**INTELLIGENT AMBULANCE**

**WITH TRAFFIC CONTROL SYSTEM**

*A Project Report Submitted in partial fulfilment of the Academic Requirement for the Award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**Submitted by**

**JANAKI RESHMA PRIYA KOLLIPARA   (17AR1A0476)**

**ASWANI MEDIKONDA                                      (17AR1A0496)**

**BHARGAVI MEKALA                                          (17AR1A0497)**

**NAGENDRA BABU MIRIYALA                           (17AR1A0499)**

**SIVALEELA NELAKUDITI                                     (17AR1A04B0)**

## Department of Electronics & Communication Engineering

## 

## 

**SAI TIRUMALA NVR ENGINEERING COLLAGE**

### NARASARAOPET – 522601.

**(**An **ISO 9001-2008** Certified Institution Affiliated to J.N.T. University, KAKINADA, Approved by AICTE)

Jonnalagadda, Guntur Road, Narasaraopet (MD),Guntur (DIST), Andhra Pradesh-

522601 2017-2021

**SAI TIRUMALA NVR ENGINEERING COLLEGE**

**(An ISO 9001-2008 Certified Institution Affiliated to J.N.T. University, KAKINADA, Approved by AICTE)**

**DEPARTMENT OF**

**ELECTRONICS & COMMUNICATION**

**ENGINEERING**

CERTIFICATE

This is to certify that the Project entitled ―**“INTELLIGENT AMBULANCE WITH TRAFFIC CONTROL SYSTEM**”“is the bonfire project work carried out the student **K.J. RESHMA PRIYA, M.ASWANI, M.BHARGAVI, M.NAGENDRABABU,N.SIVALEELA,** bearing RollNo**17AR1A0476, 17AR1A0496, 17AR1A0497, 17AR1A0499,17AR1A04B0** under the supervision, guidance and submitted in partial fulfillment of the requirements for the award of the degree of- “**Bachelor Technology “of JNTUKAKINADA**, during the year **2017-2021**

##### Internal Guide Head of the Department K.PEDARAJU.M TECH B.BALOJI NAIK M.TECH

**SAI TIRUMALA NVR ENGINEERING COLLEGE**

**(An ISO 9001-2008 Certified Institution Affiliated to J.N.T. University, KAKINADA, Approved by AICTE)**

**DEPARTMENT OF**

**ELECTRONICS & COMMUNICATION**

**ENGINEERING**

**DE****CLARATION**

We are the students of **SAI TIRUMALA NVR ENGINEERING COLLEGE** hereby declare that this dissertation entitled ―**“** INTELLIGENT AMBULANCE WITH TRAFFIC CONTROL SYSTEM**”** Being submitted to the Department of *ELECTRONICS AND COMMUNICATIONENGINEERING****,****SAI TIRUMALA NVR* ***ENGINEERING COLLEGE*** Accredited by AICTE, affiliated to **JNTU KAKINADA** for the award of **Bachelor of Technology**, is a record of confide work done by me and it has not been submitted to any other Institute or University for the award of any other degree or prize.

*BY*

**JANAKI RESHMA PRIYA KOLLIPARA   (17AR1A0476)**

**ASWANI MEDIKONDA                                 (17AR1A0496)**

**BHARGAVI MEKALA                                     (17AR1A0497)**

**NAGENDRA BABU MIRIYALA                      (17AR1A0499)**

**SIVALEELA NELAKUDITI                                (17AR1A04B0)**

**ACKNOWLEDGEMENT**

We express sincere thanks to various who were responsible for the successful completion of this project.

We express our deep felt gratitude to the management of **SAI TIRUMALA NVR ENGINEERING COLLEGE** for helping us in successful completion of the project.

We express our sincere thanks to guide **K.PEDARAJU M.TECH** and Head of the Department **B.BALOJI NAIK M.TECH** for his valuable guidance and constant encouragement which enabled us to accomplish our project successfully in time. His vast experience, profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We are graceful to **Dr. RAM MOHANRAO,** PRINCIPAL for providing us all the necessary facilities for the completion of this project in specified time.

We express our earnest thanks to other faculty members of ECE for extending their helping hands and valuable suggestions when in need.

