Long Range Wireless Traffic Light Management System

A Major Project Report

Submitted to the Rajasthan Technical University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electrical Engineering

by

Himanshu Shekhar Paliwal [18ETCEE004]

DEPARTMENT OF ELECTRICAL ENGINEERING

TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY

UDAIPUR, RAJASTHAN

June 2022

DEPARTMENT OF ELECTRICAL ENGINEERING

TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY UDAIPUR, RAJASTHAN

2021 - 22

CERTIFICATE

This is to certify that the report entitled **Long Range Wireless Traffic Light Management System** submitted by **Himanshu Shekhar Paliwal**, to the Department of Electrical Engineeringin partial fulfillment of the B.Tech. degree in **Electrical Engineering** is a bonafide record of the seminar work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Abrar Ahmed Chhipa
(Project Guide)
Assistant Professor
Dept.of EE
Techno India NJR Institute
of Technology
Udaipur, Rajasthan

Mr. Rajkumar Soni
(Project Coordinator)
Assistant Professor
Dept.of EE
Techno India NJR Institute
of Technology
Udaipur, Rajasthan

Dr. Prakash Bahrani
Professor and Head
Dept.of EE
Techno India NJR Institute of Technology
Udaipur, Rajasthan

DECLARATION

I hereby declare that the major project report **Long Range Wireless Traffic Light Management System**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the Rajasthan Technical University, Kota, Rajasthan is a bonafide work done by me under supervision of Dr. Abrar Ahmed Chhipa

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Himanshu

Himanshu Shekhar Paliwal

18ETCEE004

Udaipur

13-06-2021

Abstract

The proposed Traffic Light Management System, upgrades the existing system with wireless remote control based capability during the peak hours. We will implement different algorithms for normal hours, morning peak hours and evening peak hours along with wireless remote priority control. Main beauty of the proposed idea is that it can be integrated into existing traffic light management system.

A wireless remote device will be there which will be connected to the main traffic light management system through secured wireless connection. The main traffic light management system will be always in listening mode to receive command from the wireless mode. The Operator will have freedom to choose or give priority to any lane signal which has higher traffic conjunction.

The system is also going to have backup power for the traffic light management system so the system will never run out of operation at any time of day. It is also a very cost effective solution for the stated problem solution.

Acknowledgement

We take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. We express my sincere thanks to **Dr. Prakash Bahrani**, Head of Department, Electrical Engineering, Techno India NJR Institute of Technology Udaipur for providing me with all the necessary facilities and support.

We would like to express my sincere gratitude to **Mr. Rajkumar Soni**, department of Electrical Engineering, Techno India NJR Institute of Technology, Udaipur for their support and co-operation.

We would like to place on record my sincere gratitude to my project guide **Dr. Abrar Ahmed Chhipa**, Assistant Professor, Electrical Engineering, Techno India NJR Institute of Technology for the guidance and mentorship throughout the course.

Finally We thank my family, and friends who contributed to the succesful fulfilment of this project work.

[illegible]

Contents

List of Figures

Chapter 1

Introduction

1.1 General Introduction

Traffic light control systems are widely used to monitor and control the flow of automobiles through the junction of many roads. They aim to realize smooth motion of cars in the transportation routes. However, the synchronization of multiple traffic light systems at adjacent intersections is a complicated problem given the various parameters involved. Conventional systems do not handle variable flows approaching the junctions. In addition, the mutual interference between adjacent traffic light systems, the disparity of cars flow with time, the accidents, the passage of emergency vehicles, and the pedestrian crossing are not implemented in the existing traffic system. This leads to traffic jam and congestion.

Traffic lights are signaling devices that are conceived to control the traffic flows at road intersections, pedestrian crossings, rail trains, and other locations. Traffic lights consist of three universal colored lights: the green light allows traffic to proceed in the indicated direction, the yellow light warns vehicles to prepare for short stop, and the red signal prohibits any traffic from proceeding. Nowadays, many countries suffer from the traffic congestion problems that affect the transportation system in cities and cause serious dilemma. In spite of replacing traffic officers and flagmen by automatic traffic systems, the optimization of the heavy traffic jam is still a major issue to be faced, especially with multiple junction nodes.

The rapid increase of the number of automobiles and the constantly rising number of road users are not accompanied with promoted infrastructures with sufficient resources. Partial solutions were offered by constructing new roads, implementing

flyovers and bypass roads, creating rings, and performing roads rehabilitation.
However, the traffic problem is very complicated due to the involvement of diverse parameters. First, the traffic flow depends on the time of the day where the traffic peak hours are generally in the morning and in the afternoon; on the days of the week where weekends reveal minimum load while Mondays and Fridays generally show dense traffic oriented from cities to their outskirts and in reverse direction respectively; and time of the year as holidays and summer.

Secondly, the current traffic light system is implemented with hard coded delays where the lights transition time slots are fixed regularly and do not depend on real time traffic flow. The third point is concerned with the state of one light at an intersection that influences the flow of traffic at adjacent intersections. Also, the conventional traffic system does not consider the case of accidents, roadworks, and breakdown cars that worsen traffic congestion.
In addition, a crucial issue is related to the smooth motion through intersections of emergency vehicles of higher priorities such as ambulances, rescue vehicles, fire brigade, police, and VIP persons that could get stuck in the crowd.
Finally, the pedestrians that cross the lanes also alter the traffic system. The conventional traffic system needs to be upgraded to solve the severe traffic congestion, alleviate transportation troubles, reduce traffic volume and waiting time, minimize overall travel time, optimize cars safety and efficiency, and expand the benefits in health, economic, and environmental sectors.

This report proposes a simple, low-cost, and real time smart traffic light control system that aims to overcome many defects and improve the traffic management. The system is based on micro-controller that controls changes the lighting transition slots accordingly.
Moreover, a handheld portable device communicates wirelessly with the traffic master controller by means of Radio Frequency (RF) transceivers in order to run the appropriate subroutines and allow the smooth displacement of emergency vehicles through the intersection.

1.2 Problem Statement

The Problem we have worked on is that we develop a software/hardware automation prototype for managing the peak hours at the traffic signals. For the problem mentioned our idea proposes hardware of a secured wireless remote control of traffic light management system (TLMS). Generally, traffic signal lights are placed in open space in about 50 - 100 meter radius and controlled by a traffic light controller which is currently based on PLC.

Our idea is to replace the PLC with our cost effective traffic light management system which can be controlled from a wireless remote control during the peak hours.

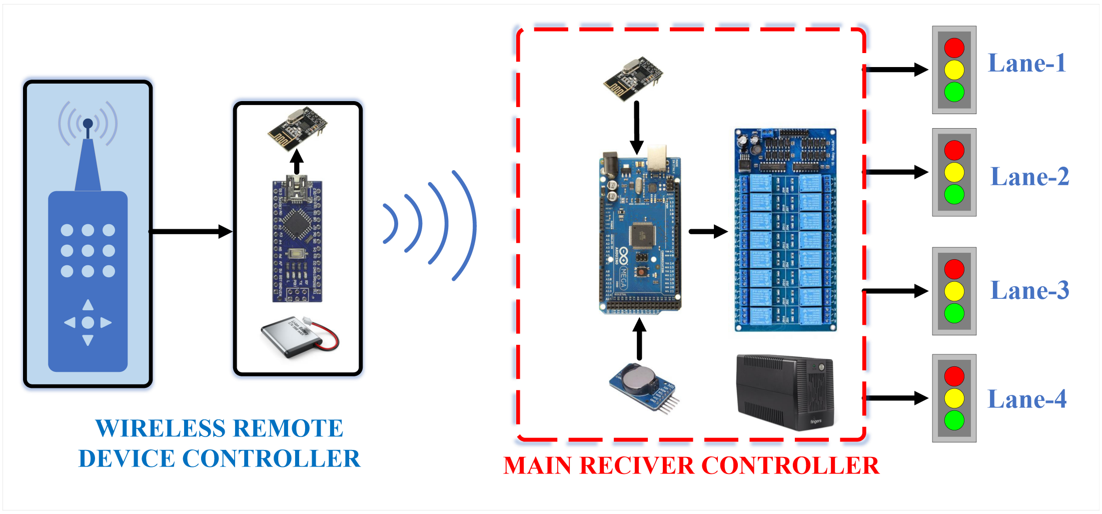


Figure 1.1: Traffic Light Controller and Signal Module

In our idea, we are going to propose a radio frequency connection between the remote and the traffic light management system. Two firmware will be designed for the complete system, one will be wireless remote control side and another will be traffic light management system side. The remote will have a button for the user input and process the input using a microcontroller and send commands to traffic light management system through the RF transmitter.

The traffic light management system will consist of a microcontroller, RF receiver, RTC and Relay board. For traffic light management system firmware we will design different algorithms to control traffic light signals during normal operational hours, morning peak hours, evening peak hours and remote priority based control. Beside these all features our system will further upgradation which have the solar based power backup so that the system will never go out of running operation at any time of the day.

1.3 Objective

The objective of this is proposed that Traffic Light Management System, upgrades the existing system with wireless remote control based capability during the peak hours. We will implement different algorithms for normal hours, morning peak hours and evening peak hours along with wireless remote priority control. Main beauty of the proposed idea is that it can be integrated into existing Traffic Light Management System.
A wireless remote device will be there which will be connected to the main Traffic Light Management System through secured wireless connection.

The main Traffic Light Management System will be always in listening mode to receive command from the wireless mode. The Operator will have freedom to choose or give priority to any lane signal which has higher traffic conjunction. This system is also going to have backup power for the Traffic Light Management System so the system will never run out of operation at any time of day. It is also a very cost effective solution for the stated problem solution.

1.4 Closure

In the end this project the Traffic lights are controlled by the remote controlled by traffic police man, where the remote controller uses the Radio frequency for the transmission the signal and data from remote to the traffic signal lights.

The Arduino nano is a microcontroller which is used as a controller for the traffic Red, Yellow and Green light LED's which can be controlled by the pushbuttons in the remote. The nRF24L01 module is a radio frequency module which is a transceiver that is it can both transmits and receives the data wirelessly for a particular distant.

Chapter 2

Literature Review

2.1 Review of Different Type of Methodology

Manpreet Singh Bhatia,Alok Aggarwal,Narendra Kumar:

A fuzzy logic-based controller is proposed with three input parameters namely Queue_length, Rem_time_green and Peak_hours and one output parameter, Time_extension. Queue_length refers to the length of the vehicles in the queue behind the signal. Sensors capture the length of queues in each lane and pass the data into the fuzzy controller. Fuzzy sets of queue length have been taken as very_less, less, medium and long. Length between 0 4 meters refers to very_less, 0 8 meters as less, 4 12 meters as medium and more than 8 meters as long. Second input parameter, rem_time_green, has 4 fuzzy sets as very_less, less, medium and large. Rem_time_green is the time remaining for the green signal before turning to red.

This is a very decisive factor for controlling the time extension of the signal. It is calculated as a fraction of the full time which is fixed as 60 seconds in all the lanes. Remaining time green as 0 0.25 is taken as very_less, 0 0.5 as less, 0.25 0.75 as medium and greater than 0.5 as large. Peak_hours refers to the different time duration of the day. This factor is also very crucial for controlling the extension of the signal as most of the conventional signals are not considering the time duration of the day for passing the vehicles due to which long waiting queues occur during the peak hours.

