**INTRODUCTION**

Through low-power, long-distance communication, the Vehicle-to-Vehicle (V2V) Communication System Using LoRa with Arduino aims to enhance road safety. Using Arduino Uno and Nano, the framework coordinates LoRa TX/RX modules, ultrasonic sensors, GPS, and a LCD. Through LoRa, vehicles exchange real-time data like location, speed, and distance to keep safe distances and monitor their surroundings. GPS and ultrasonic sensors provide crucial collision prevention data. By facilitating timely data sharing between vehicles, this system improves traffic management and safety, making it suitable for both urban and rural settings.

Arduino microcontrollers, which act as each vehicle's central processing units, are the building blocks of the system. Displays, ultrasonic sensors, GPS modules, LoRa transceivers, and other sensors and modules are all connected to these microcontrollers. The GPS module provides precise location data, while the ultrasonic sensor is used to measure the distance between vehicles. The LoRa handset works with correspondence among vehicles, and the showcase of vehicles.

The primary goal of V2V communication is to enhance road safety, traffic management, and fuel efficiency. By enabling vehicles to communicate, drivers can receive warnings of potential collisions, sudden braking, or upcoming hazards, allowing for better decision-making and quicker reaction times. In the long term, V2V is expected to support the development of fully autonomous vehicles by allowing seamless interaction between them.

Key benefits include reduced road accidents, improved traffic flow, and more efficient route planning. However, challenges such as ensuring data security, standardization, and the integration of V2V with existing infrastructure remain areas of active research and development.

**LITERATURE SURVEY**

**Evaluation of LoRaWAN for V2V communication**: by L.Cerqueira et al. (2019) This paper evaluates the performance of LoRaWAN (LoRa Wide Area Network) for V2V communication using a testbed of connected vehicles. The study examines the impact of vehicle density, network congestion, and transmission power on the reliability and latency of V2V communication. The results show that LoRaWAN can achieve reliable V2V communication at a low transmission power, making it a suitable technology for V2V applications.

**LoRaWAN-based V2V communication for intersection collision avoidance**: by Z. Zhang et al. (2020) This paper proposes a LoRaWAN-based V2V communication system for intersection collision avoidance. The system uses a pre-crash algorithm to detect potential collisions between vehicles approaching an intersection and sends a warning message to the affected vehicles via LoRaWAN. The study shows that the proposed system can effectively reduce the number of intersection collisions, making it a promising technology for improving road safety.

### 

### EXISTING SYSTEM:

### Several technologies have been developed and implemented in the current landscape of Vehicle-to-Vehicle (V2V) communication with varying degrees of success. Dedicated Short-Range Communications (DSRC) and cellular networks, such as 4G LTE and the upcoming 5G, are the most well-known of these. These systems allow vehicles to share real-time data like speed, location, and traffic conditions to enhance road safety and traffic management.

**Limitations of Existing Systems:**

* Range
* Infrastructure
* Cost
* Scalability
* Security

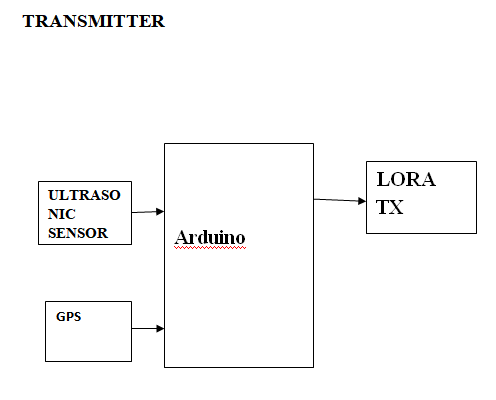
### PROPOSED SYSTEM:

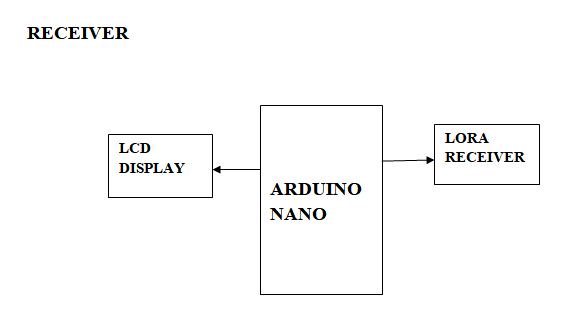
### To circumvent the limitations of the existing V2V communication systems, the proposed system makes use of LoRa (Long Range) technology. LoRa is a remote correspondence convention intended for long-range, low-power, and low-information rate applications, making it especially reasonable for V2V correspondence in both metropolitan and country settings. The proposed framework incorporates Arduino microcontrollers, ultrasonic sensors, GPS modules, LoRa handsets, ESP8266, and showcases to make a powerful and dependable V2V correspondence organization.