We express our special thanks to **Mr. A.VEERAIAH** and all the library staff of **SAI TIRUMALA NVR ENGINEERING COLLEGE**, for providing the necessary library facilities.

Last but not the least, we express our heartfelt thanks to all my friends to complete the project successfully.

**BY**

**JANAKI RESHMA PRIYA KOLLIPARA (17AR1A0476)**

**ASWANI MEDIKONDA (17AR1A0496)**

**BHARGAVI MEKALA (17AR1A0497)**

**NAGENDRABABU MIRIYALA (17AR1A0499)**

**SIVALEELA NELAKUDITI (17AR1A04B0)**

Index

**ABSTRACT**

Now-a-days many services get delayed due to traffic jams especially in large cities. Ambulance service is one of the crucial services that get delayed. Sometimes on-sight doctors are not available, so the patient does not get medical attention immediately. To overcome this situation this paper describes a solution that is 'Intelligent Ambulance with Traffic Control' which includes a traffic control system. In a traffic control system an RF transmitter on the ambulance will communicate with the RF receiver mounted on the signal post. An algorithm is used to control the traffic signals automatically based on the key pressed by the driver from the keyboard in the ambulance. The information reading the current as well as future location of the ambulance is sent from the ambulance itself. This information is used to optimally control the traffic**. RFID** reads the information. If the traffic is high then the information is passed to the nearby junction by indicating the green light.

