**CERTIFICATE**

This is to certify that the project work entitled “Wireless Multi-Utility Rover” submitted by Sanjeev (CO15547), Vishal (CO15561), Yash (CO15563) in fulfilment for the requirements of the award of Bachelor of Engineering Degree in Electronics and Communication engineering at Chandigarh college of engineering and technology (Degree Wing), Chandigarh is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University/ Institute for the award of any Degree.

Dr. Parvinder Singh

Department of ECE

CCET (Degree Wing)

Chandigarh

Date: 22/11/2018

Place: Chandigarh

**ACKNOWLEDGEMENT**

We would like to place on record my deep sense of gratitude to Dr. D.S.SAINI, HOD-Dept, of Electronics & Communication Engineering, CCET, Chandigarh, India for his generous guidance, help and useful suggestions

We express our sincere gratitude to Prof. Parminder Kaur of Electronics & Communication Engineering Department, CCET, Chandigarh, India, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work.

**Signature of Students**

**ABSTRACT**

In the constant effort to safe guard the country against enemy attacks and terrorists, defense forces have to be equally power full to the enemy countries. For the strengthening of the defense power there was immense need for the non-human war machines such as missiles and rockets, which could track the enemy target and destroy them in order to safe guard the country. This project is mainly designed for surveillance of ground in military applications. The important devices that we have used on the robot are GPS, wireless camera, two pairs of RF transmitter-receiver, temperature sensor. If the intensity of the light is less while performing video streaming, then additional lights can be switched on wirelessly. The robotic movement can be controlled wirelessly by the user with the buttons provided on the user module via RF. Along with the GPS readings, the robot also senses the temperature of the robot's surrounding. All the four wheels are energized. Hence, the robot is able to rotate about 360 degree on stand. The project presented by us is an advanced Robotic system which can be controlled through RF signals and the robot's geographical position can be continuously monitored by GPS. The proposed prototype has a wide application such as, military ground surveillance in nasal threatened area, no man's land between international borders, hijacked buildings. It can also be used to study animals, Stand-alone security systems, Safety monitoring in industries, Continuous monitoring of epidemic patients who are kept isolated.

**TABLE OF CONTENTS**

**TOPIC PAGE NO.**

Acknowledgement

Abstract

List of figures

**Chapter 1: Introduction**

* 1. GENERAL INTRODUCTION
  2. DESIGN OF SYSTEM HARDWARE
  3. L2N3D MOTOR DRIVER CIRCUIT
  4. NRF24L01 TRANSCEIVER
  5. MQ-132 AIR QUALITY CHECKER
  6. CONTROL CIRCUITRY: ARDUINO UNO
  7. DESIGN OF SYSTEM SOFTWARE

**Chapter 2: Literature Survey**

2.1 PREVIOUS WORKS

2.1.1 NECESSITY

2.1.2 SIMILAR GUIDED ROVERS

2.1.3 DRAWBACKS

2.2 DOCUMENTAION ON SIMILAR GUIDED ROVERS

**Chapter 3: Present Work**

3.1 AIM

3.2 OBJECTIVE OF THE PROJECT

3.3 METHODOLOGY

3.3.1 HARDWARE

3.3.2 SOFTWARE

3.4 COMPONENTS USED

**Chapter 4: Conclusion and Future**

4.1 CONCLUSION

4.2 FUTURE SCOPE

**REFEREMCE**

**LIST OF FIGURES**

L2N3D MOTOR DRIVER CIRCUIT

NRF24L01 TRANSCEIVER

MQ-132 AIR QUALITY CHECKER

CONTROL CIRCUITRY: ARDUINO UNO

INDUCTORS

CAPACITORS

RESISTORS

INTERFACING DIAGRAMS

**CHAPTER 1:**

**INTRODUCTION**

**1.1 GENERAL INTRODUCTION**

The main purpose of the robot we are making is to provide visual information of hard to access places, for example a building under a hostage situation. Hence the main feature of our robot is an onboard video camera. Also, the robot must be compact and self-contained in the sense it must have an onboard battery pack and wireless interface to the human controller.

Surveillance system is a type of system that been used to observe specific activities or areas in term of managing, directing or protecting purpose. By monitoring the certain activities or areas every time using the surveillance system, the users are able to know if something abnormal happens and further action can be taken after that.

According to Tom Harris, robot is an electro-mechanical machine that is controlled by a computer program or electronic apparatus. It is being used to give convenience to human by doing specific task. Robots are able to execute the task that impossible or beyond capability of human. From this advantage, it can give many advantages in term of safety, ease and efficiency.

Surveillance system and robots are two different systems that carry out their own task respectively. The main inspiration for this project is to combine these two systems to produce a mobile robot equipped with a camera as an intelligence surveillance system. By using this Surveillance Robot, the image visualize can be more efficient compared to other typical surveillance system since its multiple viewing angle can be varied. This robot can be used as a medium for a monitoring system wirelessly by remotely-controlled by human via computer.

**1.2 DESIGN OF SYSTEM HARDWARE**

System hardware design composed of L2N3D motor driver circuit, Arduino UNO, nrf24l01 transceiver, Air Quality (MQ-132) Checker.

**1.3 L2N3D MOTOR DRIVER CIRCUIT**

L293D IC is a typical Motor Driver IC which allows the DC motor to drive on any direction. This IC consists of 16-pins which are used to control a set of two DC motors instantaneously in any direction. It means, by using a L293D IC we can control two DC motors. As well, this IC can drive small and quiet big motors.

This L293D IC works on the basic principle of H-bridge, this motor control circuit allows the voltage to be flowing in any direction. As we know that the voltage must be change the direction of being able to rotate the DC motor in both the directions. Hence, H-bridge circuit using L293D ICs are perfect for driving a motor. Single L293D IC consists of two H-bridge circuits inside which can rotate two DC motors separately. Generally, these circuits are used in robotics due to its size for controlling DC motors.

* 1. **NRF24L01 TRANSCEIVER**

It uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space and with lower baud rate its range can reach up to 100 meters. The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can [communicate with up to 6 other units at the same time](https://howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules/). The power consumption of this module is just around 12mA during transmission, which is even lower than a single LED. The operating voltage of the module is from 1.9 to 3.6V, but the good thing is that the other pins tolerate 5V logic, so we can easily connect it to an Arduino without using any logic level converters.

* 1. **MQ-135 AIR QUALITY CHECKER**

A gas detector is a device that detects the presence of gases in an area, often as part of a safety system. This type of equipment is used to detect a gas leak or other emissions and can interface with a control system so a process can be automatically shut down. A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.

Gas detectors can be used to detect combustible, flammable and toxic gases, and oxygen depletion. This type of device is used widely in industry and can be found in locations, such as on oil rigs, to monitor manufacture processes and emerging technologies such as photovoltaic. They may be used in firefighting.

**1.6 CONTROL CIRCUITRY: ARDUINO UNO**

The Arduino UNO is an open-source microcontroller board based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

**1.7 DESIGN OF SYSTEM SOFTWARE**

The System software developed in Embedded C Language using Arduino IDE which has the ability of receiving the data from sensor and transmitting the data and sends information to user’s device regarding the readings from sensors.

**ARDUINO IDE**

The [**Arduino**](https://en.wikipedia.org/wiki/Arduino)**integrated development environment (**[**IDE**](https://en.wikipedia.org/wiki/Integrated_development_environment)**)** is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is used to write and upload programs to Arduino board.

The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

**CHAPTER 2:**

**LITERATURE SURVEY**

**2.1 PREVIOUS WORKS**

**2.1.1 NECESSITY**

Presently, an intelligence surveillance system is in high demand where the traditional ways of monitoring system which using CCTV resulting an ineffective image due to its static position. Several projects have been done to overcome the problem and one of the basic idea is developing a surveillance mobile robot. Several related project is reviewed as follows.