### Advantages of the Proposed System:

* Longer Range
* Interference Resistance
* Ease of Integration
* Flexibility and Modularity

**BLOCK DIAGRAM:**

****

****

**HARDWARE COMPONENTS REQUIRED:**

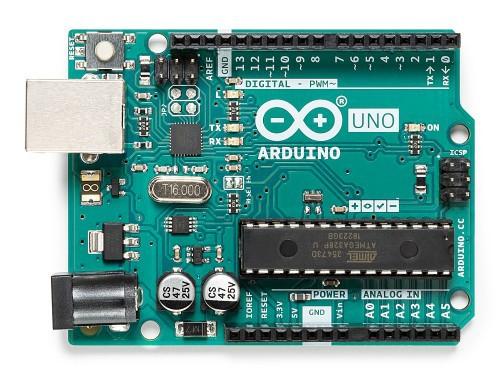
* Arduino
* Arduino nano
* Lora tx
* Lora rx
* Ultrasonic sensor
* Gps
* Lcd

**SOFTWARE REQUIRED:**

* Arduino ide

**HARDWARE DESCRIPTION:**

**Arduino**



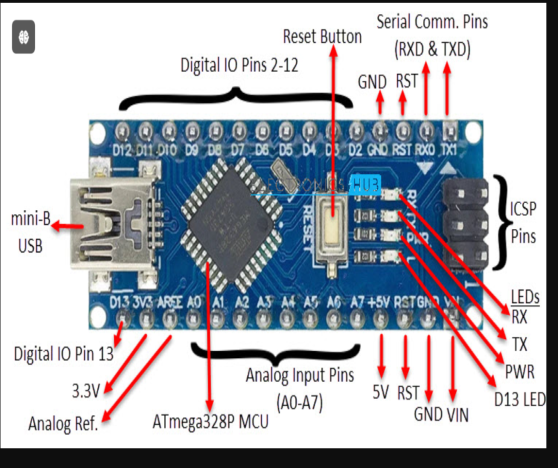
**Fig:1 ARDUINO UNO**

The Arduino Uno is an open-sourcemicrocontroller board in light of the MicrochipATmega328P microcontroller and created by Arduino.cc. Sets of digital and analog input/output (I/O) pins are provided on the board, allowing it to interface with various expansion boards (shields) and other circuits. The board is programmable using the Arduino IDE (Integrated Development Environment) via a type B USB cable and has 14 digital and 6 analog pins.

Full similarity with Safeguard sheets (Adaptation 2 is the main Arduino board that isn't viable with Safeguard sheets because of tall parts and a mistaken ICSP header position);

* AVcc LP channel to bring down the degree of commotion in the ADC;
* auto-reset empower/handicap jumper to forestall inadvertent resets;
* pin that is appropriate for the Arduino Diecimila;
* pin 13 of the installed drove, with a resistor to restrict current;
* Locally available TX and RX leds;
* power drove with fitting current limiter resistor (less 20mA of comsumption);
* jumper to handicap successive correspondence and to enable RX outside pull down resistor, to avoid "RX floating bumble". This part allows to use mechanized pin0 and pin1 as a regular pin, when consecutive correspondence isn't needed;
* "Every comparable part (diodes, semiconductors, leds, capacitors) have a similar direction on the board, simplifying mounting and lessening the probability of mistakes,"
* no wires between pads, more space between wires, greater wires, greater pads (better for cutting, restricting and entering, with no shortcircuits, securing expansions or open wires in utilization)

**Arduino nano:**

****

**Fig:2 ARDUINO NANO**

The ATmega328P microcontroller is the foundation for the small, open-source Arduino Nano board, which was created by Arduino.cc. With a smaller size factor, it functions similarly to the Arduino Uno. A collection of digital and analog input/output (I/O) pins on the Nano board allow it to be interfaced with other circuits and expansion boards. It has eight analog and fourteen digital pins, and a Mini-B USB connector that may be used to program it using the Arduino IDE. The board may be powered by an external 7–12V power source connected by the VIN pin, or it can be powered via the USB connection. It requires voltages between 7 and 12 volts to function.

**Digital Pin Usage: -** The digital pins (D0 to D13) are used in digital input/output activities.   
- If necessary, analog input pins can also be set up to operate as digital pins.   
  
**Analyzing Digital Signals: -** To determine a digital pin's current state, use {digitalRead(pinNumber)}. Either {LOW~ (0V) or {HIGH~ (5V) is returned.   
  
**Writing Digital Signals: -** To set a digital pin's state to either `LOW` (0V) or `HIGH` (5V), use `digitalWrite(pinNumber, value)}.   
  
