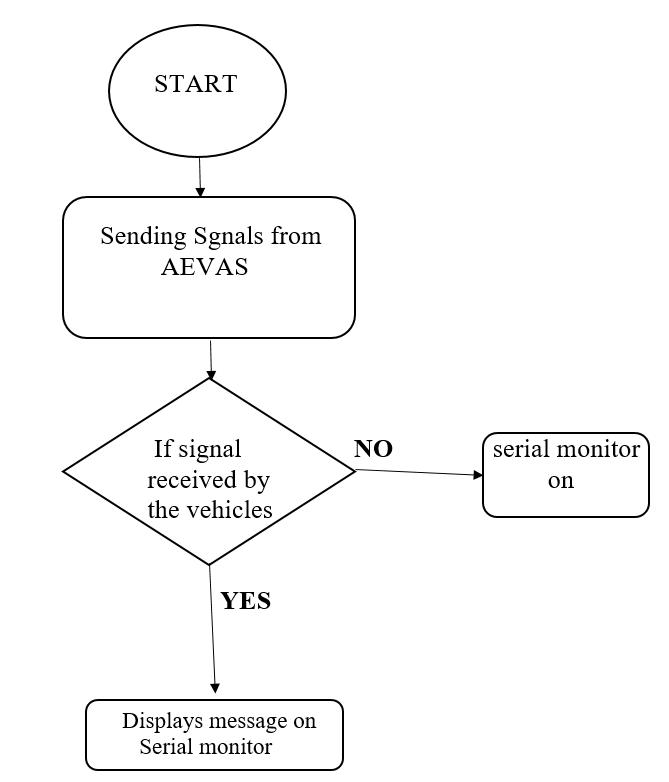
The receiver device is targeted by the LoRa Module's emission of specific frequency waves. Fig. 1.32 shows the system's reception unit, which consists of an Arduino UNO microcontroller that serves as the processing unit and a LoRa Shield for wireless communication. The LoRa Shield located in the system's reception unit detects the frequency that the transmitter unit emits, and it uses that information to activate serial monitor.

The emergency vehicle's implanted LoRa Shield transmitter never stops sending signals. The driver's attention is drawn to the receiver component of the car by serial monitor display if any other vehicle with a LoRa receiver is within the same range as it. LoRa can connect with many end devices, unlike Bluetooth, allowing for the alerting of all nearby cars. This alerts the motorist to the presence of an emergency vehicle and requests that they step aside so that the vehicle may go more quickly and easily. Examples of emergency vehicles include ambulances, fire engines, and police cars. As a result, it assists in preventing fatalities or minimising other losses that might result from the vehicle's translation being delayed.

LoRa is a method of radio signal transmission that uses chirped, multiple-symbol encoding to deliver data. Applications that carry sparse amounts of data at low bit rates are ideal for LoRa. Compared to Bluetooth, Wi-Fi, and Zigbee technologies, LoRa can transport data across a wider area. LoRa is highly suited to work in low power mode because to these properties. A software layer called LoRaWAN, often known as MAC (Media Access Control), governs how devices utilise the LoRa hardware. It is a point-to-multipoint networking protocol that makes use of the LoRa Shield.

An Arduino board is interfaced with a LoRa Shield to create the Advanced Emergency Vehicle Alert System (AEVAS), which is installed on emergency vehicles. A LoRa Shield, serial monitor, and Arduino board were used to construct the receiver side, which is installed on commuter cars. The Emergency Vehicle Alert System is created in a way that requires a LoRa transmitter to deliver the input signal. If there are any cars with LoRa receivers nearby, they will pick up the signal and show the message on serial monitor. serial monitor won't show any messages if the signal is not received.

**1.5 FLOWCHART**

****

**FIGURE.1.33 Flowchart of AEVAS**

**CHAPTER 2**

**LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TITLE OF THE PAPER** | **AUTHORS** | **YEAR OF PUBLICATION** | **DESCRIPTION** | **DRAWBACK** |
| Alert System for Emergency Vehicles Using  Software-Defined Radio | Carlos Bosquez, Ronald Moreira, Alexis De La Cruz | 2017 | SDR transmits radio signal from 88 MHz to 108 MHz using free software (such as GNU Radio and Python) with the help of USRP desired radio signal is generated and transmitted using yagi antenna. | Carlos Bosquez stated that the overlapping of frequency,poor  transmission range and the Power consumption is high |
| Evaluation of Low-Power Long Distance Radio Communication in  Urban Areas: LoRa and Impact of Spreading Factor | Selim Sagar,  Ismail Kaya,  Cem Sisman. | 2019 | LoRa is a wireless communication technology for Long-range transmission.  Data is transmitted in the form of packets  Better data rate is achieved by reducing spreading factor. | In densely populated area there is poor line of connection between transmitter and receiver. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Smart Ambulance Approach Alarm System Using Smartphone | Toru  Kobyashi,  Fukuyoshi  Kimura,  Kenichi Arai | 2019 | The position of emergency vehicle is shared with other vehicles within the range of 500m, using the live data in the cloud server | Software glitch,  Poor network  Connectivity affects the timely response of the system. |
| EVAS – Emergency Vehicle Alert System using  LoRa for automobiles | Sanjay S Tippannavar,  Puneeth K M,  Yaswanth S,  Madhu Sudan M P | 2023 | The lora based transceiver is fixed in cars and emergency vehicle it is processed using the arduino and the alert message is displayed to the nearer vehicles. | Arduino posses limited processing power ,range of transmission is not mentioned. |

