AGV

Logo, icon

Description automatically generated

# Table of Contents

[Table of Contents 1](#_Toc85765602)

[Chapter 1: Abstract 3](#_Toc85765603)

[Chapter 2: Introduction 4](#_Toc85765604)

[2.1 Introduction 4](#_Toc85765605)

[2.2 What guides an AGV? 4](#_Toc85765606)

[2.2.1 Natural Navigation AGV 5](#_Toc85765607)

[2.3 References 6](#_Toc85765608)

[Chapter 3: Robot Navigation 7](#_Toc85765609)

[3.1 Navigation Steps: 7](#_Toc85765610)

[3.1.1 Perception (Sensing): 7](#_Toc85765611)

[3.1.2 Localization: 7](#_Toc85765612)

[3.1.3 Cognition: 7](#_Toc85765613)

[3.1.4 Motion Control: 7](#_Toc85765614)

[3.2 SLAM 7](#_Toc85765615)

[3.3 Localization Techniques 7](#_Toc85765616)

[Chapter 4: Motor Driver 8](#_Toc85765617)

[4.1 Introduction 8](#_Toc85765618)

[4.1.1 Controlling DC Motor 8](#_Toc85765619)

[4.2 L298 Module 10](#_Toc85765620)

[4.2.1 L298N Module Pinout 11](#_Toc85765621)

[4.2.2 Power Supply 12](#_Toc85765622)

[4.2.3 Voltage Drop of L298N 13](#_Toc85765623)

[4.2.4 Output Pins 14](#_Toc85765624)

[4.2.5 Control Pins 14](#_Toc85765625)

[4.3 Wiring Motor Driver 16](#_Toc85765626)

[4.4 Arduino Code 17](#_Toc85765627)

[4.5 References 20](#_Toc85765628)

[Chapter 5: Communication 21](#_Toc85765629)

[5.1 Types Of Systems 21](#_Toc85765630)

[5.1.1 Wired 21](#_Toc85765631)

[5.1.2 Wireless 21](#_Toc85765632)

[5.2 Wireless Communication 22](#_Toc85765633)

[5.2.1 Why Wireless Communication? 23](#_Toc85765634)

[5.2.2 Components 23](#_Toc85765635)

[5.2.3 Types of Wireless Communication 25](#_Toc85765636)

[5.2.4 Classifications of Wireless Communication 25](#_Toc85765637)

[5.3 Chip Used 27](#_Toc85765638)

[5.3.1 RF Channel Frequency 27](#_Toc85765639)

[5.3.2 nRF24L01+ Multiceiver Network 28](#_Toc85765640)

[5.3.3 Important Features of nRF24L01 IC 29](#_Toc85765641)

[5.4 nRF24L01 Transceiver Module 30](#_Toc85765642)

[5.4.1 Components on nRF24L01 30](#_Toc85765643)

[5.4.2 Pins Configuration 31](#_Toc85765644)

[5.5 Various Modules of nRF24L01+ 31](#_Toc85765645)

[5.5.1 NRF24L01 32](#_Toc85765646)

[5.5.2 NRF24L01+PA/LNA 32](#_Toc85765647)

[5.6 Arduino Code 33](#_Toc85765648)

[5.6.1 Unidirectional Communication 33](#_Toc85765649)

[5.6.2 Key Parameters Of RF24 Library 35](#_Toc85765650)

[5.6.3 Bidirectional Communication 37](#_Toc85765651)

[5.7 Troubleshoot 39](#_Toc85765652)

[5.7.1 Power Supply Noise 39](#_Toc85765653)

[5.7.2 External Power Supply 40](#_Toc85765654)

[5.8 References 40](#_Toc85765655)

[Chapter 6: References 41](#_Toc85765656)

# Table Of Figures

[Figure ‎2.1: Common types of AGVs 4](#_Toc85765673)

[Figure ‎5.1: Radio Waves 22](#_Toc85765674)

[Figure ‎5.2: Commonly used Wireless Communication Systems 23](#_Toc85765675)

[Figure ‎5.3: Pins Configuration of nRF24L01 Transceiver Module 31](#_Toc85765676)

# Abstract

# Introduction

## Introduction

Automated Guided Vehicles (AGVs) are also known by other names such as LGV (Laser-Guided Vehicle), Mobile Robots, SGV (Self-Guided Vehicle), Guided Carts, Autonomous Vehicles, and Driverless Vehicles.

Regardless the type of automated guided vehicle (forklift, tow tractor, , etc), the AGV requires an Automated Guidance System that drives the AGV and informs the AGV Management System about the AGV positioning.

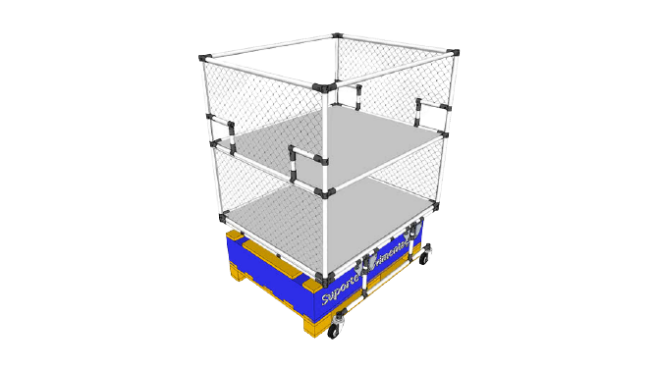


Figure .: Common types of AGVs

Vehicles and navigation are improving day after day. Mobile robots are reaching incredible performance levels that are redefining many industries, there are many AGV applications such ecommerce, warehouses, hospitals, etc, where AGVs are bringing new excellence levels.

This outstanding development is possible thanks to accurate, reliable, and cost-effective AGV navigation sensors. Depending on the application, AGVs and AMRs navigate thanks to 2D or 3D LiDAR, magnetic sensors, ultrasonic sensors, cameras, etc.

What are the main AGV Navigation Systems? AGV Navigation Technologies are Laser Guided Navigation, Magnetic Navigation, Natural Feature or Free navigation (including SLAM Navigation with LiDAR Sensors), Magnetic spot navigation, inductive wire navigation, optical navigation, vision navigation.

Choosing the right vehicle guidance technology is essential because it will influence the AGV Robot System performance. AGV performance is complex and depends on many key elements.

## What guides an AGV?

The most common AGV navigation systems are:

1. Laser-guided navigation (LGV)
2. Magnetic navigation
3. AGV with Natural navigation (SLAM or LiDAR NAVIGATION)
4. Magnetic spot navigation
5. Wire navigation
6. Optical Navigation
7. Vision Navigation

So, there are several Navigation Methods for Automated Guided Vehicles and Autonomous Mobile Robots.

### Natural Navigation AGV

Natural Navigation AGV - Free Navigation

A picture containing calendar

Description automatically generated

There are several technologies included in what’s called “Natural Navigation”.

AGV Manufacturers could buy from a Navigation Technology supplier or could develop its own Navigation Technology. The most important one is the SLAM Navigation, or Simultaneous Localization and Mapping (SLAM). It simply means that an AGV with SLAM Navigation is able to map its environment and localize where it is thanks to the information received from the surrounding environment.

The AGVs are able to map the environment with different AGV sensors such vision cameras, lidar sensors or even with the same lasers used for safety purposes.... all of this info is combined with internal inertial measurement unit (IMU) to define and recalculate the real AGV positioning. All of these calculations are made by a highly complex algorithm called SLAM.

#### AGVs with SLAM

You drive the AGV manually (for example with a joystick) or automatically along the AGV route. While running, the AGV will map the surrounding environment and creates a reference map that is used to navigate next time it will pass in the same place. You can also load an AutoCAD (or similar) map of your premises into the AGV Management System. Both sets of data, the AutoCAD map and the real mapped environment are matched to define the initial coordinates (0,0).

Based on the combined data sets, the AGV will automatically drive through the practiced route, and it will check if what it’s “seeing” is the same that was loaded on its “brain” enabling it to define its position.

All the data that the AGV acquires are combined with other data coming from odometrics, encoder, in order to improve accuracy.

#### Why natural navigation?

SLAM Navigation will substitute for other kinds of navigation such magnetic navigation, optical navigation ,etc. It’s a good solution for AGCs and Tow tractors. Many of the main AGV manufacturers are developing and including this technology on their AGVs.

#### Disadvantages of using SLAM

The main concern about Natural (SLAM, LIDAR, etc) technology is its reliability in variable environments such as production lines where there’s continuously moving of people, items, boxes, pallets, etc. In these conditions the AGV might not be able to find where it is.