**Signal States: -** There are just two potential states for digital pins: {LOW} (0V) and {HIGH~ 5V) is returned.

**ARDUINO:**

setup(): A function in every Arduino sketch that runs once before the loop() function. Typically used for setting pin modes. Syntax:

void setup() {

// code goes here

}

loop(): A function in every Arduino sketch that executes continuously after setup(). This is where the main logic of your sketch resides. Syntax:

void loop() {

// code goes here

}

**input:** Refers to pins configured to receive information.

**output:** Refers to pins configured to send information.

**HIGH:** Indicates the presence of an electrical signal (5V for Nano). Also represents ON or True in boolean logic.

**LOW:** Indicates the absence of an electrical signal (0V). Also represents OFF or False in boolean logic.

digitalRead(): Reads the state of a pin configured as input, returning HIGH or LOW.

digitalWrite(): Sets the state of a pin configured as output, to HIGH or LOW.

analogRead(): Reads a value between 0 and 1023 from an analog pin, corresponding to a voltage between 0V and 5V.

analogWrite(): Sets a PWM value between 0 and 255 on a pin, representing a signal with varying duty cycles.

PWM : PWM (Pulse Width Modulation) on the Arduino Nano is used to simulate an analog output signal using digital pins. This is particularly useful for controlling devices such as motors, LEDs, and other components where you need to vary the output signal's duty cycle. Here's a detailed description of PWM pins on the Arduino Nano:

PWM Pins on Arduino Nano

1. PWM Pins Overview:

- \*\*PWM (Pulse Width Modulation) is a technique where the digital output pin is switched between HIGH and LOW at a fast rate. By varying the proportion of time the pin is HIGH versus LOW, you can simulate different analog levels.

- On the Arduino Nano, PWM is used to provide an output that varies from 0% to 100% duty cycle, allowing you to control the intensity or speed of connected devices.

2. PWM Pins on Arduino Nano:

The Arduino Nano has 6 digital pins capable of generating PWM signals:

D3 (PWM)

D5(PWM)

D6 (PWM)

D9 (PWM)

D10(PWM)

D11(PWM)

3. PWM Functionality:

- These pins use 8-bit PWM, meaning the duty cycle can be set from 0 to 255 (where 0 is 0% duty cycle and 255 is 100% duty cycle).

- The PWM signal is generated using the `analogWrite(pin, value)` function, where:

- `pin` is the pin number you want to use for PWM.

