AUTOMATIC TRAFFIC LIGHT CONTROL SYSTEM FOR SMART AMBULANCE

A project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

M.V. Sairam (318126512154) V.N.N.S.V.A. Koushik (318126512160)

L.Rakesh (318126512150) K. Sai Chaitanya (318126512144)

Under the guidance of

Mr. A. SIVA KUMAR
BE, MTech, (Ph. D)
Assistant Professor

Mrs. V. SHIREESHA
BTech, MTech, MBA, (Ph. D)
Assistant Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'B+' Grade)

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

2021-2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'B+' Grade*)

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"AUTOMATIC TRAFFIC LIGHT CONTROL SYSTEM FOR SMART AMBULANCE"** submitted by **M.V. Sairam (318126512154), V.N.N.S.V.A. Koushik (318126512160), L. Rakesh (318126512150), K.Sai Chaitanya(318126512144)** in partial fulfillment of the requirements for the award of the Degree of ***Bachelor of Technology in Electronics & Communication Engineering*** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guides		Head of the Department
Mr. A. Siva Kumar	**Mrs. V. Shireesha**	**Dr.V. Rajyalakshmi**
BE, MTech, (Ph. D)	BTech, MTech, MBA, (Ph. D)	M.E, Ph. D, MIEEE, MIE, MIETE
Assistant Professor	Assistant Professor	Professor
Department of E.C. E	Department of E.C. E	Department of E.C. E
ANITS	**ANITS**	**ANITS**
Assistant Professor Department of E.C.E Anil Neerukonda Institute of Technology & Sciences Sangivalasa, Visakhapatnam-531 162	Assistant Professor Department of E.C.E. Anil Neerukonda Institute of Technology & Sciences Sangivalasa, Visakhapatnam-531 162	Head of the Department Department of E C E Anil Neerukonda Institute of Technology & Sciences Sangivalasa-531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guides **Mr. A. Siva kumar,** Assistant Professor and **Mrs.V. Shireesha,** Assistant Professor, Department of Electronics and Communication Engineering, ANITS for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajya Lakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. Finally, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

M.V. Sairam (318126512154),

V.N.N.S.V.A. Koushik (318126512160),

L.Rakesh (318126512150),

K. Sai Chaitanya (318126512144)

ABSTRACT

According to recent surveys conducted in India yearly 1.46 million deaths occur only due to vehicle accumulation, in which around 30 % of deaths occur only when emergency vehicles (like Ambulances & Fire engines) get stuck in traffic. Another Indian government data shows more than 50% of Heart attack cases reach hospital late, which can constitute unavailability of ambulances too but majority of it is due to patients stuck in traffic. Indian traffic signalling systems operate only on delay basis i.e., they are static and doesn't change for emergency vehicles and at the junctions, clearance to emergency vehicles is given manually. But manual assistance may not be available at all the junctions and sometimes the emergency vehicle goes unnoticed by the traffic police due to vehicles accumulated before the emergency vehicle. So, badly we need a bit smarter traffic signalling system which offers highest priority emergency vehicles. Our project is to implement such a traffic signalling system, which offers highest priority to emergency vehicles. This traffic management system mainly comprises of two subsystems one placed in the emergency vehicle and the other connected to traffic console.

Emergency vehicle subsystem and the traffic console subsystem communicate wirelessly through radio signalling as it is the best way of wireless communication. But we can't use basic radio signalling as it can be misused. So, Radio communication with LORA should be used which offers long range, low power and secure data transmission between Microcontrollers. Here, when an emergency vehicle is at a distance of 2-5 km from the traffic junction, vehicle Driver has to press a button at the ambulance to start communication between LORA modules. We used ARDUINO IDE for compiling and executing the code. In medical emergencies patient death happens half the way before reaching the hospital, this is often due to delay in the transport system. This project enables them to save the critical time lost in the traffic and there by save lives of patient.

CONTENTS **Page. No**

LIST OF TABLES AND FIGURES

TABLES: **Page. No**

FIGURES:

LIST OF ABBREVATIONS

AM	: Amplitude modulation
CSS	: Chirp Spread Spectrum
DSSS	: Direct - sequence spread spectrum
DTR	: Data Terminal Ready
EEPROM	: Electrically Erasable Programmable Read Only Memory
EV-DO	: Evolution Data Only
FHSS	: Frequency Hopping Spread Spectrum
FM	: Frequency modulation
GPRS	: General packet radio service
GPS	: Global Positioning System
HSDPA	: High speed downlink packet Access
HTML	: Hypertext Markup Language
IDE	: Integrated Development Environment
IMU	: Inertial Measurement Unit
IOT	: Internet of Things
ISM	: Industrial, Scientific and Medical Radio Spectrum
LED	: Light Emitting Diode
LORA	: Long range
LORAWAN	: Long range wide area network
LRT	: Light rail train
LTE	: Long Term Evolution
MCU	: Microcontroller Unit

MISO	: Master in Slave Out
MOSI	: Master Out Slave In
PCB	: Printed circuit Board
PM	: Phase modulation
PWM	: Pulse Width Modulation
RF	: Radio frequency
RFID	: Radio frequency Identification
RPS	: Regulated Power Supply
RSSI	: Received Signal Strength Indicator
SCK	: Serial Clock
SF	: Spread factor
SOC	: System-on-Chip
SPI	: Serial Peripheral Interface
SRAM	: Static Random Access Memory
TTL	: Transistor Transistor Logic
UART	: Universal Asynchronous Receiver- Transmitter
USB	: Universal Serial Bus
VHF	: Very high frequency
Wi-Fi	: Wireless Fidelity
WPA	: Wi-Fi Protected Access

CHAPTER-1

INTRODUCTION

Traffic management on the road has become severe problem of today's society. The desire of every individual to have private transportation mode, leads to increase the traffic on roads. Due to increase in urbanization, industrialization and population; there has been a tremendous growth in the traffic. With growth in traffic, there is occurrence of bundle of problems too; these problems include traffic jams, accidents and traffic rule violation at the heavy traffic signals. This in turn has an adverse effect on the economy of the country as well as the loss of lives. So, problem given above will become worst in the future. Traffic congestion and tidal flow management were recognized as major problems in modern urban areas. Traffic congestion is a condition on transport networks that occurs as use increases, and is characterized by slower speeds, longer trip times, and increased vehicular queuing. This caused much thwarting for emergency vehicles like ambulances and fire brigades. So, we proposed a system which pre-empts the traffic signals for emergency vehicles. The main theme behind this scheme is to provide a smooth flow for the emergency vehicles like ambulances and fire brigades to reach their destination in time and thus reduce the risk associated with emergency transportation. The idea is to implement a system which would control automatically the traffic light in the path of the emergency vehicle.

In 1868, the traffic lights are only installed in London and today these are installed in many parts of the world. Most of the traffic lights around the world follow a predetermined timing circuit. Sometime the vehicles on the red-light side have to wait for green signal even though there is little or no traffic. It results in the loss of valuable time. Traffic control at intersections is a matter of concern in large cities. Several attempts have been made to make traffic light's sequence dynamic so that these traffic lights operate according to the current volume of the traffic. Most of them use the sensor to calculate current volume of traffic but this approach has the limitation that these techniques based on counting of the vehicles and treats the emergency vehicles as the ordinary vehicles means no priority to ambulance, fire brigade or V.I.P vehicles. As a result, emergency vehicles stuck in traffic signal and waste their valuable time. So, there is a need for traffic signal pre-emption (also called as traffic signal prioritization).

Traffic pre-emption devices are implemented in a variety of ways. They can be installed on road vehicles, integrated with train transportation network management systems, or operated by remote control from a fixed location, such as a fire station, or by a dispatcher at an emergency call center. Traffic lights must be equipped to receive an activation signal to be controlled by any system intended for use in that area. A traffic signal not equipped to receive a traffic pre-emption signal will not recognize activation, and will continue to operate in its normal cycle Vehicular devices can be switched on or off as needed, though in the case of emergency vehicles, they are frequently integrated with the vehicle's emergency warning lights. When activated, the traffic pre-emption device will cause properly equipped traffic lights in the path of the vehicle to cycle immediately, to grant right-of-way in the desired direction, allowing for normal programmed time delays for signal changes.

Traffic signal pre-emption systems integrated with train transportation networks typically extend their control of traffic from the typical cross arms and warning lights to one or more nearby traffic intersections, to prevent excessive road traffic from approaching the crossing, while also obtaining the right-of-way for road traffic that may be in the way to quickly clear the crossing. This also allows buses and hazmat vehicles in the USA to proceed through the intersection without stopping at the railroad tracks.

Fixed-location systems can vary widely, but a typical implementation is for a single traffic signal in front of or near a fire station to stop traffic and allow emergency vehicles to exit the station unimpeded. Alternatively, an entire corridor of traffic signals along a street may be operated from a fixed location, such as to allow fire apparatus to quickly respond through a crowded downtown area, or to allow an ambulance faster access when transporting a critical patient to a hospital in an area with dense traffic.

Traffic signal pre-emption systems sometimes include a method for communicating to the operator of the vehicle that requested the pre-emption (as well as other drivers) that a traffic signal is under control of a pre-emption device, by means of a notifier. This device is almost always an additional light located near the traffic signals. It may be a single light bulb visible to all, which flashes or stays on, or there may be a light aimed towards each direction from which traffic approaches the intersection. In the case of multiple notifier lights at a controllable intersection, they will either flash or stay on depending on the local configuration, to communicate to all drivers from which direction a pre-empting signal is being received. This informs regular drivers which direction may need to be cleared, and informs activating vehicle drivers if they have control of the light (especially important when more than one activating vehicle approaches the same intersection). A typical installation would provide a solid notifier to indicate that an activating vehicle is approaching from behind, while a flashing notifier would indicate the emergency vehicle is approaching laterally or oncoming. There are variations of notification methods in use, which may include one or more colored lights in varying configurations.

Events leading up to an activation and notification are not experienced by drivers on a daily basis, and driver education and awareness of these systems can play a role in how effective the systems are in speeding response times. Unusual circumstances can also occur which can confuse operators of vehicles with traffic pre-emption equipment who lack proper training. For example, on January 2, 2005, a fire engine successfully pre-empted a traffic light at an intersection which included a light rail train (LRT) crossing in Hillsboro, Oregon, yet the fire engine was hit by an LRT at the crossing. A subsequent inquiry determined that the LRT operator was at fault. The accident occurred in the middle of a network of closely spaced signalized intersections where the signs and signals granted right-of-way to the LRT simultaneously, at all intersections. The LRT operator was viewing right-of-way indications from downstream signals and failed to realize that pre-emption had occurred at the nearest intersection. The fire engine, granted the green light before it arrived at the intersection, proceeded through while the LRT operator, failing to notice the unexpected signal to stop, ran into the fire engine and destroyed it.

1.1 Project Objective:

Objective of our project is to implement a bit smarter traffic signaling system which ensures safe and efficient transportation of emergency vehicles. We can observe that, traffic signals at the junctions do not change for emergency vehicles. If the traffic police is not there at the junction or if the emergency vehicle at the junction goes unnoticed (this will occur due the traffic accumulated before the emergency vehicle) then it has to wait in the traffic by losing its valuable time. Our project brings up a solution for this problem. In our project, when an emergency vehicle comes into predefined range of the junction then the traffic signal for the lane in which ambulance is present will change to green leaving the way clear for emergency vehicles.

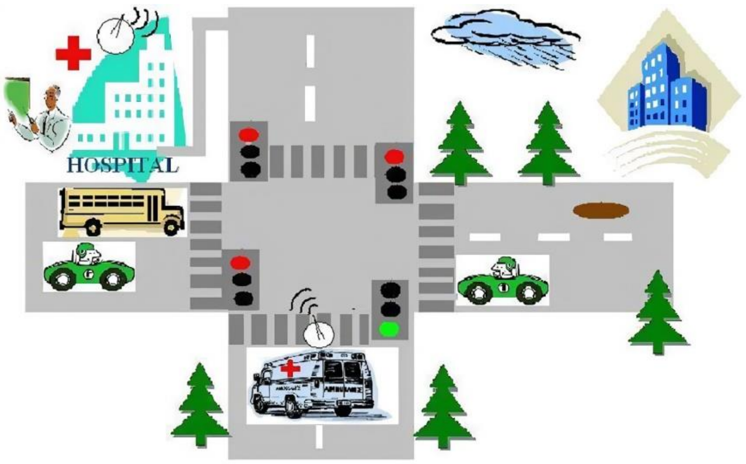


Fig-1.1 Schematic of the proposed system

We have many types of vehicular devices, which can be used to pre-empt the traffic signal. They are

a. Acoustic

b. Image Processing

1.1.1 Acoustic:

Some systems use an acoustic sensor linked to the pre-emption system. This can be used alone or in conjunction with other systems. Systems of this type override the traffic signal when a specific pattern of tweets or wails from the siren of an emergency vehicle is detected.