For this reason, SLAM with LiDAR is a great solution for AGVs where you have well defined profiles and environments with fixed structures such as walls and columns. Natural Navigation can be used in warehouses and hospitals… in general, in any environment with a low level of “confusion”.

## References

[1]

# Robot Navigation

## Navigation Steps:

### Perception (Sensing):

A model of the “real” world is captured in memory using sensors.

### Localization:

### Cognition:

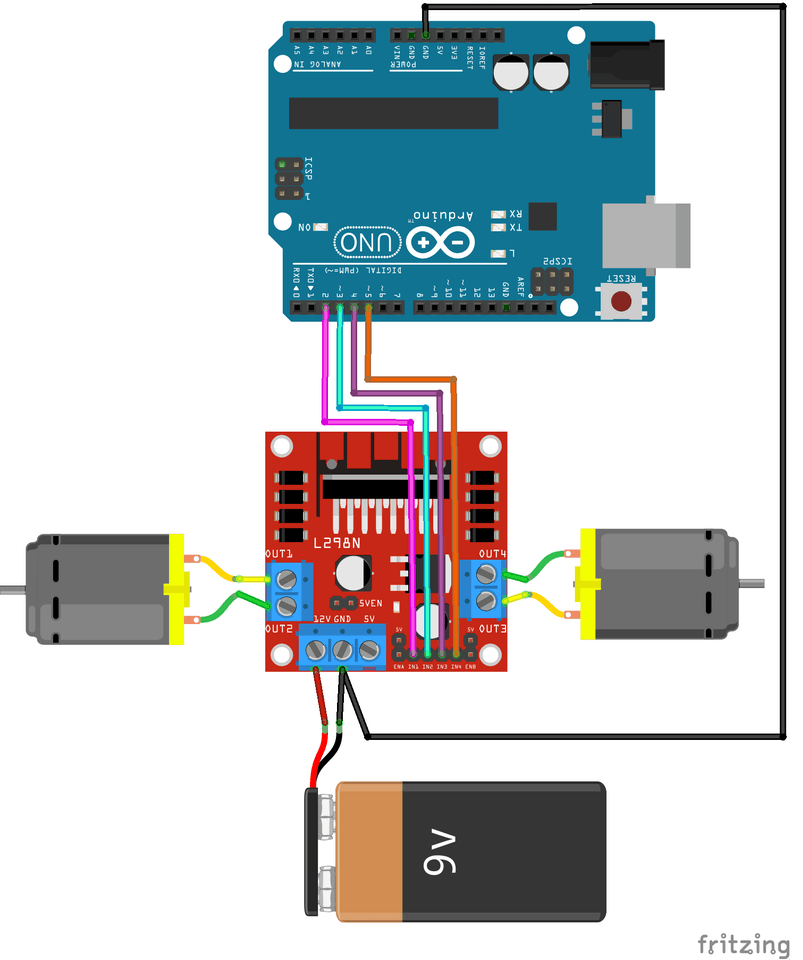
### Motion Control:

## SLAM

SLAM stands for Simultaneous Localization and Mapping

## Localization Techniques

# Motor Driver



## Introduction

No robot can move without motors. So, motors is an essential part and knowing how to control them is a must. There are many motors types, but our aim is to control DC motors.

### Controlling DC Motor

In order to have a complete control over DC motor, we have to control its speed and rotation direction which can be achieved by combining two techniques:

1. PWM – For controlling speed.
2. H-Bridge – For controlling rotation direction.

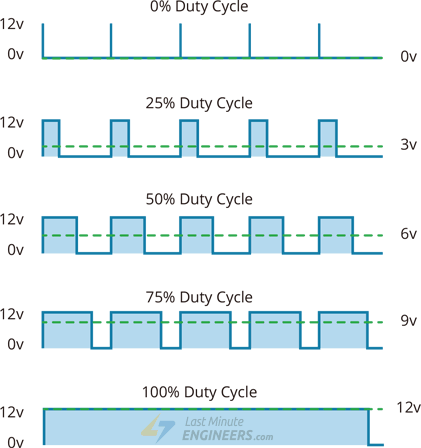
DC motors have current and voltage requirements that are beyond the capabilities of microcontrollers. It is necessary to use some external electronics to drive and control the motor, and you’ll probably need a separate power supply as well.

One of the easiest and inexpensive way to control DC motors is to interface L298N Motor Driver which is a dual H-Bridge motor driver which allows speed (PWM) and direction (H-Bridge) control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

#### PWM – For controlling speed

The speed of a DC motor can be controlled by varying its input voltage. A common technique for doing this is to use PWM (Pulse Width Modulation)

PWM is a technique where average value of the input voltage is adjusted by sending a series of ON-OFF pulses as the following diagram.



The average voltage is proportional to the width of the pulses known as Duty Cycle.

Duty Cycle = (ON-period / Off period) \* 100

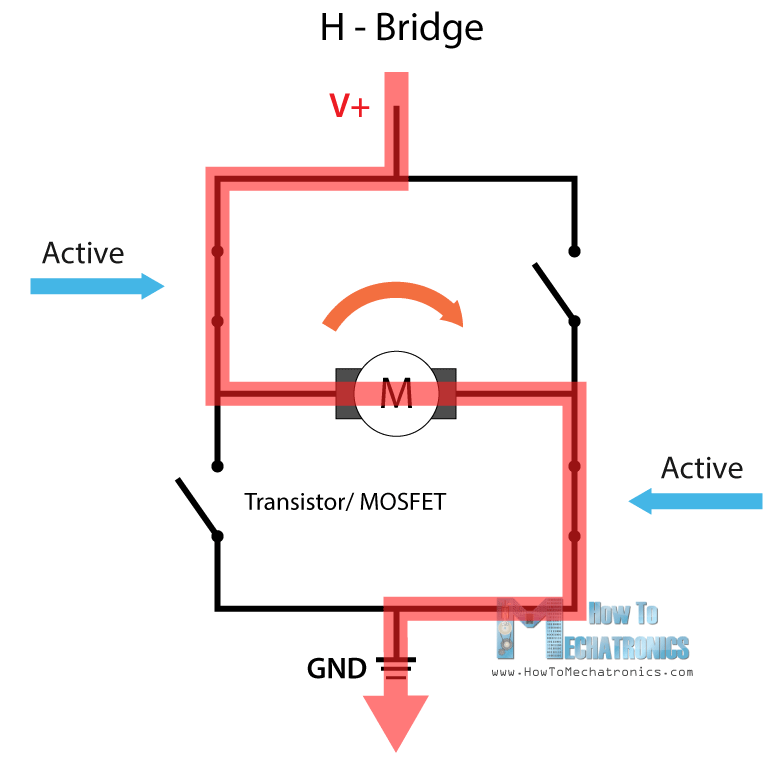
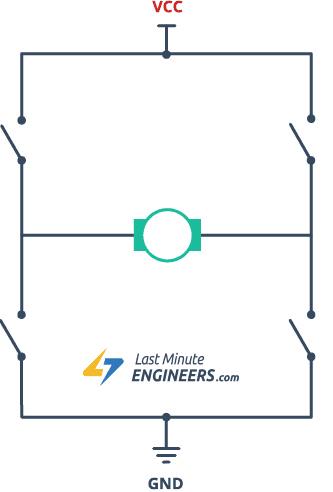
Duty cycle is proportional to the average voltage being applied to the dc motor (Speed).

Average voltage = (Duty Cycle / 100) \* Supply Voltage

#### H-Bridge – For controlling rotation direction

The DC motor’s spinning direction can be controlled by changing polarity of its input voltage. A common technique for doing this is to use an H-Bridge which contains four switches with the motor at the center forming an H-like arrangement.

Closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This causes change in spinning direction of the motor.