- `value` is the duty cycle value (0 to 255).

```

4. Pin and Timer Mapping:

The PWM pins on the Arduino Nano are controlled by timers:

Timer0: Controls PWM on pins D5 and D6.

Timer1: Controls PWM on pins D9 and D10.

Timer2: Controls PWM on pins D3 and D11.

- Each timer has a specific frequency and resolution for PWM output, typically around 490 Hz for most PWM pins on the Nano, and 976 Hz for pins D5 and D6.

Ensure you are aware of the timer conflicts when using PWM pins, especially if you are also using functions that rely on timers (e.g., `delay()`, `millis()`, or `Servo` library).

The PWM signal's frequency is fixed, so for applications needing different frequencies, you may need to use external libraries or hardware solutions.

Understanding and utilizing PWM effectively can enhance your ability to control various devices with the Arduino Nano, allowing for precise adjustments to analog-like outputs from digital pins.

**DIGITAL SIGNALS**:

Digital signals are binary and can only be HIGH or LOW, representing the presence or absence of current. On the Arduino Nano, HIGH is close to 5V and LOW is 0V.

Digital pins # D0 to # D13: Used for digital input and output. Analog pins can also be used as digital pins.

pinMode(pinNumber, value): Configures a pin as INPUT or OUTPUT.

digitalRead(pinNumber): Reads the state of a digital pin.

digitalWrite(pinNumber, value): Sets the state of a digital pin.

analogWrite(pinNumber, value): Outputs a PWM signal on digital pins #3, #5, #6, #9, #10, and #11.

Here's a concise summary of things to remember about digital pins on the Arduino NanO.

**Things to Remember about Digital Pins on Arduino Nano:**

**Digital Pins Usag:**

- Digital Input/Output operations use the digital pins (D0 to D13).

- Analog Input pins can also be configured to function as digital pins if needed.

**Reading Digital Signals:**

- Use `digitalRead(pinNumber)` to read the state of a digital pin. It returns either `HIGH` (5V) or `LOW` (0V).

**Writing Digital Signals:**

- Use `digitalWrite(pinNumber, value)` to set the state of a digital pin to either `HIGH` (5V) or `LOW` (0V).

**Signal States:**

- Digital pins only have two possible states: `HIGH` (5V) or `LOW` (0V).

**Understanding Signal:**

- All electrical signals handled by the Arduino are either analog or digital. It's crucial to understand the difference between these signal types and how to manipulate them for various applications.

**ANALOG SIGNALS:**

Analog signals vary continuously between 0V and 5V. The Arduino Nano uses analog pins for reading these signals.

Analog pins #A0 - #A7: Used for analog input.

analogRead(pinNumber): Reads an analog value from a pin, returning a number between 0 and 1023.

analogWrite(pinNumber, value): Outputs a PWM signal on pins capable of PWM.

**Things to Remember about Analog Pins on Arduino Nano:**

**Analog Input and Output:**

Analog Input: Uses the Analog In pins (A0 to A7).

Analog Output: Uses PWM (Pulse Width Modulation) pins (D3, D5, D6, D9, D10, D11).

**Reading Analog Signals:**

- Use `analogRead(pinNumber)` to read an analog signal. The value returned ranges from 0 to 1023, reflecting the 10-bit resolution (2^10 = 1024 levels).

**Writing PWM Signals:**

- Use `analogWrite(pinNumber, value)` to send a PWM signal. The value can range from 0 to 255, representing the 8-bit resolution (2^8 = 256 levels).

**Value Ranges**:

* Analog Input Values: 0 to 1023 (10-bit resolution).
* PWM Output Values: 0 to 255 (8-bit resolution).

**OUTPUT SIGNALS:**

Arduino Nano can output digital signals and PWM signals.

Digital Output: Use digitalWrite(pinNumber, value) to send HIGH or LOW signals.

PWM Output: Use analogWrite(pinNumber, value) on PWM-capable pins (#3, #5, #6, #9, #10, #11) to output varying duty cycles.

**Things to Remember about Output on Arduino Nano:**

**Output Types:**

Digital Output: Outputs a signal that is either HIGH (1) or LOW (0).

PWM Output: Provides a Pulse Width Modulation signal that varies from 0 to 255.

Sending Output Signals:

For digital output: Use digitalWrite(pinNumber, value);, where value can be HIGH or LOW.

For PWM output: Use analogWrite(pinNumber, value);, where value ranges from 0 to 255.

Setting Pin Mode:

Set the pin mode to output using pinMode(pinNumber, OUTPUT);.

Output Characteristics:

Digital Output: Always either HIGH or LOW.

PWM Output: Varies from 0 (off) to 255 (full duty cycle), representing a range of pulse widths.

**INPUT SIGNALS:**

Analog Input: Received through analog pins #A0 - #A7. Read with analogRead(pinNumber), which returns values between 0 and 1023.

Digital Input: Received through digital pins #0 - #13. Configure with pinMode(pinNumber, INPUT) and read with digitalRead(pinNumber).

**HARDWARE COMPONENTS:**

USB Connector: Mini-B USB for programming and power.

Power Pins:

VIN: Input voltage (7-12V).

5V: Output voltage (5V from regulator).

GND: Ground.

RESET Button: Resets the microcontroller.

**SOFTWARE TIPS:**

- Use the Arduino IDE for writing and uploading code.

- For specialized features or libraries, consult the Arduino documentation and community forums.

**HC-SR04 Sensor Features:**

* Operating voltage: +5V
* Theoretical Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

**Applications:**

* Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
* Used to measure the distance within a wide range of 2cm to 400cm
* Can be used to map the objects surrounding the sensor by rotating it
* Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