Advantages of a system like this are that they are fairly inexpensive to integrate into existing traffic signals and the ability to use siren equipment already installed in emergency vehicles- thus dispensing with the need for special equipment. A major disadvantage is that sound waves can easily be reflected by buildings or other large vehicles present at or near an intersection, causing the "reflected" wave to trigger a pre-emption event in the wrong direction. Reflected waves can also create unnecessary collateral pre-emption events alongside streets near the emergency vehicle's route. Yet another disadvantage is that the acoustic sensors can sometimes be sensitive enough to activate the pre-emption in response to a siren from too far away, or from an unauthorized vehicle with a horn exceeding 120 dB (many truck and bus horns exceed this threshold at close range).

1.1.2 Image Processing:

This system operates based on vehicle and emergency vehicle counting. But this system will not work efficiently in adverse weather conditions like wind, moist, smog, smoke and dusty atmosphere. And sometimes emergency vehicle may get hidden by other heavy vehicles. In this case, emergency vehicle will not be detected by the system and traffic signal pre-emption will not occur.

1.1.3 Global Positioning System:

With the advent of widespread Global Positioning System (GPS) applications became the introduction of a GPS-based traffic pre-emption system, which could also do collision avoidance. Recently some GPS pre-emption systems (see first two external links below) have found a way to overcome the nagging problem that "blinds" many GPS systems: how to prevent the system from being "blinded" by the loss of a GPS signal. In dense cities with tall buildings, GPS receivers may have difficulty obtaining the four required GPS satellite signals, required for trilateration to determine location. If the vehicle systems are not designed with a backup "IMU" (Inertial Measurement Unit), lack of GPS availability may adversely affect the system's performance (see first external link below). Extremely heavy cloud cover or severe weather can also adversely impact the ability of the GPS receiver from obtaining the four required satellites.

1.1.4 Line-of-sight:

A vehicle that uses a line-of-sight traffic signal pre-emption system is equipped with an emitter which typically sends a narrowly directed signal forward, towards traffic lights in front of the vehicle, to attempt to obtain right-of-way through controllable intersections before arriving at the intersection. These line-of-sight systems generally use an invisible infrared signal, or a visible strobe light which serves a dual purpose as an additional warning light. The emitter transmits visible flashes of light or invisible infrared pulses at a specified frequency. Traffic lights must be equipped with a compatible traffic signal pre-emption receiver to respond. Once the vehicle with the active emitter has passed the intersection, the receiving device no longer senses the emitter's signal, and normal operation resumes. Some systems can be implemented with varying frequencies assigned to specific types of uses, which would then allow an intersection's pre-emption equipment to differentiate between a fire engine and a bus sending a signal simultaneously, and then grant priority access first to the fire engine.

Drawbacks of line-of-sight systems include obstructions, lighting and atmospheric conditions, and undesired activations. Obstructions may be buildings on a curving road that block visual contact with a traffic signal until very close, or perhaps a large freight truck in front of a police car blocking the traffic signal from receiving the emitter's signal from the police car. Modifying the position of the receiver or even locating it separate from the traffic signal equipment can sometimes correct this problem. Direct sunlight into a receiver may prevent it from detecting an emitter, and severe atmospheric conditions, such as heavy rain or snow, may reduce the distance at which a line-of-sight system will function. Undesired activations may occur if an emitter's signal is picked up by many traffic lights along a stretch of road, all directed to change to red in that direction, prior to the activating vehicle turning off the road, or being parked without its emitter being deactivated. Line of sight emitters can use IR diodes. They are pulsed with a low-priority signal (10 Hz) or a high-priority signal (14 Hz).

1.1.5 Localized radio signal:

Radio-based traffic-pre-emption systems using a local, short-range radio signal in the 900MHz band, can usually avoid the weaknesses of line-of-sight systems (optical systems). A radio-based system still uses a directional signal transmitted from an emitter, but being radio-based, its signal is not blocked by visual obstructions, lighting or weather conditions. Until recently, the major drawback of radio-based traffic signal pre-emption systems was the possibility of interference from other devices that may be using the same frequency at a given time and location. The advent of FHSS (Frequency Hopping Spread Spectrum) broadcasting has allowed radio-based systems to not only overcome this limitation, but also the aforementioned limitations associated with acoustic and line of sight (optical) systems. It was not until recently that cost effective GPS pre-emption systems were introduced, supplanting FHSS radio-based pre-emption as the pre-emption method of choice, particularly for cities that had experienced the myriad of issues associated with other (acoustic and optical) pre-emption systems.

Radio-based systems also began to offer some additional benefits-adjustable range and collision avoidance. The operating range was adjusted by varying the radio signal strength so that traffic lights could be activated only nearby (if desired), or at greater distances. The downside to these pre-emption systems (which also performed collision avoidance) was that they would display the direction of impending collisions, but not be able to effectively (or accurately) calculate the distance to collision by any method other than RF signal strength, which was only a rough estimate at best.

1.1.6 RFID based system:

The problem of traffic light control can be solved by RFID based system. But this system is also having some drawbacks as follows.

- RFID systems are often more expensive than barcode systems.
- RFID technology is harder to understand.
- Tags are application specific. No one tag fits all.
- The error rate is quite high.
- Maintenance is very tedious.

•Traffic cannot be managed locally.

As all the existing systems have their own disadvantages, there is a need to develop a system which works more efficiently in all adverse conditions. This project proposes a system which can overcome the advantages of the systems mentioned above.

1.2 Project Outline:

Our project mainly comprises of two sub systems one placed in the emergency vehicle and the other is placed at the traffic console. These two sub systems communicate with each other through RF signal transmission. Basic radio frequency signalling can be achieved by transmitting a simple RF signal. So, if we use basic radio frequency signalling, pre-emption can be done by a simple radio frequency remote, which can lead to the misuse of pre-emption. So, there is a need to secure the network. So, we are using radio signalling with LORA module.

CHAPTER-2

RADIO SIGNALLING

Radio is the technology of using radio information waves to carry by systematically modulating properties of electromagnetic energy waves transmitted through space, such as their amplitude, frequency, phase, or pulse width. When radio waves strike an electrical conductor. the oscillating fields induce an alternating current in the conductor. The information in the waves can be extracted and transformed back into its original form.

Radio systems need a transmitter to modulate some property of the energy produced to impress a signal on it, for example using amplitude modulation or angle modulation (which can be frequency modulation or phase modulation). Radio systems also need an antenna to convert electric currents into radio waves, and radio waves into an electric current. An antenna can be used for both transmitting and receiving. The electrical resonance of tuned circuits in radios allow individual frequencies to be selected. The electromagnetic wave is intercepted by a tuned receiving antenna. A radio receiver receives its input from an antenna and converts it into a form that is usable for the consumer, such as sound, pictures, digital data, measurement values, navigational positions, etc. Radio frequencies occupy the range from a 3 kHz to 300 GHz, although commercially important uses of radio use only a small part of this spectrum.

A radio communication system requires a transmitter and a receiver, each having an antenna and appropriate terminal equipment such as a microphone at the transmitter and a loudspeaker at the receiver in the case of a voice-communication system.

Frequency	Wavelength	Designation	Abbreviation	IEEE Bands
3-30 Hz	10^5-10^4 km	Extremely-low frequency	ELF	-
30-300 Hz	10^4-10^3 km	Super-low frequency	SLF	-
300-3000 Hz	10^{3-}100 km	Ultra-low frequency	ULF	-
3-30 kHz	100-10 km	Very-low frequency	VLF	-
30-300 kHz	10-1 km	Low frequency	LF	-
300-3000 kHz	1 km-100 m	Medium frequency	MF	-
3-30 MHz	100-10 m	High frequency	HF	HF
30-300 MHz	10-1 m	Very-high frequency	VHF	VHF
300-3000 MHz	1 m-10 cm	Ultra-high frequency	UHF	UHF, L, S
3-30 GHz	10-1 cm	Super-high frequency	SHF	S,C,X,Ku,K,Ka
30-300 GHz	1cm-1 mm	Extremely-high frequency	EHF	Ka,V,W,mm
300-3000 GHz	1 mm-0.1 mm	Tremendously high frequency	THF	-

Table 2.1: Radio Spectrum

2.1 Origin of radio:

The term "radio" is derived from the Latin word "radius", meaning "spoke of a wheel, beam of light, ray". It was first applied to communications in 1881 when, at the suggestion of French scientist Ernest Mercadier, Alexander Graham Bell adopted "radiophone" (meaning "radiated sound") as an alternate name for his photophone optical transmission system. However, this invention would not be widely adopted.

Following Heinrich Hertz's establishment of the existence of electromagnetic radiation in the late 1880s, a variety of terms were initially used for the phenomenon, with early descriptions of the radiation itself including "Hertzian waves", "electric waves", and "ether waves", while phrases describing its use in communications included "spark telegraphy", "space telegraphy", "aerography" and, eventually and most commonly, "wireless telegraphy". However, "wireless" included a broad variety of related electronic technologies, including electrostatic induction, electromagnetic induction and aquatic and earth conduction, so there was a need for a more precise term referring exclusively to electromagnetic radiation.

The first use of radio- in conjunction with electromagnetic radiation appears to have been by French physicist Édouard Branly, who in 1890 developed a version of a coherer receiver he called a radio-conducteur. The radio- prefix was later used to form additional descriptive compound and hyphenated words, especially in Europe. For example, in early 1898 the British publication. The Practical Engineer included a reference to "the radiotelegraph" and "radiotelegraphy", while the French text of both the 1903 and 1906 Berlin Radiotelegraphic Conventions includes the phrases radiotelegraphique and radiotelegrammes.

The use of "radio" as a standalone word date back to at least December 30, 1904, when instructions issued by the British Post Office for transmitting telegrams specified that "The word Radio... is sent in the Service Instructions". This practice was universally adopted, and the word "radio" introduced internationally, by the 1906 Berlin Radiotelegraphic Convention, which included a Service Regulation specifying that "Radio telegrams shall show in the preamble that the service is "Radio

The switch to "radio" in place of "wireless" took place slowly and unevenly in the English-speaking world. Lee de Forest helped popularize the new word in the United States-in early 1907 he founded the Deforest Radio Telephone Company, and his letter in the June 22, 1907 Electrical World about the need for legal restrictions warned that "Radio chaos will certainly be the result until such stringent regulation is enforced". The United States Navy would also play a role. Although its translation of the 1906 Berlin Convention used the terms "wireless telegraph" and "wireless telegram", by 1912 it began to promote the use of "radio" instead. The term started to become preferred by the general public in the 1920s with the introduction of broadcasting. ("Broadcasting" is based upon an agricultural term meaning roughly "scattering seeds widely".) British Commonwealth countries continued to commonly use the term "wireless" until the mid-20th century, though the magazine of the British Broadcasting Corporation in the UK has been called Radio Times since its founding in the early 1920s.

2.2 Process of communication:

Radio systems used for communication have the following elements. With more than 100 years of development, each process is implemented by a wide range of methods, specialized for different communications purposes.

2.2.1 Transmitter and modulation:

Each system contains a transmitter; This consists of a source of electrical energy, producing alternating current of a desired frequency of oscillation. The transmitter contains a system to modulate (change) some property of the energy produced to impress a signal on it. This modulation might be as simple as turning the energy on and off, or altering more subtle properties such as amplitude, frequency. phase, or combinations of these properties. The transmitter sends the modulated electrical energy to a tuned resonant antenna; this structure converts the rapidly changing alternating current into an electromagnetic wave that can move through free space (sometimes with a particular polarization).

A signal may be carried by an AM or FM radio wave.

Amplitude modulation of a carrier wave works by varying the strength of the transmitted signal in proportion to the information being sent. For example, changes in the signal strength can be used to reflect the sounds to be reproduced by a speaker, or to specify the light intensity of television pixels. It was the method used for the first audio radio transmissions, and remains in use today. "AM" is often used to refer to the medium wave broadcast band (see AM radio), but it is used in various radiotelephone services such as the Citizens Band, amateur radio and especially in aviation, due to its ability to be received under very weak signal conditions and its immunity to capture effect. allowing more than one signal to be heard simultaneously.

Frequency modulation varies the frequency of the carrier. The instantaneous frequency of the carrier is directly proportional to the instantaneous value of the input signal. FM has the "capture effect" whereby a receiver only receives the strongest signal, even when others are present. Digital data can be sent by shifting the carrier's frequency among a set of discrete values, a technique known as frequency shift keying. FM is commonly used at Very high frequency (VHF) radio frequencies for high fidelity broadcasts of music and speech (see FM broadcasting). Analog TV sound is also broadcast using FM.