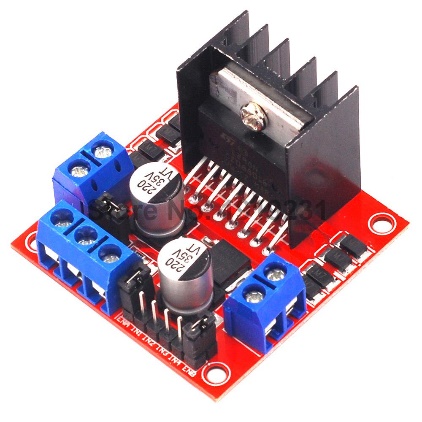


## L298 Module

L298N module is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

There are two versions of the driver modules:

1. L298: Can drive two motors with 2A for each one.
2. L298N: Can drive two motors with 3A for each one.

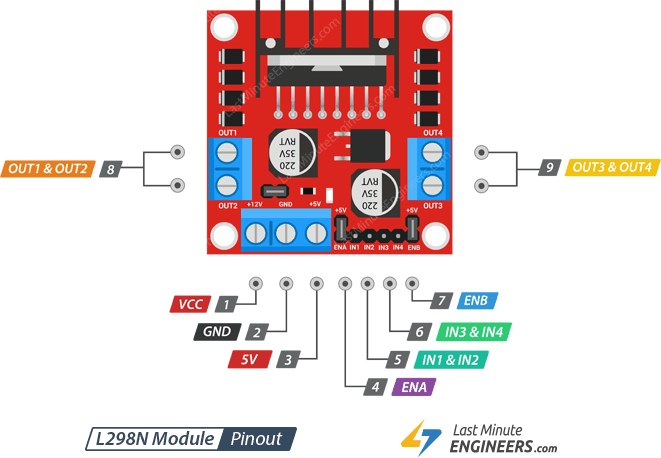


L298

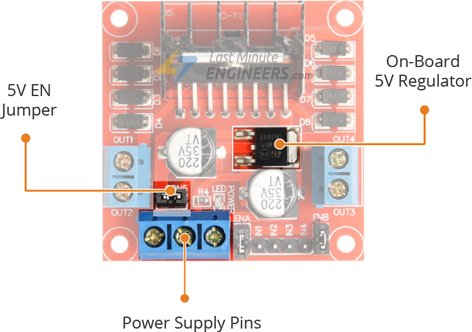
L298N

We will deal with L298 module.

### L298N Module Pinout



### Power Supply



The L298 motor driver module is powered through 3-pins:

1. Motors Power supply (VCC): supplies power for the motors. from 5 to 35V. Remember, if the 5V-EN jumper is in place, supply 2 extra volts than motor’s actual voltage requirement, to get maximum speed out of motor.
2. GND: a common ground pin.
3. 5V logic power supply (Vss) (In/Out): Module has an on-board 78M05 5V regulator.

There is a jumper called 5V-EN jumper. It is used to enable/disable regulator:

1. If the 5V-EN jumper is removed, connect 5V pin to a 5V supply to supply power for the switching logic circuitry inside L298N IC.
2. If the 5V-EN jumper is in place, this pin acts as an output and can be used to power up microcontroller, and the switching logic circuitry inside L298N IC, and other circuits with a maximum of **0.5A**.

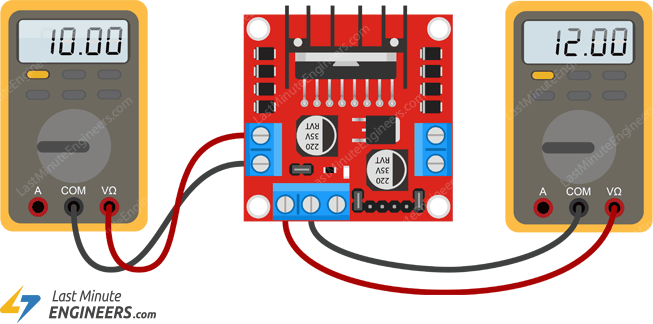
NOTE:

1. If the motor VCC > 12V, remove the 5V-EN jumper to avoid damaging the regulator, and separate 5V should be given through 5V terminal to power the internal circuitry.

Diagram, schematic

Description automatically generated

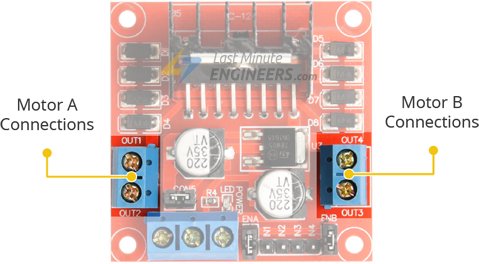
### Voltage Drop of L298N



The voltage drop of the L298 module is about 2V. This is due to the internal voltage drop in the switching transistors in the H-Bridge circuit and the voltage regulator.

So, 12V power supply terminal will be around 10V. So, to get the maximum speed of a 12V DC motor supply it with around 14V.

### Output Pins



There are two output channels for two motors having voltages between 5 to 35V:

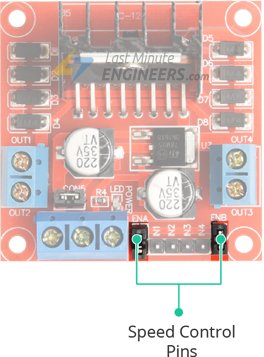
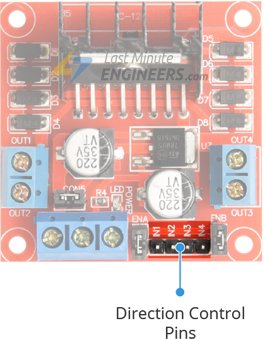
OUT1 & OUT2 for motor A.

OUT3 & OUT4 for motor B.

Each channel on the module can deliver up to 2A to the DC motor. However, the amount of current supplied to the motor depends on system’s power supply.

### Control Pins

For each channel, there are two types of control pins to control speed and spinning direction.



#### Direction Control Pins

Using the direction control pins, the motor can spin forward or backward. These pins actually control the switches of the H-Bridge circuit.

The module has two direction control pins for each channel:

IN1 & IN2 pins are used to control spinning direction of Motor A.

IN3 & IN4 pins are used to control spinning direction of Motor B.

The spinning direction of a motor is illustrated below.

Table

Description automatically generated

The next diagram shows working of case 2 in the above table for motor A.

Diagram, schematic

Description automatically generated

#### Speed Control Pins

Pins ENA and ENB are used to control speed of the motors A and B respectively:

1. Pulling these pins HIGH will make the motors spin at maximum speed. By default, there are jumpers connecting these pins to HIGH.
2. Pulling them LOW will make them stop.
3. With Pulse Width Modulation (PWM), we can control the speed of the motors by removing the jumper and connecting these pins to a PWM device.

There module come with jumpers on these pins to pull them HIGH. So, by default motors spins at the maximum speed.

So, to control the speed of motors programmatically, remove the jumpers and connect them to PWM-enabled pins on the microcontroller.

**NOTE**: Depending on the applied voltage and the motor itself, at lower speeds the motor is not able to start moving and it produces a buzzing sound. So, experiment some values and set the minimum PWM above the minimum by some offset (In 6VDC motors, use 85 minimum PWM).

## Wiring Motor Driver

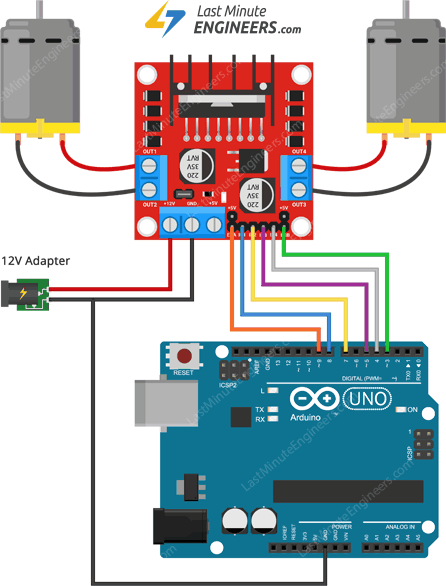
We are using DC Gearbox Motors (also known as ‘TT’ motors). They are rated for 3 to 12V.

So, to connect the motor driver to the microcontroller:

1. Connect 12V power supply to the motors at VCC and GND pins. Considering internal voltage drop of L298N IC, the motors will receive 10V and will spin at slightly lower RPM.
2. Supply 5 Volts for the L298N’s logic circuitry. On-board 5V regulator can be used and derive 5 volts from the motor power supply (VCC). so, place the 5V-EN jumper.
3. The input and enable pins(ENA, IN1, IN2, IN3, IN4 and ENB) of the L298N module are connected to six Arduino digital output pins(9, 8, 7, 5, 4 and 3).

NOTE: The Arduino output pins 9 and 3 are both PWM-enabled.