**CHAPTER 1**

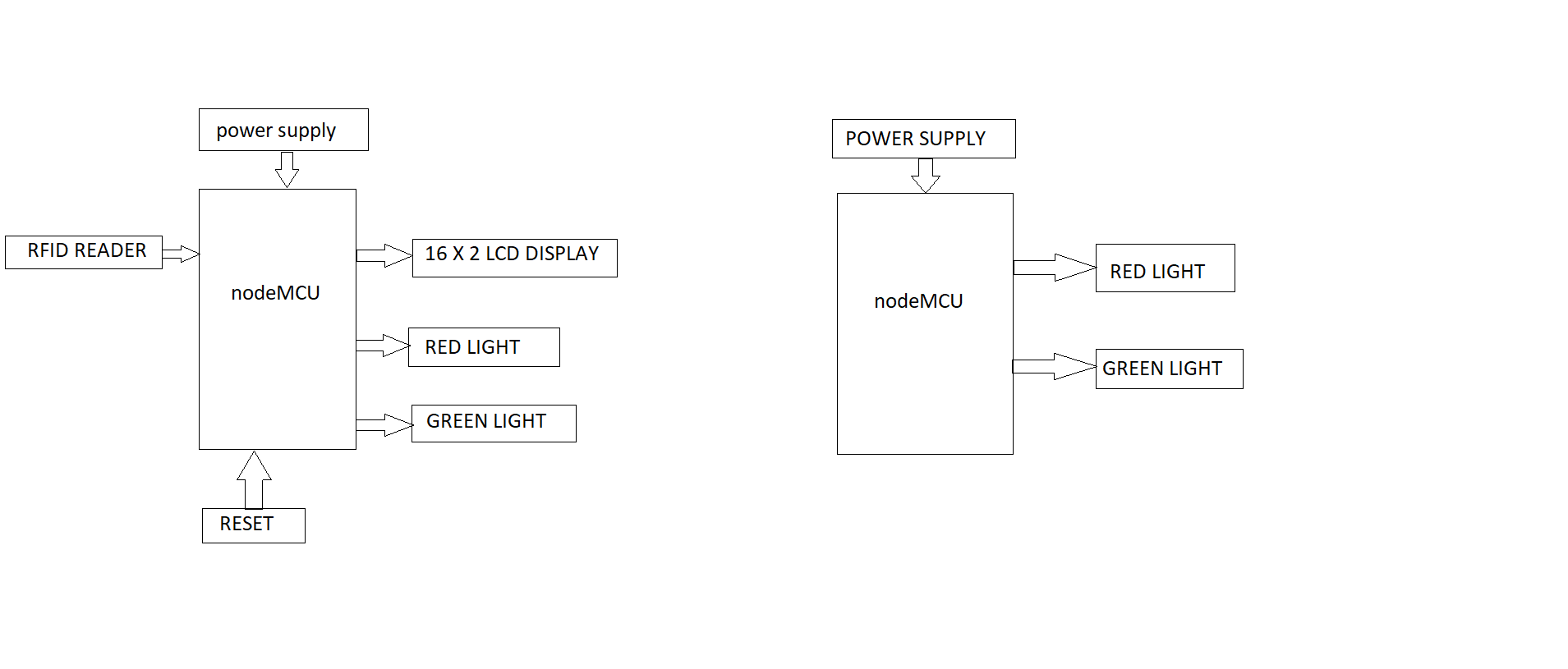
INTRODUCTION

This particular project is designed for the cities with heavy traffic .Eg: In Bangalore the roads are full jammed every time. Most of the time the traffic will at least for 100meters.In this distance the traffics police can’t hear the siren form the ambulance .so he ignores this Then the ambulance has to wait till the traffic is left. Some times to leave the traffic it takes at least 30 minutes .So by this time anything can happen to the patient .So this project avoid these disadvantages. According to this project if any ambulance comes near when the ambulance at emergency comes to any traffic post the traffic signals automatically stop the signals and give green signal for this ambulance. When the ambulance at emergency comes to any traffic post the traffic signals automatically stop the signal. The road accidents in modern urban areas are increased to uncertain level. The loss of human life due to accident is to be avoided. Traffic congestion and tidal flow are major facts that cause delay to ambulance. To bar loss of human life due to accidents we introduce a scheme called IATCS (Intelligent Ambulance Traffic control system). The main theme behind this scheme is to provide a smooth flow for the emergency vehicles like ambulance to reach the hospitals in time and thus minimizing the delay caused by traffic congestion. The idea behind this scheme is to implement IATCS which would control mechanically the traffic lights in the path of the ambulance. The ambulance is controlled by the control unit which furnishes adequate route to the ambulance and also controls the traffic light according to the ambulance location and thus reaching the hospital safely. The controller identifies the location of the accident spot through the sensor systems in the vehicle which determined the accident and thus the controller walks through the ambulance to the spot. This scheme is fully automated, thus it finds the accident spot, controls the traffic lights, helping to reach the hospital in time

**CHAPTER 2**

**BLOCK DIAGRAM DESCRIPTION**

**2.1 BLOCK DIAGRAM:**



**Fig no 2.1:** Block diagram of intelligence ambulance

**1.2 WORKING PRINCIPLE**  : The ambulance carries an IR transmitter and IR receiver will be there some few meter before the signal. The receiver will receive the signal and the module will send the command turn on green through the RF and every traffic post will have an RF receiver. So whenever the ambulance comes near the traffic, the ambulance will transmit, the receiver will receive this signal .Then it immediately switch off the other signals that is it make all the signals red and later make this particular direction signal green.