Angle modulation alters the instantaneous phase of the carrier wave to transmit a signal. It may be either FM or phase modulation (PM).

2.2.2 Antenna:

An antenna (or aerial) is an electrical device which converts electric currents into radio waves, and vice versa. It is usually used with a radio transmitter or radio receiver. In transmission, a radio transmitter supplies an electric current oscillating at radio frequency (i.e. high frequency AC) to the antenna's terminals, and the antenna radiates the energy from the current as electromagnetic waves (radio waves). In reception, an antenna intercepts some of the power of an electromagnetic wave in order to produce a tiny voltage at its terminals that is applied to a receiver to be amplified. Some antennas can be used for both transmitting and receiving, even simultaneously, depending on the connected equipment.

2.2.3 Propagation

Once generated, electromagnetic waves travel through space either directly, or have their path altered by reflection, refraction or diffraction. The intensity of the waves diminishes due to geometric dispersion (the inverse-square law); some energy may also be absorbed by the intervening medium in some cases. Noise will generally alter the desired signal; this electromagnetic interference comes from natural sources, as well as from artificial sources such as other transmitters and accidental radiators. Noise is also produced at every step due to the inherent properties of the devices used. If the magnitude of the noise is large enough, the desired signal will no longer be discernible; the signal-to-noise ratio is the fundamental limit to the range of radio communications.

2.2.4 Resonance:

Electrical resonance of tuned circuits in radios allows individual stations to be selected. A resonant circuit will respond strongly to a particular frequency and much less so to differing frequencies. This allows the radio receiver to discriminate between multiple signals differing in frequency.

2.2.5 Receiver and demodulation:

A crystal receiver, consisting of an antenna, adjustable electromagnetic coil, crystal rectifier, capacitor, headphones and ground connection.

The electromagnetic wave is intercepted by a tuned receiving antenna; this structure captures some of the energy of the wave and returns it to the form of oscillating electrical currents. At the receiver, these currents are demodulated, which is conversion to a usable signal form by a detector sub system. The receiver is "tuned" to respond preferentially to the desired signals, and reject undesired signals.

Early radio systems relied entirely on the energy collected by an antenna to produce signals for the operator. Radio became more useful after the invention of electronic devices such as the vacuum tube and later the transistor, which made it possible to amplify weak signals. Today radio systems are used for applications from walkie-talkie children's toys to the control of space vehicles, as well as for broadcasting, and many other applications.

A radio *receiver* receives its input from an antenna, uses electronic filters to separate a wanted radio signal from all other signals picked up by this antenna, amplifies it to a level suitable for further processing, and finally converts through demodulation and decoding the signal into a form usable for the consumer, such as sound, pictures, digital data, measurement values, navigational positions, etc.

2.3 Special properties of RF current:

Electric currents that oscillate at radio frequencies have special properties not shared by direct current or alternating current of lower frequencies.

- Energy from RF currents in conductors can radiate into space as electromagnetic waves (radio waves). This is the basis of radio technology.

- RF current does not penetrate deeply into electrical conductors but tends to flow along their surfaces; this is known as the skin effect.
- RF currents applied to the body often do not cause the painful sensation and muscular contraction of electric shock that lower frequency currents produce. This is because the current changes direction too quickly to trigger depolarization of nerve membranes. However, this does not mean RF currents are harmless. They can cause internal injury as well as serious superficial burns called RF burns.

- RF current can easily ionize air, creating a conductive path through it. This property is exploited by "high frequency" units used in electric arc welding, which use currents at higher frequencies than power distribution uses.
- Another property is the ability to appear to flow through paths that contain insulating material, like the dielectric insulator of a capacitor. This is because capacitive reactance in a circuit decreases with frequency.

- In contrast, RF current can be blocked by a coil of wire, or even a single turn or bend in a wire. This is because the inductive reactance of a circuit increases with frequency.

- When conducted by an ordinary electric cable, RF current has a tendency to reflect from discontinuities in the cable such as connectors and travel back down the cable toward the source, causing a condition called standing waves. Therefore, RF current must be carried by specialized types of cable called transmission line.

2.4 Radio communication:

To receive radio signals an antenna must be used. However, since the antenna will pick up thousands of radio signals at a time, a radio tuner is necessary to tune into a particular frequency (or frequency range). This is typically done via a resonator in its simplest form, a circuit with a capacitor and an inductor form a tuned circuit. The resonator amplifies oscillations within a particular frequency band, while reducing oscillations at other frequencies outside the band. Another method to isolate a particular radio frequency is by oversampling (which gets a wide range of frequencies) and picking out the frequencies of interest, as done in software defined radio.

The distance over which radio communications is useful depends significantly on things other than wavelength, such as transmitter power, receiver quality, type, size, and height of antenna, mode of transmission, noise, and interfering signals. Ground waves, tropospheric scatter and sky waves can all achieve greater ranges than line-of-sight propagation. The study of radio propagation allows estimates of useful range to be made.

2.5 Types of wireless protocols:

People sometimes refer to wireless networking as "Wi-Fi" even when the network uses a totally unrelated kind of wireless technology. While it might seem ideal that all of the world's wireless devices should use one common network protocol such as Wi-Fi, today's networks support a wide variety of different protocols instead. The reason: No one protocol in existence provides an optimal solution for all of the different wireless usages people want. Some are better optimized to conserve battery on mobile devices, while others offer higher speeds or more reliable and longer-distance connections.

The below wireless network protocols have proven especially useful in consumer devices and business environments.

Fig-2.1 Types of wireless protocols

2.5.1 LTE:

Before newer smartphones adopted so-called fourth-generation ("4G") wireless networking. phones utilized a dizzying variety of older generation cellular communication protocols with names such as HSDPA, GPRS, and EV-DO. Phone carriers and the industry have invested large sums of money to upgrade cell towers and other network equipment to support 4G, standardizing on a communication protocol called Long Term Evolution (LTE) that emerged as a popular service starting in 2010.

LTE technology was designed to significantly improve the low data rates and roaming issues with older phone protocols. The protocol can carry more than 100 Mbps of data, although the network bandwidth is normally regulated to levels below 10 Mbps for individual users. Due to the significant cost of equipment, plus some government regulatory challenges, phone carriers have not yet deployed LTE in many locations, LTE is also not suitable for home and other local area networking, being designed to support a larger number of customers across much longer distances (and corresponding higher cost).

2.5.2 Wi-Fi:

Wi-Fi is widely associated with wireless networking as it has become the de facto standard for home networks and public hotspot networks. Wi-Fi became popular starting in the late 1990s as the networking hardware required to enable PCs, printers, and other consumer devices became widely affordable and the supported data rates were improved to acceptable levels (from 11 Mbps to 54 Mbps and above).

Although Wi-Fi can be made to run over longer distances in carefully controlled environments, the protocol is limited to work within single residential or commercial buildings and outdoor areas within short walking distances. Wi-Fi speeds are also lower than for some other wireless protocols. Mobile devices increasingly support both Wi-Fi and LTE (plus some older cellular protocols) to give users more flexibility in the kinds of networks they can use.

Wi-Fi Protected Access security protocols add network authentication and data encryption capabilities to Wi-Fi networks. Specifically, **WPA2** is recommended for use on home networks to prevent unauthorized parties from logging into the network or intercepting personal data sent over the air.

2.5.3 Bluetooth:

One of the oldest wireless protocols still broadly available, Bluetooth was created in the 1990s to synchronize data between phones and other battery-powered devices. Bluetooth requires a lower amount of power to operate than Wi-Fi and most other wireless protocols. In return, Bluetooth connections only function over relatively short distances, often 30 feet (10 m) less and support relatively low data rates, usually 1-2 Mbps, Wi-Fi has replaced Bluetooth on some newer equipment. but many phones today still support both of these protocols.

2.5.4 60 GHz Protocols - WirelessHD and WiGig:

One of the most popular activities on computer networks is streaming of video data, and several wireless protocols that run on 60 Gigahertz (GHz) frequencies have been built to better support this and other usages which require large amounts of network bandwidth. Two different industry standards called WirelessHD and WiGig were created in the 2000s both using 60 GHz technology to support high bandwidth wireless connections: WiGig offers between 1 and 7 Gbps of bandwidth while WirelessHD supports between 10 and 28 Gbps.

Although basic video streaming can be done over Wi-Fi networks, best quality high-definition video streams demand the higher data rates these protocols offer. The very high signalling frequencies of WirelessHD and WiGig compared to Wi-Fi (60 GHz versus 2.4 or 5 GHz) greatly limit connection range, generally shorter than Bluetooth, and typically to within a single room (as 60 GHz signals do not penetrate walls effectively).

2.5.5 Wireless Home Automation Protocols-Z-Wave and Zigbee:

Various network protocols have been created to support home automation systems that allow remote control of lights, home appliances, and consumer gadgets. Two prominent wireless protocols for home automation are Z-Wave and Zigbee. To achieve the extremely low energy consumption required in home automation environments, these protocols and their associated hardware support only low data rates - 0.25 Mbps for Zigbee and only about 0.01 Mbps for Z-Wave. While such data rates are obviously unsuitable for general-purpose networking, these technologies work well as interfaces to consumer gadgets which have simple and limited communication requirements.

2.5.6 LoRa and LoRaWAN:

LoRa is developed in the year 2009 by accompany called Semtech, since the invention of LoRa, the Long Range, low power Internet of Things (IOT) platform currently implemented in over 100 million devices worldwide and accelerating the global adoption of IoT.LoRa is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. LoRa creates only a physical layer method of wireless transport, such as a transceiver chip. That means it lacks the proper network protocols to manage traffic for data collection and endpoint device management. This is where Long-Range WAN -- or LoRaWAN -- comes into the picture.

LoRa devices and networks such as the LoRaWAN enable smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, and disaster prevention. Semtech's LoRa devices have amassed several hundred known uses cases for smart cities, homes and buildings, communities, metering, supply chain and logistics, agriculture, and more.

CHAPTER-3

METHODOLOGY

As said earlier this project mainly comprises of two subsystems. One placed in the emergency vehicle and the other placed at the traffic console. These two subsystems communicate with each other through radio signalling. Here, we are using LoRa modules for wireless communication. We choose LoRa modules because it provides Long Range and consumes low power. Communication is established between the LoRa modules, one placed along with Arduino in emergency vehicle and other placed along with Arduino in traffic control subsystem.AT Commands are used to configure the LoRa modules. As LoRa is a TransReceiver, an acknowledgement is also possible for the driver in the emergency vehicle through feedback mechanism.

3.1 Overview of the system proposal:

.

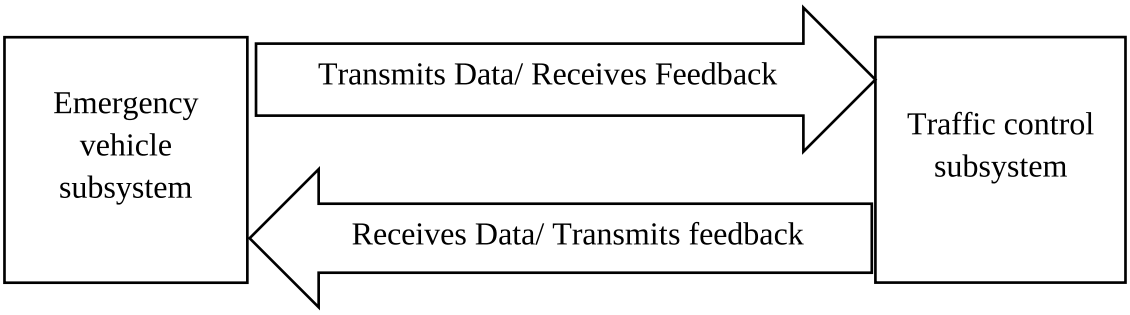


Fig 3.1: Overview of the system

3.2 EMERGENCY VEHICLE SUBSYSTEM:

In this sub system, LoRa trans receiver is connected with Arduino and placed in Emergency vehicle, whenever the driver is at a distance of up to 2-5 km from the traffic lights, he presses the button to send signal from the LoRa transReceiver.Lora actually sends a payload which contains the address of the receiver Lora. After successfully receiving the data, an

acknowledgement is sent by the receiver to the transmitter which signifies the successful transmission of the data from the transmitter.