Peak_hours consists of five fuzzy sets namely,very_light, heavy_morning, medium, heavy_evening, light. Time duration from 11:00 PM 7:30 AM is taken as very_light traffic conditions as most of the roads don't have much traffic density during this period, heavy_morning timings are 7:00 AM 11:30 AM in which most of the traffic is active, medium timings are 11:00 AM 4:30 PM, 4:00 PM 8:30 PM is taken as heavy_evening as this is that duration of the day where most of the traffic density occurs and 8:00 PM 12:30 AM as light traffic conditions. Output of the fuzzy control system is Time_extension which is to be controlled by the use of the three input parameters. All four lanes have been allocated a fixed green signal time of 60 seconds at the start. Time_extension is done dynamically in the running queues after analyzing the three input parameters.

Extensiondecrease of the green light can be done up to 35 seconds, i.e. the range of time duration of green time is in the interval 25, 95 seconds. Five fuzzy sets have been taken for the Time_extension as more_decrease, decrease, do_not_change, increase and more_increase. If the Rem_time_green and Queue_length is high in the running queue, time extension should be decreased and vice versa. Similarly, during heavy_morning and heavy_evening timings extension should be increased up to 95 seconds while during very_light conditions it should decrease to 25 seconds. Extension of time is done by keeping in mind the state of other queues as well to provide a better optimized solution to the problem.

Rongrong Tian, Xu Zhang:

They suggested to use the TRANSYT traffic modeling software to find the optimal fixedtime signal plan and VISSIM microsimulation software to affirm and evaluate the TRANSYT model and to help assess the optimal signal plan; build an adaptive frame signal plan and refined and evaluated the plan using VISSIM with VS-PLUS emulator. Through microsimulation, it was shown that delay in the adaptive signal control was shortened noticeably than that in the fixed time control. Jianhua Guo et al introduced a new method for areawide traffic signal timing optimization under user equilibrium traffic.

The optimization model was formulated as a multi-dimensional search problem aimed to achieve minimized product of the total travel time associated with urban street network and the variance of travel time for unit distance of travel. A genetic algorithm was developed to derive the model solution. A simulation control protocol embedded in PARAMICS software tool capable of conducting area-wide micro simulation is adopted to design the logic frame and function module of the area-wide traffic signal control system. His results shown that mobility improvements are achieved after applying the proposed model along with the genetic algorithm for area-wide signal timing optimization, assessed by extended capacity ratio, and reductions in through and turning movement delays, as well as average and variance of travel time for unit distance of Travel.

The design of intelligent traffic control system is an active research topic. Researchers around the world are inventing newer approaches and innovative systems to solve this stressful problem. Models based on mathematical equations are applied to estimate the car waiting time at a junction, the number of cars in the waiting queue, the extension of the waiting cars along the lane, the optimal timing slots for green, yellow, and red lights that best fit the real and veritable situation and the efficient combination of routing. In fact, the mutual dependencies between nearby intersections lead to a complicated formulation with cumbersome parameters. These parameters are accidental, hazardous, dependent, and the worse point is the variance of these parameters with time.

Thus, finding a dynamic, consistent, and convenient solution is quite impossible. Researchers from different disciplines are collaborating to explore feasible solutions that reduce traffic congestion. Therefore, various methodologies are constantly proposed in the literature and many techniques are implemented profiting from the technological advances of microcomputers, recent manufactured devices and sensors, and innovative algorithms modeling, as much as possible, the complication of traffic lights.

The IR sensors are employed in numerous traffic systems. The IR transmitter and the IR receiver are mounted on either sides of a road. When an automobile passes on the road between the IR sensors, the system is activated and the car counter is incremented. The collected information about the traffic density of the different roads of a junction is analyzed in order to modify dynamically the delays of green light at the lane having the significant traffic volume. The whole system could be controlled by PIC microcontroller or even by PLC.

To inform the traffic system about the arrival of the emergency vehicles toward the junction, they are supported by RF emitters that send warning signals to RF transceivers disposed at every traffic light intersection. The triggering sequences of the traffic lights are modified correspondingly in order to provide a special route to the emergency vehicles. Other researchers use the Global Positioning System (GPS) to communicate with the traffic light controllers and send preemption signals. The ambulance was equipped with both RF to communicate with traffic light controller and the GSM module to report to hospital doctors about the patient status and to receive messages concerning the kind of therapy or first aid recovery that should be done to the injured patient.

Many works predict the density of the traffic based on image processing methodology. But these techniques require the acquisition of good images whose quality are weather dependent, especially with the rain and the fog. Other researchers use sophisticated algorithms to model the various states of the traffic such as fuzzy logic and genetic algorithms.

Most published works are dedicated to one junction or intersection where the influence of the adjacent intersections is not examined. Thus, the situation becomes more complicated and widely dependent. Further efforts should be made to achieve complete modeling, monitoring, and control for multiple synchronized junctions.

2.2 Different Ways of Traffic Passing

Before you drive any motor vehicle on the road for which a driving licence is required, you must have a valid driving licence authorising you to drive that motor vehicle, and be covered by valid insurance in respect of that motor vehicle.Before you move off, look around, even though you may have looked in your mirror, to see that no one is about to overtake you. Give the proper signal before moving off, and only move off when you can do so safely and without inconvenience to other road users. Give way to passing and overtaking vehicles.

On a three-lane carriageway, you may keep to the central lane when the left-hand lane is occupied by slower moving vehicles. The outer (right-hand) lane is for overtaking only; do not stay in it longer than necessary after overtaking vehicles in the centre lane.Never overtake unless you can do so without danger to yourself or others. Be specially careful at night, and in heavy rain and mist, as it is more difficult to judge speed and distance at such times.

When approaching a junction, you must observe the following:

(a) If you are going straight across the junction, you must give way to traffic going straight from the right.

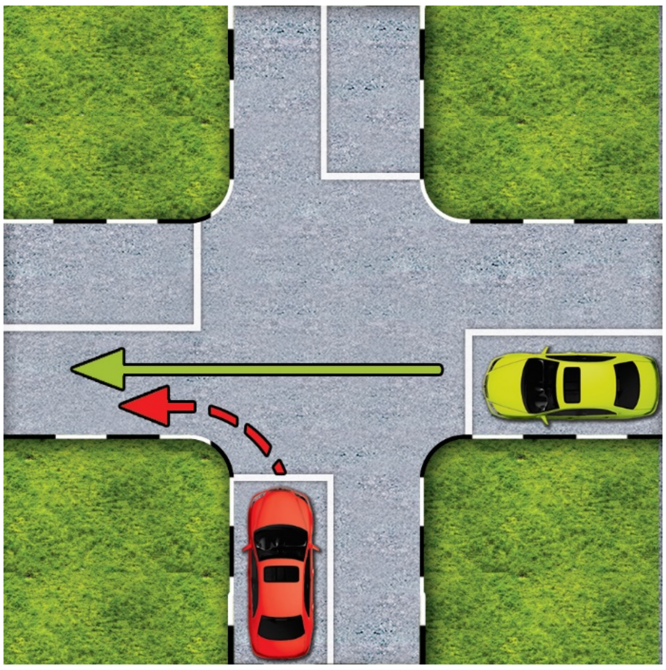


Figure 2.1: Traffic Passing

(b) If you are turning right, you must give way to traffic going from all directions. You must also give way to traffic turning right from the right, and to traffic turning left from the opposite direction.

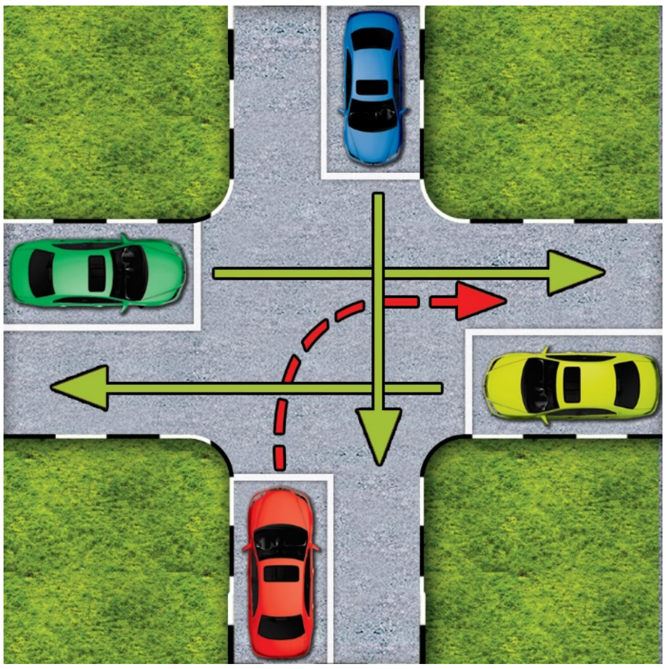


Figure 2.2: Traffic Passing

(c) If you are turning left, you must give way to traffic going straight from the right.

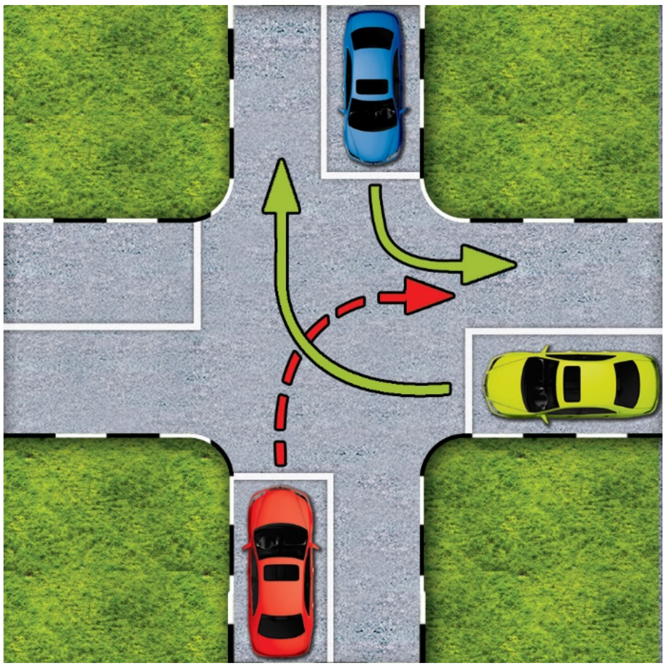


Figure 2.3: Traffic Passing

Well before you turn right at a junction, take full account of the position and movement of traffic behind you. Signal your intention early, and drive cautiously towards the centre of the junction. Give way to approaching vehicles and crossing pedestrians adjacent to you. Wait until it is safe to cross or wait for the green arrow signal to appear. Turn swiftly to the correct lane, keeping a look-out for pedestrians crossing at the junction.

Chapter 3

Methodology

3.1 Method

In our idea, we are going to propose a radio frequency connection between the remote and the traffic light management system. Two firmware will be designed for the complete system, one will be wireless remote control side and another will be traffic light management system side. The remote will have a button for the user input and process the input using a microcontroller and send commands to traffic light management system through the RF transmitter. The traffic light management system will consist of a microcontroller, RF receiver, RTC and Relay board.

For traffic light management system firmware we will design different algorithms to control traffic light signals during normal operational hours, morning peak hours, evening peak hours and remote priority based control. Beside these all features our system will have the solar based power backup so that the system will never go out of running operation at any time of the day.

Above figure shows that we have got four different lanes with different traffic densities. In existing traffic light management system the timings for outgoing traffic is the same for each lane. Let the time assigned for each lane is 60 seconds.
According to traffic density lane 1 should get more outgoing traffic time compared to other lanes and lane 4 needs very less time amongst all lanes. In this scenario traffic at lane 1 will keep increasing and traffic conjunction problems will occur at this lane.

Here Two firmware will be designed for the complete system, one will be wireless remote control side and another will be traffic light management system side.