1. Connect one motor to terminal A (OUT1 & OUT2) and the other motor to terminal B (OUT3 & OUT4).



## Arduino Code

// Motor A connections

int enA = 9;

int in1 = 8;

int in2 = 7;

// Motor B connections

int enB = 3;

int in3 = 5;

int in4 = 4;

void setup() {

// Set all the motor control pins to outputs

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

// Turn off motors - Initial state

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

void loop() {

directionControl();

delay(1000);

speedControl();

delay(1000);

}

// This function lets you control spinning direction of motors

void directionControl() {

// Set motors to maximum speed

// For PWM maximum possible values are 0 to 255

analogWrite(enA, 255);

analogWrite(enB, 255);

// Turn on motor A & B

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

delay(2000);

// Now change motor directions

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(2000);

// Turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

// This function lets you control speed of the motors

void speedControl() {

// Turn on motors

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

// Accelerate from zero to maximum speed

for (int i = 0; i < 256; i++) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Decelerate from maximum speed to zero

for (int i = 255; i >= 0; --i) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Now turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

Code Explanation

It doesn’t require any libraries to get it working.

1. Declaring Arduino pins to which L298N’s control pins are connected.

// Motor A connections

int enA = 9;

int in1 = 8;

int in2 = 7;

// Motor B connections

int enB = 3;

int in3 = 5;

int in4 = 4;

1. motor control pins are declared as digital OUTPUT and pulled LOW.

void setup() {

// Set all the motor control pins to outputs

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

// Turn off motors - Initial state

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

1. In loop section, we call two user defined functions at an interval of a second.

void loop() {

directionControl();

delay(1000);

speedControl();

delay(1000);

}

* 1. directionControl(): spins both motors forward at maximum speed for two seconds, then reverses the motor’s spinning direction and spins for another two seconds, finally turns the motors off.

void directionControl() {

// Set motors to maximum speed

// For PWM maximum possible values are 0 to 255

analogWrite(enA, 255);

analogWrite(enB, 255);

// Turn on motor A & B

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

delay(2000);

// Now change motor directions

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(2000);

// Turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

* 1. speedControl():accelerates both the motors from zero to maximum speed by producing PWM signals using analogWrite() function, then it decelerates them back to zero, finally it turns the motors off.

void speedControl() {

// Turn on motors

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

// Accelerate from zero to maximum speed

for (int i = 0; i < 256; i++) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Decelerate from maximum speed to zero

for (int i = 255; i >= 0; --i) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Now turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

## References

[2], [3]

# Communication

## Types Of Systems

Communication systems can be:

### Wired

The medium is a physical path like Co-axial Cables, Twisted Pair Cables and Optical Fiber Links etc. which guides the signal to propagate from one point to other.

Such type of medium is called Guided Medium.

### Wireless

The medium is space which does not require any physical medium but propagates the signal through space. Since, space only allows for signal transmission without any guidance, it is called Unguided Medium. the transmission and reception of signals is accomplished with Antennas.

Antenna: and electrical device transforms the electrical signals to radio signals in the form of Electromagnetic (EM) Waves and vice versa. These Electromagnetic Waves propagates through space. Hence, both transmitter and receiver consist of an antenna.

Electromagnetic Waves carry the electromagnetic energy of electromagnetic field through space. They include Gamma Rays (γ – Rays), X – Rays, Ultraviolet Rays, Visible Light, Infrared Rays, Microwave Rays and Radio Waves.

Electromagnetic Waves used in wireless communication to carry the signals are usually Radio Waves.

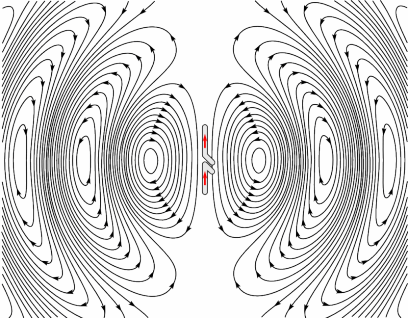
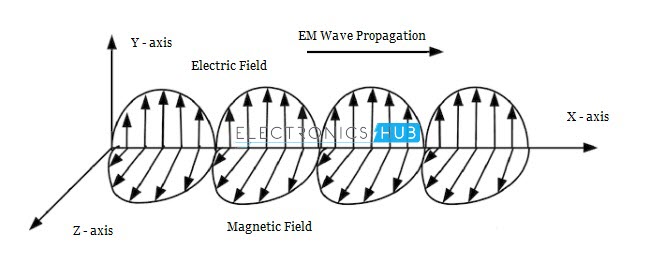


Figure .: Radio Waves

Electromagnetic Wave consists of (electric + magnetic fields) in the form of time varying sinusoidal waves. Both are oscillating perpendicular to each other and the direction of propagation is perpendicular to both these fields.



Mathematically, an Electromagnetic Wave can be described using Maxwell’s equations. Pictorial representation shows the Electric Field in the Y – axis, magnetic field in the Z – axis and the Electromagnetic Wave propagates in X – axis.

## Wireless Communication

Wireless Communication is a method of transmitting information from one point to another, without using any connection wires, cables, or any physical medium.

Generally, information is transmitted from a transmitter to a receiver placed over a limited distance.

Wireless Communication lets transmitter and receiver can be placed anywhere between:

1. few meters (like a T.V. Remote Control).
2. few thousand kilometers (Satellite Communication).

The commonly used Wireless Communication Systems:

1. Mobile Phones.
2. GPS Receivers.
3. Remote Controls.
4. Bluetooth Audio.
5. Wi-Fi.
6. etc…

A picture containing text, electronics, circuit

Description automatically generated

Figure .: Commonly used Wireless Communication Systems

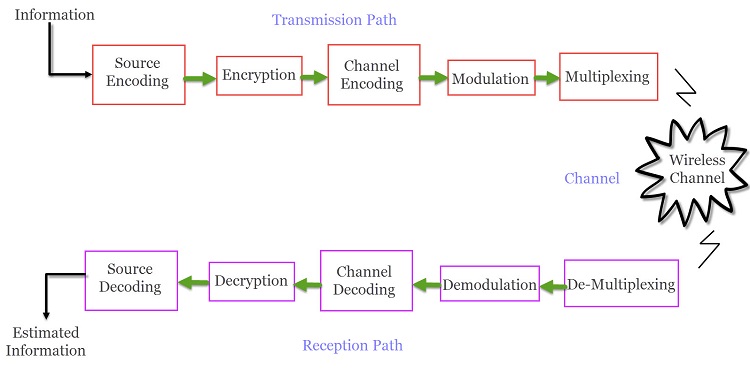
Wired communication can do most of the tasks that a wireless communication can.

### Why Wireless Communication?

1. The primary and important benefit is mobility.
2. flexibility and ease of use; can be made anywhere and anytime with a considerably high throughput performance.
3. Infrastructure: the infrastructure setup and installation of wired communication systems is an expensive and time-consuming job. The infrastructure for wireless communication can be installed easily and low cost.
4. In emergency situations and remote locations, where the setup of wired communication is difficult.

### Components

Elements of a Wireless Communication System.



A typical Wireless Communication System can be divided into three elements:

#### Transmitter

Transmission path of a Wireless Communication System consists of:

1. Source Encoder: converts source signal to a suitable form to apply signal processing. Redundant information is removed to maximize resources utilization.
2. Encryption: The signal is then encrypted using an Encryption Standard so that the signal and the information is secured and doesn’t allow any unauthorized access.
3. Modulation: reduce the impairments like noise, interference, etc. During this process, a small amount of redundancy is introduced to the signal so that it becomes robust against noise. The signal is modulated using a suitable Modulation Technique (like PSK, FSK and QPSK etc.), to be easily transmitted using antenna.
4. Multiplexing: The modulated signal is multiplexed with other signals using different Techniques like Time Division Multiplexing (TDM) or Frequency Division Multiplexing (FDM) to share the valuable bandwidth.

#### Channel

The medium of transmission of the signal i.e., open space. A wireless channel is unpredictable and also highly variable and random in nature. It may be subject to interference, distortion, noise, scattering etc. and the result is that the received signal may be filled with errors.

#### Receiver

Collects the signal from the channel and reproduce it as the source signal.

The reception path of a Wireless Communication System consists of:

1. Demultiplexing: separates the signal from the channel from other signals.
2. Demodulation: Demodulated the individual signals using appropriate Techniques to recover the original message signal.
3. Channel Decoding: removes the redundant bits from the message.
4. Decryption: removes encryption security and turns it into simple sequence of bits.
5. Source Decoding: gets back the original transmitted message or signal.

From the components of the reception path it is clear that the task of the receiver is just the inverse to that of transmitter.