3.2.1 Block Diagram:

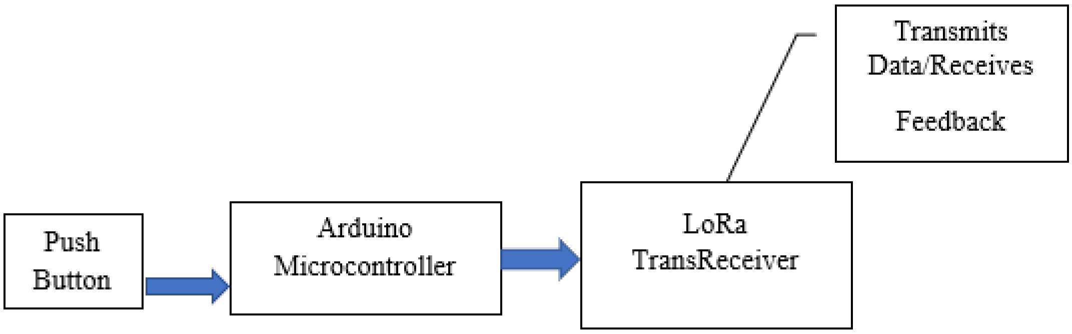


Fig 3.2: Emergency vehicle sub system

3.3 TRAFFIC CONSOLE SUBSYSTEM:

This traffic console subsystem acts upon the commands of paired Bluetooth devices. This sub system consists of a LoRa TransReceiver along with Arduino microcontroller connected with traffic lights, if LoRa receives data from the transmitter, then it sends data to Arduino. This Arduino processes this information and turns red light to green in that particular direction and also sends feedback to the transmitter signifying the successful communication between the LoRa modules.

3.3.1 Block diagram

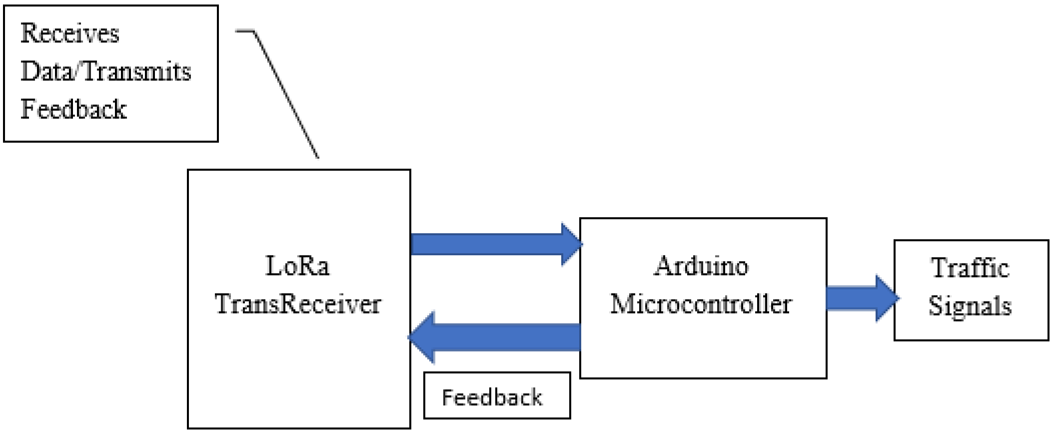


Fig 3.3: Traffic console subsystem

CHAPTER-4

LORA TECHNOLOGY

LoRa technology was developed by a company called Semtech and it is a new wireless protocol designed specifically for long-range, low-power communications. LoRa stands for Long Range Radio and is mainly targeted for M2M and IoT networks. This technology will enable public or multi-tenant networks to connect a number of applications running on the same network.

LoRa is a physical layer technology that works in unlicensed sub-GHz ISM band and is based on chirped spread spectrum (CSS) technique. CSS is a wideband linear frequency modulation in which carrier frequency varies for the defined extent of time. LoRa works on pure ALOHA principles and supports different ISM frequencies, namely 868 (Europe), 915 (North America), and 433 MHz (Asia). It is basically single-hop technology, which relays the messages received from LoRa sensor nodes to the central server via gateways. The data transmission rate supported by LoRa varies from 300 bps to 50 kbps, depending on spreading factor (SF) and channel bandwidth settings. LoRa transmissions with different SFs are quasiorthogonal and allow multiple transmissions with different SFs simultaneously. To support LoRa on Internet, LoRa alliance has developed long-range wide-area network (LoRaWAN), which includes the network and upper layer functionalities.

LoRa also features an adaptive data rate algorithm to help maximize the nodes battery life and network capacity. The LoRa protocol includes a number of different layers including encryption at the network, application and device level for secure communications.

4.1 About LoRa:

LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. Semtech's LoRa is a long range, low power wireless platform that has become the de facto wireless platform of Internet of Things (IoT). LoRa devices and networks such as the LoRaWAN enable smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, and disaster prevention. Semtech's LoRa devices have amassed several hundred known uses cases for smart cities, homes and buildings, communities, metering, supply chain and logistics, agriculture, and more. With hundreds of millions of devices connected to networks in more than 100 countries and growing, LoRa is creating a smarter planet.

LoRa devices and the LoRaWAN standard offer compelling features for IoT applications including long range, low power consumption and secure data transmission. The technology is utilized by public, private or hybrid networks and provides greater range than Cellular networks. Deployments can easily integrate into existing infrastructure and enable low-cost battery-operated IoT applications. Semtech's LoRa chipsets are incorporated into devices manufactured by a large ecosystem of IoT solution providers, and connected to networks around the globe. Simply stated, LoRa connects devices to the Cloud, providing a "voice" to things — making the world a better place to live, work and play.

4.2 History of LORA:

- The story of LoRa began in 2009, when two friends in France aimed at developing a long range, low power modulation technology. Despite encountering resistance, as with most disruptive technologies, Nicolas Sornin and Olivier Seller continued dedicating their time to turn the idea into a reality.
- In 2010, Nicolas and Olivier met their third partner, François Sforza, and together they started the company *Cycleo*. Initially, the three founders were targeting the metering industry and aimed at adding wireless communication capabilities for gas, water and electricity meters.
- In May 2012. Semtech collaborated with Nicolas, Olivier and François to further improve the technology and finalize the chips required for the end devices (SX1272 and SX1276), as well as for the gateways (SX1301).
- In February 2015, the LoRa Alliance® was founded and the networking protocol was renamed "LoRaWAN." The LoRa Alliance's goals were, and still are, to "*support and promote the global adoption of the LoRaWAN standard by ensuring interoperability of all LoRaWAN products and technologies.*"

4.3 Operating frequencies and data rates:

LoRa transmits over license-free megahertz radio frequency bands:

- 169 MHz, 433 MHz (Asia)
- 868 MHz (Europe)
- 915 MHz (North America)

The 868MHz band is used in Europe for a number of applications, including long range wireless networking. The 915MHz band is a part of industrial, scientific and medical (ISM) frequency bands. The 915MHz frequency band is used mainly in North America.

LoRa uses a proprietary spread spectrum modulation that is similar to and a derivative of chirp spread spectrum (CSS) modulation. The spread spectrum LoRa modulation is performed by representing each bit of payload information by multiple chirps of information. The rate at which the spread information is sent is referred to as the symbol rate, the ratio between the nominal symbol rate and chirp rate is the spreading factor (SF) and represents the number of symbols sent per bit of information. The result is an M- array digital modulation, where the $\mathbf{M=2^{SF}}$ possible waveforms at the output of the modulator are chirp modulated signals over the frequency interval $\mathbf{(f_0 - B/2, f_0 + B/2)}$ with M different initial frequencies: the instantaneous frequency is linearly increased, and then wrapped to $\mathbf{f_0 - B/2}$ when it reaches the maximum frequency $\mathbf{f_0 + B/2}$.

LoRa can trade off data rate for sensitivity with a fixed channel bandwidth by selecting the amount of spread used (a selectable radio parameter from 6 to 12). Lower SF means more chirps are sent per second; hence, you can encode more data per second. Higher SF implies fewer chirps per second; hence, there are fewer data to encode per second. Compared to lower SF, sending the same amount of data with higher SF needs more transmission time, known as airtime. More airtime means that the modem is up and running longer and consuming more energy. The benefit of high SF is that more extended airtime gives the receiver more opportunities to sample the signal power, which results in better sensitivity. The LoRa modem allows changing of the transmission power from 2dBm to 14dBm (433 MHz) or as high as 20dBm (865 MHz to 867 MHz, 915 MHz, and 923 MHz) as per the regulations of each country.

4.4 Chirp spread spectrum (CSS):

Chirp:

Chirp stands for 'Compressed High Intensity Radar Pulse'. It is a signal which frequency either increase or decrease with time. It is very common is sonar and radar. It is also used in spread spectrum.

In digital communications, **chirp spread spectrum** (**CSS**) is a spread spectrum technique that uses wideband linear frequency modulated chirp pulses to encode information. A chirp is a sinusoidal signal whose frequency increases or decreases over time.

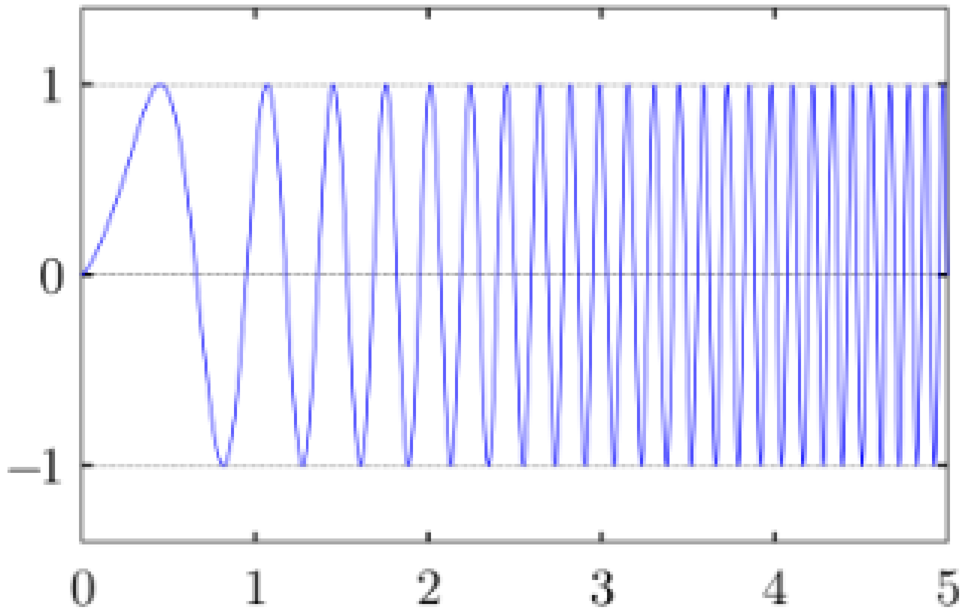
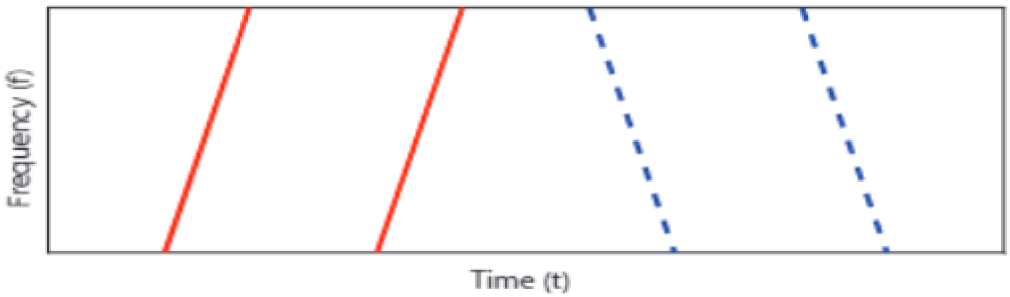


Fig 4.1: A linear frequency modulated upchirp in the time domain. Other types of upchirps may increase exponentially over

As with other spread spectrum methods, chirp spread spectrum uses its entire allocated bandwidth to broadcast a signal, making it robust to channel noise. Further, because the chirps utilize a broad band of the spectrum, chirp spread spectrum is also resistant to multi-path fading even when operating at very low power. However, it is unlike direct - sequence spread spectrum (DSSS) or frequency-hopping spread spectrum (FHSS) in that it does not add any pseudo-random elements to the signal to help distinguish it from noise on the channel, instead relying on the linear nature of the chirp pulse. Additionally, chirp spread spectrum is resistant to the Doppler effect, which is typical in mobile radio applications.

Chirp Spread Spectrum was developed for radar applications. Chirp signals have constant amplitude and pass the whole bandwidth in a linear or non-linear way from one end to another end in a certain time. Chirp spread spectrum uses complete bandwidth to transmit signals. If the frequency changes from lowest to highest, it is call up-chirp and if the frequency changes from highest to lowest, we call it down-chirp. A chirp is a tone in which frequency changes with time.

Fig 4.2: In the below figure shows chirps, the solid lines represent an up-chirp whereas the dotted lines represent a down-chirp.