**2.1.2** **SIMILAR PRODUCTS**

**2.2.2.1 Autonomous Explorer Mobile Robot**

The previous project by Csongor Márk Horváth and Róbert Tamás Fekete in their article "Development of Autonomous Explorer Mobile Robot for a Specific Environment" on 2011 is using the same basic concept as the Surveillance Robot where a mobile robot is equipped with a camera but the robot is used for exploring purpose instead of monitoring. Figure 2.1 below shows the Autonomous Explorer Mobile Robot. Figure In this project, the robot is controlled by ATMEL ATmega128 microcontroller. It is a fully automated type of robot where it consists of sonar sensor for obstacle avoiding and infrared sensor for a wall following function [4]. The mobile robot used a wireless camera in order to transmit the image captured on-site to a monitoring station. This robot does not use a communicating device such as RF transceiver to manually control by a computer. It only used a Wi-Fi network to connect the robot and computer. This mobile robot is fully depends on the Wi-Fi access point that determine its covering area which gives some disadvantages where this robot only able to operate in a Wi-Fi network environment.

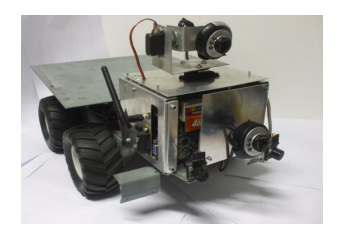


Figure 2.1: AUTONOMOUS EXPLORER MOBILE ROBOT

**2.2.2.2 Remote Controlled Surveillance Mobile Robot with IP Camera**

The Remote Controlled Surveillance Mobile Robot with IP Camera is developed by Gilbert, Martin and Janssen in 2011. This manually-operated mobile robot using PIC 16F877A microcontroller and equipped with wireless IP Camera. As shown in Figure 2.2, this surveillance mobile robot also being control by user using a GUI console created on a computer. As a connecting device, this robot used an RF Transceiver. Other additional function such as battery level indicator also included in this project.

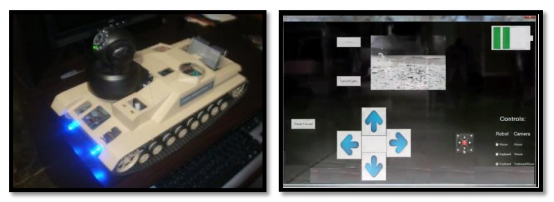


FIGURE 2.2: RC Surveillance Mobile Robot with GUI Console

**2.1.3 DRAWBACKS**

* Limited Use.
* Risk of Breaking Down.
* Risk of Short Circuiting.

**2.2 DOCUMENTATION ON SIMILAR GUIDED ROVERS**

**2.2.1 Patent No.7436143**: Miniature Surveillance Robot is very relevant to this project as it shares a lot of the same concepts that we are hoping to implement. Among them is the concept of a logic controller operating a drive motor to move a robot. This project also includes inputs from various sensors such as proximity, sound, and chemical sensors to make decisions as to where the robot moves to. These are consistent with the patent claims #12 and #13. The surveillance robot should also be able to communicate wirelessly with a network to store the camera footage captured, which is also consistent with various claims of the patent.

**2.2.2 IEEE Explorer Article Search Publication Number:** US 2010/085946 shows how to be able to program a robot or override a robot. The idea is to be able to interact with the robot remotely. In this design we are going to develop an interface to interact with the robot by programming it or controlling it with certain functions. This will help the user to set up the robot easily and efficiently. For part of the marketing requirements a GUI is the best way to setup a robot easily. The idea of the GUI is to keep things simple with the design so the user can have some control. If the costumer or user wants to expand the robot capabilities they can update the interface for the robot’s plug-ins and add-ons.

**CHAPTER 3: PRESENT WORK**

**3.1 AIM**

WIRELESS SURVEILLANCE ROVER.

**3.2 OBJECTIVES OF THE PROJECT**

* Scans for various smells.
* Remove the workload from human .i.e. efficient resource management.

**3.3 METHODOLOGY**

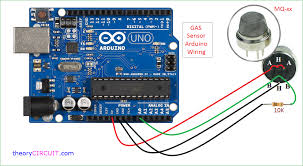
For the implementation of project both hardware and software requirements are specified below. Familiarity with coding is required for using Arduino UNO IDE available for programming IDE.

**3.3.1 HARDWARE**

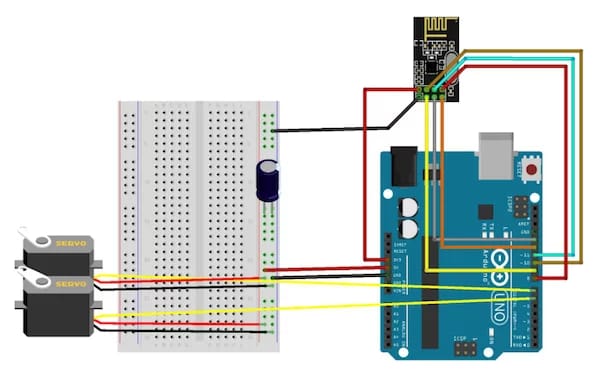
* Interfacing MQ135 GAS sensor to Arduino UNO.
* Interfacing NRF24L01 TRANSCEIVER to Arduino UNO.
* Interfacing L2N3D MOTOR DRIVER CIRCUIT.

**3.3.2 SOFTWARE**

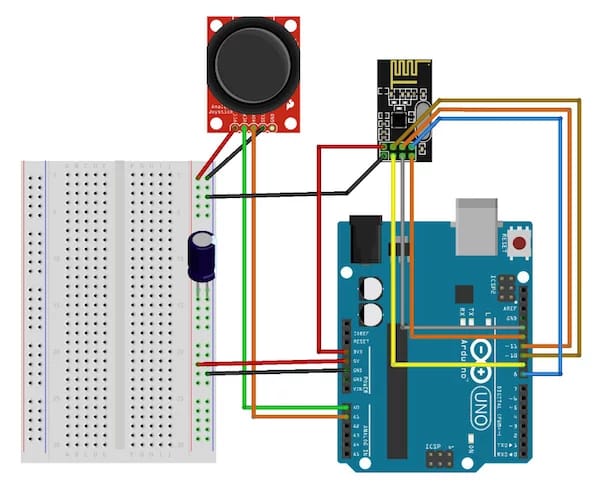
* ARDUINO UNO IDE is an integrated development environment which uses coding for developing programs for required purpose.

****

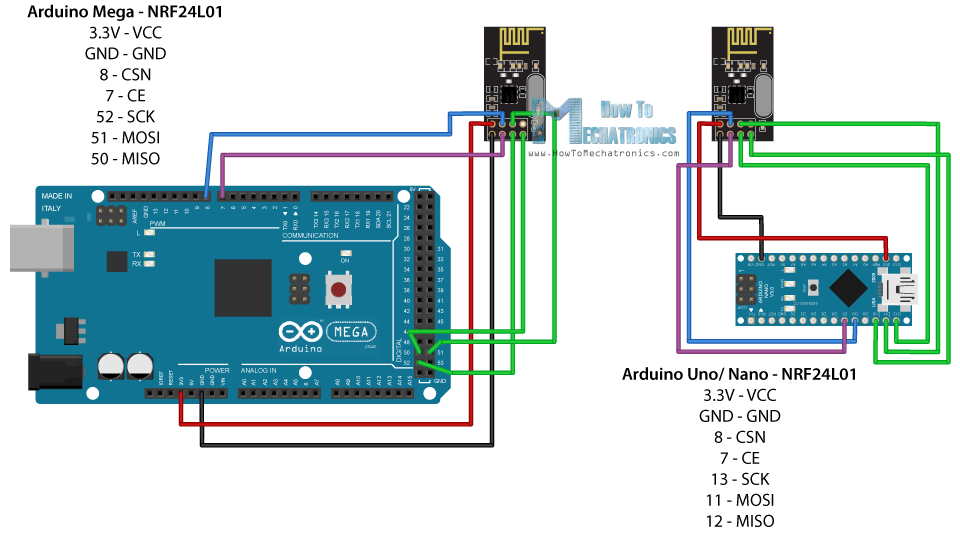
MQ-135 GAS SENSOR Interfacing with ARDUINO



INTERFACING OF SERVO MOTORS WITH ARDUINO



INTERFACING OF JOYSTICK MODULE WITH ARDUINO



INTERFACING OF NRF24L01 WITH ARDUINO UNO

**3.4 COMPONENTS USED**

**3.4.1 ARDUINO UNO**

**INTRODUCTION**

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project. The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.