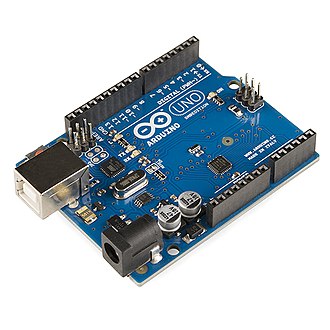
**FIGURE.2.1. Literature survey**

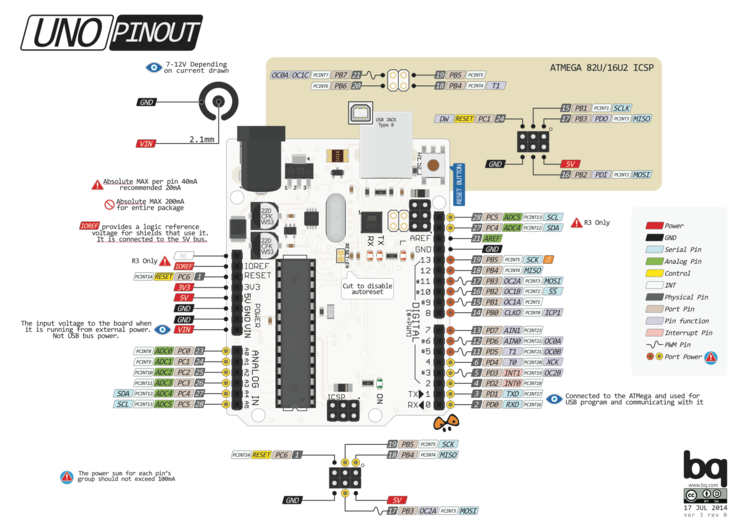
**CHAPTER 3**

**ARDUINO**

**3.1 OVERVIEW**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller(MCU) and developed by Arduino.cc and initially released in 2010. The microcontroller board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE(Integrated Development Environment), via a type B USB cable. It can be powered by a USB cable or a barrel connector that accepts voltages between 7 and 20 volts, such as a rectangular 9-volt battery. It has the same microcontroller as the Arduino Nano board, and the same headers as the Leonardo board.The hardware reference design is distributed under a Creative CommonsAttribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

****

**Figure.3.11.ArduinoUNO**

**Figure.3.12. Pin Configuration of Arduino UNO**

**3.2 SPECIFICATIONS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Microcontroller | ATmega328P –8bit AVR family  microcontroller | | Operating Voltage | 5V | | Recommended Input Voltage | 7-12V | | Input Voltage Limits | 6-20V | | Analog Input Pins | 6 (A0 – A5) | | Digital I/O Pins | 14 (Out of which 6  provide PWM output) | | DC Current on I/O Pins | 40 mA | | DC Current on 3.3V Pin | 50 mA | | Flash Memory | 32 KB (0.5 KB is used for  Bootloader) | | SRAM | 2 KB | | EEPROM | 1 KB | | Frequency (Clock Speed) | 16 MHz | |

**Table 3.21. Specifications of Arduino UNO**

**3.3 POWER**

|  |  |  |
| --- | --- | --- |
| Power | Vin,  3.3V,  5V,  GND | Vin: Input voltage to Arduino when using an external power source.  5V: Regulated power supply used to power microcontroller and other components on the board.  3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA.  GND: ground pins. |

**Table 3.31. Power Supply of Arduino UNO**

**3.4 PINOUT CONFIGURATION**

|  |  |  |
| --- | --- | --- |
| Reset | Reset | Resets the microcontroller. |
| Analog Pins | A0 – A5 | Used to provide analog input in the range of 0-5V |
| Input/Output Pins | Digital Pins 0 - 13 | Can be used as input or output pins. |
| Serial | 0(Rx), 1(Tx) | Used to receive and transmit TTL serial data. |
| External Interrupts | 2, 3 | To trigger an interrupt. |
| PWM | 3, 5, 6, 9, 11 | Provides 8-bit PWM output. |
| SPI | 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK) | Used for SPI communication. |
| Inbuilt LED | 13 | To turn on the inbuilt LED. |
| TWI | A4 (SDA), A5 (SCA) | Used for TWI communication. |
| AREF | AREF | To provide reference voltage for input voltage. |

**Table 3.41. Pinout Configuration of Arduino UNO**

**3.5 COMMUNICATION**

Communication with an Arduino Uno typically involves using its serial communication capabilities. You can communicate with it via USB using the Serial Monitor in the Arduino IDE, or through other serial communication methods like Bluetooth or Wi-Fi modules. Additionally, you can communicate with other devices or microcontrollers using protocols like I2C or SPI.