The chirps in LoRa modulation are cyclic and the frequency jumps determine how data is encoded. A symbol represents one or more bits of data. If the symbol has a spreading factor of 7, it carries 7 bits and the values can range from 0-127.These divisions of frequency are called chips. The chips/s are defined by the bandwidth. For a bandwidth of 125 kHz, we can transfer 125000 chips/s. The symbol rate can be calculated as

$$\mathbf{R_s = BW/2^{SF} = R_C/2^{SF}\ [symbols/s]}$$

where BW is the bandwidth, R_C is the chip rate and SF is the spreading factor. The time-on-air is directly affected by the spreading factor. A larger spreading factor signifies a higher symbol duration (or in other words, more raw bits) and more time-on-air, hence mapping a larger physical range. This relation of SF and signal range has direct implications on the RSSI to distance mapping.

Chirp spread spectrum was originally designed to compete with ultra-wideband for precision ranging and low-rate wireless networks in the 2.45 GHz band. It is ideal for applications requiring low power usage and needing relatively low data rates (1 Mbit/s or less). It is also be used in the future for military applications as it is very difficult to detect and intercept when operating at low power.

4.5 LORA Architecture:

A LoRa device is typically based on reference hardware architecture, as shown below:

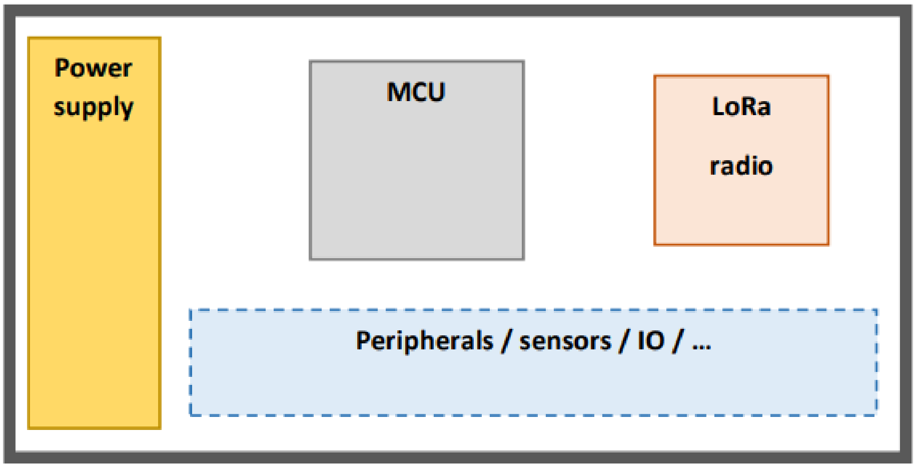


Fig 4.3: Lora Architecture

- Power supply may be provided through a power plug or a battery
- The MCU is the microcontroller managing all device functionalities, and implementing the Lora WAN stack.
- The LoRa radio is composed by the LoRa transceiver, the antenna matching circuitry and the antenna itself.
- Peripherals may be sensors like accelerometers or temperature sensors, or I/O such as relay or display.
- Depending on design and production constraints, several options are available to build a LoRa device
- Design based on a LoRa chipset
- Design based on a LoRa module
- Design based on a RF-MCU
- Design based on LoRa modem
- Existing device with external LoRa modem

The choice of the target architecture needs to be made based on criteria such as expected production quantities, RF engineering skills available and the development timeline available to complete the project. In order to select the optimal architecture, it is preferable to start with a LoRa starter-kit and create a mock-up.

The generic device software architecture can be described as below:

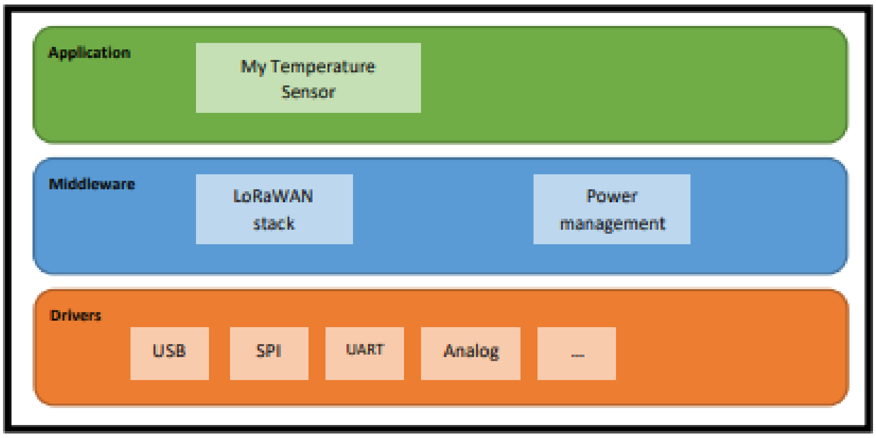


Fig 4.4: Generic device software architecture

The driver layer provides the hardware adaptation and implements all the drivers to manage the device peripherals. It abstracts the hardware as simple functions exposed to the middleware. The middleware implements the communication protocol libraries (LoRaWAN, LowPAN…), it implements also complex drivers like screen, GPS driver. The application layer contains all functional applications where the device behavior and functionalities are implemented.

4.5.1 Design based on a LORA chipset:

This design follows the generic architecture with the LoRa radio part based on the LoRa transceiver from Semtech and the antenna plus associated circuitry from any vendor.

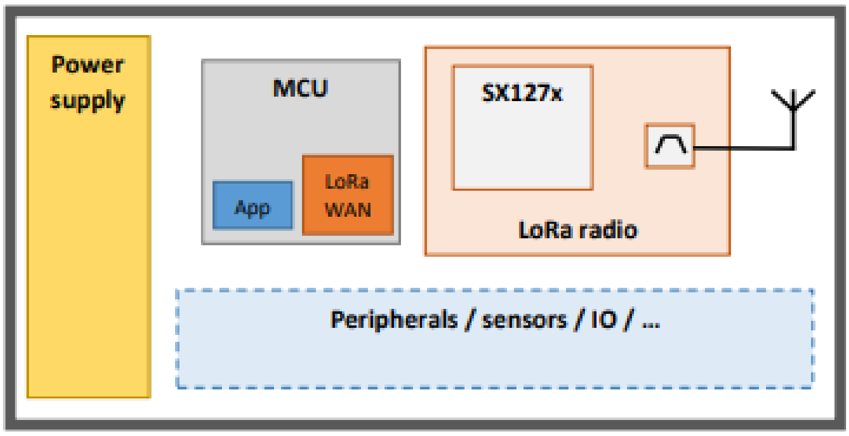


Fig 4.5: Lora chipset device architecture

In this architecture, the device developer must take responsibility for the entire RF design, including PCB routing constraints, antenna tuning and emission/immunity issues. The entire LoRaWAN stack here must be managed by the main MCU, and implemented by the device developer or manufacturer.

4.5.2 Design based on a LORA module:

A LoRa module is a component containing the MCU and the LoRa radio. The MCU is available for software programming in order to run the application and the LoRaWAN stack. The main benefit of this design approach is that all RF hardware development is implemented by the module manufacturer. The antenna tuning and matching is mostly done inside the module. The module manufacturer provides a reference design to connect or redesign an antenna.

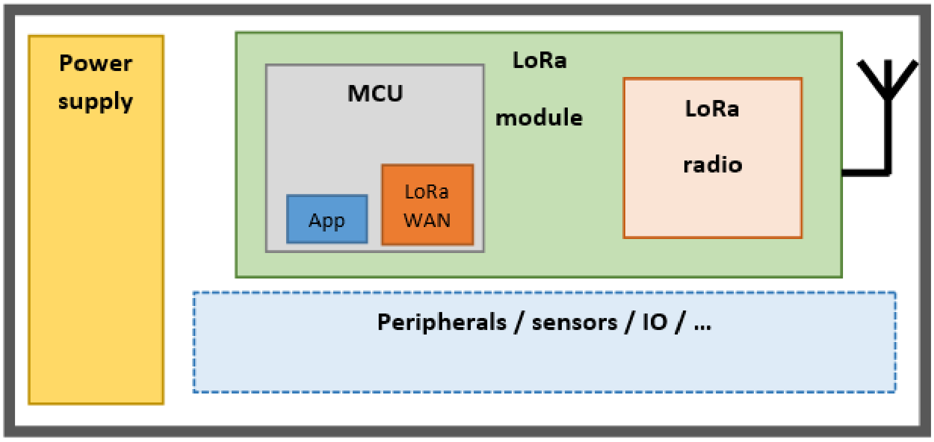


Fig 4.6: Lora module device architecture

Compared to the previous architecture, as the MCU is available, the device manufacturer is only responsible for the integration of the LoRaWAN stack. Depending of the module provider, the LoRaWAN stack may be delivered as a library.

Design based on a RF-MCU:

A RF-MCU is a SoC (System-on-Chip) including a MCU and a LoRa transceiver in a same silicon package. The RF constraints remain the same as in previous RF design but this architecture offers major gains in overall device size. From the developer point of view, this design is similar to a module.

Design based on a LORA modem:

A LoRa modem is a component containing all the radio related components: stack + RF circuitry. Some modems may also include an integrated antenna, or a reference design is provided by the modem maker onhow to design or connect an external antenna.

The modem integrates the entire LoRaWAN stack and acts as a slave so it requires a host in order to manage it.The communication is usually done through AT commands to configure it or to send messages.

The hardware interface could be usually an UART, USB or SPI.

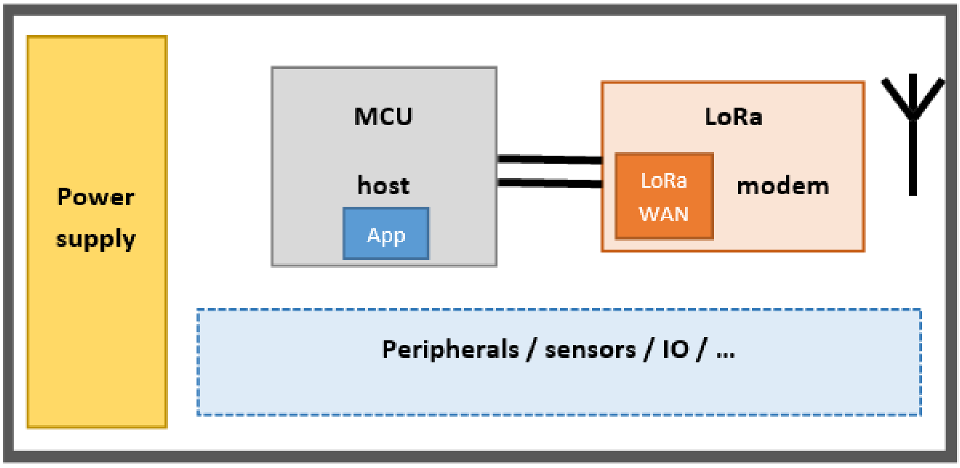


Fig 4.7: Lora modem device architecture

Several manufacturers propose a LoRa modem such as Microchip RN2483, MultiTech or also ATIM.

Existing device with external LORA modem:

A LoRa modem can also be an external stand-alone device, connected to an existing device through a USBconnection for example. In that case, the existing device must have an available external connection and also be able to power the external LoRa modem. The modem integrates the entire Lora WAN stack and acts as a slave so it requires a host in order to manage it.

The communication is usually done through AT commands to configure it or to send messages.

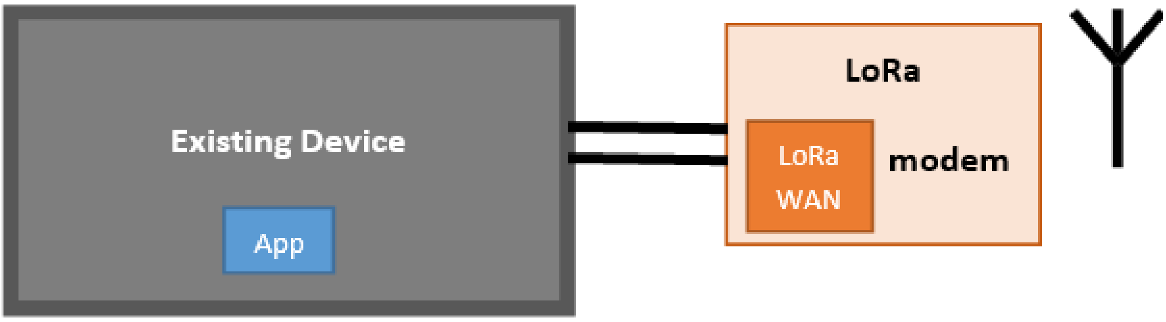


Fig 4.8: Lora external modem architecture

4.6 Architecture selection:

The choice between a homegrown design and a module or modem-based architecture is mostly done dependingon the following parameters:

- Maturity of the device specifications: feasibility test phase, market test phase or mass production phase
- Electronic and RF development skills
- Budget for development
- Project timeline
- Expected quantities and expected market price
- Software development resources availability
- RF region (ISM band)

The ISM band where the device will be deployed will impact the architecture as the antenna matching and the antenna itself must be tuned to fit the correct frequency band. The module and modem architecture, which already includes the RF part, will specify in which band it is available.