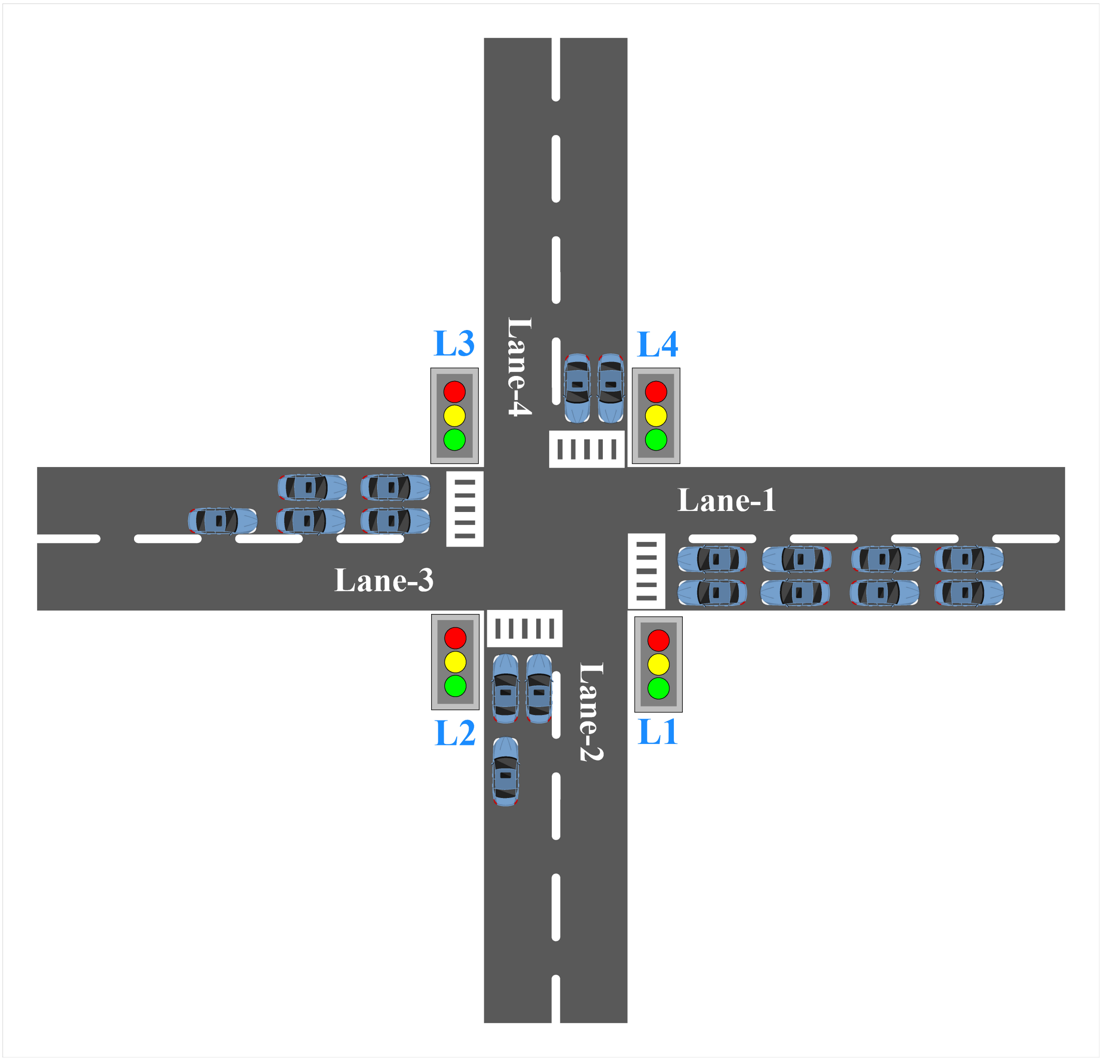


Figure 3.1: Traffic Signal with Lane

Now in this case, if we use the proposed wireless remote control based traffic light management system then the operator can prioritize lane 1 for outgoing traffic. Which will eventually solve the traffic conjunction problem.

After operating through one or two cycles of traffic signals the operation goes to its normal mode operation. The block diagram of the proposed wireless remote control of traffic light management system is shown in Fig. 3.1. This system consists of a remote control unit and the main traffic light management system unit which are wirelessly connected to each other over the radio frequency communication.

3.2 Block Diagram

The transmitter Remote has some push buttons which when pressed give some input to the Arduino nano of the Remote traffic signal light where the traffic light Red, Yellow and Green are connected to the Arduino nano (a microcontroller). so when these pushbuttons are pressed the respective lights which glow and blink with particular time slot given as per the guidelines by RTO and rules and regulations of it.

The Arduino nano is connected with the nRF24L01 module which send the input data or signal when we pressed the pushbuttons which are on the breadboard. these pushbuttons send the data to the microcontroller and the microcontroller send the data or signal to another Arduino nano with the help of nRF24L01 module which has inbuilt both transmitter and receiver.

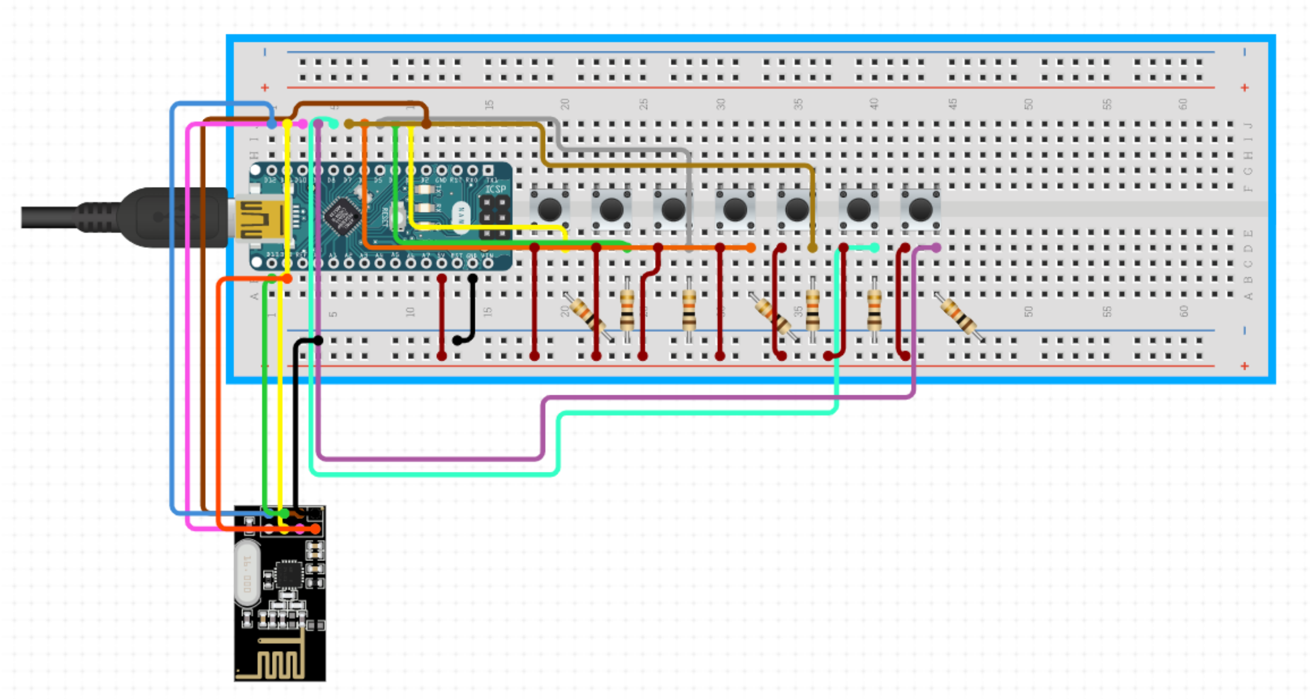


Figure 3.2: Transmitter Remote Block Diagram

The receiver side has also a Microcontroller that is Arduino nano which controls the respective lane traffic lights through the remote which was in the hand of traffic police man. There are Three LED's i.e. Red, Yellow and Green which will on of as per the input.

The nRF24L01 Module is connected to the Arduino nano for the receiving the signals send by the remote antenna and transmitter.

The nRF24L01 Module converts and decode the the data and signal and produces the specific output. This output may results the controlling on and off LED's in particular lane.

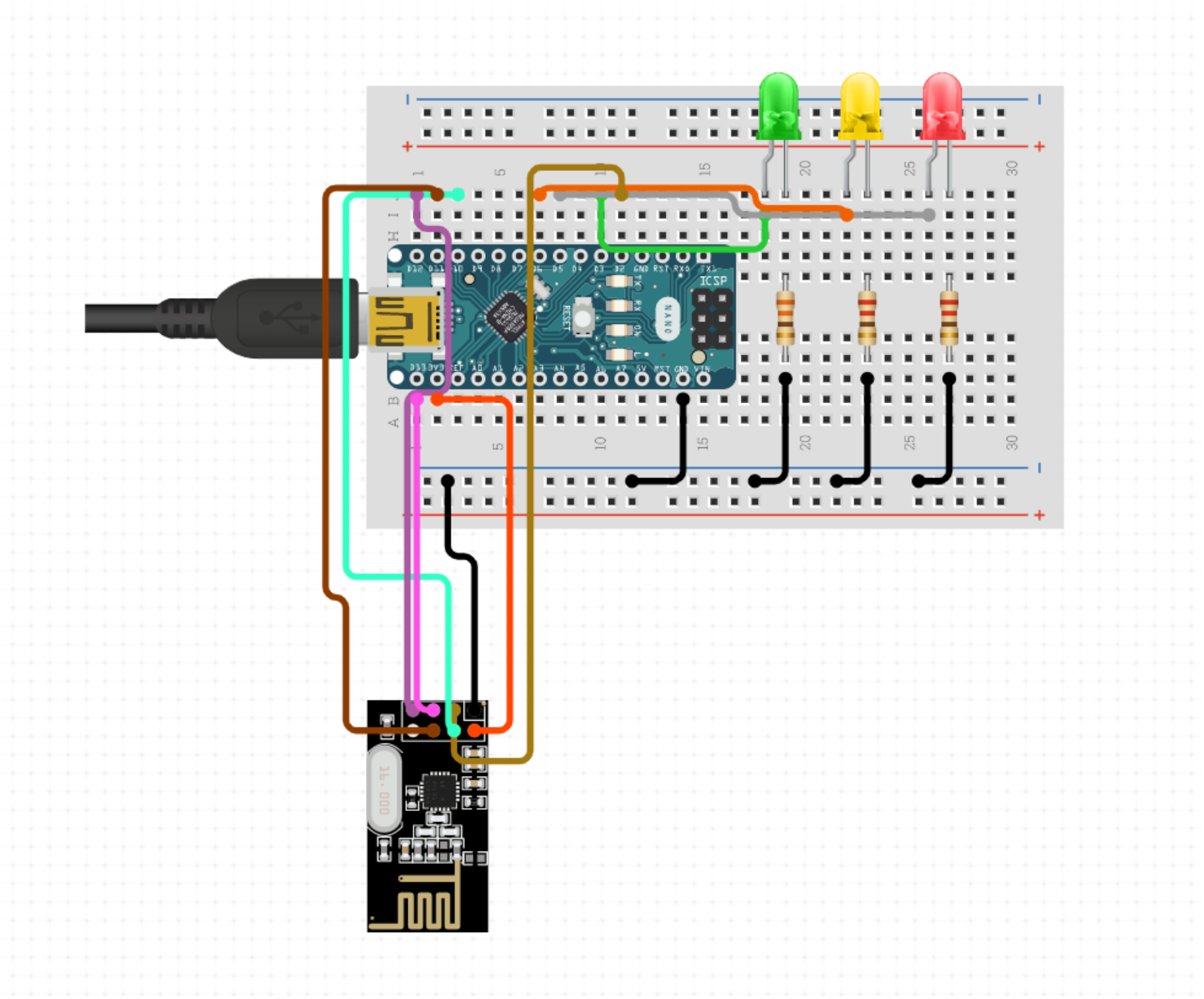


Figure 3.3: Receiver Side Arduino Based Diagram

3.3 Programming Codes

The Below code is written for communication of transmitter and receiver of the traffic signals and the master that is our remote controller. The below RF24 library includes the code for this communication and various other problems like sending more than one data to the receiver's like Traffic signal installed Arduino nano.