**CHAPTER 3**

**SYSTEM SPECIFICATION**

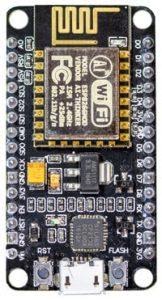
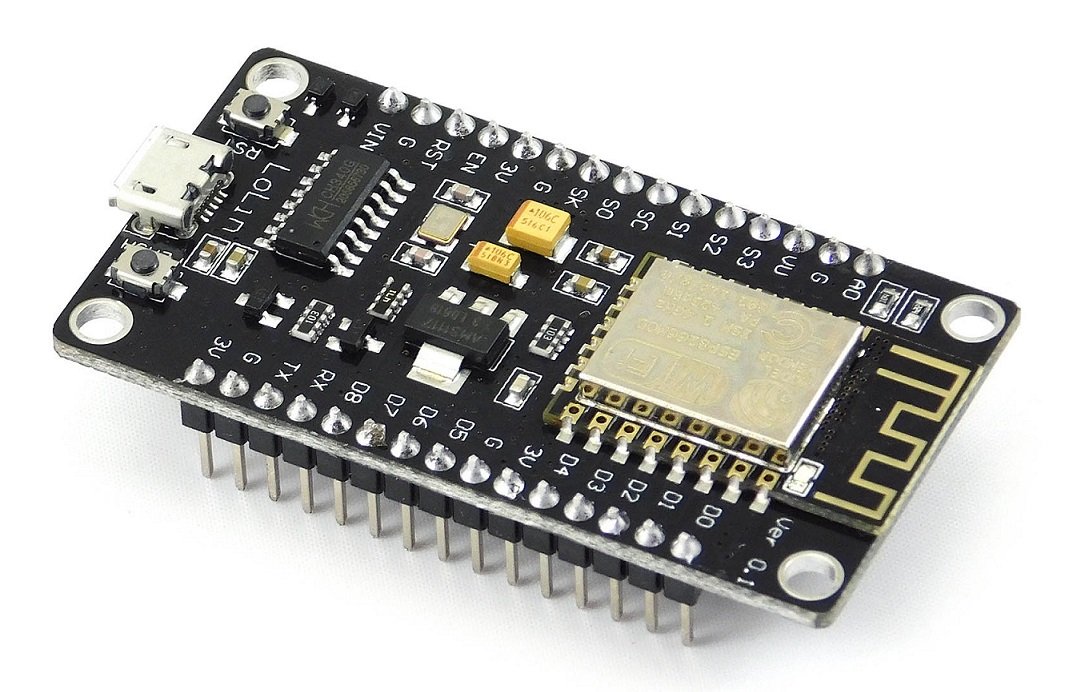
* 1. **COMPONENT SPECIFICATION**

* NodeMCU
* RFID READER EMI8
* LCD DISPLAY/OLED DISPLAY
* LED
* PCB READERS
* POWER 5V-20AMP
* BUTTONS
* RF TRANSMITER RECIEVER

**3.2 SOFTWARE USED**

* Keil software for  c programming
* Express PCB for lay out design
* Express SCH for schematic design

**NodeMCU**

****

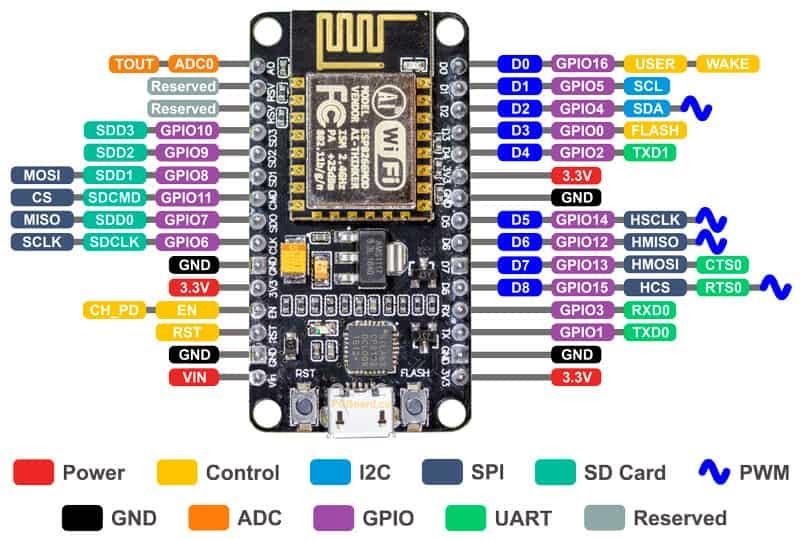
The NodeMCU (*N*ode *M*icro*C*ontroller *U*nit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espress if Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for the Internet of Things (IoT) projects of all kinds.

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

But, what about Arduino? The Arduino project created an open-source hardware design and software SDK for their versatile IoT controller. Similar to NodeMCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike NodeMCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

## NodeMCU Pin out and Functions Explained

****

* Power Pins There are four power pins. VIN pin and three 3.3V pins.

VIN can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the NodeMCU module – you can also supply 5V regulated to the VIN pin

3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components.

* GND are the ground pins of NodeMCU/ESP8266
* I2C Pins are used to connect I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.
* GPIO Pins NodeMCU/ESP8266 has 17 GPIO pins which can be assigned to functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.
* ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.
* UART Pins NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

* SPI Pins NodeMCU/ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

4 timing modes of the SPI format transfer

Up to 80 MHz and the divided clocks of 80 MHz

Up to 64-Byte FIFO

* SDIO Pins NodeMCU/ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

* PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs (100 Hz and 1 kHz).