**3.6 PROGRAMMING**

Programming for the Arduino Uno involves writing code in the Arduino IDE, which uses a simplified version of C/C++. Here's a basic outline to get you started:

**1 Setup:**

This function runs once when you power up or reset the Arduino. It's where you initialize variables, pin modes, etc.

1. **Loop:**

This function runs repeatedly after setup. It's where you put the main logic of your program.

1. **Serial Communication:**

You can communicate with your computer using serial communication. For example, sending data from Arduino to the Serial Monitor.

1. **Reading Inputs:**

You can read from digital or analog pins to get sensor data, button states, etc.

1. **Control Outputs:**

You can control LEDs, motors, etc., by writing digital or PWM values to pins.

**3.7 AUTOMATIC RESET**

The Arduino Uno has an auto-reset feature that resets the microcontroller whenever a new sketch is uploaded from the Arduino IDE or when the serial communication port is opened. This reset is triggered by the DTR (Data Terminal Ready) signal from the USB-to-Serial converter chip on the Uno board.

The auto-reset function is useful because it puts the Arduino into a known state when a new sketch is uploaded, allowing the bootloader to take control and receive the new code. However, if you're using the serial port for communication with external devices and you don't want the Uno to reset each time you open the serial port, you may need to disable the auto-reset feature.

**3.8 PHYSICAL CHARACTERISTICS AND SHIELD COMPATABILITY**

**PHYSICAL CHARACTERISTICS:**

The Arduino Uno is a popular microcontroller board known for its ease of use and versatility. Here are its key physical characteristics:

**1 Size:**

The Arduino Uno has a standard form factor, measuring approximately 68.6mm (2.7 inches) in length and 53.4mm (2.1 inches) in width.

**2 Microcontroller:**

It is based on the ATmega328P microcontroller, which operates at 16 MHz.

**3 Digital I/O Pins:**

The Uno has a total of 14 digital input/output pins, of which 6 can be used as PWM (Pulse Width Modulation) outputs.

**4 Analog Input Pins:**

It features 6 analog input pins, marked A0 through A5, which can also be used as digital I/O pins.

**5 Voltage Regulator:**

The board includes a voltage regulator that allows it to be powered from an external power source or through the USB connection.

1. **USB Interface:**

It has a USB Type-B connector for connecting to a computer or other USB host devices for programming and serial communication.

1. **Power Connector:**

The board can be powered via a DC power jack, which accepts a voltage range of 7 to 12 volts DC.

1. **Reset Button:**

There is a reset button on the board that can be used to restart the microcontroller.

1. **LEDs:**

The Uno has several onboard LEDs for indicating power status, serial communication, and pin activity.

1. **ICSP Header:**

It includes an ICSP (In-Circuit Serial Programming) header for programming the microcontroller with an external programmer.

1. **Operating Voltage:**

The Uno operates at 5 volts, which is the standard voltage level for most of its components.

1. **Memory:**

It has 32 KB of flash memory for storing sketches (programs), of which 0.5 KB is used by the bootloader. Additionally, it has 2 KB of SRAM and 1 KB of EEPROM.

**SHIELD COMPATABILITY:**

The Arduino Uno is compatible with a wide range of shields, which are add-on boards that can be stacked on top of the Uno to add functionality or features to your projects. Here's what you need to know about shield compatibility with the Arduino Uno:

**Physical Compatibility:** Shields are designed to match the physical dimensions and pin layout of the Arduino Uno, making them easy to stack on top of each other using the standard Arduino headers.

**Pin Compatibility:** Shields typically use the standard digital and analog pins of the Arduino Uno, so they should work seamlessly without any additional wiring or configuration.

**Power Compatibility:** Shields can draw power from the Arduino Uno's power supply or may require an external power source, depending on their power requirements. Make sure to check the power specifications of the shield and ensure it is compatible with the Uno's power capabilities.

**Library Compatibility:** Some shields may require specific libraries or software to be installed on your Arduino IDE in order to function properly. Make sure to check the documentation provided with the shield for any additional setup requirements.

**Stacking Limitations:** While the Arduino Uno can stack multiple shields on top of each other, there may be limitations on the number of shields you can use simultaneously due to factors such as power consumption, pin conflicts, and available memory

**Custom Shields:** If you're designing your own shield, make sure to follow the Arduino Uno's hardware specifications and pinout to ensure compatibility. You can find detailed information about the Uno's pinout in the Arduino documentation.