Fig.2 Arduino Uno Board

**ARDUINO UNO PIN DISCRIPTION**

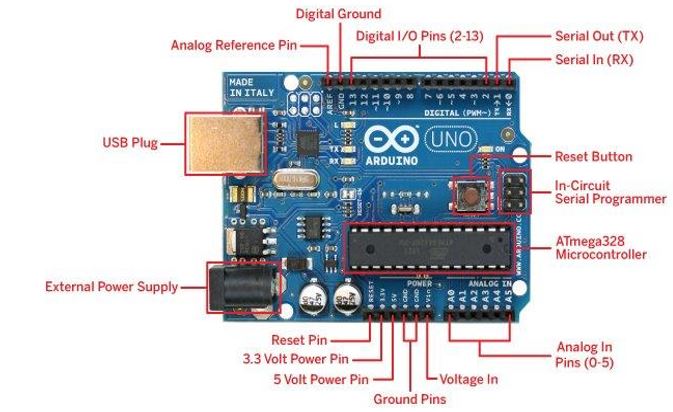


Fig.3 ARDUINO UNO PIN DISCRIPTION

Starting clockwise from the top center:

Analog Reference pin (orange)

Digital Ground (light green)

Digital Pins 2-13 (green)

Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communication (e.g. Serial.begin). Reset Button - S1 (dark blue)

In-circuit Serial Programmer (blue-green)

Analog In Pins 0-5 (light blue)

Power and Ground Pins (power: orange, grounds: light orange)

External Power Supply In (9-12VDC) - X1 (pink)

Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)

USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

**DIGITAL PINS**

Input and Output

Each of the 14 digital pins on the Arduino Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:

Serial Pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-toTTL Serial chip.

External Interrupts Pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. Use the attachInterrupt() function .

PWM 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it’s off. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

TWI A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board: BT Reset: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

**ANALOG PINS**

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins. I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library.

**Power Pins**

VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input. 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply. 3V3. (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip. GND. Ground pins.

**Other Pins AREF.**

Reference voltage for the analog inputs. Used with analogReference(). Reset. (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

AREF

Reference voltage for the analog inputs. Used with analogReference().

Reset

Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. To reset Arduino program Prepare the basic empty program (empty setup, loop, etc.)

1. Compile it.
2. Reset the Arduino using the hardware button on the chip.
3. Press Ctrl + U to upload your code.
4. If unsuccessful - got to 3.

2.13.4 What's on the board?

**Reset Button**

Just like the original Nintendo, the Arduino has a reset button . Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any problems. Power LED Indicator Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit!

**TX RX LEDs**

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we’re loading a new program onto the board).

**Main IC**

The black thing with all the metal legs is an IC, or Integrated Circuit . Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC’s, reading the datasheets is often a good idea.

**Voltage Regulator**

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don’t hook up your Arduino to anything greater than 20 volts.

**The Arduino Family**

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you’re not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well-suited to someone new to the world of Arduino: Arduino Uno (R3) The Uno is a great choice for your first Arduino. It’s got everything you need to get started, and nothing you don’t. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs.

**Arduino Uno coding Using IDE software**



Arduino Uno IDE software

The “void setup()” section is widely used to initialize variables, pin modes, set the serial baud rate and related. The software only goes though the section once.

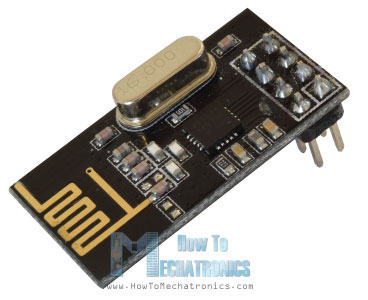
The “void loop()” section is the part of the code that loops back onto itself and is the main part of the code. In the Arduino examples, this is called “Bare Minimum” under File-> Examples -> Basics. Note that you are free to add subroutines using the same syntax:

void subroutinename() {} Almost every line of code needs to end with a semicolon ‘;’ (there are a few exceptions which we will see later). To write single line comments in the code, type two back slashes followed by the text: //comments are overlooked when compiling your program To write multi-line comments, start the comment with /\* and end with \*/

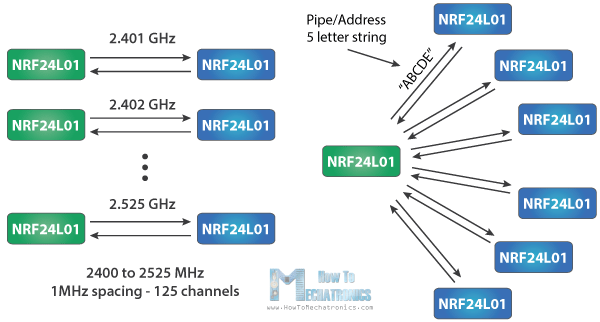
/\* This is a multi-line comment and saves you having to always use double slashes at the beginning of every line. Comments are used used to explain the code textually. Good code always has a lot of comments.\*/

**3.4.2 NRF24L01 TRANSCEIVER**

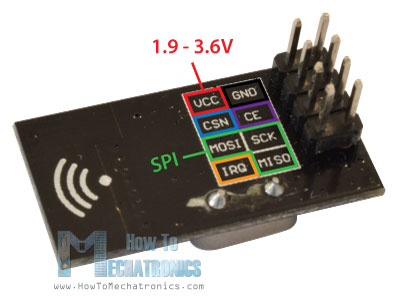
Let’s take a closer look at the NRF24L01 transceiver module. It uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space and with lower baud rate its range can reach up to 100 meters.



The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can [communicate with up to 6 other units at the same time](https://howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules/).



The power consumption of this module is just around 12mA during transmission, which is even lower than a single LED. The operating voltage of the module is from 1.9 to 3.6V, but the good thing is that the other pins tolerate 5V logic, so we can easily connect it to an Arduino without using any logic level converters.



Three of these pins are for the SPI communication and they need to be connected to the SPI pins of the Arduino, but note that each Arduino board have different SPI pins. The pins CSN and CE can be connected to any digital pin of the Arduino board and they are used for setting the module in standby or active mode, as well as for switching between transmit or command mode. The last pin is an interrupt pin which doesn’t have to be used.

So once we connect the NRF24L01 modules to the Arduino boards we are ready to make the codes for both the transmitter and the receiver.

**3.4.3 MQ-135 QUALITY CHECKER**

A **gas detector** is a device that detects the presence of [gases](https://en.wikipedia.org/wiki/Gas) in an area, often as part of a [safety system](https://en.wikipedia.org/w/index.php?title=Safety_system&action=edit&redlink=1). This type of equipment is used to detect a [gas leak](https://en.wikipedia.org/wiki/Gas_leak) or other emissions and can interface with a [control system](https://en.wikipedia.org/wiki/Control_system) so a process can be automatically shut down. A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.

Gas detectors can be used to detect [combustible](https://en.wikipedia.org/wiki/Combustible), [flammable](https://en.wikipedia.org/wiki/Flammable) and [toxic](https://en.wikipedia.org/wiki/Toxic) gases, and [oxygen](https://en.wikipedia.org/wiki/Oxygen) depletion. This type of device is used widely in industry and can be found in locations, such as on oil rigs, to monitor manufacture processes and emerging technologies such as [photovoltaic](https://en.wikipedia.org/wiki/Photovoltaic). They may be used in [firefighting](https://en.wikipedia.org/wiki/Firefighting).