### Types of Wireless Communication

People need Mobile Phones for many things like talking, internet, multimedia etc. All these services must be made available to the user on the go i.e. while the user is mobile. With the help of these wireless communication services, we can transfer voice, data, videos, images etc.

Wireless Communication Systems also provide different services like video conferencing, cellular telephone, paging, TV, Radio etc. Due to the need for variety of communication services, different types of Wireless Communication Systems are developed. Some of the important Wireless Communication Systems available today are:

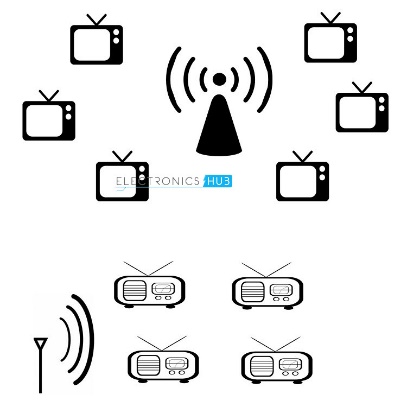
1. Television and Radio Broadcasting.
2. Satellite Communication.
3. Radar
4. Mobile Telephone System (Cellular Communication)
5. Global Positioning System (GPS)
6. Infrared Communication
7. WLAN (Wi-Fi)
8. Bluetooth
9. Zigbee
10. Paging
11. Cordless Phones
12. Radio Frequency Identification (RFID)

### Classifications of Wireless Communication

1. **Simplex**: one way communication. An example is Radio broadcast system.
2. **Half Duplex**: two-way communication but not simultaneous one. i.e., walkie – talkie (civilian band radio).
3. **Full Duplex**: two-way communication and simultaneous. i.e., mobile phones.

Examples:

1. Television and Radio Broadcasting:
   1. Simplex Communication System where information is transmitted in one direction and all users receiving the same data.



1. Satellite Communication:
   1. worldwide coverage independent to population density.
   2. offer telecommunication (Satellite Phones), positioning and navigation (GPS), broadcasting, internet, etc. Other wireless services like mobile, television broadcasting and other radio systems are dependent of Satellite Communication.
2. Radar:
3. Mobile Telephone System (Cellular Communication): mobile phones are not limited to just making calls but are integrated with numerous other features like Bluetooth, Wi-Fi, GPS, and FM Radio.
   1. The latest generation of Mobile Communication Technology is 5G. Apart from increased data transfer rates, 5G Networks are also aimed at Internet of Things (IoT) related applications and future automobiles.
4. Global Positioning System (GPS): a subcategory of satellite communication. It provides different wireless services like navigation, positioning, location, speed etc. with the help of dedicated GPS receivers and satellites.
5. Infrared Communication: uses the infrared waves of the Electromagnetic (EM) spectrum. It is used in remote controls of Televisions, cars, audio equipment etc.
6. Bluetooth: low range wireless communication system provides data, voice and audio transmission with a transmission range of 10 meters. Almost all mobile phones, tablets and laptops are equipped with Bluetooth devices. They can be connected to wireless Bluetooth receivers, audio equipment, cameras etc.
7. Paging:
   1. An obsolete technology. It was a major success before the widespread use of mobile phones. It provides information in the form of messages in a simplex system i.e., the user can only receive the messages.
8. Wireless Local Area Network (WLAN):
   1. An internet related wireless service. WLAN enables laptops and mobile phones connect to an access point (like a Wi-Fi Router) and access internet.
   2. Wi-Fi is one of the widely used wireless network, usually for internet access (but sometimes for data transfer within the Local Area Network). It is very difficult to imagine the modern World without Wi-Fi.

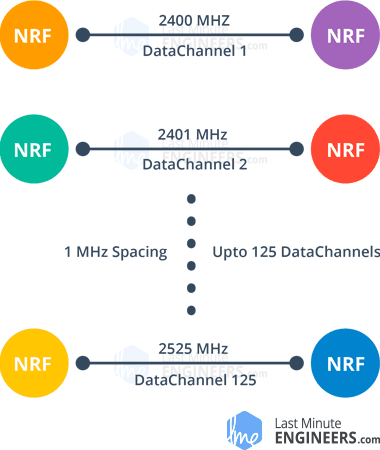
## Chip Used

A single chip RF Transceiver IC developed by Nordic Semiconductor. It operates in the license-free 2.4GHz ISM band (ISM – Industrial, Scientific and Medical) with support for data rates of 250kbps, 1Mbps and 2Mbps.

### RF Channel Frequency

nRF24L01+ transceiver module transmits and receives data on a certain frequency called Channel. Also in order for two or more transceiver modules to communicate with each other, they need to be on the same channel. This channel could be any frequency in the 2.4 GHz ISM band (2.400 to 2.525 GHz).

Each channel occupies a bandwidth of less than 1MHz. This gives us 125 possible channels with 1MHz spacing. So, the module can use 125 different channels which give a possibility to have a network of 125 independently working modems in one place.



250kbps and 1Mbps data rates channels bandwidth = 1MHz.

2Mbps data rate channel bandwidth should be 2MHz or more to ensure non-overlapping.

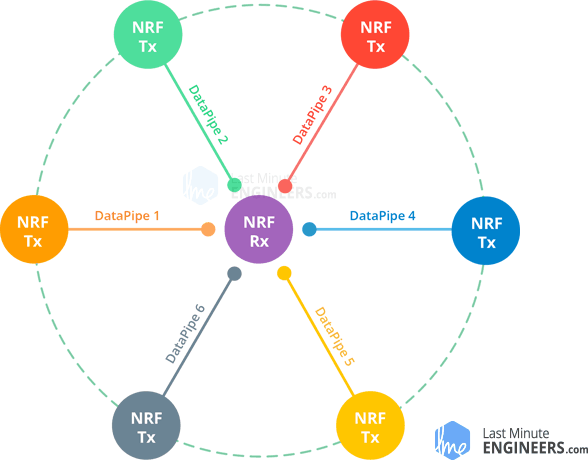
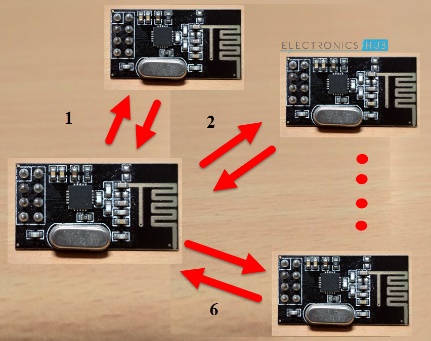
RF channel frequency of your selected channel is set according to the following formula:

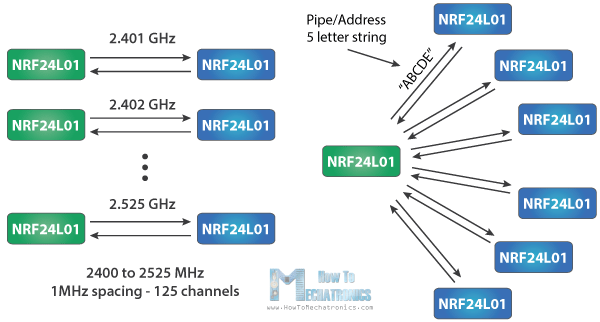
Freq(Selected) = 2400 + CH(Selected)

* 1. Example .: if you select channel 108 for data transmission, RF channel frequency of channel = 2508MHz 🡺 (2400 + 108)

### nRF24L01+ Multiceiver Network

nRF24L01+ IC has MultiCeiver(Multiple Transmitters Single Receiver) feature enables each RF Channel to be logically divided into 6 unique addressed parallel data channels called Data Pipes. So that each module can communicate with 6 other modules in the same RF Channel.





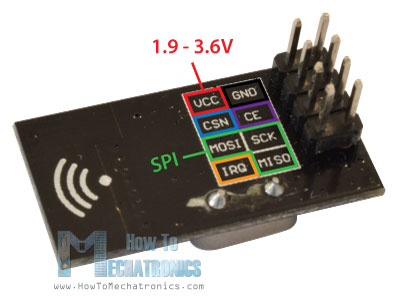
Imagine the primary receiver acting as a hub receiver:

1. collects information from 6 different transmitter nodes simultaneously.
2. can stop listening any time and acts as a transmitter. But this can only be done one pipe/node(send to one node) at a time.

To design a Wireless Communication system using nRF24L01 Module, all you need is a Microcontroller, which is interfaced through Serial Peripheral Interface (SPI).