The Arduino transmitter code send the data input to the receiver side that is it sends the data signal from the remote controller with push buttons for various lane switching to the receiver that is our main master receiver which further sends data to traffic signal lights to on/off by the particular input given through the controller remote.

The Arduino receiver code is written for the communication between the Arduino nano's for transmitting and receiving the data signals through the remote controller. This code helps to setup communication between transmitter Arduino nano and receiver Arduino nano to receive the input and on/off the traffic led's as per the input.

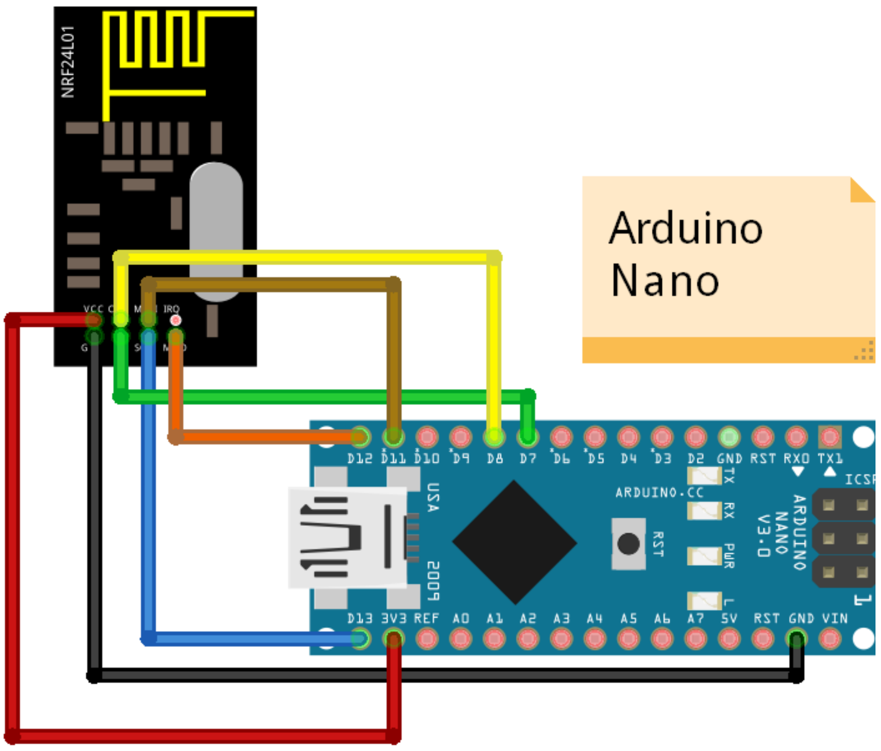

NRF24L01
VCC
IRQ
Arduino Nano
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
ARDUINO.CC
ARDUINO NANO V3.0
ICSP
RST
TX
RX
PWR
L
USA
2009

Figure 3.4: Arduino Nano Connected to nRF24L02 Module for Radio frequency Communication

3.3.1 RF Library

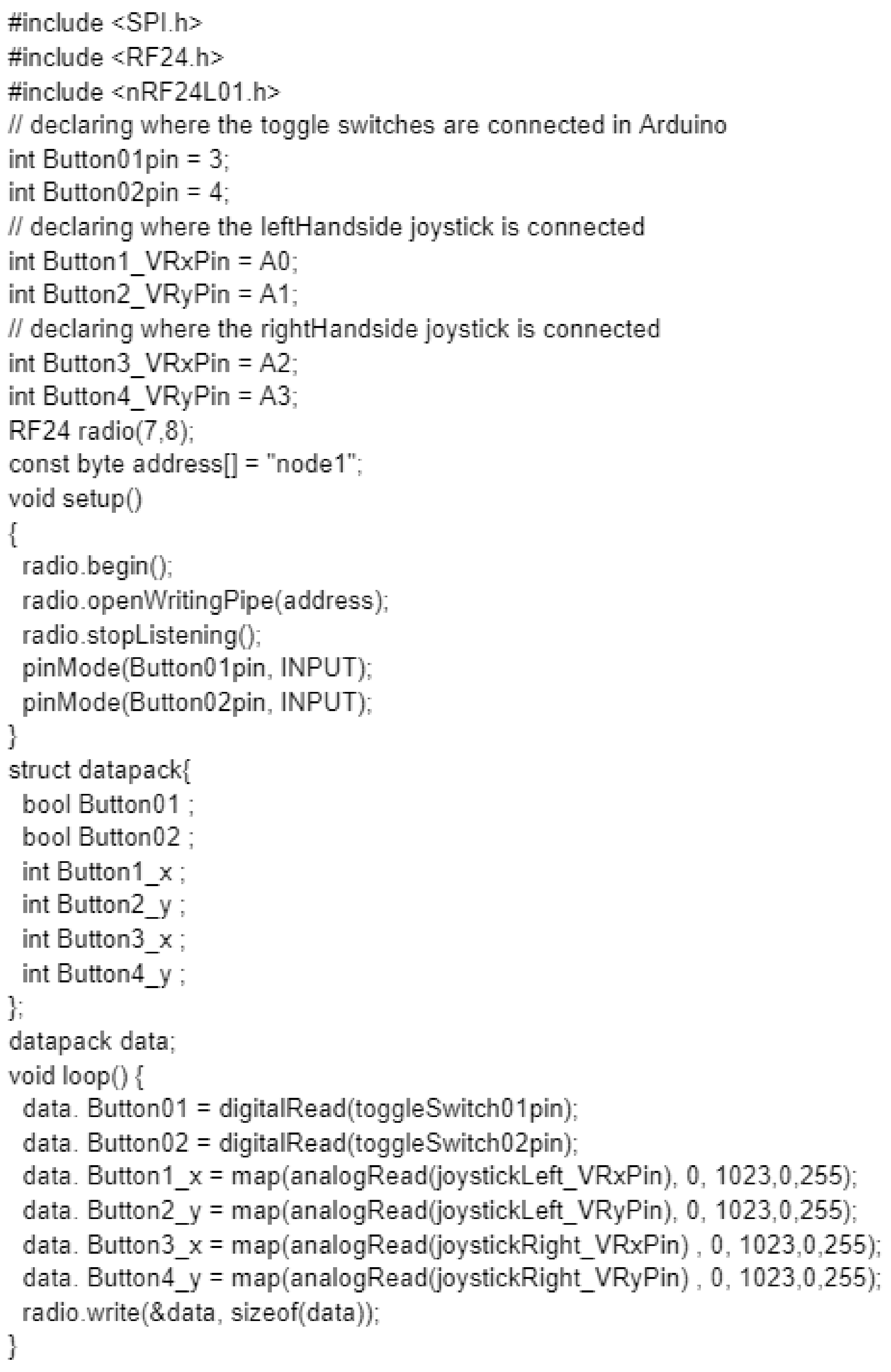
```
#include "nRF24L01.h"
#include "RF24_config.h"
#include "RF24.h"
void RF24::csn(bool mode)
{
   #if defined(RF24_TINY)
   if (ce_pin != csn_pin) {
      digitalWrite(csn_pin, mode);
   }
   else {
      if (mode == HIGH) {
         PORTB |= (1<<PINB2);                          // SCK->CSN HIGH
         delayMicroseconds(RF24_CSN_SETTLE_HIGH_DELAY); // allow csn to settle.
      }
      else {
         PORTB &= ~(1<<PINB2);                        // SCK->CSN LOW
         delayMicroseconds(RF24_CSN_SETTLE_LOW_DELAY); // allow csn to settle
      }
   }
   // Return, CSN toggle complete
   return;
   #elif defined (ARDUINO) && !defined(RF24_SPI_TRANSACTIONS)
      #if !defined(SOFTSPI)
      // applies to SPI_UART and inherent hardware SPI
         #if defined (RF24_SPI_PTR)
   _spi->setBitOrder(MSBFIRST);
   _spi->setDataMode(SPI_MODE0);
            #if !defined(F_CPU) || F_CPU < 20000000
   _spi->setClockDivider(SPI_CLOCK_DIV2);
            #elif F_CPU < 40000000
   _spi->setClockDivider(SPI_CLOCK_DIV4);
            #elif F_CPU < 80000000
   _spi->setClockDivider(SPI_CLOCK_DIV8);
            #elif F_CPU < 160000000
   _spi->setClockDivider(SPI_CLOCK_DIV16);
            #elif F_CPU < 320000000
   _spi->setClockDivider(SPI_CLOCK_DIV32);
            #elif F_CPU < 640000000
   _spi->setClockDivider(SPI_CLOCK_DIV64);
            #elif F_CPU < 1280000000
   _spi->setClockDivider(SPI_CLOCK_DIV128);
            #else // F_CPU >= 1280000000
               #error "Unsupported CPU frequency. Please set correct SPI divider."
            #endif // F_CPU to SPI_CLOCK_DIV translation
```

```
        #else // !defined(RF24_SPI_PTR)
    _SPI.setBitOrder(MSBFIRST);
    _SPI.setDataMode(SPI_MODE0);

            #if !defined(F_CPU) || F_CPU < 20000000
    _SPI.setClockDivider(SPI_CLOCK_DIV2);
            #elif F_CPU < 40000000
    _SPI.setClockDivider(SPI_CLOCK_DIV4);
            #elif F_CPU < 80000000
    _SPI.setClockDivider(SPI_CLOCK_DIV8);
            #elif F_CPU < 160000000
    _SPI.setClockDivider(SPI_CLOCK_DIV16);
            #elif F_CPU < 320000000
    _SPI.setClockDivider(SPI_CLOCK_DIV32);
            #elif F_CPU < 640000000
    _SPI.setClockDivider(SPI_CLOCK_DIV64);
            #elif F_CPU < 1280000000
    _SPI.setClockDivider(SPI_CLOCK_DIV128);
            #else // F_CPU >= 1280000000
                #error "Unsupported CPU frequency. Please set correct SPI divider."
            #endif // F_CPU to SPI_CLOCK_DIV translation
        #endif // !defined(RF24_SPI_PTR)
      #endif // !defined(SOFTSPI)

    #elif defined (RF24_RPi)
    if(!mode)
        _SPI.chipSelect(csn_pin);
    #endif // defined(RF24_RPi)

    #if !defined(RF24_LINUX)
    digitalWrite(csn_pin, mode);
    delayMicroseconds(csDelay);
    #else
    static_cast<void>(mode); // ignore -Wunused-parameter
    #endif // !defined(RF24_LINUX)
}
void RF24::ce(bool level)
{
    //Allow for 3-pin use on ATTiny
    if (ce_pin != csn_pin) {
        digitalWrite(ce_pin, level);
    }
}
```

```
inline void RF24::beginTransaction()
{
   #if defined (RF24_SPI_TRANSACTIONS)
      #if defined (RF24_SPI_PTR)
         #if defined (RF24_RP2)
   _spi->beginTransaction(spi_speed);
         #else // ! defined (RF24_RP2)
   _spi->beginTransaction(SPISettings(spi_speed, MSBFIRST, SPI_MODE0));
         #endif // ! defined (RF24_RP2)
      #else // !defined(RF24_SPI_PTR)
   _SPI.beginTransaction(SPISettings(spi_speed, MSBFIRST, SPI_MODE0));
      #endif // !defined(RF24_SPI_PTR)
   #endif // defined (RF24_SPI_TRANSACTIONS)
   csn(LOW);
}
inline void RF24::endTransaction()
{
   csn(HIGH);
   #if defined (RF24_SPI_TRANSACTIONS)
      #if defined (RF24_SPI_PTR)
   _spi->endTransaction();
      #else // !defined(RF24_SPI_PTR)
   _SPI.endTransaction();
      #endif // !defined(RF24_SPI_PTR)
   #endif // defined (RF24_SPI_TRANSACTIONS)
}
void RF24::read_register(uint8_t reg, uint8_t* buf, uint8_t len)
{
   #if defined (RF24_LINUX) || defined (RF24_RP2)
   beginTransaction(); //configures the spi settings for RPi, locks mutex and setting csn low
   uint8_t * prx = spi_rxbuff;
   uint8_t * ptx = spi_txbuff;
   uint8_t size = static_cast<uint8_t>(len + 1); // Add register value to transmit buffer

   *ptx++ = (R_REGISTER | reg);

   while (len--){ *ptx++ = RF24_NOP; } // Dummy operation, just for reading

      #if defined (RF24_RP2)
   _spi->transfernb((const uint8_t*)spi_txbuff, spi_rxbuff, size);
      #else // !defined (RF24_RP2)
   _SPI.transfernb(reinterpret_cast<char *>(spi_txbuff), reinterpret_cast<char *>(spi_rxbuff),
size);
      #endif // !defined (RF24_RP2)
```