In terms of a project timeline, the key tasks to be estimated are:

- Feasibility study, is dev-kit available with the target architecture?

- Design specifications
- Schematics
- PCB routing
- Software development
- Proto v1 debug - If modification is needed: Proto v1 review (schematics + routing)
- Manufacturing test bench development + test software

4.7 Antenna design:

The antenna is a critical part in a communicating device and such a device where the sensitivity is a major issue in the LoRa technology.

In the 868MHz band, a λ/4 antenna is 8.2cm length, so depending on the product this has to be taken into account during the mechanical design to place correctly the antenna to avoid closed disturbance from the product itself.

There are 3 typical antenna architectures:

- A dedicated OEM antenna tuned for the LoRa frequency: this solution is the easiest one as the dedicated antenna is the perfect fit to achieve the best results in terms of transmission and sensitivity. However, this solution is the most expensive.

- A PCB-Antenna which is a simple trace on the circuit however parameters like copper width and thickness must be taken into account during the antenna design.

- A simple quarter wave wire antenna (λ/4 length) could be the easiest implementation. The main challenge is to ensure repeatability in its production and to deliver the correct antenna length on every product. Such design must also have a correct mechanical part to maintain the wire properly inside the product in order to ensure a consistent RF performance on all manufactured products.

4.8 Device autonomy:

Device autonomy requirements must be taken into account before hardware and software development phases as both may have a huge impact on power consumption.

From the hardware point of view:

- Choice of the MCU
- Choice of the battery and its power management
- Hardware issues (pull-up and pull-down resistors, pins not connected, mA leak inside components)
- Sleeping mode of peripherals

In order to estimate the lifetime of a battery, 5 major consumption modes must be known:

- Sleeping: everything is OFF or sleeping
- Idle: everything is sleeping except the MCU
- Running: the device is awake and running its functionalities
- LoRa Tx: the device is sending a data
- LoRa Rx: the device is listening or receiving a data

Once each of these 5 power modes are known, the developer needs to estimate how long the device will stay in each mode to calculate the average power consumption per hour.

CHAPTER 5

HARDWARE USED

Hardware is the general term used to specify the physical parts of a system. The hardware components used in our project are

1) Arduino Uno R3 ---02

2) LoRa Reyax Rylr998 ---02

3) Push to ON buttons ---01

4)RS232 USB to TTL Converter ---01

5) Green LEDs ---05

6) Yellow LEDs ---05

7) Red LEDs ---05

8) Resistors

8) Jumper wires as per our requirement

A Brief Review on the Components Used in the Proposed Work and their Specifications are provided below.

5.1 ARDUINO UNO REV3:

5.1.1 Why Arduino?

We have wide range of microcontrollers available; still we selected Arduino as it is a key note to learn new things. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges. As this project is first hands on project for all four of us, we will be able to do things better with the Arduino. Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino microcontrollers are available in large variants. We are using Arduino Uno R3 in our project as it is meeting all our requirements. Arduino boards are able to read inputs and can turn it into an output. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

5.1.2 Specifications of Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (0-13) (of which 6 can be used as PWM outputs), 6 analog inputs (A0-A5), 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 an AC-to-DC adapter or battery to get started.

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	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock speed	16MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 5.1: Summarizes the main specifications of Arduino Uno.

5.2 Arduino Uno R3 Pin Diagram

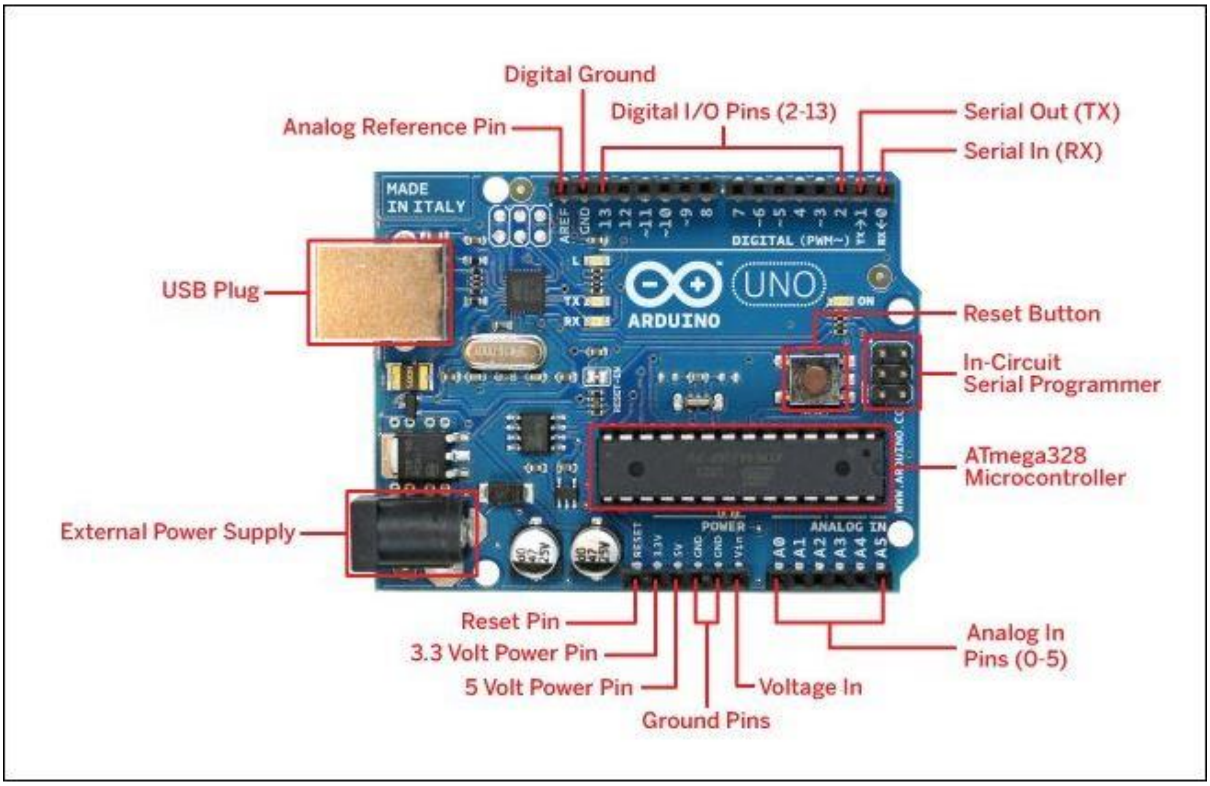


Fig 5.1: Arduino Uno Pin Diagram

Power Supply

The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection. The exterior power supply (6 to 20 volts) mainly includes a battery or an AC to DC adapter. The connection of an adapter can be done by plugging a center-positive plug (2.1mm) into the power jack on the board. The battery terminals can be placed in the pins of Vin as well as GND. The power pins of an Arduino board include the following.

Vin

The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply). By using this pin, one can supply the voltage.

5Volts

The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.

3V3

A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.

GND

GND (ground) pins

Memory

The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilized for the Boot loader), and also it includes SRAM-2 KB as well as EEPROM-1KB.

Input and Output

We know that an arguing Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read (), and digital Write (). These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm pull up resistor. The maximum current on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

Serial Pins

The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data. The connection of these pins can be done with the equivalent pins of the ATmega8 U2 USB to TTL chip.

External Interrupt Pins

The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value.

PWM Pins

The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write ().

SPI (Serial Peripheral Interface) Pins

The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the SPI communication with the help of the SPI library.

LED Pin

An arguing board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high, the LED will glow otherwise it will not glow.

TWI (2-Wire Interface) Pins

The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI with the help of Wire library.

AREF (Analog Reference) Pin

An analog reference pin is the reference voltage to the inputs of an analog i/ps using the function like analog Reference ().

Reset (RST) Pin

This pin brings a low line for resetting the microcontroller, and it is very useful for using an RST button toward shields which can block the one over the Arduino R3 board.

Communication

The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 1602 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a.inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pin.

UART

An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.

I2C

An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.

SPI Pins

The SPI communication includes MOSI, MISO, and SCK.

MOSI (Pin11)

This is the master out slave in the pin, used to transmit the data to the devices

MISO (Pin12)

This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which is produced by the master.

SCK (Pin13)

The CLK pulse synchronizes data transmission that is generated by the master. Equivalent pins with the SPI library are employed for the communication of SPI. ICSP (in-circuit serial programming) headers can be utilized for programming ATmega microcontroller directly with the boot loader.

5.2.1 Programming:

The Arduino Uno can be programmed with the Arduino IDE software. The ATmega328 on the Arduino Uno comes preburned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK.500 protocol.

5.2.2 Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano-farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e., anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line; see this forum thread for details.

5.2.3 USB Overcurrent Protection:

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

5.3 LoRa RYLR998:

The RYLR998 transceiver module feature the Lora long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption.

FEATURES

- NUVOTON MCU & Semtech LoRa Engine
- Excellent blocking immunity
- Smart receiving power saving mode
- High sensitivity
- Control easily by AT commands
- Built-in antenna

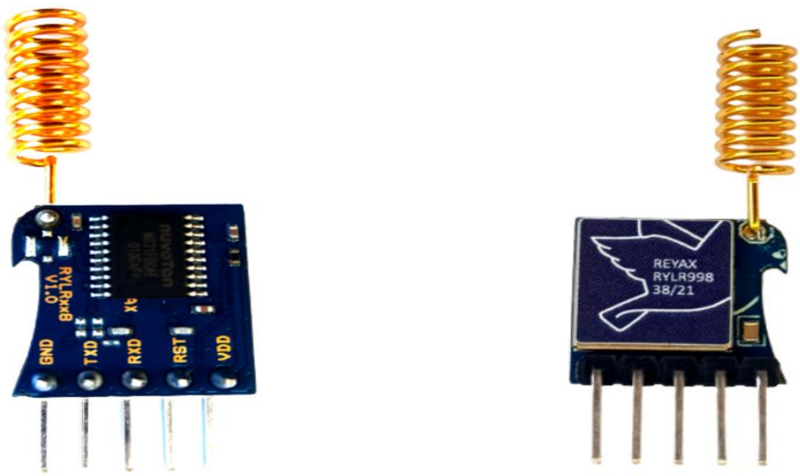


Fig 5.2: LoRa RYLR998 TransReceiver:

PIN	INPUT/OUTPUT	Condition
VDD	I	Power Supply
NRST	I	RESET (Active Low) 100KΩ Internal pull up, Pull down at least 100ms
RXD	I	UART Data Input
TXD	O	UART Data Output
GND	-	Ground

Fig 5.3: LoRa RYLR998 Pin Description

5.3.1 Hardware features:

- VDD Power supply - 3.3V (Typical),3.6V (Max)
- RF Output power range - 22dBm
- RF Sensitivity - -129dBm
- Frequency range - 868/915 MHz
- Transmit Mode current - 140 mA
- Receive Mode current - 17.5 mA

5.3.2 Software features:

AT Commands are used to configure LoRa module

THE SEQUENCE OF USING AT COMMAND:

- Use “AT+ADDRESS” to set ADDRESS. The ADDRESS is regard as the identification of transmitter or specified receiver.
- Use “AT+NETWORKID” to set the ID of Lora network. This is a Group function. Only by setting the same NETWORKID can the modules communicate with each other. If the

ADDRESS of specified receiver belongs to different group, it is not able to communicate with each other.

- Use” AT+BAND” to set the center frequency of wireless band. The transmitter and the receiver are required to use the same frequency to communicate with each other.
- Use” AT+PARAMETER” to set the RF wireless parameters. The transmitter and the receiver are required to set the same parameters to communicate with each other. The parameters of which as follows:
    - The larger the SF is, the better the sensitivity is. But the transmission time will take longer.
    - The smaller the bandwidth is, the better the sensitivity is. But the transmission time will take longer.
    - The coding rate will be the fastest if setting it as 1.
    - Preamble code. If the preamble code is bigger, it will result in the less opportunity of losing data. Generally, preamble code can be set above 10 if under the permission of the transmission time. Recommend to set “AT + PARAMETER = 9,7,1,12”
- Use “AT+SEND” to send data to the specified ADDRESS. Please use “Lora Modem Calculator Tool” to calculate the transmission time. Due to the program used by the module, the payload part will increase more 8 bytes than the actual data length.

5.4 Settings to configure LoRa Module:

RS232 USB to TTL converter: The TTL-232R cables are a family of USB to TTL serial UART converter cables incorporating FTDI’s FT232RQ USB to Serial UART interface IC device which handles all the USB signalling and protocols. The cables provide a fast, simple way to connect devices with a TTL level serial interface to USB.