**CHAPTER 4**

**LORA SHIELD**

**4.1 OVERVIEW**

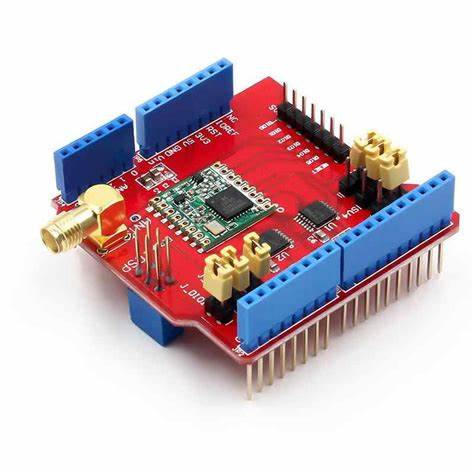
The Dragino Lora Shield is a long range transceiver on a Arduino shield form factor and based on Open source library. The Lora Shield allows the user to send data and reach extremely long ranges at low data-rates. It provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

The Lora Shield based on SX1276/SX1278 targets professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, smartphone detection, building automation, and so on.

Using Semtech's patented LoRa™ modulation technique the Lora Shield can achieve a sensitivity of over - 148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields

These devices also support high performance (G)FSK modes for systems including WMBus, IEEE802. The Lora Shield deliver exceptional phase noise, selectivity, receiver linearity and IIP3 for significantly lower current consumption than competing

devices.



**Figure 4.11. LoRa Shield**

**4.2 SPECIFICATIONS**

**﻿**

* 168 dB maximum link budget.
* ﻿﻿+20 dBm - 100 mW constant RF output vs.
* +14 dBm high efficiency PA.
* Programmable bit rate up to 300 kbps.
* High sensitivity: down to -148 dBm.
* Bullet-proof front end: 1P3 = -12.5 dBm.
* Excellent blocking immunity.
* Low RX current of 10.3 mA, 200 nA register retention.
* Fully integrated synthesizer with a resolution of 61 Hz.
* FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
* Built-in bit synchronizer for clock recovery.
* Preamble detection.
* 127 dB Dynamic Range RSSI.
* Automatic RF Sense and CAD with ultra-fast AFC.
* Packet engine up to 256 bytes with CRC.
* Built-in temperature sensor and low battery indicator.

**4.3 FEATURES**

1.Compatible with 3.3v or 5v I/O Arduino Board.

2.Frequency Band: 915MHz/868 MHZ/433 MHZ (Pre-configure in a factory)

3.Low power consumption

4.Compatible with Arduino Leonardo, Uno, Mega, DUE

5.External Antenna via I-Pex connector

6.Excellent blocking immunity.

7.Built-in bit synchronizer for clock recovery.

8.Preamble detection.

9.127 dB Dynamic Range RSSI.

10.Automatic RF Sense and CAD with ultra-fast AFC.

11.Packet engine up to 256 bytes with CRC.

12.Built-in temperature sensor and low battery indicator.

**4.4 POWER CONSUMPTION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SYMBOL** | **DESCRIPTION** | **CONDITIONS** | **TYP** | **MAX** | **UNIT** |
| IDDSL | Supply current in sleep mode |  | 0.2 | 1 | uA |
| IDDIDLE | Supply current in idle mode | RC oscillator enabled | 1.5 | - | uA |
| IDDST | Supply current in standby mode | Crystal oscillator enabled | 1.6 | 1.8 | mA |
| IDDFS | Supply current in synthesizer mode | FSRx | 5.8 | **-** | mA |
| IDDR | Supply current in receive mode | LnaBoost off, band 1  LnaBoost on, band 1  Bands 2&3 | 10.8  11.5  12.0 | **-**  **-**  **-** | mA  mA  mA |
| IDDT | Supply current in transmit mode with impedance matching | RFOP = +20 dBm, on PA\_BOOST  ROP = +17 dBm, on PA\_BOOST  ROP = +13 dBm, on RFO\_LF/HF pin | 120  87  29  20 | **-**  **-**  **-**  **-** | mA  mA  mA  mA |

**Table 4.41. Power Consumption of LoRa Shield**

**4.5 USAGE NOTICE**

You have to be aware that Radio link quality and performances are highly dependent of environment.

Better performances can be reached with:

* ﻿﻿Outdoor environment.
* ﻿﻿No obstacles.
* ﻿﻿No high level radio interferer in the ISM 868MHz band.
* ﻿﻿At least 1 meter above the ground.
* Radio performances are degraded with:

﻿﻿Obstacles: buildings, trees...

* ﻿﻿Inner buildings environments.
* ﻿﻿High ISM 868MHz band usage by other technologies.
* Radio comery sinain are usually killed with bad topographic conditions. It is usually not possible to communicate through a hill,even very small.

**4.6 ORDER INFORMATION**

Lora Shield 868: Load with RFM95W-8682, support 868M frequency

﻿﻿Lora Shield 915: Load with RFM95W-9152, support 915M frequency

﻿﻿Lora Shield 433: Load with RFM98W-433S2, support 433M frenquency.