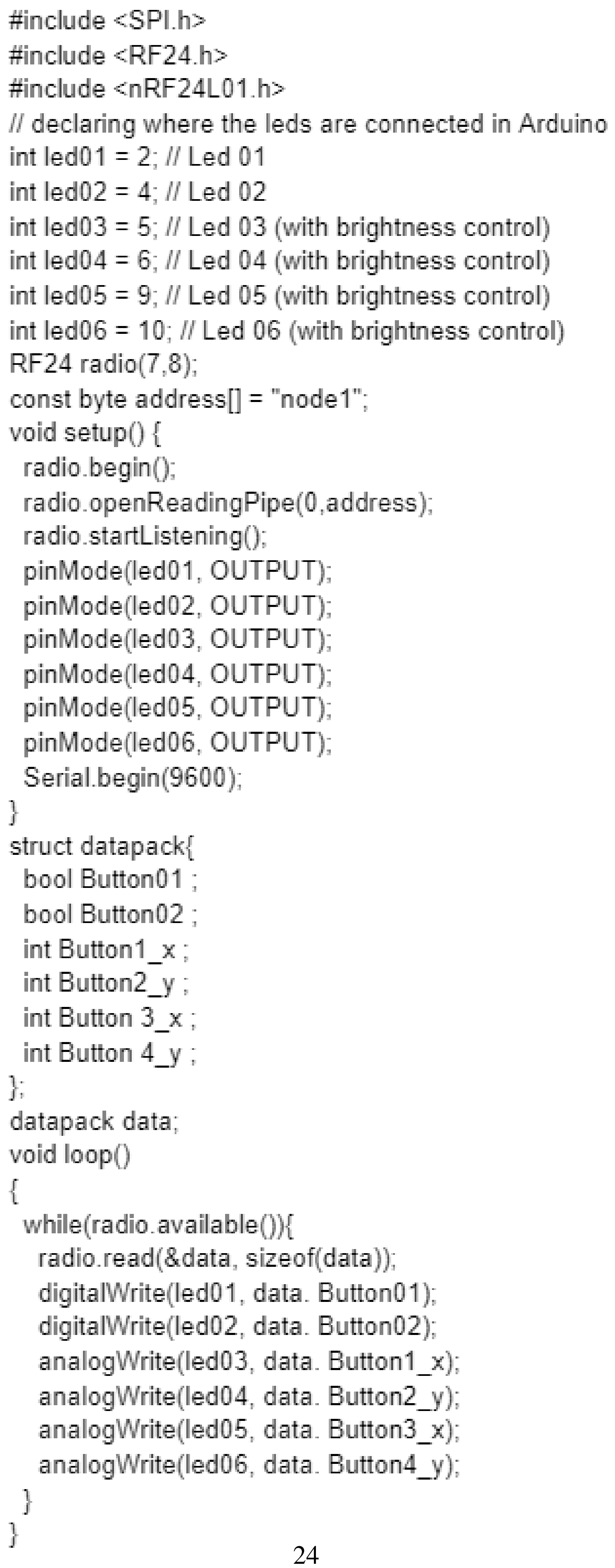
```
    status = *prx++; // status is 1st byte of receive buffer

    // decrement before to skip status byte
    while (--size) { *buf++ = *prx++; }

    endTransaction(); // unlocks mutex and setting csn high

    #else // !defined(RF24_LINUX) && !defined(RF24_RP2)

    beginTransaction();
        #if defined (RF24_SPI_PTR)
    status = _spi->transfer(R_REGISTER | reg);
    while (len--) { *buf++ = _spi->transfer(0xFF); }

        #else // !defined(RF24_SPI_PTR)
    status = _SPI.transfer(R_REGISTER | reg);
    while (len--) { *buf++ = _SPI.transfer(0xFF); }

        #endif // !defined(RF24_SPI_PTR)
    endTransaction();
    #endif // !defined(RF24_LINUX) && !defined(RF24_RP2)
}

uint8_t RF24::read_register(uint8_t reg)
{
    uint8_t result;

    #if defined (RF24_LINUX) || defined (RF24_RP2)
    beginTransaction();

    uint8_t * prx = spi_rxbuff;
    uint8_t * ptx = spi_txbuff;
    *ptx++ = (R_REGISTER | reg);
    *ptx++ = RF24_NOP ; // Dummy operation, just for reading

        #if defined (RF24_RP2)
    _spi->transfernb((const uint8_t*)spi_txbuff, spi_rxbuff, 2);
        #else // !defined(RF24_RP2)
    _SPI.transfernb(reinterpret_cast<char *>(spi_txbuff), reinterpret_cast<char *>(spi_rxbuff), 2);
        #endif // !defined(RF24_RP2)

    status = *prx;     // status is 1st byte of receive buffer
    result = *++prx;   // result is 2nd byte of receive buffer
    endTransaction();
    #else // !defined(RF24_LINUX) && !defined(RF24_RP2)

    beginTransaction();
        #if defined (RF24_SPI_PTR)
    status = _spi->transfer(R_REGISTER | reg);
    result = _spi->transfer(0xff);

        #else // !defined(RF24_SPI_PTR)
    status = _SPI.transfer(R_REGISTER | reg);
    result = _SPI.transfer(0xff);

        #endif // !defined(RF24_SPI_PTR)
    endTransaction();
    #endif // !defined(RF24_LINUX) && !defined(RF24_RP2)

    return result;
}
void RF24::write_register(uint8_t reg, const uint8_t* buf, uint8_t len)
{
    #if defined (RF24_LINUX) || defined (RF24_RP2)
    beginTransaction();
    uint8_t * prx = spi_rxbuff;
    uint8_t * ptx = spi_txbuff;
    uint8_t size = static_cast<uint8_t>(len + 1); // Add register value to transmit buffer

    *ptx++ = (W_REGISTER | (REGISTER_MASK & reg));
    while (len--) { *ptx++ = *buf++; }

        #if defined (RF24_RP2)
    _spi->transfernb((const uint8_t*)spi_txbuff, spi_rxbuff, size);
        #else // !defined(RF24_RP2)
    _SPI.transfernb(reinterpret_cast<char *>(spi_txbuff), reinterpret_cast<char *>(spi_rxbuff),
size);
        #endif // !defined(RF24_RP2)

    status = *prx; // status is 1st byte of receive buffer
    endTransaction();
    #else // !defined(RF24_LINUX) && !defined(RF24_RP2)

    beginTransaction();
        #if defined (RF24_SPI_PTR)
    status = _spi->transfer(W_REGISTER | reg);
    while (len--) { _spi->transfer(*buf++); }

        #else // !defined(RF24_SPI_PTR)
    status = _SPI.transfer(W_REGISTER | reg);
    while (len--) { _SPI.transfer(*buf++); }

        #endif // !defined(RF24_SPI_PTR)
    endTransaction();
    #endif // !defined(RF24_LINUX) && !defined(RF24_RP2)
}
void RF24::write_register(uint8_t reg, uint8_t value, bool is_cmd_only)
{
    if (is_cmd_only) {
        if (reg != RF24_NOP) { // don't print the get_status() operation
            IF_SERIAL_DEBUG(printf_P(PSTR("write_register(%02x)\r\n"), reg));
        }
        beginTransaction();
        #if defined (RF24_LINUX)
        status = _SPI.transfer(W_REGISTER | reg);
        #else // !defined(RF24_LINUX) || defined (RF24_RP2)
            #if defined (RF24_SPI_PTR)
        status = _spi->transfer(W_REGISTER | reg);
            #else // !defined (RF24_SPI_PTR)
        status = _SPI.transfer(W_REGISTER | reg);
            #endif // !defined (RF24_SPI_PTR)
        #endif // !defined(RF24_LINUX) || defined(RF24_RP2)
        endTransaction();
    }
    else {
        IF_SERIAL_DEBUG(printf_P(PSTR("write_register(%02x,%02x)\r\n"), reg, value));
        #if defined (RF24_LINUX) || defined (RF24_RP2)
        beginTransaction();
        uint8_t * prx = spi_rxbuff;
        uint8_t * ptx = spi_txbuff;
        *ptx++ = (W_REGISTER | reg);
        *ptx = value;

            #if defined (RF24_RP2)
        _spi->transfernb((const uint8_t*)spi_txbuff, spi_rxbuff, 2);
            #else // !defined(RF24_RP2)
        _SPI.transfernb(reinterpret_cast<char *>(spi_txbuff), reinterpret_cast<char *>(spi_rxbuff),
2);
            #endif // !defined(RF24_RP2)
        status = *prx++; // status is 1st byte of receive buffer
        endTransaction();
        #else // !defined(RF24_LINUX) && !defined(RF24_RP2)

        beginTransaction();
            #if defined (RF24_SPI_PTR)
        status = _spi->transfer(W_REGISTER | reg);
        _spi->transfer(value);
            #else // !defined(RF24_SPI_PTR)
        status = _SPI.transfer(W_REGISTER | reg);
        _SPI.transfer(value);
            #endif // !defined(RF24_SPI_PTR)
        endTransaction();
        #endif // !defined(RF24_LINUX) && !defined(RF24_RP2)
    }
}
                void RF24::setRadiation(uint8_t level, rf24_datarate_e speed, bool lnaEnable)
{
    uint8_t setup = _data_rate_reg_value(speed);
    setup |= _pa_level_reg_value(level, lnaEnable);
    write_register(RF_SETUP, setup);
}
```

3.3.2 Transmitter Code



```
#include <SPI.h>
#include <RF24.h>
#include <nRF24L01.h>
// declaring where the toggle switches are connected in Arduino
int Button01pin = 3;
int Button02pin = 4;
// declaring where the leftHandside joystick is connected
int Button1_VRxPin = A0;
int Button2_VRyPin = A1;
// declaring where the rightHandside joystick is connected
int Button3_VRxPin = A2;
int Button4_VRyPin = A3;
RF24 radio(7,8);
const byte address[] = "node1";
void setup()
{
  radio.begin();
  radio.openWritingPipe(address);
  radio.stopListening();
  pinMode(Button01pin, INPUT);
  pinMode(Button02pin, INPUT);
}
struct datapack{
  bool Button01 ;
  bool Button02 ;
  int Button1_x ;
  int Button2_y ;
  int Button3_x ;
  int Button4_y ;
};
datapack data;
void loop() {
  data. Button01 = digitalRead(toggleSwitch01pin);
  data. Button02 = digitalRead(toggleSwitch02pin);
  data. Button1_x = map(analogRead(joystickLeft_VRxPin), 0, 1023,0,255);
  data. Button2_y = map(analogRead(joystickLeft_VRyPin), 0, 1023,0,255);
  data. Button3_x = map(analogRead(joystickRight_VRxPin) , 0, 1023,0,255);
  data. Button4_y = map(analogRead(joystickRight_VRyPin) , 0, 1023,0,255);
  radio.write(&data, sizeof(data));
}
```