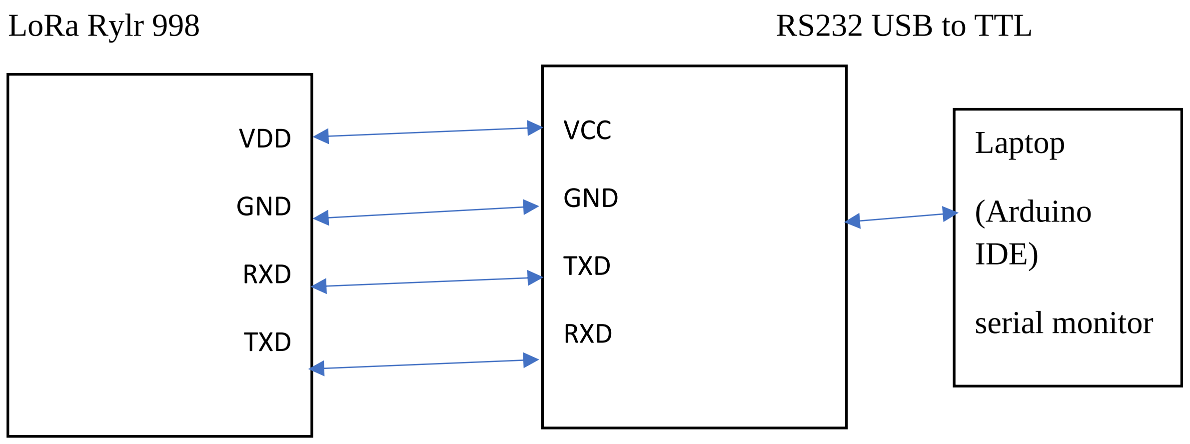


Fig 5.3: Configuring LoRa Module

Note: For every AT command, LoRa module will respond +OK.After the response type another command.

Connect RS232 to LoRa as showed in the above figure:

- Select Port in Arduino IDE
- Open Serial Monitor
- Select BAUD Rate to 115200 and select BOTH NL AND CR

Configuring LoRa Module (In transmitter side):

- AT+ADDRESS=1
- AT+NETWORKID=5
- AT+BAND=865000000 i.e., 865MHz

Configuring LoRa Module (In Receiver side):

- AT+ADDRESS=2
- AT+NETWORKID=5
- AT+BAND=865000000

Note:

- To check the parameters, we have set during configuring the LoRa modules through AT commands.

  Type AT+ADDRESS?

  AT+NETWORKID?
- If there is error while typing the AT commands the modules respond with +ERR.
- While sending serial data from Arduino to LoRa, we use

  AT + SEND = RECEIVER ADDRESS, LENGTH OF THE MESSAGE, MESSAGE.
- LENGTH OF THE MESSAGE must be same as the number of the characters of the MESSAGE, otherwise communication won't work.

CHAPTER 6

SOFTWARE TOOLS

In our project we used Arduino IDE Software.

6.1 ARDUINO IDE:

Arduino IDE is an open-source Arduino software. This makes it easy to write and upload code on to the board. It runs on Windows, Mac OS X and Linux. 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.

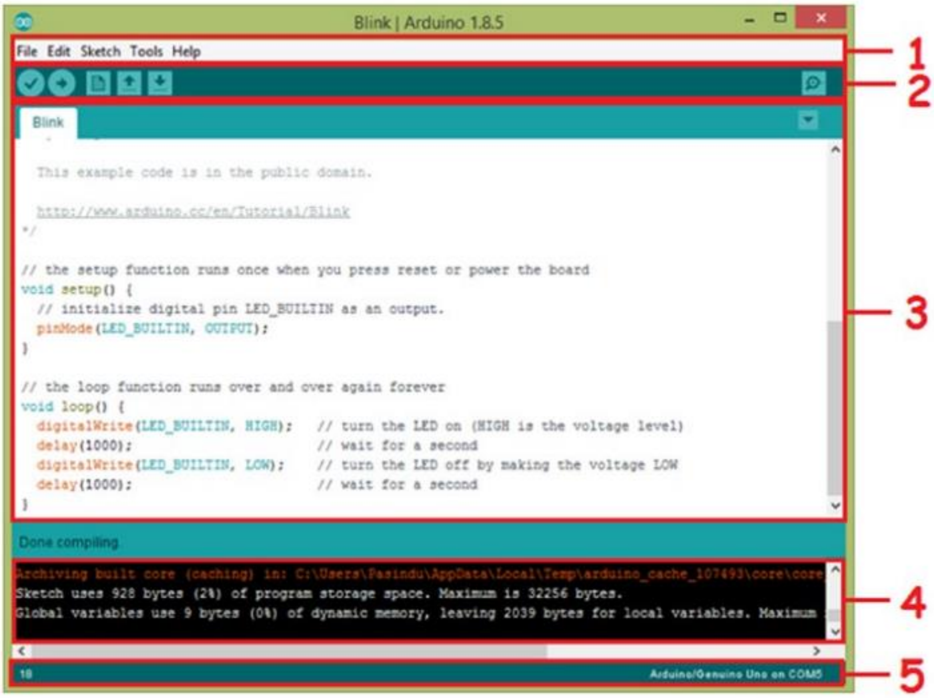


Fig 6.1: Arduino IDE

1. Menu bar: Provides useful commands

2. Toolbar: Provides the most important commands

3. Sketch editor: The area you can use to type your code

4. Text console: Shows status and error messages

5. Status bar: Displays the currently configured microcontroller board with the COM port

6.1.1 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 right and 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.

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.

6.1.2 Files:

New

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

Open

Allows to load a sketch file browsing through the computer drives and folders.

Open Recent

Provides a short list of the most recent sketches, ready to be opened.

Sketchbook

Shows the current sketches within the sketchbook folder structure; clicking the corresponding sketch in a new editor instance.

Examples

Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

Close

Closes the instance of the Arduino Software from which it is clicked.

Save

Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as..." window.

Save as...

Allows to save the current sketch with a different name.

Page Setup

It shows the Page Setup window for printing.

Print

Sends the current sketch to the printer according to the settings defined in Page Setup.

Preferences

Opens the Preferences window where some settings of the IDE may be customized. as the language of the IDE interface.

Quit

Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE

Comment/Uncomment

Puts or removes the // comment marker at the beginning of each selected line.

Increase/Decrease Indent

Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

Find

Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

Find Next

Highlights the next occurrence - if any of the string specified as the search item in the Find window, relative to the cursor position.

Find Previous

Highlights the previous occurrence - if any of the string specified as the search item in the Find window relative to the cursor position.

6.1.3 Sketch Edit:

Undo/Redo

Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

Cut

Removes the selected text from the editor and places it into the clipboard.

Copy

Duplicates the selected text in the editor and places it into the clipboard.

Copy for Forum

Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

Copy as HTML

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

Paste

Puts the contents of the clipboard at the cursor position, in the editor

Select All

Selects and highlights

Verify/ Compile

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area

Upload

Compiles and loads the binary file onto the configured board through the configured Port.

Upload Using Programmer

This will overwrite the bootloader on the board; you will need use Tools Bum Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools Burn Bootloader command must be executed.

Export Compiled Binary

Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder

Opens the current sketch folder.

Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from zip files.

Add File:

Adds a source file to the sketch (it will be copied from its current location). The new file appears. in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

6.1.4 Tools:

Auto Format

This formats your code nicely, Le indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more

Archive Sketch

Archives a copy of the current sketch in zip format. The archive is placed in the same directory as the sketch.

Fix Encoding & Reload

Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

Serial Monitor

Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

Board

Select the board that you're using. See below for descriptions of the various boards.

Port

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

Programmer

For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

Burn Bootloader

The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally comes without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

6.1.5 Help:

Here you find easy access to a number of documents that come with the Arduino Software (IDE) You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online cines and may link hack to our online website.

6.1.6 Sketchbook:

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

6.1.7 Tabs, Multiple Files and Compilation:

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.e extension). C++ files (.cpp). or header files (h).

6.1.8 Uploading:

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty. usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial IBI (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1bIPLI (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COMI or COM2 (for a serial board) or COM4, COMS, COM7, or higher (for a USB board)- to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/tty ACMx, /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets, then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will think the of board (pin 13) LED when it starts (i.e., when the board resets)

6.1.9 Libraries:

Libraries provide extra functionality for use in sketches, e.g., working with hardware of manipulating data. To use a library in a sketch, select it from the Sketch> Import Library mems. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the hoard with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #includestatements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library,

6.1.10 Third-Party Hardware:

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu). core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

6.1.11 Serial Monitor:

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial. Begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

6.1.12 Preferences:

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

6.2 Arduino Programming:

Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions.

Structure

Let us start with the Structure. Software structure consist of two main functions: divided into three parts. They are structure, values

The basic structure of Arduino programming language is simple and runs in two parts. These two parts, or functions, enclose blocks of statements.

```
Void setup ()

{

Statements;

}

Void loop ()

{

}
```

where setup () is the preparation, loop () is the execution. Both functions are required for the program to work. The setup function should follow the declaration of any variables at the beginning of the program. It is the first function to run the program, is run only once and is used to set the Statements i.e., pinMode or initialize the serial communication.

The loop function follows next and includes the code to be executed continuously by reading inputs, triggering outputs. This function is the core of Arduino programs and does a lot of work.

6.2.1 SETUP

The setup function is called once when the program starts. We use it to initialize pin modes or to begin serial communication. It must be included in the program even if there are no statements to run.

Syntax:

```
Void setup ()

{

pinMode (pin, OUTPUT); // sets the 'pin' as output

}
```

6.2.2 LOOP

The loop () function does precisely what its name suggests and loops consecutively allowing the program to change, respond, and control the Arduino board.

Syntax:

```
Void loop ()

{

digitalWrite (pin, HIGH);          // Turns pin on

delay (1000);                      // pauses for one second

}
```

6.2.3 OTHER STRUCTURES

CONTROL STRUCTURES

There are many control structures which are used in the programming language like if, if else, while, do while, for, switch case, break, continue, return, goto. They are mainly used for controlling after a certain condition is reached. Some control structures used in this project are listed below. IF If statements test whether a certain condition has been reached, such as a value being above a certain value, and executes the statements inside the brackets if it is true. If false the program skips over the statement.

Syntax:

```
 If (some variable ?? value)

{

Statements;

}
```

While

While loops will loop continuously, and infinitely, until the expression inside the parenthesis becomes false. Something must change the tested variable, or the while loop will never exit. This could be in the code such as an incremented variable, or an external condition, such as testing a sensor.

Syntax:

```
While (variable<200)

 {

Control statement;

Variable++

}
```

6.2.4 Functions:

pinMode () Function

The pinMode () function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

pinMode () Function Syntax

```
Void setup ()

{

 pinMode (pin, mode);

}
```

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT, or INPUT_PULLUP.

digitalWrite () Function

The digitalWrite () function is used to write a HIGH or a LOW value to a digital pin. If the Pin has been configured as an OUTPUT with pinMode (), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW. If the pin is configured as an INPUT, digitalWrite () will enable (HIGH) or disable (LOW) the internal pullup on the input pin. It is recommended to set the pinMode () to INPUT_PULLUP to enable the internal pull-up resistor.

If you do not set the pinMode () to OUTPUT, and connect an LED to a pin, when calling digitalWrite (HIGH), the LED may appear dim. Without explicitly setting pinMode (), digitalWrite () will have enabled the internal pull-up resistor, which acts like a large current limiting resistor.

digitalWrite () Function Syntax

```
Void loop ()
{
 digitalWrite (pin, value);
}
```

pin: the number of the pin whose mode you wish to set

value: HIGH, or LOW.

CHAPTER 7

CIRCUIT DIAGRAM

This project consists of two subsystems, one placed in the emergency vehicle and the other placed at the traffic console. These two subsystems communicate with each other through radio signalling. Here, we are using LoRa modules for wireless communication.

7.1 Emergency vehicle subsystem

When the ambulance driver is at a distance of 2-5 km from the traffic lights, he has to press a button which acts as a input to Arduino. Then, Arduino sends serial data to LoRa TransReceiver. This serial data consists of Address of the receiver LoRa and a message. In the transmitter LoRa circuit, we made a voltage divider using 4.7k and 10k resistors to drop down 5V logic level to 3.3V logic level. This is because Arduino can send the signal at 5V logic level but the Lora can only receive the signal at 3.3V logic level.