**GPS:**

**What is GPS**

The Global Positioning System (GPS) is a satellite-based navigation system made up of at least 24 satellites. GPS works in any weather conditions, anywhere in the world, 24 hours a day, with no subscription fees or setup charges.



**How GPS works**

GPS satellites circle the Earth twice a day in a precise orbit. Each satellite transmits a unique signal and orbital parameters that allow GPS devices to decode and compute the precise location of the satellite. GPS receivers use this information and trilateration to calculate a user's exact location. Essentially, the GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a user's position and display it.

To calculate your 2-D position (latitude and longitude) and track movement, a GPS receiver must be locked on to the signal of at least 3 satellites. With 4 or more satellites in view, the receiver can determine your 3-D position (latitude, longitude and altitude). Generally, a GPS receiver will track 8 or more satellites, but that depends on the time of day and where you are on the earth.

Once your position has been determined, the GPS unit can calculate other information, such as:

* Speed
* Bearing
* Track
* Trip dist
* Distance to destination

**What's the signal?**

GPS satellites transmit at least 2 low-power radio signals. The signals travel by line of sight, meaning they will pass through clouds, glass and plastic but will not go through most solid objects, such as buildings and mountains. However, modern receivers are more sensitive and can usually track through houses.

A GPS signal contains 3 different types of information:

* Pseudorandom code is an I.D. code that identifies which satellite is transmitting information. You can see which satellites you are getting signals from on your device's satellite page.
* Ephemeris data is needed to determine a satellite's position and gives important information about the health of a satellite, current date and time.
* Almanac data tells the GPS receiver where each GPS satellite should be at any time throughout the day and shows the orbital information for that satellite and every other satellite in the system.

**Overview of NEO-6M GPS Module:**

**1.NEO-6M GPS Chip**

The heart of the module is a NEO-6M GPS chip from u-blox. It can track up to 22 satellites on 50 channels and achieves the industry’s highest level of sensitivity i.e. -161 dB tracking, while consuming only 45mA supply current. The u-blox 6 positioning engine also boasts a Time-To-First-Fix (TTFF) of under 1 second. One of the best features the chip provides is Power Save Mode(PSM). It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just 11mA making it suitable for power sensitive applications like GPS wristwatch. The necessary data pins of NEO-6M GPS chip are broken out to a "0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART.

**Note**: - The module supports baud rate from 4800bps to 230400bps with default baud of 9600.



**Position Fix LED Indicator:**

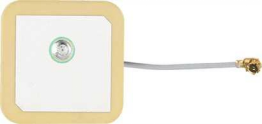
There is an LED on the NEO-6M GPS Module which indicates the status of Position Fix. It’ll blink at various rates depending on what state it’s in

* No Blinking ==> means It is searching for satellites
* Blink every 1s – means Position Fix is found

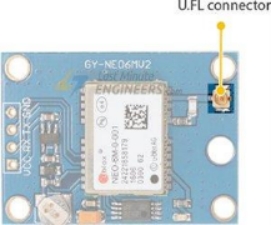


**Antenna:**

An antenna is required to use the module for any kind of communication. So, the module comes with a patch antenna having -161 dBm sensitivity.



You can snap-fit this antenna to small U.FL connector located on the module.



**ultrasonic sensor:**

****

**Fig:3 ULTRASONIC SENSOR**

Using ultrasonic waves, ultrasonic sensors calculate distance. The ultrasonic wave is sent out by the sensor head, and it is reflected back by the target. Ultrasonic sensors track the duration between an emission and a reception to determine the target's distance.

**HC-SR04 Sensor Features:**

* Operating voltage: +5V
* Theoretical Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

**HC-SR04 Ultrasonic Sensor – Working:**

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

**Distance = Speed × Time**

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below



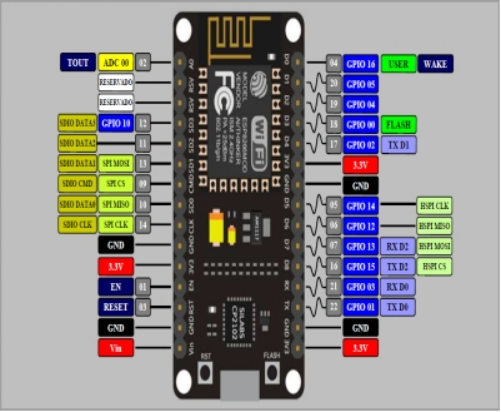
**Specifications:**

|  |  |
| --- | --- |
| Receiver Type | 50 channels, GPS L1(1575.42Mhz) |
| Horizontal Position Accuracy | 2.5m |
| Navigation Update Rate | 1HZ (5Hz maximum) |
| Capture Time | Cool start: 27sHot start: 1s |
| Navigation Sensitivity | -161dBm |
| Communication Protocol | NMEA, UBX Binary, RTCM |
| Serial Baud Rate | 4800-230400 (default 9600) |
| Operating Temperature | -40°C ~ 85°C |
| Operating Voltage | 2.7V ~ 3.6V |
| Operating Current | 45mA |
| TXD/RXD Impedance | 510Ω |

**Nodemcu:**

**Introduction**

General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behaviour can be controlled at the run time.



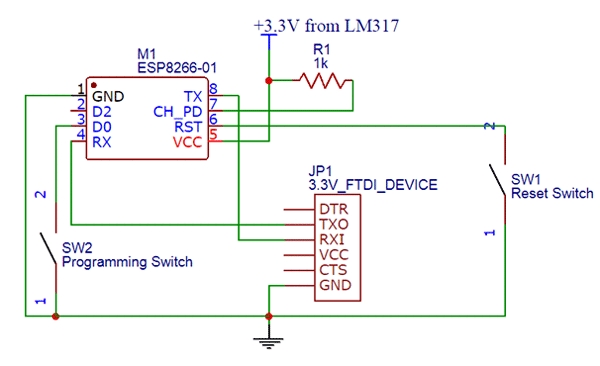
**Where to use ESP8266-01**

The ESP8266 is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making Internet of Things as easy as possible. It can also fetch data from internet using API’s hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. However this version of the module has only 2 GPIO pins (you can hack it to use upto 4) so you have to use it along with another microcontroller like [Arduino](https://components101.com/microcontrollers/arduino-uno), else you can look onto the more standalone ESP-12 or ESP-32 versions. So if you are looking for a module to get started with IOT or to provide internet connectivity to your project then this module is the right choice for you.