3.3.3 Receiver Code



```
#include <SPI.h>
#include <RF24.h>
#include <nRF24L01.h>
// declaring where the leds are connected in Arduino
int led01 = 2; // Led 01
int led02 = 4; // Led 02
int led03 = 5; // Led 03 (with brightness control)
int led04 = 6; // Led 04 (with brightness control)
int led05 = 9; // Led 05 (with brightness control)
int led06 = 10; // Led 06 (with brightness control)
RF24 radio(7,8);
const byte address[] = "node1";
void setup() {
  radio.begin();
  radio.openReadingPipe(0,address);
  radio.startListening();
  pinMode(led01, OUTPUT);
  pinMode(led02, OUTPUT);
  pinMode(led03, OUTPUT);
  pinMode(led04, OUTPUT);
  pinMode(led05, OUTPUT);
  pinMode(led06, OUTPUT);
  Serial.begin(9600);
}
struct datapack{
  bool Button01 ;
  bool Button02 ;
  int Button1_x ;
  int Button2_y ;
  int Button 3_x ;
  int Button 4_y ;
};
datapack data;
void loop()
{
  while(radio.available()){
    radio.read(&data, sizeof(data));
    digitalWrite(led01, data. Button01);
    digitalWrite(led02, data. Button02);
    analogWrite(led03, data. Button1_x);
    analogWrite(led04, data. Button2_y);
    analogWrite(led05, data. Button3_x);
    analogWrite(led06, data. Button4_y);
  }
}
```

3.4 Components

The components used in the this projects are given below :

3.4.1 Arduino Nano

We have used five Arduino nano to the communication and input - output communication and controlling. Arduino Nano is a small, complete, flexible and breadboard-friendly Microcontroller board, based on AT-mega328p, developed by Arduino.cc in Italy in 2008 and contains 30 male I/O headers, configured in a DIP30 style.

Arduino Nano Pin out contains 14 digital pins, 8 analog Pins, 2 Reset Pins 6 Power Pins.Arduino Nano is simply a smaller version of Arduino UNO, thus both have almost the same functionalities. It comes with an operating voltage of 5V, however, the input voltage can vary from 7 to 12V.

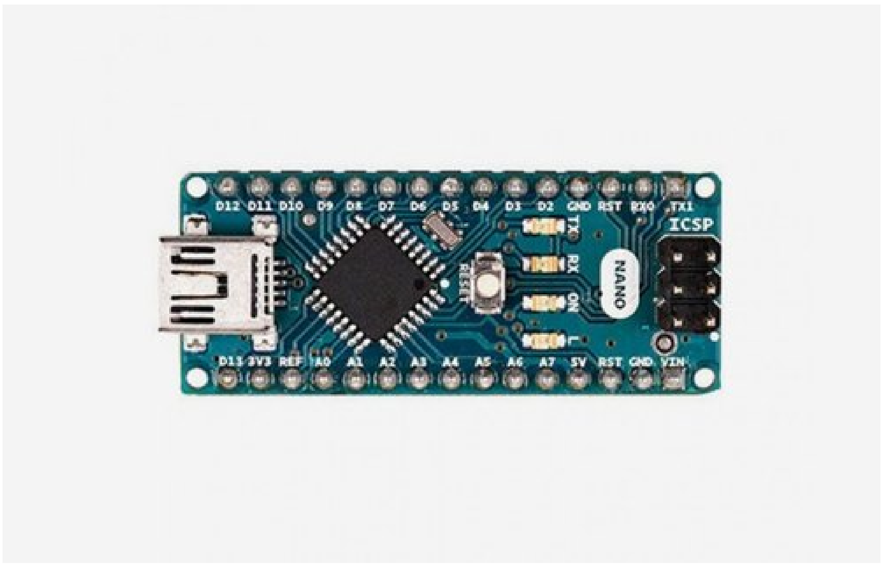

D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
TX
RX
ON
L
RESET
NANO
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

Figure 3.5: Arduino Nano Image

Arduino Nano's maximum current rating is 40mA, so the load attached to its pins shouldn't draw current more than that.
Each of these Digital Analog Pins is assigned with multiple functions but their main function is to be configured as Input/Output. Arduino Pins are acted as Input Pins when they are interfaced with sensors, but if you are driving some load then we need to use them as an Output Pin.Functions like pinMode() and digitalWrite() are used to control the operations of digital pins while analogRead() is used to control analog pins.

The analog pins come with a total resolution of 10-bits which measures the value from 0 to 5V. Arduino Nano comes with a crystal oscillator of frequency 16 MHz. It is used to produce a clock of precise frequency using constant voltage.

There is one limitation of using Arduino Nano i.e. it doesn't come with a DC power jack, which means you can not supply an external power source through a battery.This board doesn't use standard USB for connection with a computer, instead, it comes with Type-B Micro USB.

The tiny size and breadboard-friendly nature make this device an ideal choice for most applications where the size of the electronic components is of great concern.Flash memory is 16KB or 32KB that all depends on the Atmega board i.e Atmega168 comes with 16KB of flash memory while Atmega328 comes with a flash memory of 32KB. Flash memory is used for storing code.
The 2KB of memory out of total flash memory is used for a bootloader.The SRAM memory of 2KB is present in Arduino Nano.Arduino Nano has an EEPROM memory of 1KB.

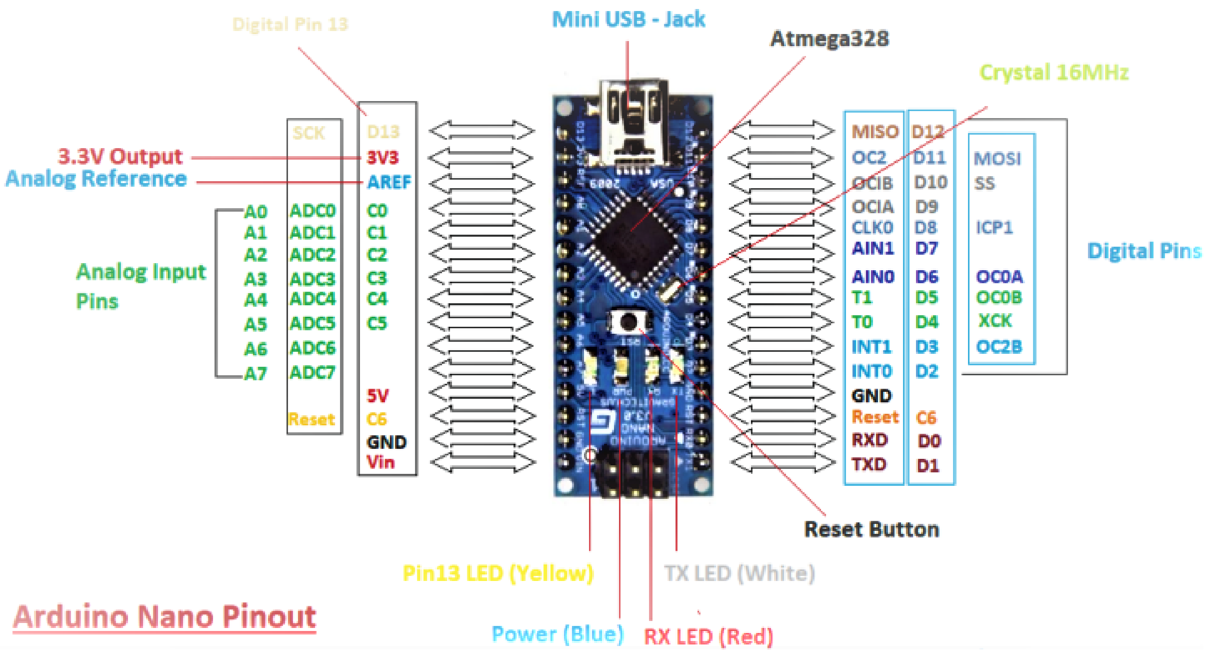
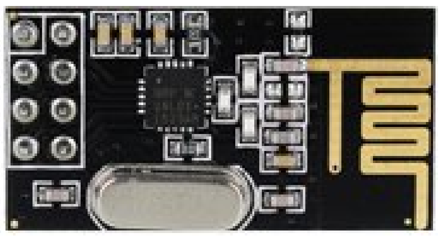


Figure 3.6: Arduino Nano Pin Layout Image

3.4.2 nRF24L01 Module

Being able to have two or more Arduino's communicate with each other wirelessly, opens up many possibilities such as remotely monitoring sensor data, controlling robots, home automation, etc.

And when it comes to an affordable but reliable 2-way RF solution, none does a better job than the nRF24L01+ transceiver module from Nordic Semiconductor.The nRF24L01+ module can often be obtained online for less than two dollars, making it one of the most affordable data communication options you can find. And the best part is that these modules are very small, allowing you to incorporate a wireless interface into almost any project.


(a) nRF24L01+ Module

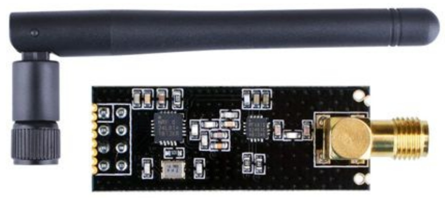

(b) nRF24L01+ P/A LNA Module

Figure 3.7: nRF24L01 Module Types

Radio Frequency

The nRF24L01+ module is designed to operate in 2.4 GHz worldwide ISM frequency band and uses GFSK modulation for data transmission. The data transfer rate is configurable and can be one of 250kbps, 1Mbps and 2Mbps.

Power

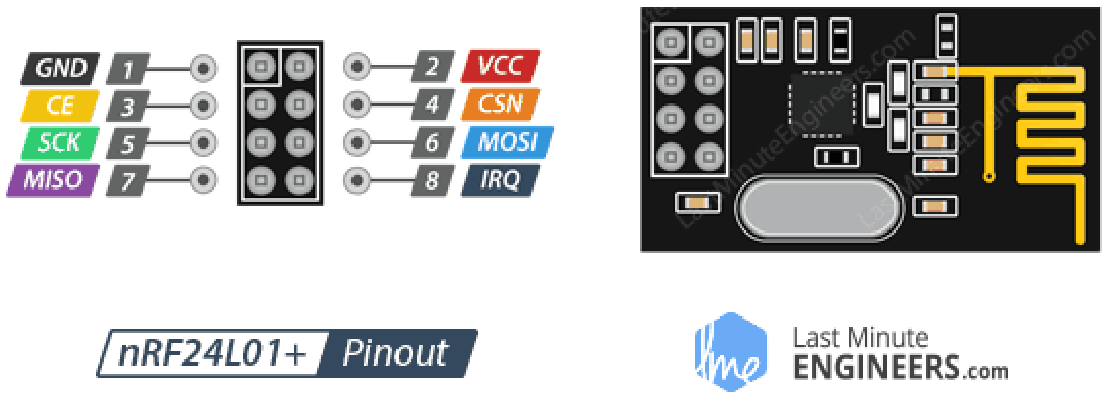
The module's operating voltage ranges from 1.9 to 3.6V, but the good news is that the logic pins are 5-volt tolerant, so you can use it with your favorite 3.3V or 5V microcontroller without worry.The module supports programmable output power i.e. 0 dBm, -6 dBm, -12 dBm or -18 dBm. At 0 dBm the module consumes only 12 mA during transmission which is less than the consumption of a single LED.