### Important Features of nRF24L01 IC

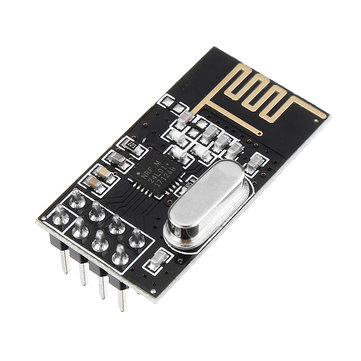
1. Ultra-low power operation (26μA Standby-I mode, 900nA power down mode).
2. 12mA during transmission.
3. SPI Interface with Microcontroller.
4. Integrated RF Transmitter, Receiver and Synthesizer.
5. Operating voltage is 1.9V – 3.6V (3.3v).
6. Other 6 data pins tolerate 5V logic, so we can easily connect it to an Arduino without using any logic level converters.
7. nRF24L01 Transceiver Module.



## nRF24L01 Transceiver Module

### Components on nRF24L01

nRF24L01+ Transceiver IC, Antenna Trace (on the PCB), Pins for communication and power and a few passive components.



### Pins Configuration

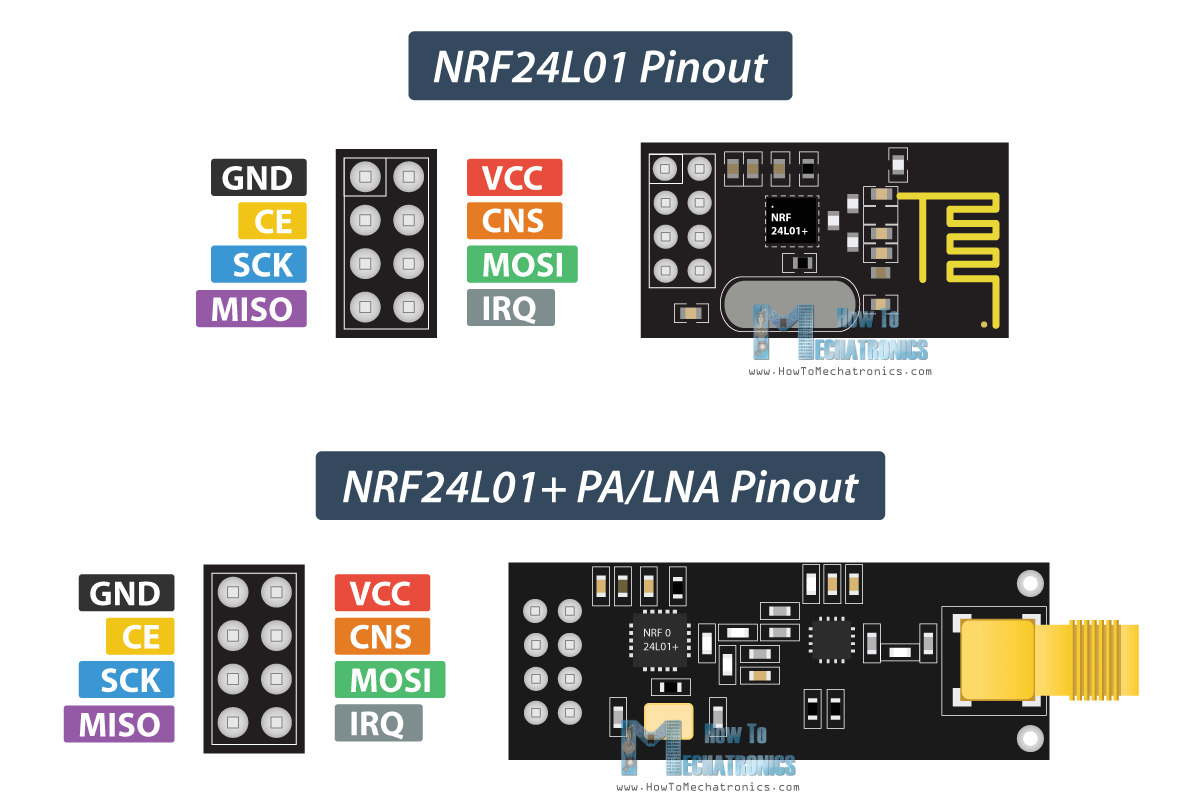
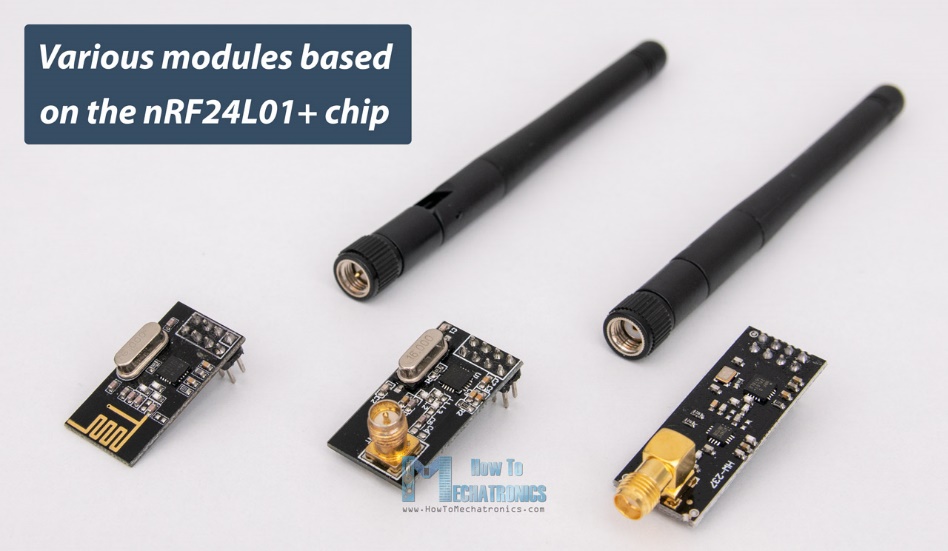


Figure .: Pins Configuration of nRF24L01 Transceiver Module

The pins are:

1. VCC : 3.3V must be given.
2. GND : Ground.
3. SCK : SPI Clock Pin.
4. MOSI : SPI Slave Data Input Pin.
5. MISO : SPI Slave Data Output Pin.
6. IRQ : Active LOW Interrupt Pin.
7. CE : Chip Enable Pin.
8. CSN : SPI Chip Select Pin.

## Various Modules of nRF24L01+



### NRF24L01



It uses on-board antenna. This allows for a more compact version of the breakout. However, the smaller antenna also means a lower transmission range; over a distance of 100 meters. Of course, that is outdoors in an open space. Your range indoors, especially through walls, will be slightly weakened.

### NRF24L01+PA/LNA

It comes with a SMA connector and a duck-antenna but that’s not the real difference.

The real difference is that it comes with RFX2401C chip which integrates the PA, LNA, and transmit-receive switching circuitry.

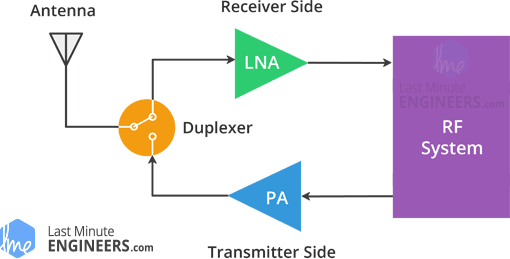
This range extender chip along with a duck-antenna helps the module achieve a significantly larger transmission range about 1000m.

#### What is PA/LNA?

PA (Power Amplifier): merely boosts the power of the signal being transmitted from the nRF24L01+ chip.

LNA (Low-Noise Amplifier): take the extremely weak and uncertain signal from the antenna (usually on the order of microvolts or under -100 dBm) and amplify it to a more useful level (usually about 0.5 to 1V)

LNA of the receive path and PA of the transmit path connect to the antenna via a duplexer, which separates the two signals and prevents the relatively powerful PA output from overloading the sensitive LNA input.



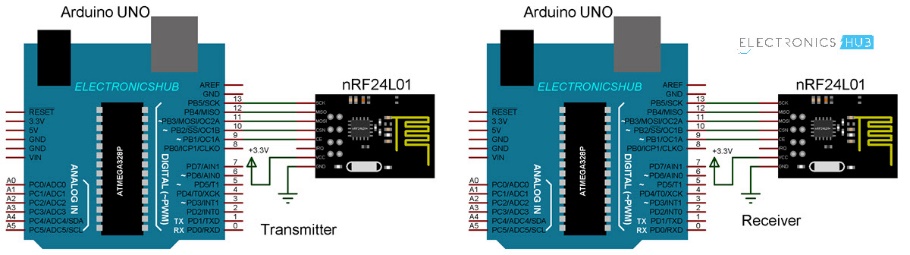
## Arduino Code

nRF24L01 Transceiver Module communicates over SPI Interface. So, to interface with Arduino, you need to use the SPI Pins of the Arduino board.