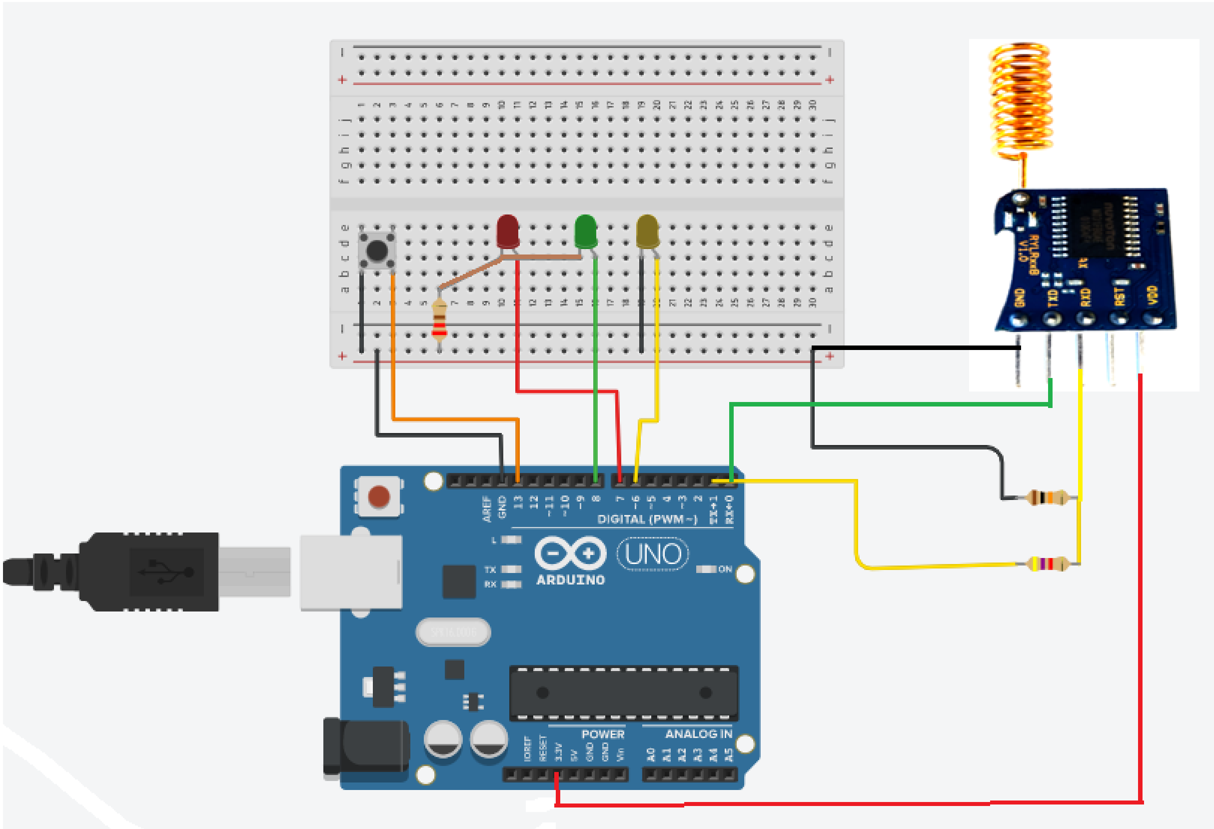


Fig 7.1: Transmitter Circuit

7.2 Traffic console subsystem

When The traffic signals are running normally, if the Lora TransReceiver receives payload which contains the address of itself, it transmits the payload to Arduino. Then, the Arduino processes this payload and halt the current operations and turns red signal to green in that direction. After sometime when ambulance passes the traffic junction, the traffic lights return to normal condition.

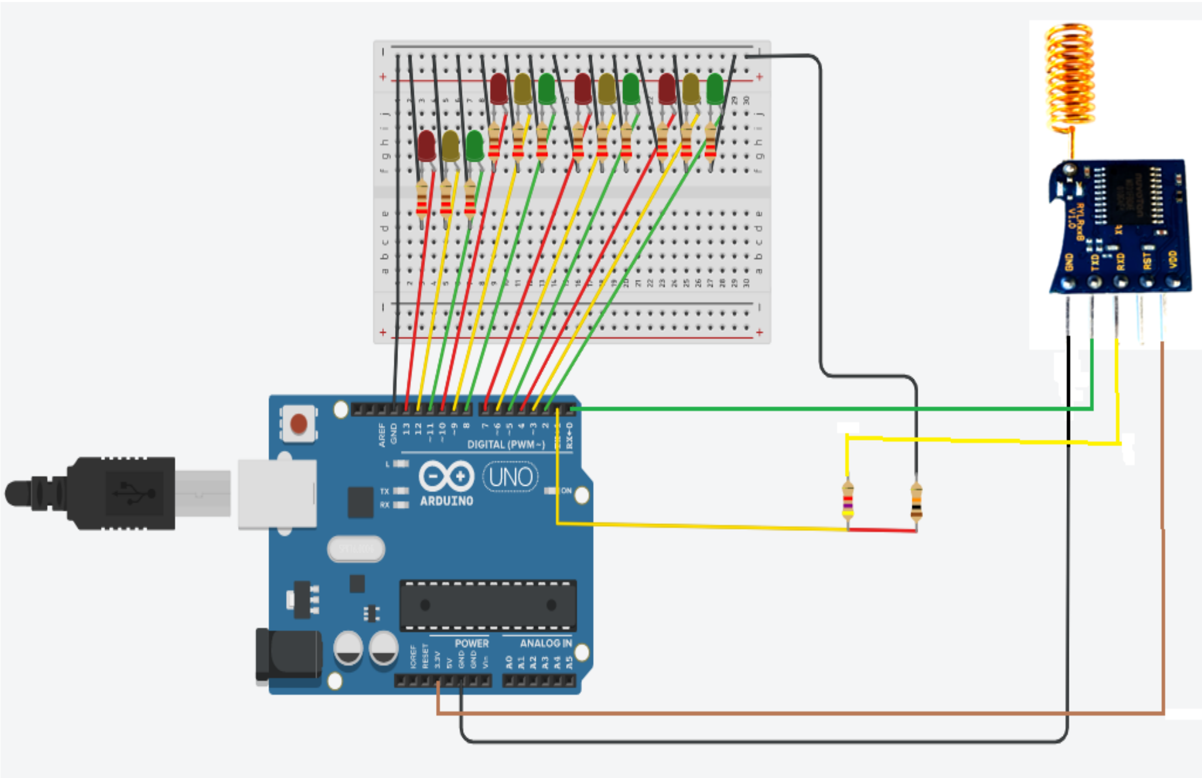


Fig 7.2: Receiver Circuit

7.3 Acknowledgement

One of the best features of Lora is, it can act as a transreceiver which helps to implement a feedback mechanism.Through this feedback we can notify the ambulance driver about the successful transmission of the payload from the transmitter to the receiver.Arduino present in the receiver circuit transfers payload to Receiver Lora to send feedback.This payload contains address of the transmitter Lora.After receiving payload,an LED glows at the transmitter circuit indicating the Acknowledgement.

CHAPTER 8

RESULTS

If we want to pre-empt traffic signals in the lane, where the Ambulance is present the driver has to press a button which acts as an input to the Arduino (Any microcontroller) present in the Ambulance.Then,After receiving the input Arduino will send the serial data to the Lora Transreceiver.Lora transmits this data to the another Lora present in the traffic junction.Then,Receiver Lora passes this information to the Arduino which currently operating the Traffic signal in that junction.After receiving the data,Arduino checks the data and according to our code it turns off the red signal and turns on the Green signal until the Ambulance Passes by.After changing the signal,Arduino at the receiver end send serial data to the Lora.Further,transmission takes place and Lora at the transmitter end receives this data.Upon receiving the data,Arduino process this data and illuminates a LED indicating successful data transmission.This acts as a Feeback to the Driver.

First we have to configure our Lora modules with the AT Commands with separate Addresses, same NetworkId and same Band.The Serial Data which is transmitted/received by the Lora must consist the Address of the another Lora such that both Lora modules communicate with each other.Since,Lora has a problem of line of sight we introduced a real time feedback mechanism which helps the Driver about successful Data Transmission.We tested our circuits at a distance of 1-2 KM in the Highway , tested it in different test situations and all of them worked successfully.

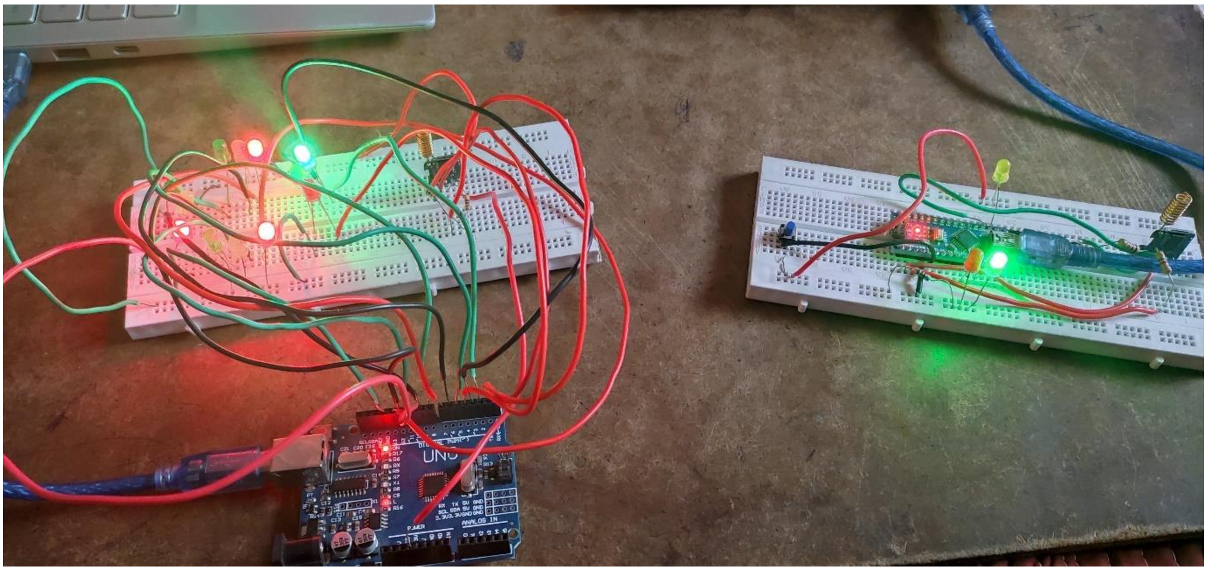


Fig 8.1: Traffic Signal Pre-emption and Real time Feedback

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

Congestion of traffic is an important problem which is to be addressed as early as possible. This is a challenging situation which is to be resolved quickly. This project can resolve this problem by overcoming the traffic congestion whenever emergency vehicles are to be passed. By initiating this project, it not only saves a lot of human effort involving traffic police but also helps the emergency vehicles or precisely ambulances to make way for them as early as possible and save lives. “The first step towards being a truly responsible society is the day we learn to allow an ambulance to pass through in traffic”. This project might be the biggest step towards making a better society for humanity.

9.2 Future Scope

- At present, we implemented LoRa to achieve maximum distance up to 2-5KM in urban areas but, from a networking perspective, it creates only a physical layer method of wireless transport such as a transceiver chip. That means it lack the network protocols to manage traffic for data collection and endpoint device management. This is where Long Range WAN or LoRaWAN comes into picture.

- Basic LoRa end devices lack security but LoRaWAN takes LoRa wireless technology and adds a networking component to it, while also incorporating node authentication and data encryption for security.

- LoRaWAN networks are ideal for IoT devices that continuously monitor the status of something and then trigger alerts back to gateways when the monitored data surpasses a specified threshold. These types of IoT devices require little bandwidth and can run on battery power for months or even years.

REFERENCES:

- LoRa Based Network for Accident Detection and providing Quicker Ambulance Services for Medical Assistance. International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Published by, www.ijert.org NCESC - 2018 Conference Proceedings.

- Automatic Accident Alert and Safety System using Embedded GSM Interface, Kajal Nandaniya, Viraj Choksi, International Journal of Computer Applications (0975 – 8887) Volume 85 – No 6, January 2014

- M. P. Karthikeyan, S. R, M. K and K. G, "Smart Ambulance for Traffic Management System," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021, pp. 747-753, doi: 10.1109/ICESC51422.2021.9532613.

- “SMART AMBULANCE WITH TRAFFIC CONTROL ABILITY” International Journal of Engineering Research & Technology (IJERT) e-ISSN: 2395-0056 Published by, www.ijert.org -2019.

- Smart Ambulance with Traffic Control © 2020 JETIR June 2020, Volume 7, Issue 6 www.jetir.org (ISSN-2349-5162).

- Girish H. R , J. Vinay Kumar , Kumara Swamy N. R , Sachin Kumar. M, Sudhakara H M, 2020, A Review: Smart Ambulance and Traffic Controlling System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 04 (April 2020).