SPI Interface

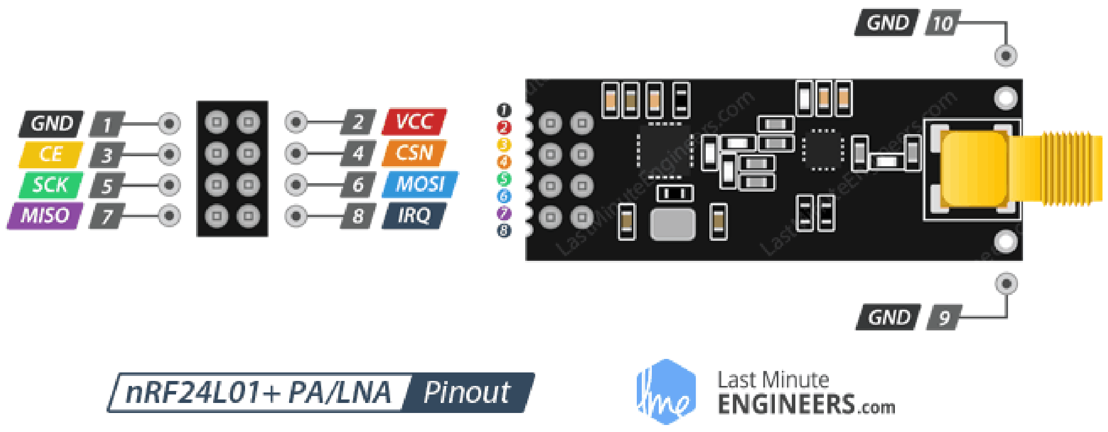
The nRF24L01+ communicates over a 4-pin SPI (Serial Peripheral Interface) with a maximum data rate of 10Mbps. All parameters such as frequency channel (125 selectable channels), output power (0 dBm, -6 dBm, -12 dBm or -18 dBm), and data rate (250kbps, 1Mbps, or 2Mbps) can be configured through the SPI interface.The SPI bus uses the concept of a master and a slave.

In most of our projects the Arduino is the master and the nRF24L01+ module is the slave.Unlike the I2C bus, the SPI bus has a limited number of slaves. That's why you can use up to two SPI slaves i.e. two nRF24L01+ modules on one Arduino.

nRF24L01+ Wireless module



nRF24L01+ PA LNA Wireless Transceiver Module



3.4.3 LiPo Battery

Lithium Polymer (AKA “LiPo”) batteries are a type of battery now used in many consumer electronics devices. They have been gaining in popularity in the radio control industry over the last few years and are now the most popular choice for anyone looking for long run times and high power.

LiPo batteries offer a wide array of benefits, but each user must decide if the benefits outweigh the drawbacks. For more and more people, they do. In my personal opinion, there is nothing to fear from LiPo batteries, so long as you follow the rules and treat the batteries with the respect they deserve.

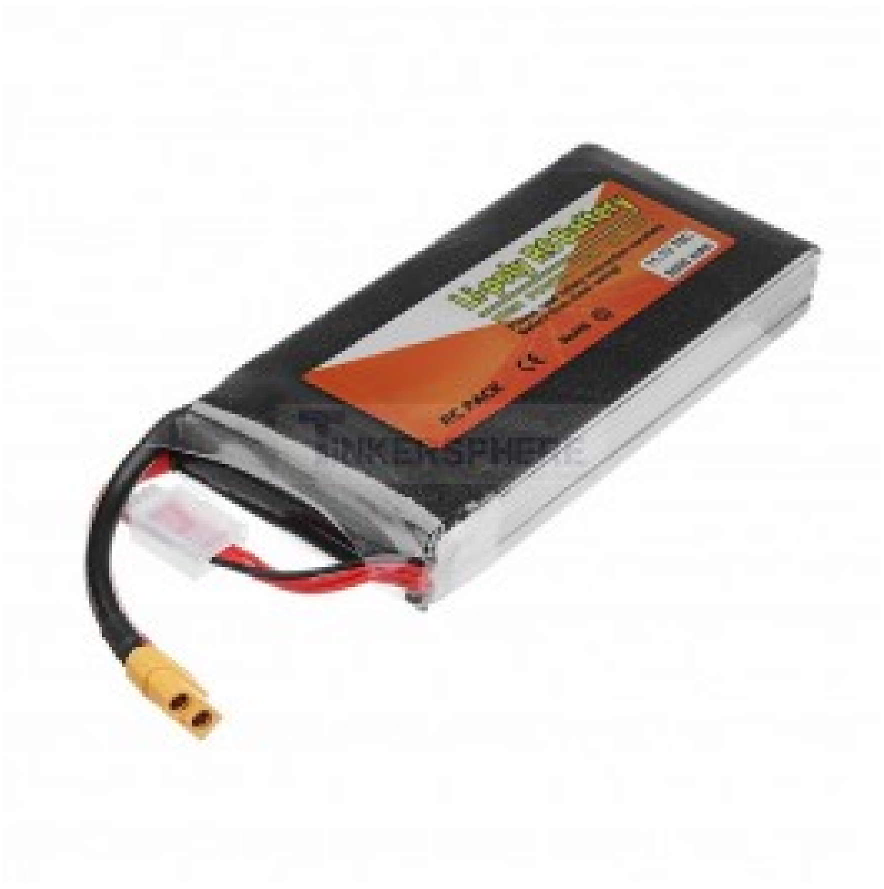


Figure 3.8: LiPo Battery 5v

3.4.4 Breadboard

A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode).

The connections are not permanent, so it is easy to remove a component if you make a mistake, or just start over and do a new project. This makes breadboards great for beginners who are new to electronics. You can use breadboards to make all sorts of fun electronics projects, from different types of robots or an electronic drum set, to an electronic rain detector to help conserve water in a garden, just to name a few.

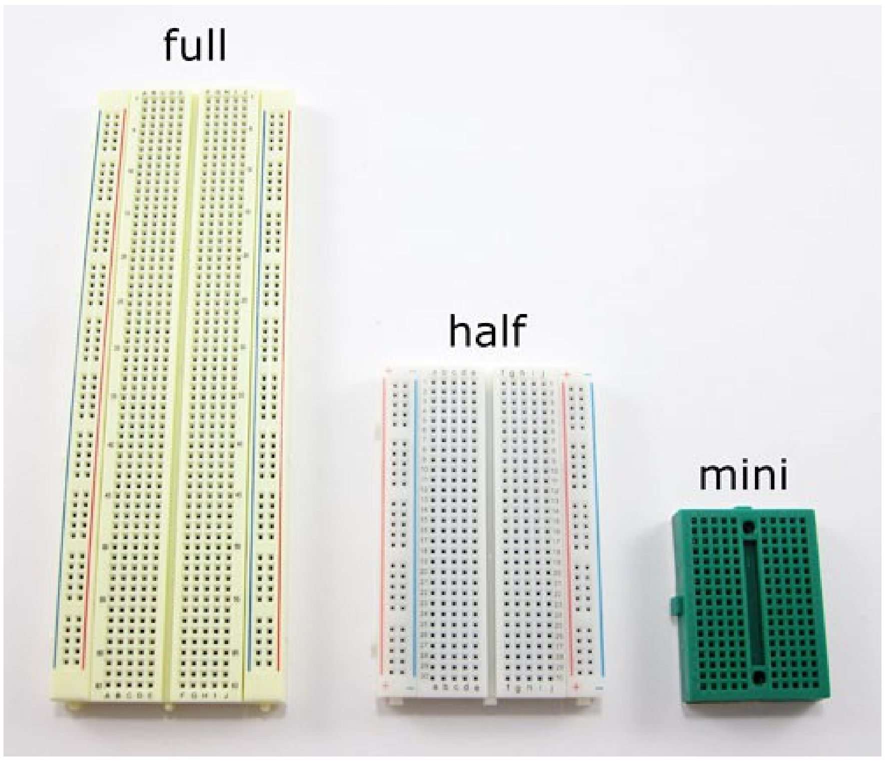


Figure 3.9: Different Sizes of Breadboard

3.4.5 LED's

LED stands for Light Emitting Diode, and this light source should not be confused with a light fixture or luminaire. An LED is a component of the entire fixture. LED lighting can also be referred to as solid-state lighting (SSL) because an LED is solid-state technology similar to the memory in your computer.

A Light Emitting Diode (LED) is a semiconductor device, which can emit light when an electric current passes through it. To do this, holes from p-type semiconductors recombine with electrons from n-type semiconductors to produce light. The wavelength of the light emitted depends on the band gap of the semiconductor material. Harder materials with stronger molecular bonds generally have wider band gaps. Aluminum Nitride semiconductors are known as ultra-wide band gap semiconductors.

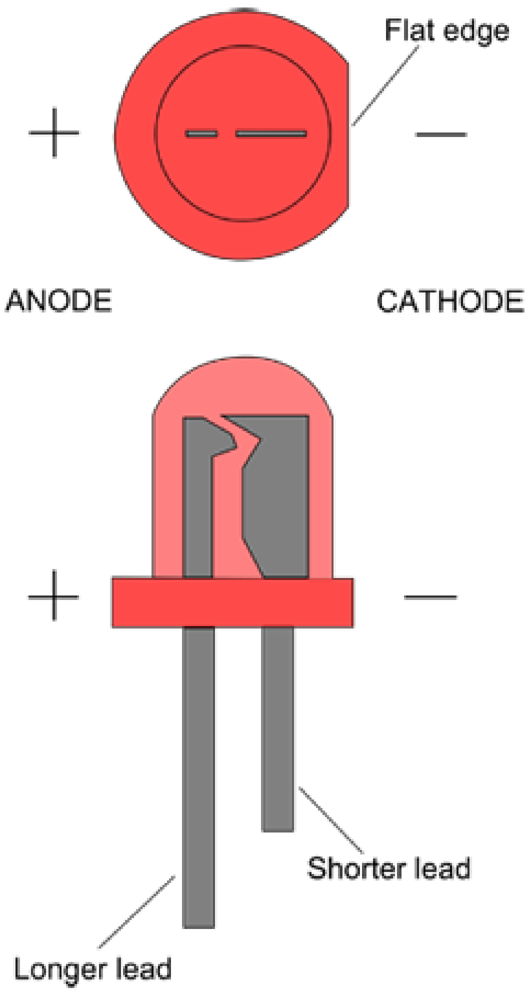


Figure 3.10: LED’s Diagram

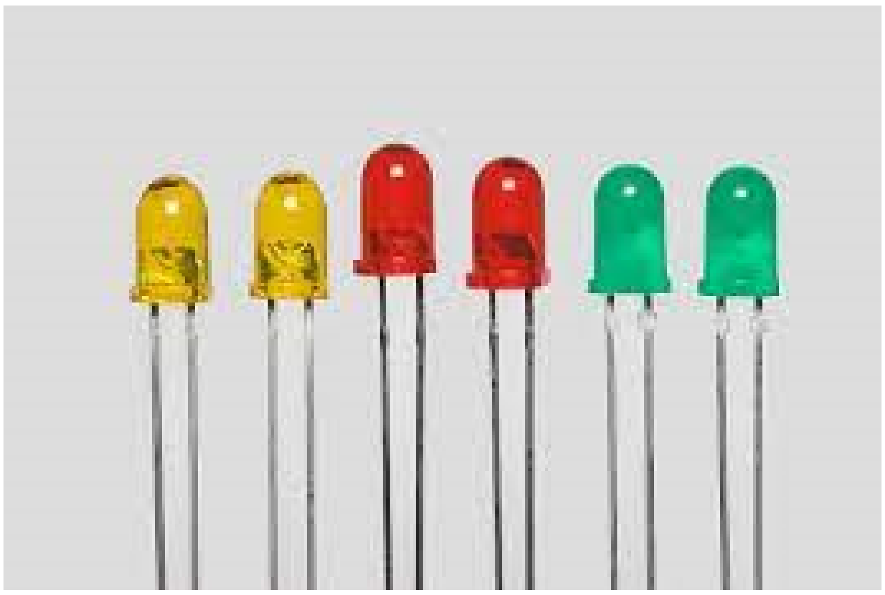


Figure 3.11: Red, Yellow and Green LED’s

3.4.6 Push Button

The push button switch is usually used to turn on and off the control circuit, and it is a kind of control switch appliance that is widely used. It is used in electrical automatic control circuits to manually send control signals to control contactors, relays, electromagnetic starters, etc.