**CHAPTER 5**

**RADIOHEAD LIBRARY WITH ARDUINO BOARDS**

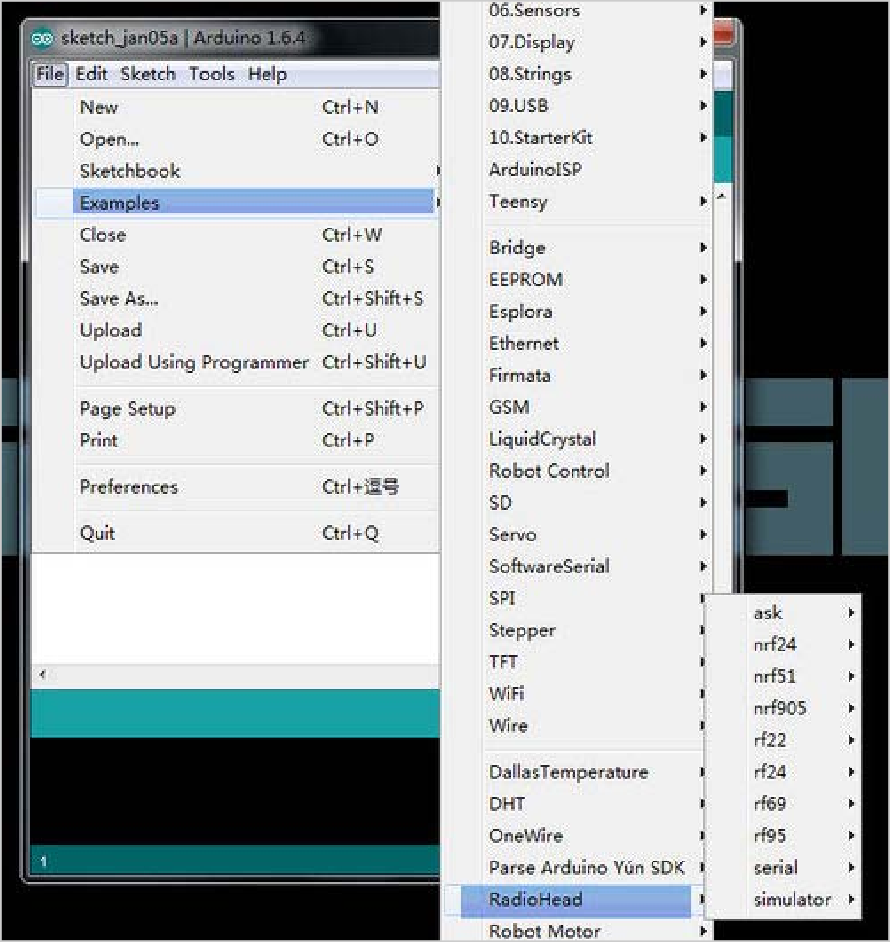
**5.1 SOURCE FILE**

The RadioHead Library simplifies communication between the Arduino Board and the Lora Shield.We can find it here: <http://www.airspayce.com/mikem/arduino/RadioHead/> To use it in your project you just have to copy it in your Arduino project folder (next to your main Arduino project file .ino).

**5.2 COMPILE NOTICE**

* After putting the llbrary in the right place, you have to also modify the frequency to the frequency you want to use, the position of this is setFrequency() in the file: arduino-xxx\libraries\RadioHead\RH RF95.cpp;
* The RF95-server sketch use the led=9 to show the status. please change it to other LED , such 8. because 9 is connected to the lora module's RESET pin. you will see 'init fail' in the default server sketch

Then ,we can easily to use it to do some experimentation.



**Figure.5.21. RadioHead Example**

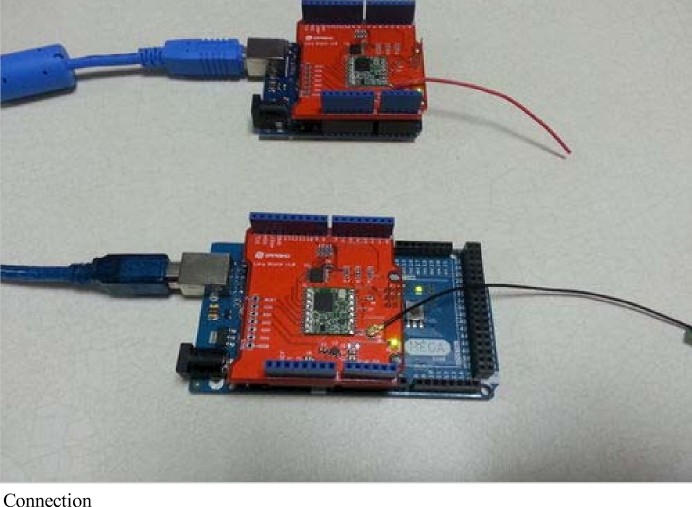
In this example,we use two Lora Shields to transmit and receive signa1,there are some project in the RadioHead Library,the operation is as follows:

* Connect two Lora Shields with Arduino board and connect them to computer via the USB cable;

We used the Lora Shield+Arduino UNO as the server to transmit signal and the Lora Shield+Aduino MEGA2560 as the client to receive signal here.