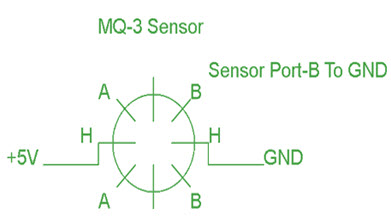
**Gas leak detection** is the process of identifying potentially hazardous [gas leaks](https://en.wikipedia.org/wiki/Gas_leaks) by sensors. These sensors usually employ an audible alarm to alert people when a dangerous gas has been detected. Exposure to toxic gases can also occur in operations such as painting, fumigation, fuel filling, construction, excavation of contaminated soils, landfill operations, entering confined spaces, etc. Common sensors include combustible gas sensors, photoionization detectors, [infrared point sensors](https://en.wikipedia.org/wiki/Infrared_point_sensor), [ultrasonic sensors](https://en.wikipedia.org/wiki/Ultrasonic_sensor), [electrochemical gas sensors](https://en.wikipedia.org/wiki/Electrochemical_gas_sensor), and [semiconductor sensors](https://en.wikipedia.org/w/index.php?title=Semiconductor_sensor&action=edit&redlink=1). More recently, infrared imaging sensors have come into use. All of these sensors are used for a wide range of applications and can be found in industrial plants, refineries, pharmaceutical manufacturing, fumigation facilities, paper pulp mills, aircraft and shipbuilding facilities, hazmat operations, waste-water treatment facilities, vehicles, indoor air quality testing and homes.

**** MQ-135 QUALITY CHECKER

The MQ-135 gas sensor senses the gases like ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulfide and smoke. The [boost converter](https://www.elprocus.com/dc-dc-converter-types/) of the chip MQ-3 gas sensor is PT1301. The operating voltage of this gas sensor is from 2.5V to 5.0V. The MQ-3 gas sensor has a lower conductivity to clean the air as a gas sensing material. In the atmosphere we can find polluting gases, but the conductivity of gas sensor increases as the concentration of polluting gas increases. MQ-135 gas sensor can be implementation to detect the smoke, benzene, steam and other harmful gases. It has potential to detect different harmful gases. The MQ-135 gas sensor is low cost to purchase.

**Basic Pin Configuration Of Alcohol Sensor**

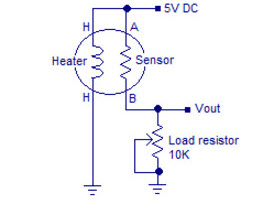
The MQ-3 alcohol gas sensor consists of total 6-pins including A, H, B and the other three pins are A, H, B out of the total 6-pins we use only 4 pins. The two pins A, H are used for the heating purpose and the other two pins are used for the ground and power. There is a heating system inside the sensor, which is made up of aluminium oxide, tin dioxide.  It has heat coils to produce heat, and thus it is used as a [heat sensor](https://www.elprocus.com/heat-sensor-circuit-and-working-operation/). The below diagram shows the pin diagram and the configuration of the MQ-3 alcohol sensor.



Pin Configuration Of Alcohol Sensor

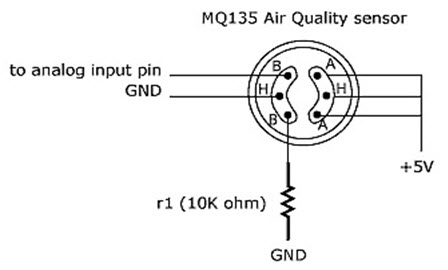
**Working Principle And Circuit Diagram**

The MQ-135 alcohol sensor consists of a tin dioxide (SnO2), a perspective layer inside aluminium oxide micro tubes (measuring electrodes) and a heating element inside a tubular casing. The end face of the sensor is enclosed by a stainless steel net and the back side holds the connection terminals. Ethyl alcohol present in the  breath is oxidized into acetic acid passing through the heat element. With the ethyl alcohol cascade on the tin dioxide sensing layer, the resistance decreases. By using the external load resistance the resistance variation is converted into a suitable voltage variation. The circuit diagram and the connection arrangement of an MQ 135 alcohol is shown below.

MQ-135 Circuit Diagram

**MQ – 135 Air Quality Sensor**

The air quality sensor is also a MQ-135 sensor for detecting venomous gases that are present in the air in homes and offices. The gas sensor layer of the sensor unit is made up of tin dioxide (SnO2); it has lower conductivity compare to clean hair and due to air pollution the conductivity is increases. The air quality sensor detects ammonia, nitrogen oxide, smoke, CO2 and other harmful gases. The air quality sensor has a small potentiometer that permits the adjustment of the load resistance of the sensor circuit. The 5V [power supply](https://www.elprocus.com/types-power-supplies/)is used for air quality quality sensor.



MQ – 135 Air Quality Sensor

The air quality sensor is a signal output indicator instruction. It has two outputs: analog output and [TTL output](https://www.elprocus.com/transistor-transistor-logic-ttl/). The TTL output is low signal light which can be accessed through the IO ports on the Microcontroller. The analog output is an concentration, i.e. increasing voltage is directly proportional to increasing concentration. This sensor has a long life and reliable stability as well.

**Applications Of MQ 135 Gas Sensor**

The following are the applications of the MQ 135 gas sensor:

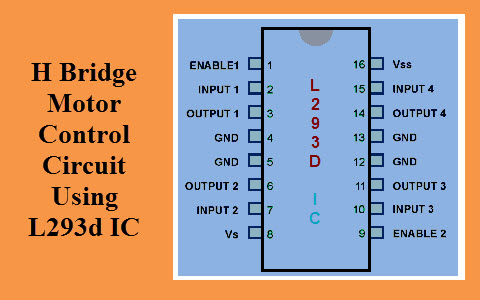
* Air quality monitor
* Detection of harmful gases
* Domestic air pollution detection
* Industrial pollution detection
* Portable air pollution detection

**Characteristics Of MQ 135**

* Good sensitivity to harmful gases in wide range.
* It has long life and low cost.
* Possesses high sensitivity to ammonia, benzene, sulfide gases.
* It is a simple drive circuit

**3.4.4 L2N93D MOTOR DRIVER CIRCUIT**

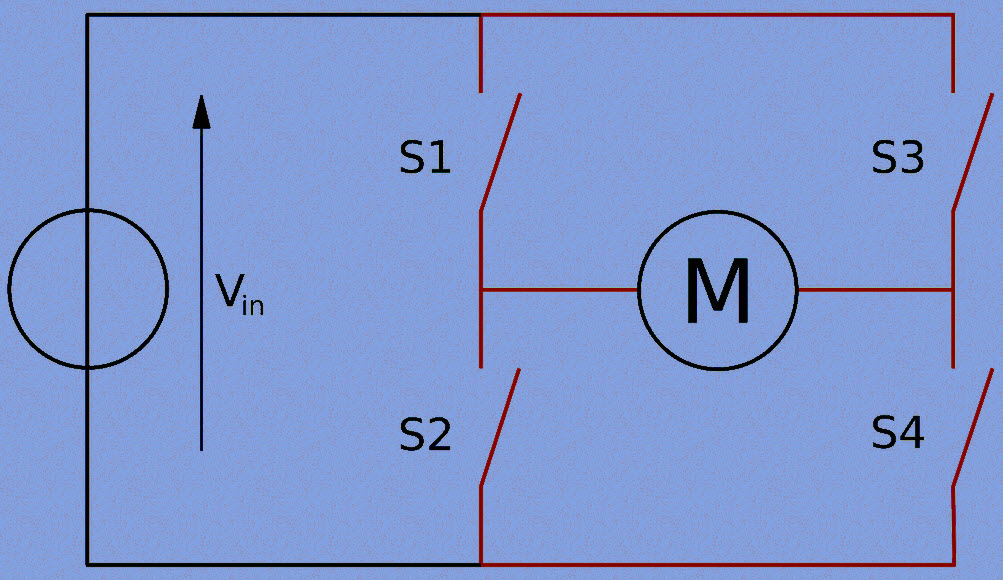
Common  DC gear head motors need current above 250mA. There are many integrated circuits like ATmega16 [Microcontroller](https://www.elprocus.com/8051-microcontroller-architecture-and-applications/), [555 timer IC](https://www.elprocus.com/555-timer-circuits-for-engineering-students/). But, IC 74 series cannot supply this amount of current. When the motor is directly connected to the o/p of the above ICs then, they might damaged. To overcome this problem, a motor control circuit is required, which can act as a bridge between the above motors and ICs ([integrated circuits](https://www.elprocus.com/different-types-of-integrated-circuits/)). There are various ways of making H-bridge motor control circuit such as using transistor, relays and using L293D/L298.