**How to use the ESP8266 Module**

There are so many methods and IDEs available to with ESP modules, but the most commonly used on is the Arduino IDE. So let us discuss only about that further below.

The ESP8266 module works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with your circuits. The best way to program an ESP-01 is by using the FTDI board that supports 3.3V programming. If you don’t have one it is recommended to buy one or for time being you can also use an Arduino board. One commonly problem that every one faces with ESP-01 is the powering up problem. The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider. So it is important to make a small voltage regulator for 3.31v that could supply a minimum of 500mA. One recommended regulator is the [LM317](https://components101.com/lm317-pinout-equivalent-datasheet) which could handle the job easily. A simplified circuit diagram for using the ESP8266-01 module is given below



### The switch SW2 (Programming Switch) should be held pressed to hold the GPIO-0 pin to ground. This way we can enter into the programming mode and upload the code. Once the code is released the switch can be released.

### GPIO PINS:

### NodeMCU Development kit provides access to these GPIOs of ESP8266. The only thing to take care is that NodeMCU Dev kit pins are numbered differently than internal GPIO notations of ESP8266 as shown in below figure and table. For example, the D0 pin on the NodeMCU Dev kit is mapped to the internal GPIO pin 16 of ESP8266.The Internet of Things (IoT) has been a trending field in the world of technology. It as changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Expressive Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized Wi-Fi enabled microcontroller – ESP8266, it can monitor and control things from anywhere in the world

### C:\Users\dell\AppData\Local\Temp\ksohtml12732\wps11.jpg

|  |  |
| --- | --- |
| Pin Names on NodeMCU Development Kit | ESP8266 Internal GPIO Pin number |
| D0 | GPIO16 |
| D1 | GPIO5 |
| D2 | GPIO4 |
| D3 | GPIO0 |
| D4 | GPIO2 |
| D5 | GPIO14 |
| D6 | GPIO12 |
| D7 | GPIO13 |
| D8 | GPIO15 |
| D9/RX | GPIO3 |
| D10/TX | GPIO1 |
| D11/SD2 | GPIO9 |
| D12/SD3 | GPIO10 |

### Installing the ESP8266 Core on Windows OS

### Let’s proceed with installing ESP8266 Arduino core.

### The first thing is having latest Arduino IDE (Arduino 1.6.4 or higher) installed on your PC. If don’t have it, we recommend upgrading now.

### Latest Arduino IDE

### To begin, we’ll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window:

### <http://arduino.esp8266.com/stable/package_esp8266com_index.json>

### 

### 

### Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.

### 

### 

### The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, a small INSTALLED text will appear next to the entry. You can now close the Board Manager.

### Arduino Example: Blink

### To make sure ESP8266 Arduino core and the NodeMCU are properly set up, we’ll upload the simplest sketch of all – The Blink!

### We will use the on-board LED for this test. As mentioned earlier in this tutorial, D0 pin of the board is connected to on-board Blue LED & is user programmable. Perfect!

### Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select NodeMCU 0.9 (ESP-12 Module) option under your Arduino IDE > Tools > Board menu.

### 

### 

### Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed : 115200

### 

### WARNING

### More attention needs to be given to selecting board, choosing COM port and selecting Upload speed. You may get espcomm\_upload\_mem error while uploading new sketches, if failed to do so.

### Once you are done, try the example sketch below.

### void setup()

### {

### pinMode(D0, OUTPUT);

### }

### void loop()

### {

### digitalWrite(D0, HIGH);

### delay(500);

### digitalWrite(D0, LOW);

### delay(500);

### }

### Once the code is uploaded, LED will start blinking. You may need to tap the RST button to get your ESP8266 to begin running the sketch.

### 

### LCD (Liquid Crystal Display):

### 

### LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16×2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments](https://www.engineersgarage.com/content/seven-segment-display) and other multi segment [LED](https://www.engineersgarage.com/content/led)s. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even [custom characters](https://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](https://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so on.

### A 16×2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5×7 pixel matrix. This LCD has two registers, namely, Command and Data.

### The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a [LCD](https://www.engineersgarage.com/insight/how-lcd-works).

### 

### 

### 

### 

### 

### How LCDs are Constructed?

### 

### The basic structure of LCD should be controlled by changing the applied current.

### We must use a polarized light.

### Liquid crystal should able be to control both of the operation to transmit or can also able to change the polarized light.

### 

### As mentioned above that we need to take two polarized glass pieces filter in the making of the liquid crystal. The glass which does not have a polarized film on the surface of it must be rubbed with a special polymer which will create microscopic grooves on the surface of the polarized glass filter. The grooves must be in the same direction of the polarized film. Now we have to add a coating of pneumatic liquid phase crystal on one of the polarized filter of the polarized glass. The microscopic channel cause the first layer molecule to align with filter orientation. When the right angle appears at the first layer piece, we should add a second piece of glass with the polarized film. The first filter will be naturally polarized as the light strikes it at the starting stage.