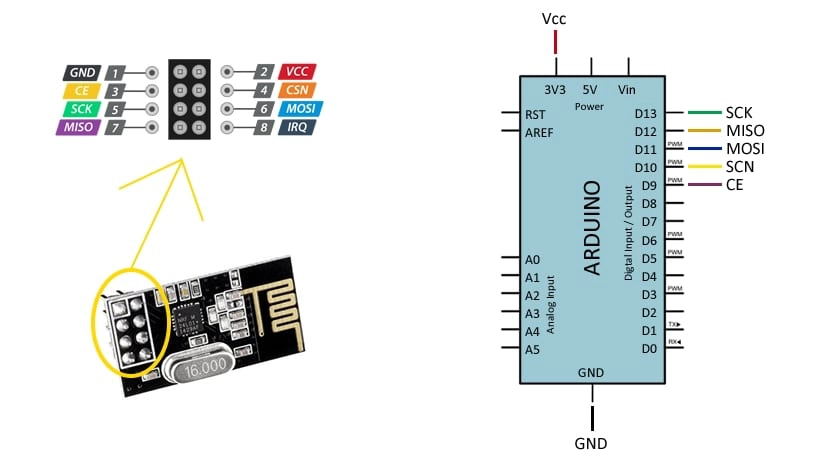
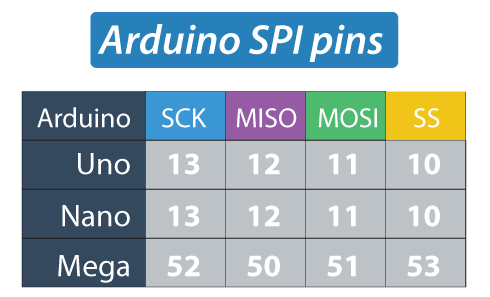
Here is a simple Arduino Wireless Communication using nRF24L01 Transceiver Module:

### Unidirectional Communication

#### Circuit Diagram



#### Connections



NOTE .:

1. MOSI, MISO, and SCK pins must be digital pins 11, 12, and 13 respectively.
2. CE and CSN pins can be any two digital pins! They are used to:
   1. Setting the board in standby or active mode.
   2. Switching between transmit and receive command mode.

#### Code

##### Transmitter Code

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

RF24 radio(9, 10); // CE, CSN Pins

const uint64\_t address = 0xF0F0F0F0E1LL;

void setup()

{

radio.begin();

radio.openWritingPipe(address);

radio.setPALevel(RF24\_PA\_MIN);

radio.stopListening();

}

void loop()

{

const char sendData[] = "This is a test!";

radio.write(&sendData, sizeof(sendData));

delay(500);

}

##### Receiver Code

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

RF24 radio(9, 10); // CE, CSN Pins

const uint64\_t address = 0xF0F0F0F0E1LL;

void setup()

{

Serial.begin(9600);

radio.begin();

radio.openReadingPipe(1, address);

radio.setPALevel(RF24\_PA\_MIN);

radio.startListening();

}

void loop()

{

if (radio.available()) {

char recvData[32] = "";

radio.read(&recvData, sizeof(recvData));

Serial.println(recvData);

}

}

##### Code Explanation

1. We need to include the basic SPI and the newly installed RF24 libraries and create an RF24 object. The two arguments here are the CSN and CE pins.

RF24 radio(9, 10); // CE, CSN Pins

1. We need to create a byte array represents the address, or the so called pipe through which the two modules will communicate.

const uint64\_t address = 0xF0F0F0F0E1LL;

OR:

const byte address[6] = "00001";

We can change the value of this address to any 5 letter string and this enables to choose to which receiver we will talk, so in our case we will have the same address at both the receiver and the transmitter.

1. To initialize the radio object and transmit using the radio.openWritingPipe(),, we set the address of the receiver has the 5 letter string we previously set.

radio.openWritingPipe(address);

1. At the receiver, using the radio.setReadingPipe() function we set the same address which enables the communication between the two modules.

radio.openReadingPipe(1, address);

1. radio.setPALevel() sets the Power Amplifier level in case of PA/LNA module.
   1. If modules are very close to each other, use:

radio.setPALevel(RF24\_PA\_MIN);

OR

radio.setPALevel(RF24\_PA\_LOW);

* 1. If modules are far from each other, use:
     1. Bypass capacitor (104) across GND and 3.3V of the modules.

radio.setPALevel(RF24\_PA\_HIGH);

1. radio.stopListening() sets module as transmitter. On the other side, radio.startListening() sets the module as receiver.

radio.stopListening(); // at the Transmitter

radio.startListening(); // at the Receiver

1. In loop section:
2. Transmitter
3. create an array of characters to which we assign the message text. Using the radio.write() function we will send that message to the receiver.
4. radio.write() takes:
   * + 1. The address of the variable to be sent.
       2. Number of bytes to be sent.
5. we will add 0.5 second delay.

void loop()

{

const char sendData[] = "This is a test!";

radio.write(&sendData, sizeof(sendData));

delay(500);

}

1. Receiver:
2. radio.available() check whether there is data to be received. If that’s true, first we create an array of 32 elements, called “text”, in which we will save the incoming data.

void loop() {

if (radio.available()) {

char recvData[32] = "";

radio.read(&recvData, sizeof(recvData));

Serial.println(recvData);

}

}

### Key Parameters Of RF24 Library

In RF24 library documentation, there are many parameters that can be set. Some key parameters are:

1. Channel: the specific frequency channel that communication will occur on (frequencies are mapped to integers between 0 and 125
2. Reading pipe: a unique 24, 32, or 40-bit address the module reads data from.
3. Writing pipe: a unique address to which the module writes data
4. Power Amplifier (PA) level: the PA level sets the power draw of the chip and thereby the transmission power. For the purposes of this tutorial (use with the Arduino) we will use the minimum power setting.

In general, when using the RF module with an Arduino board, it is probably a good idea to keep the PA level as low as possible to reduce the current draw on the Arduino’s regulated power supply.

At higher levels (further distances), 104 ceramic bypass capacitor at module i/p.

A good programming practice for storing the writing and reading pipe addresses is to put the two values inside an array (byte array named “addresses.”)

Note that the openReadingPipe() method must be passed an additional integer parameter that describes which reading pipe is being initialized. This is because the RF module can receive from up to 6 pipes at a time!

Listening is done by calling the RF24::startListening() method.

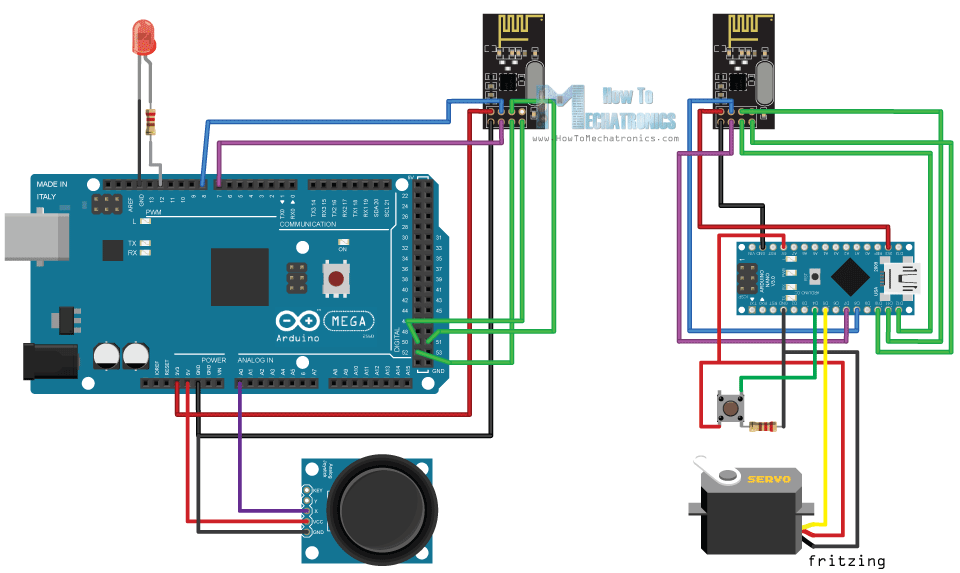
Note: the reading pipe must be initialized before start listening for data (openReadingPipe() method must be called before the startListening() method!)