[](https://www.elprocus.com/wp-content/uploads/2015/03/H-Bridge-Motor-Control-Circuit-Using-L293d-IC.jpg)

H Bridge Motor Control Circuit Using L293d IC

**H-Bridge Circuit**

A H bridge is an electronic circuit that allows a voltage to be applied across a load in any direction. H-bridge circuits are frequently used in robotics and many other applications to allow DC motors to run forward  & backward. These motor control circuits are mostly used in different converters like DC-DC, DC-AC, AC-AC converters and many other types of [power electronic converters](https://www.elprocus.com/power-electronic-converters/). In specific, a bipolar stepper motor is always driven by a motor controller having two H-bridges

[](https://www.elprocus.com/wp-content/uploads/2015/03/H-Bridge-Circuit.jpg)

H-Bridge Circuit

A H-bridge is fabricated with[four switches](https://www.elprocus.com/switches-types-working/) like S1, S2, S3 and S4. When the S1 and S4 switches are closed, then a +ve voltage will be applied across the motor. By opening the switches S1 and S4 and closing the switches S2 and S3, this voltage is inverted, allowing invert operation of the motor.

Generally, the H-bridge motor driver circuit is used to reverse the direction of the motor and also to break the motor. When the motor comes to a sudden stop, as the terminals of the motor are shorted. Or let the motor run free to a stop, when the motor is detached from the circuit. The table below gives the different operations with the four switches corresponding to the above circuit.

[](https://www.elprocus.com/wp-content/uploads/2015/03/Operation-of-the-H-Bridge.jpg)

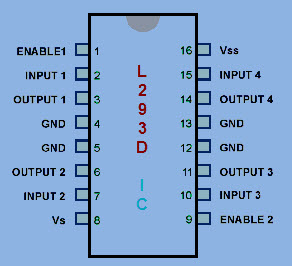
Operation of the H-Bridge

1. **L293D Motor Driver IC**

L293D IC  is a typical Motor Driver IC which allows the [DC motor](https://www.elprocus.com/dc-motor-basics-types-application/) to drive on any direction. This IC consists of  16-pins which are used to control a set of two DC motors instantaneously in any direction. It means, by using a L293D IC we can control two DC motors. As well, this IC can drive small and quiet big motors.

This L293D IC works on the basic principle of H-bridge, this [motor control](https://www.elprocus.com/what-are-the-best-ways-to-control-the-speed-of-dc-motor/) circuit allows the voltage to be flowing in any direction. As we know that the voltage must be change the direction of being able to rotate the DC motor in both the directions. Hence, H-bridge circuit using L293D ICs are perfect for driving a motor. Single L293D IC consists of two H-bridge circuits inside which can rotate two DC motors separately. Generally, these circuits are used in robotics due to its size for controlling DC motors.

Pin Diagram of a L293D Motor Driver IC Controller

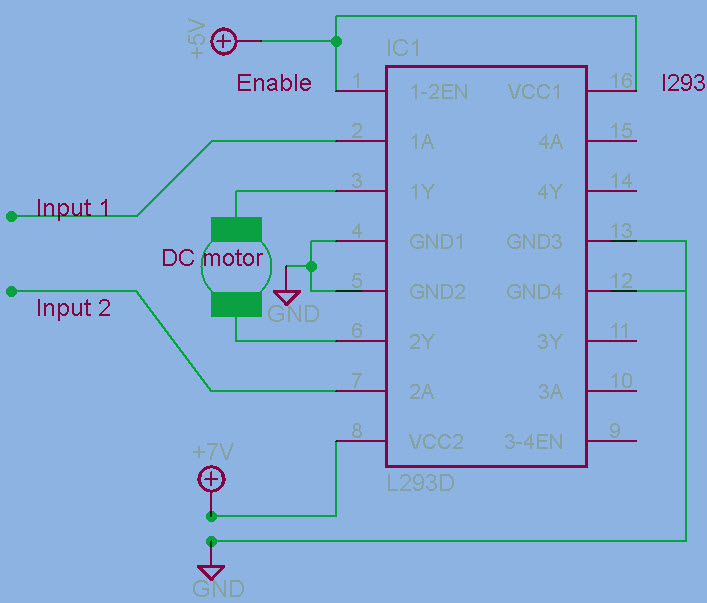
[](https://www.elprocus.com/wp-content/uploads/2015/03/L293D-IC-Pin-Configuration.jpg)

L293D IC Pin Configuration

* Pin-1 (Enable 1-2):  When the enable pin is high, then the left part of the IC will work otherwise it won’t work. This pin is also called as a master control pin.
* Pin-2 (Input-1):  When the input pin is high, then the flow of current will be through output 1
* Pin-3 (Output-1): This output-1 pin must be connected to one of the terminals of the motor
* Pin4 &5: These pins are ground pins
* Pin-6 (Output-2):  This pin must be connected to one of the terminals of the motor.
* Pin-7 (Input-2): When this pin is HIGH then the flow of current will be though output 2
* Pin-8 (Vcc2): This is the voltage pin which is used to supply the voltage to the motor.
* Pin-16 (Vss): This pin is the power source to the integrated circuit.
* Pin-15 (Input-4): When this pin is high, then the flow of current will be through output-4.
* Pin-14 (Output-4): This pin must be connected to one of the terminals of the motor
* Pin-12 & 13: These pins are ground pins
* Pin-11 (Output-3): This pin must be connected to one of the terminals of the motor.
* Pin-10 (Input-3): When this pin is high, then the flow of current will through output-3
* Pin-9 (Enable3-4): When this pin is high, then the right part of the IC will work & when it is low the right part of the IC won’t work. This pin is also called as a master control pin for the right part of the IC.

H Bridge Motor Control Circuit Using L293d IC

The IC LM293D consists of 4-i/p pins where, pin2 and 7 on the left side of the IC and Pin 10 and 15 on the right side of the IC. Left input pins on the IC will control the rotation of a motor. Here, the motor is connected across side and right i/p for the motor on the right hand side. This motor rotates based on the i/ps we provided across the input pins as Logic 0 and Logic 1.

[](https://www.elprocus.com/wp-content/uploads/2015/03/H-bridge-Motor-Circuit-with-L293D-IC.jpg)

H-bridge Motor Circuit with L293D IC

Let’s consider, when a motor is connected to the o/p pins 3 and 6 on the left side of the IC. For rotating of the motor in clockwise direction, then the i/p pins have to be provided with Logic 0 and Logic 1.

When Pin-2= logic 1 & pin-7 = logic 0, then it rotates in clockwise direction.  
Pin-2=logic 0 & Pin7=logic 1, then it rotates in anti clock direction  
Pin-2= logic 0 & Pin7=logic 0, then it is idle (high impedance state)  
Pin-2= logic 1 & Pin7=logic 1, then it is idle

In a similar way the motor can also operate across input pin-15 and pin-10 for the motor on the right hand side.