### Thus the light travels through each layer and guided on the next with the help of molecule. The molecule tends to change its plane of vibration of the light in order to match their angle. When the light reaches to the far end of the liquid crystal substance, it vibrates at the same angle as that of the final layer of the molecule vibrates. The light is allowed to enter into the device only if the second layer of the polarized glass matches with the final layer of the molecule.

### 

### Working:

### The principle behind the LCD’s is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist. This causes the angle of light which is passing through the molecule of the polarized glass and also cause a change in the angle of the top polarizing filter. As a result a little light is allowed to pass the polarized glass through a particular area of the LCD. Thus that particular area will become dark compared to other. The LCD works on the principle of blocking light. While constructing the LCD’s, a reflected mirror is arranged at the back. An electrode plane is made of indium-tin oxide which is kept on top and a polarized glass with a polarizing film is also added on the bottom of the device. The complete region of the LCD has to be enclosed by a common electrode and above it should be the liquid crystal matter.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### Next comes to the second piece of glass with an electrode in the form of the rectangle on the bottom and, on top, another polarizing film. It must be considered that both the pieces are kept at right angles. When there is no current, the light passes through the front of the LCD it will be reflected by the mirror and bounced back. As the electrode is connected to a battery the current from it will cause the liquid crystals between the common-plane electrode and the electrode shaped like a rectangle to untwist. Thus the light is blocked from passing through. That particular rectangular area appears blank.

### 

### 16X2 LCD pinout diagram

### 

### 

### 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No | Pin No. | Pin Name | Pin Type | Pin Description | Pin Connection |
| 1 | Pin 1 | Ground | Source Pin | This is a ground pin of LCD | Connected to the ground of the MCU/ Power source |
| 2 | Pin 2 | VCC | Source Pin | This is the supply voltage pin of LCD | Connected to the supply pin of Power source |
| 3 | Pin 3 | V0/VEE | Control Pin | Adjusts the contrast of the LCD. | Connected to a variable POT that can source 0-5V |
| 4 | Pin 4 | Register Select | Control Pin | Toggles between Command/Data Register | Connected to a MCU pin and gets either 0 or 1.0 -> Command Mode1-> Data Mode |
| 5 | Pin 5 | Read/Write | Control Pin | Toggles the LCD between Read/Write Operation | Connected to a MCU pin and gets either 0 or 1.0 -> Write Operation1-> Read Operation |
| 6 | Pin 6 | Enable | Control Pin | Must be held high to perform Read/Write Operation | Connected to MCU and always held high. |
| 7 | Pin 7-14 | Data Bits (0-7) | Data/Command Pin | Pins used to send Command or data to the LCD. | In 4-Wire ModeOnly 4 pins (0-3) is connected to MCUIn 8-Wire ModeAll 8 pins(0-7) are connected to MCU |
| 8 | Pin 15 | LED Positive | LED Pin | Normal LED like operation to illuminate the LCD | Connected to +5V |
| 9 | Pin 16 | LED Negative | LED Pin | Normal LED like operation to illuminate the LCD connected with GND. | Connected to ground |

### 

### 

### Features of 16×2 LCD module:

### Operating Voltage is 4.7V to 5.3V

### Current consumption is 1mA without backlight

### Alphanumeric LCD display module, meaning can display alphabets and numbers

### Consists of two rows and each row can print 16 characters.

### Each character is build by a 5×8 pixel box

### Can work on both 8-bit and 4-bit mode

### It can also display any custom generated characters

### Available in Green and Blue Backlight

### Advantages :

### LCD’s consumes less amount of power compared to CRT and LED

### LCD’s are consist of some microwatts for display in comparison to some mill watts for LED’s

### LCDs are of low cost

### Provides excellent contrast

### LCD’s are thinner and lighter when compared to cathode ray tube and LED

### Disadvantages :

### Require additional light sources

### Range of temperature is limited for operation

### Low reliability

### Speed is very low

### LCD’s need an AC drive

### Applications :

### Liquid crystal technology has major applications in the field of science and engineering as well on electronic devices.

### Liquid crystal thermometer

### Optical imaging

### The liquid crystal display technique is also applicable in visualization of the radio frequency waves in the waveguide

### Used in the medical applications

**LORA (LONG RANGE):**

****

**Fig:5 LORA MODULE**

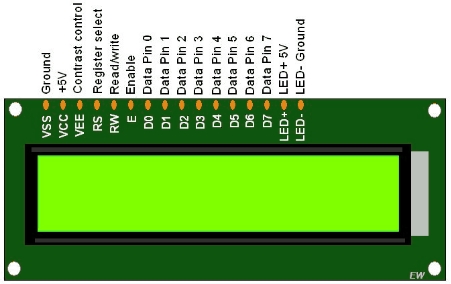
LoRa (Long Range) is a low-power, long-range wireless platform used for IoT networks, introduced by Semtech. It uses spread spectrum modulation based on chirp spread spectrum (CSS) technology. Common frequencies include 433MHz, 915MHz, and 868MHz, enabling bi-directional communication over distances of 15-20km with minimal power consumption. LoRa supports public, private, and hybrid networks, offering greater range than cellular networks and facilitating low-cost, battery-operated IoT applications.