Similarly, stopListening() method must be called before writing.

If data is available over RF connection, available()returns true, and data can be read.

### Bidirectional Communication

#### Circuit Diagram



#### Code

##### Transmitter Code

/\*

\* Arduino Wireless Communication Tutorial

\* Example 2 - Transmitter Code

\*

\* by Dejan Nedelkovski, www.HowToMechatronics.com

\*

\* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/

\*/

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#define led 12

RF24 radio(7, 8); // CE, CSN

const byte addresses[][6] = {"00001", "00002"};

boolean buttonState = 0;

void setup() {

pinMode(12, OUTPUT);

radio.begin();

radio.openWritingPipe(addresses[1]); // 00002

radio.openReadingPipe(1, addresses[0]); // 00001

radio.setPALevel(RF24\_PA\_MIN);

}

void loop() {

delay(5);

radio.stopListening();

int potValue = analogRead(A0);

int angleValue = map(potValue, 0, 1023, 0, 180);

radio.write(&angleValue, sizeof(angleValue));

delay(5);

radio.startListening();

while (!radio.available());

radio.read(&buttonState, sizeof(buttonState));

if (buttonState == HIGH) {

digitalWrite(led, HIGH);

}

else {

digitalWrite(led, LOW);

}

}

##### Receiver Code

/\*

\* Arduino Wireless Communication Tutorial

\* Example 2 - Transmitter Code

\*

\* by Dejan Nedelkovski, www.HowToMechatronics.com

\*

\* Library: TMRh20/RF24, https://github.com/tmrh20/RF24/

\*/

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

#define led 12

RF24 radio(7, 8); // CE, CSN

const byte addresses[][6] = {"00001", "00002"};

boolean buttonState = 0;

void setup() {

pinMode(12, OUTPUT);

radio.begin();

radio.openWritingPipe(addresses[1]); // 00002

radio.openReadingPipe(1, addresses[0]); // 00001

radio.setPALevel(RF24\_PA\_MIN);

}

void loop() {

delay(5);

radio.stopListening();

int potValue = analogRead(A0);

int angleValue = map(potValue, 0, 1023, 0, 180);

radio.write(&angleValue, sizeof(angleValue));

delay(5);

radio.startListening();

while (!radio.available());

radio.read(&buttonState, sizeof(buttonState));

if (buttonState == HIGH) {

digitalWrite(led, HIGH);

}

else {

digitalWrite(led, LOW);

}

}

##### Code Explanation

What’s different here from the previous example is that:

1. Create two pipes or addresses for the bi-directional communication.

const byte addresses[][6] = {"00001", "00002"};

* 1. In the setup section, define both pipes, and note that the writing address at the first Arduino needs to be the reading address at the second Arduino, and vice versa, the reading address at the first Arduino needs to be the writing address at the second Arduino.
     1. Transmitter:

radio.openWritingPipe(addresses[1]); // 00001

radio.openReadingPipe(1, addresses[0]); // 00002

* + 1. Receiver:

radio.openWritingPipe(addresses[0]); // 00002

radio.openReadingPipe(1, addresses[1]); // 00001

* 1. In the loop section using the radio.stopListening() function we set the first Arduino as transmitter, read and map the value of Joystick from 0 to 180, and using the radio.write() function send the data to the receiver.

radio.stopListening();

int potValue = analogRead(A0);

int angleValue = map(potValue, 0, 1023, 0, 180);

radio.write(&angleValue, sizeof(angleValue));

* 1. On the other side, using the radio.startListening() function we set the second Arduino as receiver and we check whether there is available data. While there is data available we will read it, save it to the “angleV” variable and then use that value to rotate the servo motor.

radio.startListening();

if ( radio.available()) {

while (radio.available()) {

int angleV = 0;

radio.read(&angleV, sizeof(angleV));

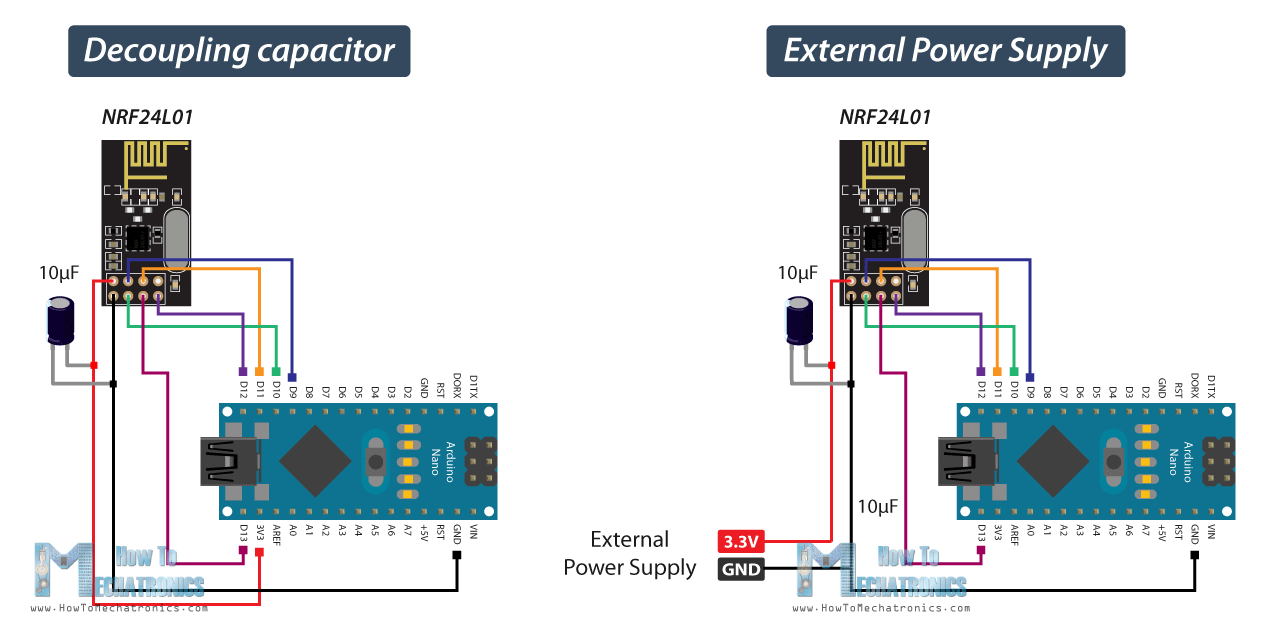
myServo.write(angleV);

* 1. Next, at the transmitter, we set the first Arduino as receiver and with an empty “while” loop we wait for the second Arduino the send data, and that’s the data for the state of the push button whether is pressed or not. If the button is pressed the LED will light up. So, these processes constantly repeats and both Arduino boards are constantly sending and receiving data.

## Troubleshoot

### Power Supply Noise

It is one of the most common issues people experience when trying to make successful communication with the NRF24L01 modules. Generally, RF circuits or radio frequency signals are sensitive to power supply noise. Therefore, it’s always a good idea to include a decoupling capacitor across the power supply line. The capacitor can be anything from 10uF to 100uF.



### External Power Supply

The 3.3V pin of the Arduino boards cannot always supply enough power to the NRF24L01 module. So, powering the module with an external power source is also a good idea.

## References

[4]–[7]

# References

[1] “AGV Navigation: Methods, Comparison, Pros and Cons - Illustrated Guide.” https://www.agvnetwork.com/types-of-navigation-systems-automated-guided-vehicles (accessed Oct. 11, 2021).

[2] “L298N Motor Driver - Arduino Interface, How It Works, Codes, Schematics.” https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/ (accessed Oct. 22, 2021).

[3] “In-Depth: Interface L298N DC Motor Driver Module with Arduino.” https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/ (accessed Oct. 22, 2021).

[4] “nRF24L01 Transceiver Module.” https://www.electronicshub.org/nrf24l01-transceiver-module/ (accessed Oct. 22, 2021).

[5] “nRF24L01 - How It Works, Arduino Interface, Code, Schematic.” https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/ (accessed Oct. 22, 2021).

[6] “In-Depth: How nRF24L01 Wireless Module Works & Interface with Arduino.” https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/ (accessed Oct. 22, 2021).

[7] “Wireless Communication: Introduction, Types and Applications.” https://www.electronicshub.org/wireless-communication-introduction-types-applications/ (accessed Oct. 22, 2021).