The L4293D motor driver IC deals with huge currents, due to this reason, this circuit uses a heat sink to decrease the heat. Therefore, there are 4-ground pins on  the L293D IC. When we solder these pins on the PCB (printed circuit board), then we can get a huge metallic area between the ground pins where the heat can be produced.

**3.4.5 SERVO MOTOR**



SERVO MOTOR

A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, the angular position of the shaft changes. [8] In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

Servo Motor Servos are extremely useful in robotics. The motors are small, have built-in control circuitry, and are extremely powerful for their size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, does not consume much energy. The guts of a servo motor is shown in the following picture. You can see the control circuitry, the motor, a set of gears, and the case. You can also see the 3 wires that connect to the outside world. One is for power (+5volts), ground, and the white wire is the control wire.

2.3.1Working of a Servo Motor

The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) connected to the output shaft. In the picture above, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor until it is at a desired angle. The output shaft of the servo is capable of traveling somewhere around 180 degrees. Usually, it is somewhere in the 210-degree range, however, it varies depending on the manufacturer. A normal servo is used to control an angular motion of 0 to 180 degrees. It is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear. The power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control.

How Do You Communicate the Angle at Which the Servo Should Turn?

The control wire is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse Coded Modulation. The servo expects to see a pulse every 20 milliseconds (.02 seconds). The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90-degree position (often called as the neutral position). If the pulse is shorter than 1.5 milliseconds, then the motor will turn the shaft closer to 0 degrees. If the pulse is longer than 1.5 milliseconds, the shaft turns closer to 180 degrees.

**JOYSTICK MODULE**

This is a dual axis high quality JoyStick Module . It can be used to sense movements in 2 directions(axes). It also has a inbuilt switch which can be activated by pressing the stick.Directional movements are simply two potentiometers - one for each axis. Pots are ~10k each.

With the help of this Joystick Module, you can measure position coordinates on the X and Y axis by moving the "hat". It also contains a switch that is press-able by pushing the "hat".It also contains a switch that is press-able by pushing the "hat" down. Similar to the XBOX controller.

The X and Y axes are two 10k potentiometers which control 2D movement by generating analog signals. When the module is in working mode, it will output two analog values, representing two directions. This module uses the 5V power supply, and value, when reading through analog input, would be about 2.5V, a value will increase with joystick movement and will go up till maximum 5V; the value will decrease when the joystick is moved in other direction till 0V.



JOYSTICK MODULE

**Specifications and Features:-**

* Dimensions: 40 x 27 x 15 (LxWxH) mm
* Weight: 10gm (without Hat).
* 2.54mm pin interface leads
* Operating Voltage: 5V.
* Long service life and stable performance
* Standard interface and electronic building blocks
* Widely use in Arduino DIY projects
* Cross rocker as a two-way 10K resistor, with the rocker in a different direction

**RESISTORS**

The resistor's function is to reduce the flow of electric current. There are two classes of resistors; fixed resistors and the variable resistors. They are also classified according to the material from which they are made.

The typical resistor is made of either carbon film or metal film. There are other types as well, but these are the most common. The resistance value of the resistor is not the only thing to consider when selecting a resistor for use in a circuit. The "tolerance" and the electric power ratings of the resistor are also important.

The tolerance of a resistor denotes how close it is to the actual rated resistance value. For example, a ±5% tolerance would indicate a resistor that is within ±5% of the specified resistance value.

**3.5.4.1 TYPES**

* Fixed resistors
* Variable Resistors

**Fixed Resistors**-A fixed resistor is one in which the value of its resistance cannot change. Such as Carbon film resistors.

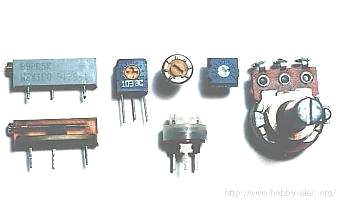
**Carbon film resistors**-This is the most general purpose, cheap resistor. Usually the tolerance of the resistance value is ±5%. Power ratings of 1/8W, 1/4W and 1/2W are frequently used.Carbon film resistors have a disadvantage; they tend to be electrically noisy. Metal film resistors are recommended for use in analog circuits.



Figure(14) Fixed Resistors

**Variable Resistors**-There are two general ways in which variable resistors are used. One is the variable resistor which value is easily changed, like the volume adjustment of Radio. The other is semi-fixed resistor that is not meant to be adjusted by anyone but a technician.

It is used to adjust the operating condition of the circuit by the technician. Semi-fixed resistors are used to compensate for the inaccuracies of the resistors, and to fine-tune a circuit. The rotation angle of the variable resistor is usually about 300 degrees. Some variable resistors must be turned many times to use the whole range of resistance they offer. This allows for very precise adjustments of their value. These are called "Potentiometers" or "Trimmer Potentiometers."

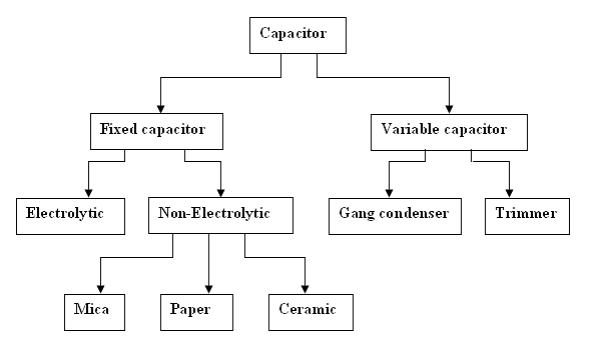


Figure(15) Variable Resistors

**CAPACITORS**

The capacitor's function is to store electricity, or electrical energy. The capacitor also functions as a filter, passing alternating current (AC), and blocking direct current (DC). This symbol ‘F’ is used to indicate a capacitor The capacitor is constructed with two electrode plates facing each other, but separated by an insulator. When DC voltage is applied to the capacitor, *an electric charge* is stored on each electrode. While the capacitor is charging up, current flows. The current will stop flowing when the capacitor has fully charged.

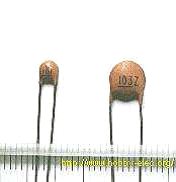
**3.5.5.1 TYPES OF CAPACITOR:**



Figure(16) Types of capacitor

**1. Electrolytic Capacitors (Electrochemical type capacitors)-**Aluminum is used for the electrodes by using a thin oxidization membrane. Large values of capacitance can be obtained in comparison with the size of the capacitor, because the dielectric used is very thin. The most important characteristic of electrolytic capacitors is that they have polarity. They have a positive and a negative electrode.

**2. Ceramic Capacitors-**Ceramic capacitors are constructed with materials such as titanium acid barium used as the dielectric. Internally, these capacitors are not constructed as a coil, so they can be used in high frequency applications. Typically, they are used in circuits which bypass high frequency signals to ground. These capacitors have the shape of a disk. Their capacitance is comparatively small.. The diameter of the disk is about 6 mm. Ceramic capacitors have no polarity. Ceramic capacitors should not be used for analog circuits, because they can distort the signal.



Ceramic Capacitors

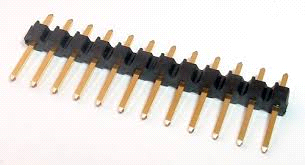
**3.Variable Capacitors-** Variable capacitors are used for adjustment etc. of frequency mainly. On the left in the photograph is a "trimmer," which uses ceramic as the dielectric. Next to it on the right is one that uses polyester film for the dielectric. The pictured components are meant to be mounted on a printed circuit board.