In LoRa technology, messages from devices are received by gateways and forwarded to a central network for processing. The LoRa Alliance, a non-profit organization, standardizes and advances this LP-WAN technology, driven by IoT demands.

**Features:**

* LoRaTM spread spectrum communication
* +20dBm - 10mW. Stable RF output power when input voltage changed
* Half-duplex SPI communication
* Programmable bit rate can reach to 300KBPS

**Lcd:**



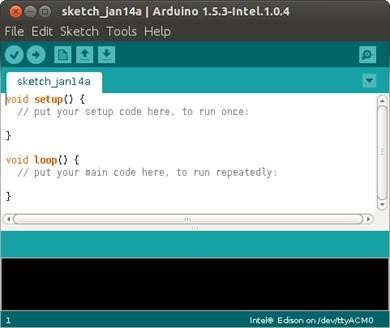
**Fig:6 LCD**

An electronic display module known as an LCD (Liquid Crystal Display) screen is utilized in numerous contexts. A very fundamental module, a 162 LCD display is utilized frequently in numerous circuits and devices. These modules are liked north of seven portions and other multi section LEDs. These are the reasons: LCDs are prudent; effectively programmable; have no constraint of showing extraordinary and even custom characters (in contrast to in seven portions), activitys, etc.A 16\*2 LCD has two such lines and can display 16 characters per line. In this LCD each character is shown in 5×7 pixel lattice. The Command and Data registers on this LCD are its two registers.

### Features of 16×2 LCD module

* Operating Voltage is 4.7V to 5.3V
* Current consumption is 1mA without backlight
* Alphanumeric LCD display module, meaning can display alphabets and numbers

# **SOFTWARE DESCRIPTION:**

ArduinoSoftware(IDE)

Arduino is an open source, PC equipment and programming organization, task, and client local area that plans and makes microcontroller units for building computerized gadgets and intuitive items that can detect and control objects in the actual world. The venture's items are dispersed as open-source equipment and programming, which are authorized under the GNU Lesser Overall population Permit (LGPL) or the GNU Overall population Permit (GPL), allowing the production of Arduino sheets and programming conveyance by anybody. Pre-assembled Arduino boards can be purchased commercially or as DIY kits. The designs of Arduino boards make use of a variety of controllers and microprocessors.

The sheets are furnished with sets of advanced and simple information/yield (I/O) sticks that might be communicated to different development sheets (safeguards) and different circuits.

**CONCLUSION :**

The proposed Vehicle-to-Vehicle (V2V) communication system utilizing LoRa technology offers a promising alternative to existing communication methods. By leveraging LoRa's long-range capabilities, low power consumption, and cost-effective nature, this system addresses the key limitations of current V2V technologies such as DSRC, cellular networks, and Wi-Fi. The integration of ultrasonic sensors, GPS modules, Arduino microcontrollers, and ESP8266 provides a comprehensive solution for real-time data exchange, collision avoidance, and enhanced road safety.

This system is particularly well-suited for deployment in both urban and rural environments, where traditional infrastructure may be lacking or insufficient. Its scalability and flexibility make it adaptable to a wide range of vehicular communication needs, from basic safety alerts to more complex traffic management systems. The use of open-source hardware and unlicensed frequency bands further reduces costs and encourages widespread adoption. LoRa-based V2V communication system represents a significant advancement in intelligent transportation systems. It not only enhances the safety and efficiency of road networks but also provides a viable pathway for the future of connected and autonomous vehicles. As the system continues to be refined and optimized, it holds the potential to revolutionize how vehicles communicate, ultimately leading to safer and smarter roads for all users.

**REFERENCES:**

**[1]** K. F. Haque,Venkata P. Yanambaka, Kumar Yelamarthi and A. Abdelgawad, “A LoRa Based Reliable and Low Power Vehicle to Everything (V2X) Communication” 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS) 2021.

**[2]** H. R. Abdulshaheed, Z. T. Yaseen, A. M. Salman, andI. Al Barazanchi, “A survey on the use of wimax and wi-fi on vehicular ad-hoc networks (vanets),” in IOP Conference Series: Materials Science and Engineering, vol. 870, no. 1. IOP Publishing, 2020, p.012122.

**[3]** C. Dong, X. Chen, H. Dong, K. Yang, J. Guo, and Y. Bai, “Research on intelligent vehicle infrastructure cooperative system based on zigbee,” in 2019 5th International Conference on Transportation Information and Safety (ICTIS). IEEE, 2019, pp. 1337–1343.

**[4]** Y. Cheung, M. Qiu, and M. Liu, “Autonomous vehicle communication in v2x network with lora protocol,” in International Conference on Smart Computing and Communication. Springer, 2019, pp. 398–410.

**[5]** J. Wang, Y. Shao, Y. Ge, and R. Yu, “A survey of vehicle to everything(v2x) testing,” Sensors, vol. 19, no. 2, p. 334, 2019.

**[6]** X. Xu, Z. Zeng, Y. Wang, and J. Ash, “A framework of a v2x communication system for enhancing vehicle and pedestrian safety at unsignalized intersections,” in International Conference on Management Science and Engineering Management. Springer, 2018, pp. 51–63.

**[7]** O. Alsaif, I. Saleh, and D. Ali, “Evaluating the performance of nodes mobility for zigbee wireless sensor network,” in 2019 International Conference on Computing and Information Science and Technology and Their Applications (ICCISTA). IEEE, 2019, pp. 1–5.

**[8]** A. Lavric and V. Popa, “Performance evaluation oflorawan communication scalability in large-scale wireless sensor networks,” Wireless Communications and Mobile Computing, vol. 2018, 2018.