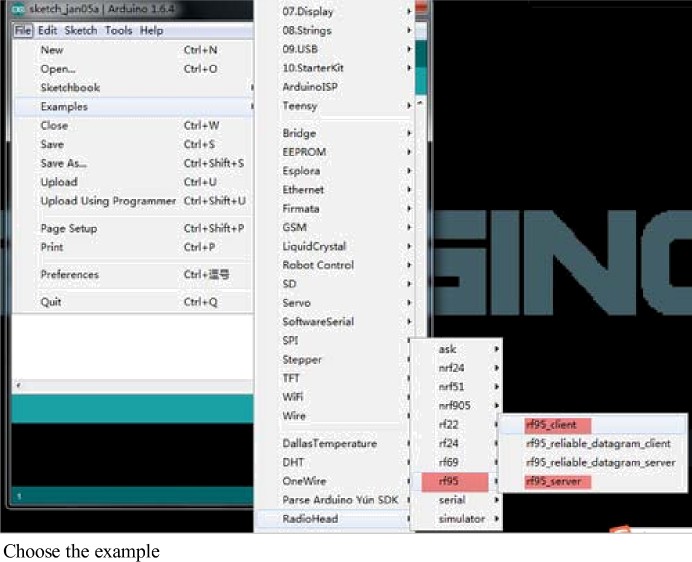


**Figure.5.22. Lora Shields and Arduino boards**

 Connect the Lora Shield and Arduino,and connect them to computer.

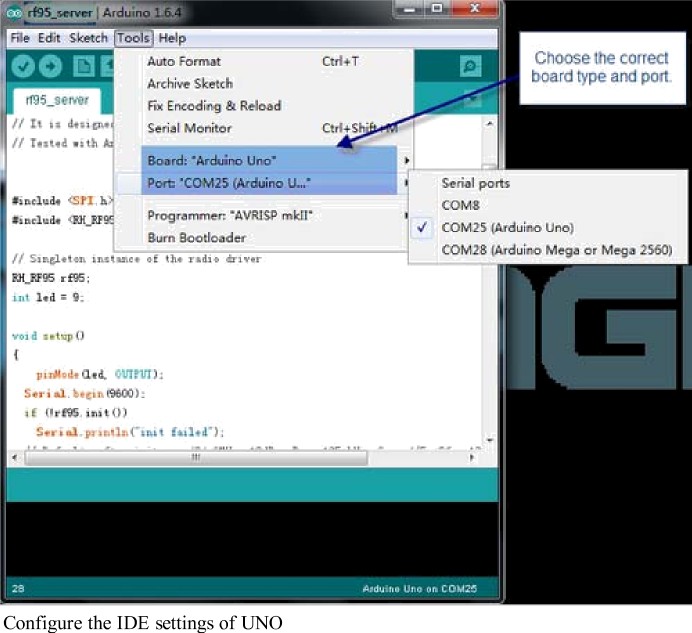
**Figure.5.23. Connections**

Open the IDE and choose the right example;



**Figure.5.24. Choose the example**

Configure the IDE settings and upload the two skteches to Arduino.



**Figure.5.25. Configure the IDE settings of UNO**

After configure the IDE settings,up1oad the skteche to Aduino.Check the Serial Moniter and we can view the running result of the two boards.

**CHAPTER 6**

**LORA ANTENNA**

**6.1 OVERVIEW**

LoRa stands for long range — its basis is spread spectrum modulation technology. A LoRa antenna is a device that extends the range of low-power [wide-area networks](https://novotech.com/learn/m2m-blog/blog/2023/02/06/want-some-cant-miss-tips-on-how-to-configure-a-wan-router/). It facilitates long-distance communications with minimal power consumption. The primary application of the antenna is in low-power wireless [devices](https://novotech.com/learn/m2m-blog/blog/2019/03/28/picking-the-best-iot-network-for-your-application-part-2-7-non-cellular-connection-technologies-webinar/). These devices are useful in the IoT and Machine to Machine (M2M) communications.



**FIGURE.6.11. LoRa Antenna**

**6.2 WORKING**

The design of LoRa antennas makes them work in the sub-GHz frequency range, where radio signals travel longer distances with less power. They use a patented spread spectrum modulation technique. The Chirp Spread Spectrum (CSS) technique maximizes the range of communication. CSS is a process that spreads the signal across a wide frequency range, allowing for more efficient use of the radio spectrum. The antenna receives and transmits the signal over a wide area with low power.