* Control Pins are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

EN: The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.

RST: RST pin is used to reset the ESP8266 chip.

WAKE: Wake pin is used to wake the chip from deep-sleep.

* Tiny Sine WaveControl Pins are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

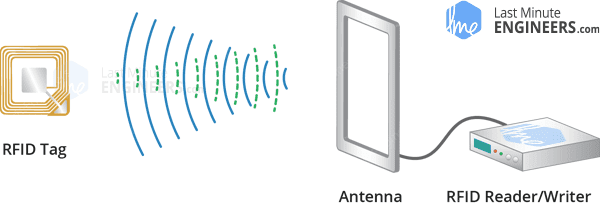
EN: The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.

RST: RST pin is used to reset the ESP8266 chip.

WAKE: Wake pin is used to wake the chip from deep-sleep.

**RFID READER EMI 8**

RFID or [Radio Frequency Identification](https://en.wikipedia.org/wiki/Radio-frequency_identification) system consists of two main components, a transponder/tag attached to an object to be identified, and a Transceiver also known as interrogator/Reader.



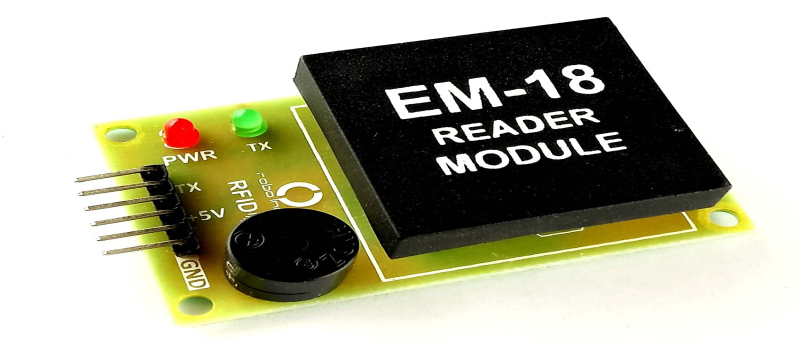
A Reader consists of a Radio Frequency module and an antenna which generates high frequency electromagnetic field. On the other hand, the tag is usually a passive device, meaning it doesn’t contain a battery. Instead it contains a microchip that stores and processes information, and an antenna to receive and transmit a signal.

To read the information encoded on a tag, it is placed in close proximity to the Reader (does not need to be within direct line-of-sight of the reader). A Reader generates an electromagnetic field which causes electrons to move through the tag’s antenna and subsequently power the chip.

The powered chip inside the tag then responds by sending its stored information back to the reader in the form of another radio signal. This is called backscatter. The backscatter, or change in the electromagnetic/RF wave, is detected and interpreted by the reader which then sends the data out to a computer or microcontroller.

**Hardware Overview – RC522 RFID Reader/Writer Module**

The RC522 RFID module based on [MFRC522 IC from NXP](https://www.nxp.com/products/identification-and-security/nfc/nfc-reader-ics/standard-performance-mifare-and-ntag-frontend:MFRC52202HN1) is one of the most inexpensive RFID options that you can get online for less than four dollars. It usually comes with a RFID card tag and key fob tag having 1KB memory. And best of all, it can write a tag, so you can store your some sort of secret message in it.





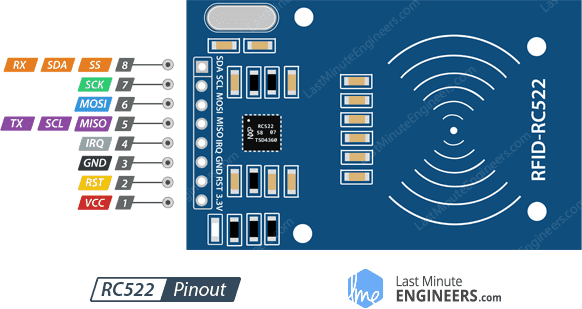
The RC522 RFID Reader module is designed to create a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags (ISO 14443A standard tags). The reader can communicate with a microcontroller over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. It also supports communication over I2C and UART protocols.

The module comes with an interrupt pin. It is handy because instead of constantly asking the RFID module “is there a card in view yet? “, the module will alert us when a tag comes into its vicinity.