Variable Capacitor

**3.5.7 APPLIANCE CONNECTOR**

Berg strip male and female come with a predefined number of pins. The designer then cut off only the required number of pins and assembles to the circuit board. It can act as wire-to-board connector also; we can solder the wire to either of the connector pins and fix the other connector to the circuit board. When we solder wires to the pin of both the pair it can function as a wire-to-wire connector also. Normally they are used as board-to-board connectors with the male connector are soldered at the periphery of the circuit board and the female connector is assembled wherever another board is intending to be connected.

MALE CONNECTOR FEMALE CONNETOR

Connectors

They are designed for high reliability and quick assembly. They are also very cheap connectors. The Berg strips are used to carry both power and data. Berg strip found its application in almost all kind of circuit modules, like GSM module, GPS module, Xbee module, LCD module and in all kind of circuit boards.

**INDUCTOR**

An inductor, also called a coil, choke, or reactor, is a [passive](https://en.wikipedia.org/wiki/Incremental_passivity) [two-terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [electrical component](https://en.wikipedia.org/wiki/Electronic_component) that stores energy in a [magnetic field](https://en.wikipedia.org/wiki/Magnetic_field) when [electric current](https://en.wikipedia.org/wiki/Electric_current) flows through it.[[1]](https://en.wikipedia.org/wiki/Inductor#cite_note-1) An inductor typically consists of an insulated wire wound into a [coil](https://en.wikipedia.org/wiki/Electromagnetic_coil) around a core.

When the current flowing through an inductor changes, the time-varying magnetic field induces an [electromotive force](https://en.wikipedia.org/wiki/Electromotive_force) (*e.m.f.*) ([voltage](https://en.wikipedia.org/wiki/Voltage)) in the conductor, described by [Faraday's law of induction](https://en.wikipedia.org/wiki/Faraday%27s_law_of_induction). According to [Lenz's law](https://en.wikipedia.org/wiki/Lenz%27s_law), the induced voltage has a polarity (direction) which opposes the change in current that created it. As a result, inductors oppose any changes in current through them.

An inductor is characterized by its [inductance](https://en.wikipedia.org/wiki/Inductance), which is the ratio of the voltage to the rate of change of current. In the [International System of Units](https://en.wikipedia.org/wiki/International_System_of_Units) (SI), the unit of inductance is the [henry](https://en.wikipedia.org/wiki/Henry_(unit)) (H) named for 19th century American scientist [Joseph Henry](https://en.wikipedia.org/wiki/Joseph_Henry). In the measurement of magnetic circuits, it is equivalent to [weber](https://en.wikipedia.org/wiki/Weber_(unit))/[ampere](https://en.wikipedia.org/wiki/Ampere). Inductors have values that typically range from 1 µH (10−6 H) to 20 H. Many inductors have a [magnetic core](https://en.wikipedia.org/wiki/Magnetic_core) made of iron or [ferrite](https://en.wikipedia.org/wiki/Ferrite_(magnet)) inside the coil, which serves to increase the magnetic field and thus the inductance. Along with [capacitors](https://en.wikipedia.org/wiki/Capacitor) and [resistors](https://en.wikipedia.org/wiki/Resistor), inductors are one of the three passive [linear](https://en.wikipedia.org/wiki/Linear_circuit) [circuit elements](https://en.wikipedia.org/wiki/Circuit_element) that make up electronic circuits. Inductors are widely used in [alternating current](https://en.wikipedia.org/wiki/Alternating_current) (AC) electronic equipment, particularly in [radio](https://en.wikipedia.org/wiki/Radio) equipment. They are used to block AC while allowing DC to pass; inductors designed for this purpose are called [chokes](https://en.wikipedia.org/wiki/Choke_(electronics)). They are also used in [electronic filters](https://en.wikipedia.org/wiki/Electronic_filter) to separate signals of different [frequencies](https://en.wikipedia.org/wiki/Frequency), and in combination with capacitors to make [tuned circuits](https://en.wikipedia.org/wiki/Tuned_circuit), used to tune radio and TV receivers.



TYPES OF INDUCTORS

**POWER SUPPLY**

A power supply is an electrical device that supplies [electric power](https://en.wikipedia.org/wiki/Electric_power) to an [electrical load](https://en.wikipedia.org/wiki/Electrical_load). The primary function of a power supply is to convert [electric current](https://en.wikipedia.org/wiki/Electric_current) from a source to the correct [voltage](https://en.wikipedia.org/wiki/Voltage), [current](https://en.wikipedia.org/wiki/Electric_current), and [frequency](https://en.wikipedia.org/wiki/Frequency) to power the load. As a result, power supplies are sometimes referred to as [electric power converters](https://en.wikipedia.org/wiki/Electric_power_converter). Some power supplies are separate standalone pieces of equipment, while others are built into the load appliances that they power. Examples of the latter include power supplies found in [desktop computers](https://en.wikipedia.org/wiki/Desktop_computer) and  electronic devices. Other functions that power supplies may perform include limiting the current drawn by the load to safe levels, shutting off the current in the event of an [electrical fault](https://en.wikipedia.org/wiki/Electrical_fault), power conditioning to prevent [electronic noise](https://en.wikipedia.org/wiki/Electronic_noise) or [voltage surges](https://en.wikipedia.org/wiki/Voltage_surge) on the input from reaching the load, [power-factor correction](https://en.wikipedia.org/wiki/Power-factor_correction), and storing energy so it can continue to power the load in the event of a temporary interruption in the source power ([uninterruptible power supply](https://en.wikipedia.org/wiki/Uninterruptible_power_supply)).

All power supplies have a *power input* connection, which receives energy in the form of electric current from a source, and one or more *power output* connections that deliver current to the load. The source power may come from the [electric power grid](https://en.wikipedia.org/wiki/Electric_power_grid), such as an [electrical outlet](https://en.wikipedia.org/wiki/Electrical_outlet), [energy storage](https://en.wikipedia.org/wiki/Energy_storage) devices such as [batteries](https://en.wikipedia.org/wiki/Battery_(electricity)) or [fuel cells](https://en.wikipedia.org/wiki/Fuel_cell), [generators](https://en.wikipedia.org/wiki/Electrical_generators) or [alternators](https://en.wikipedia.org/wiki/Alternator), [solar power](https://en.wikipedia.org/wiki/Solar_power) converters, or another power supply. The input and output are usually hardwired circuit connections, though some power supplies employ [wireless energy transfer](https://en.wikipedia.org/wiki/Wireless_power) to power their loads without wired connections. Some power supplies have other types of inputs and outputs as well, for functions such as external monitoring and control.