**6.3 SPECIFICATIONS**

* **Frequency**: Antenna frequency is one of the most important factors to consider when buying a LoRa antenna. For example, the most common frequencies are 868 MHz (Europe), LoRa antenna 915 range (US), and 433 MHz (Asia). Picking an antenna that supports this range is important.
* **Gain**: LoRa antenna gain is a measure of its efficiency. The higher numbers in decibels represent better signal strength. Choosing a higher gain is important if you want a better range from your LoRa antenna:
  1. The 4 dBi range is common and suitable for most applications.
  2. The 8 dBi range, like LoRa antenna helium, provides great coverage and signal strength.
  3. A 3 dBi range provides a lower coverage.
* **Bandwidth**: The bandwidth of a LoRa antenna determines its data transfer rate. So it’s important to look for an antenna with a high bandwidth depending on the data you’re transferring.
* **Return Loss**: Return loss measures the quality of an antenna’s signal reflection. Lower numbers indicate better return loss, and it’s important to find an antenna with low return loss for better signal strength.
* **VSWR**: VSWR stands for Voltage Standing Wave Ratio and measures how well an antenna will match the impedance of your equipment. Lower numbers show better VSWR.

**6.4 MOUNT AND INSTALLATIONS**

The two most common LoRa antenna mounts are pole and wall mounts. Pole-mounted antennas have a vertical orientation, which is ideal for long-range applications. Contrary, wall-mounted antennas are ideal for indoor deployments. They can be mounted horizontally or vertically, depending on your application needs.

Installing a LoRa antenna is not difficult.

Before installation, it’s important to check the antenna for tight connections.

Also, ensure the mounting structure is secure.

After installation, you should check the antenna’s orientation and adjust it to ensure optimal performance.

**CHAPTER 7**

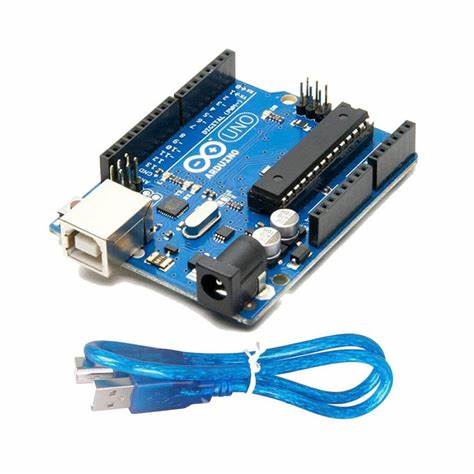
**USB CABLE**

**7.1 OVERVIEW**

This is a Cable For Arduino UNO/MEGA (USB A to B)-1feet, you can use it to connect[“Arduino Uno”](https://robu.in/product-category/arduino-2/arduino/arduino-uno/), “[Arduino Mega 2560″](https://robu.in/product-category/arduino-2/arduino/arduino-mega/) or any board with the USB female A port of your computer. Length is approximately 52 cm. Cable color and shape may vary slightly from image as our stock rotates.

This is a standard-issue USB 2.0 cable. the kind that’s usually used for printers, [Arduino](https://robu.in/product-category/arduino-2/arduino/), etc. Compatible with most SFE designed USB boards as well as USB Arduino boards like the Uno.

It is very important to understand the cheap USB cables or faulty USB cables may harm your Printers and cause errors again and again while printing important data. For the error-free and long life of your printers, you must use high-quality USB A to B printer cables. We are selling highly robust and high-quality USB 2.0 Printer cable at a reasonable price.Connect your USB printer, scanner and more to your computer. Transmits data at high speeds with error-free, high-performance transmission.



**FIGURE.7.11. USB Cable**

**7.2 SPECIFICATIONS**

The USB cable used with an Arduino Uno typically has a USB Type-A connector on one end (which plugs into your computer or power source) and a USB-B connector on the other end (which plugs into the Arduino Uno board). It's usually a standard USB 2.0 cable with a Type-A to Type-B connection.

**7.3 FEATURES**

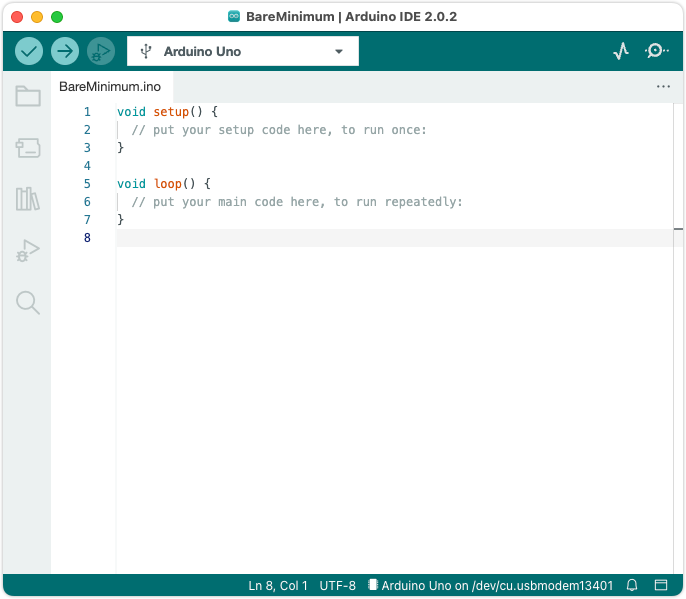
1. Fully compatible with the PC.
2. Molded strain relief and PVC over molding to ensure a lifetime of error-free data transmissions.
3. Aluminum under mold shield helps meet FCC requirements on KMI/RFI interference.