The operating voltage of the module is from 2.5 to 3.3V, but the good news is that the logic pins are 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

## RC522 RFID Module Pinout

The RC522 module has total 8 pins that interface it to the outside world. The connections are as follows:



VCC supplies power for the module. This can be anywhere from 2.5 to 3.3 volts. You can connect it to 3.3V output from your Arduino. Remember connecting it to 5V pin will likely destroy your module!

RST is an input for Reset and power-down. When this pin goes low, hard power-down is enabled. This turns off all internal current sinks including the oscillator and the input pins are disconnected from the outside world. On the rising edge, the module is reset.

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

IRQ is an interrupt pin that can alert the microcontroller when RFID tag comes into its vicinity.

MISO / SCL / Tx pin acts as Master-In-Slave-Out when SPI interface is enabled, acts as serial clock when I2C interface is enabled and acts as serial data output when UART interface is enabled.

MOSI (Master Out Slave In) is SPI input to the RC522 module.

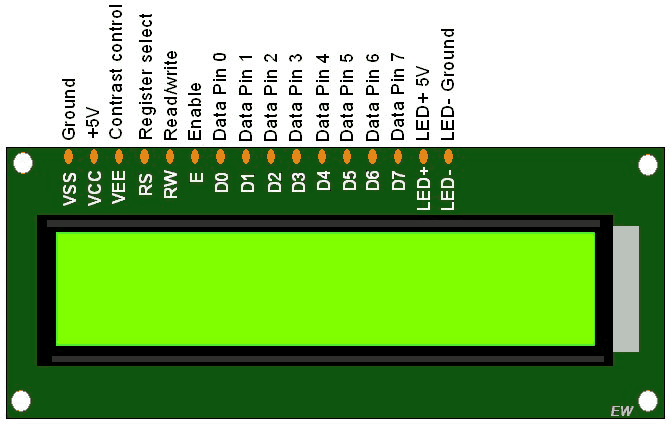
SCK (Serial Clock) accepts clock pulses provided by the SPI bus Master i.e. Arduino.

SS / SDA / Rx pin acts as Signal input when SPI interface is enabled, acts as serial data when I2C interface is enabled and acts as serial data input when UART interface is enabled. This pin is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins.

**LCD DISPLAY**

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Since LCD panels produce no light of their own, they require external light to produce a visible image. LCD uses a liquid crystal to produce a visible image. Liquid Crystal Display is very important to check the status of any automated and semi automated devices. This can be done by displaying their status on a display module such as an LCD (Liquid Crystal Display). 16x2 LCD module is one of the most common devices on the market. The liquid crystal display has been replacing many other displays like 7 segment and others. This is because of the multiple benefits of LCDs i.e. that they are economical. It can be programmed easily and doesn’t have any limitations of displaying special and even custom characters. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

These **pins** are used to send data to the **display**. These **pins** are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four **pins** are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-**pins** are connected to microcontroller unit like 0 to 7.



**POWER SUPPLY**

Power supplies in recent times have greatly improved in reliability but, because they have to handle considerably higher voltages and currents than any or most of the circuitry they supply, they are often the most susceptible to failure of any part of an electronic system. Modern power supplies have also increased greatly in their complexity, and can supply very stable output voltages controlled by feedback systems. Many power supply circuits also contain automatic safety circuits to prevent dangerous over voltage or over current situations. A DC Power Supply Unit (commonly called a PSU) deriving power from the AC mains (line) supply performs a number of tasks: 1. It changes (in most cases reduces) the level of supply to a value suitable for driving the load circuit. 2. It produces a DC supply from a pure AC wave. 3. It prevents any AC from appearing at the supply output. 4. It will ensure that the output voltage is kept at a constant level, independent of changes in:

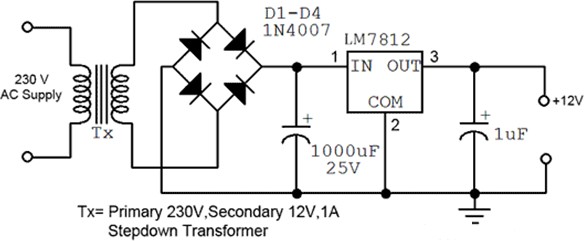
a. The AC supply voltage at the supply input.

b. The Load current drawn from the supply output.

c. Temperature.