**PROGRAM CODE:**

**/\*nrf24l01-joy-rcv-demo.ino**

**nRF24L01+ Receiver with Joystick Decode**

**Use with Joystick Transmitter Demo**

**\*/**

**// Include RadioHead ReliableDatagram & NRF24 Libraries**

**#include <RHReliableDatagram.h>**

**#include <RH\_NRF24.h>**

**// Include dependant SPI Library**

**#include <SPI.h>**

**// Define addresses for radio channels**

**#define CLIENT\_ADDRESS 1**

**#define SERVER\_ADDRESS 2**

**// Create an instance of the radio driver**

**RH\_NRF24 RadioDriver;**

**// Sets the radio driver to NRF24 and the server address to 2**

**RHReliableDatagram RadioManager(RadioDriver, SERVER\_ADDRESS);**

**// Define a message to return if values received**

**uint8\_t ReturnMessage[] = "JoyStick Data Received";**

**// Define the Message Buffer**

**uint8\_t buf[RH\_NRF24\_MAX\_MESSAGE\_LEN];**

**void setup()**

**{**

**// Setup Serial Monitor**

**Serial.begin(9600);**

**// Initialize RadioManager with defaults - 2.402 GHz (channel 2), 2Mbps, 0dBm**

**if (!RadioManager.init())**

**Serial.println("init failed");**

**}**

**void loop()**

**{**

**if (RadioManager.available())**

**{**

**// Wait for a message addressed to us from the client**

**uint8\_t len = sizeof(buf);**

**uint8\_t from;**

**if (RadioManager.recvfromAck(buf, &len, &from))**

**//Serial Print the values of joystick**

**{**

**Serial.print("got request from : 0x");**

**Serial.print(from, HEX);**

**Serial.print(": X = ");**

**Serial.print(buf[0]);**

**Serial.print(" Y = ");**

**Serial.print(buf[1]);**

**Serial.print(" Z = ");**

**Serial.println(buf[2]);**

**// Send a reply back to the originator client, check for error**

**if (!RadioManager.sendtoWait(ReturnMessage, sizeof(ReturnMessage), from))**

**Serial.println("sendtoWait failed");**

**}**

**}**

**}**

**/\***

**nRF24L01+ Joystick Transmitter**

**nrf24l01-joy-xmit-car.ino**

**nRF24L01+ Transmitter with Joystick for Robot Car.0.jj**

**Use with Joystick Receiver for Robot Car**

**\*/**

**// Include RadioHead ReliableDatagram & NRF24 Libraries**

**#include <RHReliableDatagram.h>**

**#include <RH\_NRF24.h>**

**// Include dependant SPI Library +**

**#include <SPI.h>**

**// Define Joystick Connections**

**#define joyVert A0**

**#define joyHorz A1**

**// Define Joystick Values - Start at 512 (middle position)**

**int joyposVert = 512;**

**int joyposHorz = 512;**

**// Define addresses for radio channels**

**#define CLIENT\_ADDRESS 1**

**#define SERVER\_ADDRESS 2**

**// Create an instance of the radio driver**

**RH\_NRF24 RadioDriver;**

**// Sets the radio driver to NRF24 and the client address to 1**

**RHReliableDatagram RadioManager(RadioDriver, CLIENT\_ADDRESS);**

**// Declare unsigned 8-bit motorcontrol array**

**// 2 Bytes for motor speeds plus 1 byte for direction control**

**uint8\_t motorcontrol[3];**

**// Define the Message Buffer**

**uint8\_t buf[RH\_NRF24\_MAX\_MESSAGE\_LEN];**

**void setup()**

**{**

**// Setup Serial Monitor**

**Serial.begin(9600);**

**// Initialize RadioManager with defaults - 2.402 GHz (channel 2), 2Mbps, 0dBm**

**if (!RadioManager.init())**

**// Serial.println("init failed");**

**// Set initial motor direction as forward**

**motorcontrol[2] = 0;**

**}**

**void loop()**

**{**

**// Print to Serial Monitor**

**Serial.println("Reading motorcontrol values ");**

**// Read the Joystick X and Y positions**

**joyposVert = analogRead(joyVert);**

**joyposHorz = analogRead(joyHorz);**

**// Determine if this is a forward or backward motion**

**// Do this by reading the Verticle Value**

**// Apply results to MotorSpeed and to Direction**

**if (joyposVert < 460)**

**{**

**// This is Backward**

**// Set Motors backward**

**motorcontrol[2] = 1;**

**//Determine Motor Speeds**

**// As we are going backwards we need to reverse readings**

**motorcontrol[0] = map(joyposVert, 460, 0, 0, 255);**

**motorcontrol[1] = map(joyposVert, 460, 0, 0, 255);**

**}**

**else if (joyposVert > 564)**

**{**

**// This is Forward**

**// Set Motors forward**

**motorcontrol[2] = 0;**

**//Determine Motor Speeds**

**motorcontrol[0] = map(joyposVert, 564, 1023, 0, 255);**

**motorcontrol[1] = map(joyposVert, 564, 1023, 0, 255);**

**}**

**else**

**{**

**// This is Stopped**

**motorcontrol[0] = 0;**

**motorcontrol[1] = 0;**

**motorcontrol[2] = 0;**

**}**

**// Now do the steering**

**// The Horizontal position will "weigh" the motor speed**

**// Values for each motor**

**if (joyposHorz < 460)**

**{**

**// Move Left**

**// As we are going left we need to reverse readings**

**// Map the number to a value of 255 maximum**

**joyposHorz = map(joyposHorz, 460, 0, 0, 255);**

**motorcontrol[0] = motorcontrol[0] - joyposHorz;**

**motorcontrol[1] = motorcontrol[1] + joyposHorz;**

**// Don't exceed range of 0-255 for motor speeds**

**if (motorcontrol[0] < 0)motorcontrol[0] = 0;**

**if (motorcontrol[1] > 255)motorcontrol[1] = 255;**

**}**

**else if (joyposHorz > 564)**

**{**

**// Move Right**

**// Map the number to a value of 255 maximum**

**joyposHorz = map(joyposHorz, 564, 1023, 0, 255);**

**motorcontrol[0] = motorcontrol[0] + joyposHorz;**

**motorcontrol[1] = motorcontrol[1] - joyposHorz;**

**// Don't exceed range of 0-255 for motor speeds**

**if (motorcontrol[0] > 255)motorcontrol[0] = 255;**

**if (motorcontrol[1] < 0)motorcontrol[1] = 0;**

**}**

**// Adjust to prevent "buzzing" at very low speed**

**if (motorcontrol[0] < 8)motorcontrol[0] = 0;**

**if (motorcontrol[1] < 8)motorcontrol[1] = 0;**

**//Display the Motor Control values in the serial monitor.**

**Serial.print("Motor A: ");**

**Serial.print(motorcontrol[0]);**

**Serial.print(" - Motor B: ");**

**Serial.print(motorcontrol[1]);**

**Serial.print(" - Direction: ");**

**Serial.println(motorcontrol[2]);**

**//Send a message containing Motor Control data to manager\_server**

**if (RadioManager.sendtoWait(motorcontrol, sizeof(motorcontrol), SERVER\_ADDRESS))**

**{**

**// Now wait for a reply from the server**

**uint8\_t len = sizeof(buf);**

**uint8\_t from;**

**if (RadioManager.recvfromAckTimeout(buf, &len, 2000, &from))**

**{**

**Serial.print("got reply from : 0x");**

**Serial.print(from, HEX);**

**Serial.print(": ");**

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

**}**

**else**

**{**

**Serial.println("No reply, is nrf24\_reliable\_datagram\_server running?");**

**}**

**}**

**else**

**Serial.println("sendtoWait failed");**

**// \(10); // Wait a bit before next transmission**

**}**

**CHAPTER 4:**

**CONCLUSION AND FUTURE SCOPE**

**4.1 CONCLUSION**

The surveillance robot will be designed to deliver a reasonable level of efficiency and simplicity, providing each user with a streamlined user experience. The surveillance robot is aimed at providing monitoring inclusive of vision, motion, fire, and carbon monoxide with limited setup. The surveillance robot can be customized to fuse seamlessly to any home, apartments or multi-dwelling units. Based on modular designs and complete scalability, the surveillance robot is designed to be expandable and allow for future home control upgrades, thus enhancing the protection of your home as time and lifestyles change.

**4.2 FUTURE SCOPE**

1.The most important use of this project will be for surveillance.

2.It can be used to infiltrate at various locations where it would be hard for a human to

infiltrate without getting noticed.

3.Can be used for household surveillance.

4.Can be used by traffic security to assess fast moving cars

5.Drone can be detected using radar system whereas it is hard for a land rover to be detected

while it is moving on land.

6.Can be used to hijacked areas/buildings.

7.Can be used under tunnel with introduction of sensors like gas and pollution sensor.

**REFERENCE**

1. Sridhar Lakshmanan, Vin Joe Varghese, Narasimhamurthi Natarajan. “Miniature surveillance robot” (<http://www.google.com/patents/US7436143>).

2. <https://www.electronicshub.org/arduino-line-follower-robot>.

3.https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/

4. <https://www.elprocus.com/h-bridge-motor-control-circuit-using-l293d-ic/>

5. <https://create.arduino.cc/projecthub/aritro/smoke-detection-using-mq-2-gas-sensor-79c54a>