**CHAPTER 8**

**SOFTWARE REQUIRED**

## ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT(IDE)

Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. Use the Arduino IDE on your computer (picture following) to create, open, and change sketches (Arduino calls programs "Sketches"). Sketches define what the board will do.



**Figure.8.1. Screenshot of Arduino IDE**

**SIMULATION IN ARDUINO IDE**

**TRANSMITTER(AMBULANCE)**

****

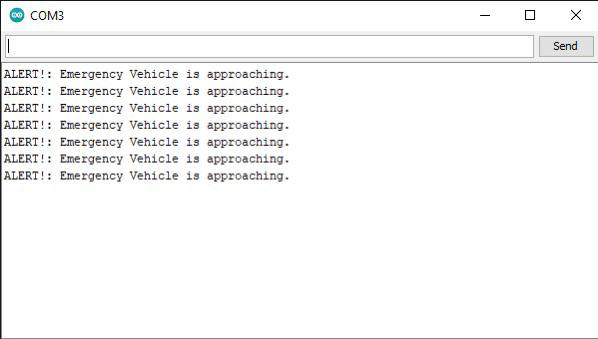
**FIGURE.8.2. Transmitter simulation in Arduino IDE**

**RECEIVER(OTHER VEHICLE)**

****

**FIGURE.8.3. Receiver simulation in Arduino IDE**

**RESULT**



**FIGURE.8.4. Result of the simulation**

**CHAPTER 9**

**CONCLUSION**

This proposed work involves the development of an emergency vehicle warning system that enables emergency vehicle operators to warn other cars in front of them to move aside. This system will be ideal for low-range, medium-range, and long-range communications between emergency vehicles and other on-road vehicles. Future generations of autonomous cars might use the suggested Emergency Vehicle Alert System. It is possible to extend the communication range between the transmitters and receivers. Future automobile radios may be programmed to switch on instantly in case of an emergency, protecting all moving automobiles.

**APPENDIX**

**TRANSMITTER**

#include <SPI.h>

#include <RH\_RF95.h>

RH\_RF95 rf95;

void setup()

{

Serial.begin(9600);

while (!Serial) ;

if (!rf95.init())

Serial.println("init failed");

}

void loop()

{

Serial.println("Sending to rf95\_server");

uint8\_t data[] = "Emergency Vehicle is approaching.";

rf95.send(data, sizeof(data));

rf95.waitPacketSent();

uint8\_t buf[RH\_RF95\_MAX\_MESSAGE\_LEN];

uint8\_t len = sizeof(buf);

if (rf95.waitAvailableTimeout(3000))

{

if (rf95.recv(buf, &len))

{

Serial.print("ALERT!: ");

Serial.println((char\*)buf);

}

else

{

Serial.println("recv failed");

}

}

else

{

Serial.println("No reply, is rf95\_server running?");

}

delay(40);

}

**RECEIVER**

#include <SPI.h>

#include <RH\_RF95.h>

RH\_RF95 rf95;

void setup()

{

Serial.begin(9600);

while (!Serial) ;

if (!rf95.init())

Serial.println("init failed");

}

void loop()

{

if (rf95.available())

{

uint8\_t buf[RH\_RF95\_MAX\_MESSAGE\_LEN];

uint8\_t len = sizeof(buf);

if (rf95.recv(buf, &len))

{

Serial.print("ALERT!: ");

Serial.println((char\*)buf);

}

else

{

Serial.println("recv failed");

}

}

}

**REFERENCE**

[1] Bosquez, C., Moreira, R., & De La Cruz, A. (2017, November). Alert system for emergency vehicles using software-defined radio. In *2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)* (pp. 1-5). IEEE.

[2] Kobayashi, T., Kimura, F., Imai, T., & Arai, K. (2019). Smart ambulance approach alarm system using smartphone. *IEICE TRANSACTIONS on Information and Systems*, *102*(9), 1689-1692.

[3] Roopashree, V., Malavika, D. N., Bai, E. N., Suman, A., & Shashikala, D. R. (2020). Traffic congestion detection and Alerting Ambulance using IOT. *Int J Eng Res Tech*, *9*(7), 1339-1343.

[4] Widianto, E. D., Pakpahan, M. S., & Septiana, R. (2018, October). LoRa QoS performance analysis on various spreading factor in Indonesia. In *2018 International Symposium on Electronics and Smart Devices (ISESD)* (pp. 1-5). IEEE.