Related image

**Fig 3.1: BLOCK DIAGRAM OF POWER SUPPLY**



**Fig 3.2: CIRCUIT DIAGRAM OF POWER SUPPLY**

### 

### RFID READER

Radio frequency identification reader (RFID reader) is a device used to gather information from an RFID tag, which is used to track individual objects. Radio waves are used to transfer data from the tag to a reader.

RFID is a technology similar in theory to bar codes. However, the RFID tag does not have to be scanned directly, nor does it require line-of-sight to a reader.

The RFID tag it must be within the range of an RFID reader, which ranges from 3 to 300 feet, in order to be read. RFID technology allows several items to be quickly scanned and enables fast identification of a particular product, even when it is surrounded by several other items.

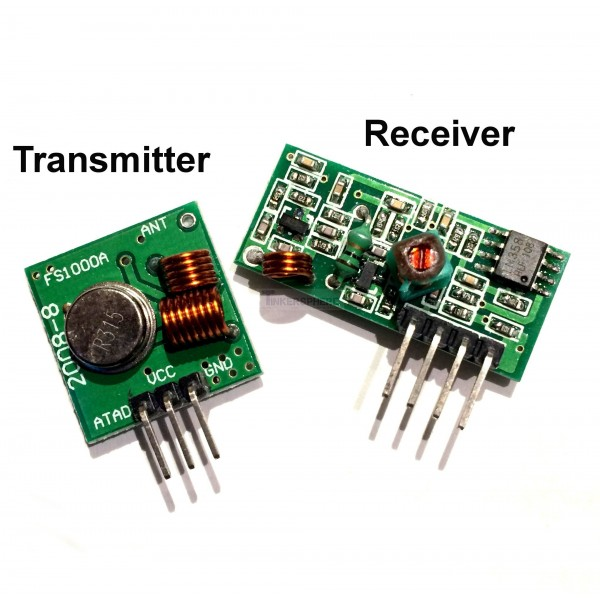
RFID tags have not replaced bar codes because of their cost and the need to individually identify every item.



**Fig. 3.3: RFID Reader Board**

RFID technology may be used in a variety of applications including:

* Passports
* Smart cards
* Airplane luggage
* Toll booth passes
* Home appliances
* Merchandise tags
* Animal and pet tags
* Automobile key-and-lock
* Monitoring heart patients
* Pallet tracking for inventory
* Telephone and computer network



**CHAPTER 4**

## SOFTWARE EXPLANATION

**4.1 ARDUINO IDE:**

Arduino is an open-source electronics platform based on easy-to-use hardware and

software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming](https://www.arduino.cc/en/Reference/HomePage) [language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE),](https://www.arduino.cc/en/Main/Software) based on [Processing](https://processing.org/).

The 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) [mac OS,](https://en.wikipedia.org/wiki/MacOS) [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in functions from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B_(programming_language)).It is used to write and upload programs to [Arduino](https://en.wikipedia.org/wiki/Arduino) compatible boards, but also, with the help of third-party cores, other vendor development boards.



FIG 4.1. ARDUINO IDE

If you want to program your Arduino Uno while offline you need to install the [Arduino](https://www.arduino.cc/en/Main/Software) [Desktop IDE](https://www.arduino.cc/en/Main/Software) The Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards. Before you can move on,

you must have installed the Arduino Software (IDE) on your PC.

### Why Arduino?

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.

Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

### .1 INSTALLING THE ARDUINO IDE :

1. Visit <http://www.arduino.cc/en/main/software>to download the latest Arduino IDE version for your computer’s operating system. There are versions for Windows, Mac, and Linux systems. At the download page, click on the ―Windows Installer‖ option for the easiest installation.
2. Save the .exe file to your hard drive.
3. Open the .exe file.
4. Click the button to agree to the licensing agreement:



FIG 4.1.1 ARDUINO SETUP LICENSE AGREEMENT

1. Decide which components to install, then click ―Next‖:



FIG 4.1.2 ARDUINO SETUP INSTALLATION OPTIONS

1. Select which folder to install the program to, then click ―Install‖:



FIG 4.1.3 ARDUINO SETUP INSTALLATION FOLDER

1. Wait for the program to finish installing, then click ―Close‖:

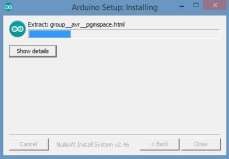


FIG 4.1.4 ARDUINO SETUP INSTALLING

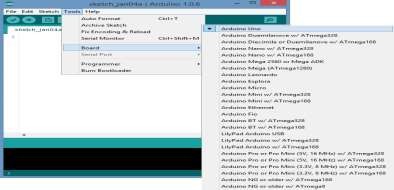
1. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:



FIG 4.1.5 SKETCH OFARDUINO

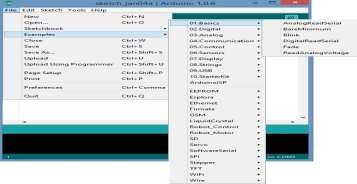
#### CONFIGURING THE ARDUINO IDE

The next thing to do is to make sure the software is set up for your particular Arduino board. Go to the ―Tools‖ drop-down menu, and find ―Board‖. Another menu will appear, where you can select from a list of Arduino models. I have the Arduino Uno R3, so I chose ―Arduino Uno‖.



### EXPLORING THE ARDUINO IDE

If you want, take a minute to browse through the different menus in the IDE. There is a good variety of example programs that come with the IDE in the ―Examples‖ menu. These will help you get started with your Arduino right away without having to do lots of research:



#### EXPERIMENTING WITH THE ARDUINO

Play around with the example programs and try changing parts of the code to see what happens. But if you want to learn programming as a skill, it’s best not to rely too much on these examples in your projects. You’ll learn much more by experimenting and writing your own code from scratch. A good way to learn programming is to get a book and work through the example projects. A great book for both beginning and advanced Arduino users is the [Arduino](http://www.amazon.com/gp/product/1449313876/ref%3Das_li_qf_sp_asin_il_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1449313876&linkCode=as2&tag=circbasi-20&linkId=LUNBKOV6L6NTBC6Q) [Cookbook by Michael Margolis.](http://www.amazon.com/gp/product/1449313876/ref%3Das_li_qf_sp_asin_il_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1449313876&linkCode=as2&tag=circbasi-20&linkId=LUNBKOV6L6NTBC6Q) It is a handy resource, and covers almost everything you can

do with the Arduino from a programming perspective.

###### Install the board drivers

If you used the Installer, Windows - from XP up to 10 - will install drivers automatically as soon as you connect your board.

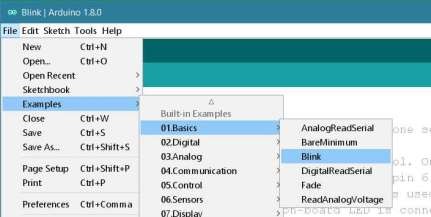
If you downloaded and expanded the Zip package or, for some reason, the board wasn't properly recognized, please follow the procedure below.

* + - Click on the Start Menu, and open up the Control Panel.

While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.

* + - Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
    - Right click on the "Arduino UNO (COmxx)" port and choose the "Update Driver Software" option.
    - Next, choose the "Browse my computer for Driver software" option.
    - Finally, navigate to and select the driver file named **"arduino.inf"**, located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub- directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named **"Arduino UNO.inf"**
    - Windows will finish up the driver installation from there.
    - Open your first sketch

Open the LED blink example sketch: **File > Examples >01.Basics > Blink**.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

FIG 4.2 BUILT IN EXAMPLES

Select your board type and port

You'll need to select the entry in the **Tools > Board** menu that corresponds to your Arduino board.

## 

FIG.4.2.1 SELECT YOUR BOARD TYPE

Select the serial device of the board from the Tools | Serial Port menu. This is likely to

be **COM3** or higher (**COM1** and **COM2** are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

## 

FIG.4.2.2 SELECTING PORTS

Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

## 

**FIG 4.2.3 UPLOADING**

## A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running. If you have problems, please see the [troubleshooting suggestions](https://www.arduino.cc/en/Guide/Troubleshooting).