Its characteristic is that it is installed in the machine and instrument in the process of work, most of the time is in the initial free state position, and only when needed, it is converted to the second state (position) under the action of external force. Once the external force is removed, due to With the action of the spring, the switch returns to the initial position.

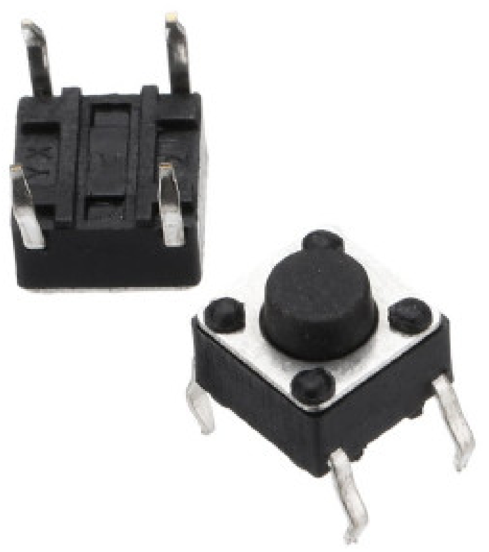


Figure 3.12: Tactile Push Button

A Push Button switch is a type of switch which consists of a simple electric mechanism or air switch mechanism to turn something on or off. Depending on model they could operate with momentary or latching action function. The button itself is usually constructed of a strong durable material such as metal or plastic.Push button switches are used throughout industrial and medical applications and are also recognisable in everyday life.

For uses within the Industrial sector, push buttons are often part of a bigger system and are connected through a mechanical linkage. This means that when a button is pressed it can cause another button to release.

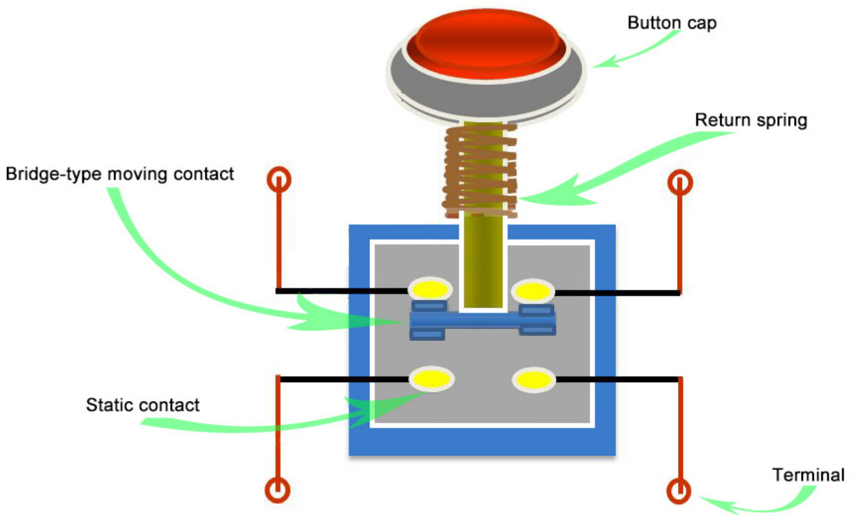


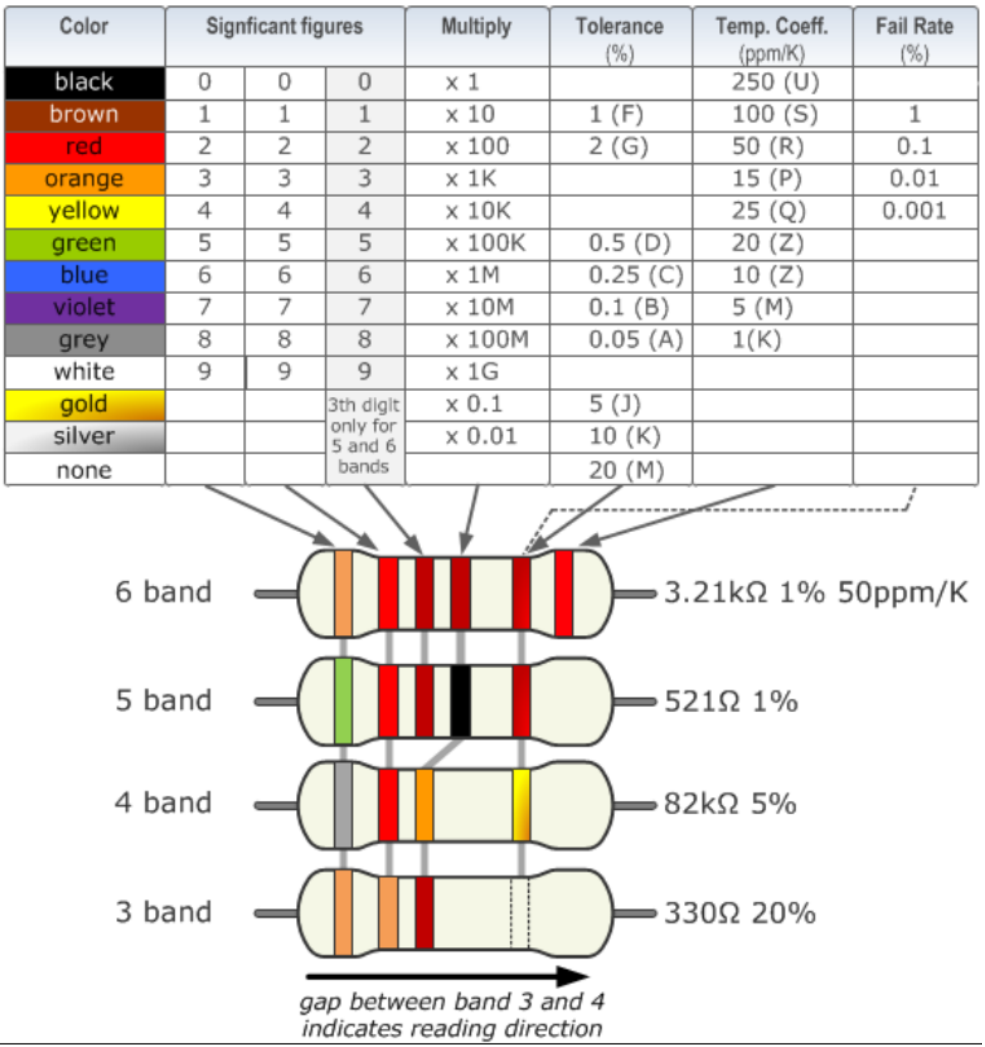
Figure 3.13: Push Button Mechanism

3.4.7 Resistance

a 1k resistor has a value of 1,000 ohms and the number we will code is 1,000. There are three steps for coding a 1k resistor.the first three colours being brown, black and red, the value of resistance is 10*100 = 1000 ohms or 1K.

The 2.2k Ohm resistor is one of the most common resistors in electronics. The four band 2.2k resistor is very easy to recognize with its' distinctive pattern of three red color bands. The 5 and 6 band versions have a pattern of red, red, black, brown, followed by the tolerance band and temperature coefficient band on the 6 band version.

33K Ohm 0.25W High Quality Metal Film Resistor (MFR) with ±1% Tolerance and Tin Plated Copper Leads. 33K Ohm Resistor Color Code: Orange, Orange, Black, Red, Brown. Resistance: 33K Ohm, Power Rating: 0.25 Watt, Approximate Maximum Current: 2.75mA.



Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
black	0	0	0	x 1		250 (U)	
brown	1	1	1	x 10	1 (F)	100 (S)	1
red	2	2	2	x 100	2 (G)	50 (R)	0.1
orange	3	3	3	x 1K		15 (P)	0.01
yellow	4	4	4	x 10K		25 (Q)	0.001
green	5	5	5	x 100K	0.5 (D)	20 (Z)	
blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
violet	7	7	7	x 10M	0.1 (B)	5 (M)	
grey	8	8	8	x 100M	0.05 (A)	1(K)	
white	9	9	9	x 1G			
gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
silver				x 0.01	10 (K)		
none					20 (M)		

Figure 3.14: Resistance Code Table

3.4.8 Male, Female Jumper Wires

Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into.

A jumper can be set to enable or disable it. Jumper wires are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering. You can use jumper wires to modify a circuit or diagnose problems in a circuit.

Figure 3.15: Male Female Jumper Wires

3.4.9 Battery Connectors

The connector is a device that joins electric circuits together. Most battery packs require more than one connector. The main battery connector is both the mechanical and electrical part that interfaces the battery to the PDA or other electronic device.

Figure 3.16: Battery Connectors

3.5 Case Study

Traffic light optimization is a complex problem. Even for single junctions there might be no obvious optimal solution. With multiple junctions, the problem becomes even more complex, as the state of one light influences the flow of traffic towards many other lights. Another complication is the fact that flow of traffic constantly changes, depending on the time of day, the day of the week, and the time of year. Roadwork and accidents further influence complexity and performance.In practice most traffic lights are controlled by fixed-cycle controllers.

A cycle of configurations is defined in which all traffic gets a green light at some point. The split time determines for how long the lights should stay in each state. Busy roads can get preference by adjusting the split time. The cycle time is the duration of a complete cycle. In crowded traffic, longer cycles lead to better performance. The offset of a cycle defines the starting time of a cycle relative to other traffic lights. Offset can be adjusted to let several lights cooperate, and for example create green waves. Fixed controllers have to be adapted to the specific situation to perform well. Often a table of time-specific settings is used to enable a light to adapt to recurring events like rush hour traffic.
Setting the control parameters for fixed controllers is a lot of work, and controllers have to be updated regularly due to changes in traffic situation. Unique events cannot be handled well, since they require a lot of manual changes to the system. Fixed controllers could respond to arriving traffic by starting a cycle only when traffic is present, but such vehicle actuated controllers still require lots of fine-tuning.

Most research in traffic light control focuses on adapting the duration or the order of the control cycle. In our approach we do not use cycles, but let the decision depend on the actual traffic situation around a junction, which can lead to much more accurate control. Of course, our approach requests that information about the actual traffic situation can be obtained by using different sensors or communication systems.
We will first describe related work on intelligent traffic light control, and then describe our car-based reinforcement learning algorithm.

3.6 Flow Chart

The Flowchart shows the communication and the corresponding result of the communication between the remote controller and the traffic signal lights. The push button is on remote controller and when pressed, sends a data to Arduino Nano which will transmits by nRF24L01 module to the traffic light side receiver nRF24L01 module and it sends data to Arduino Nano to decode which lane light should on or off as per the master gave the input through remote. The particular lane light will on or off as per the signal.

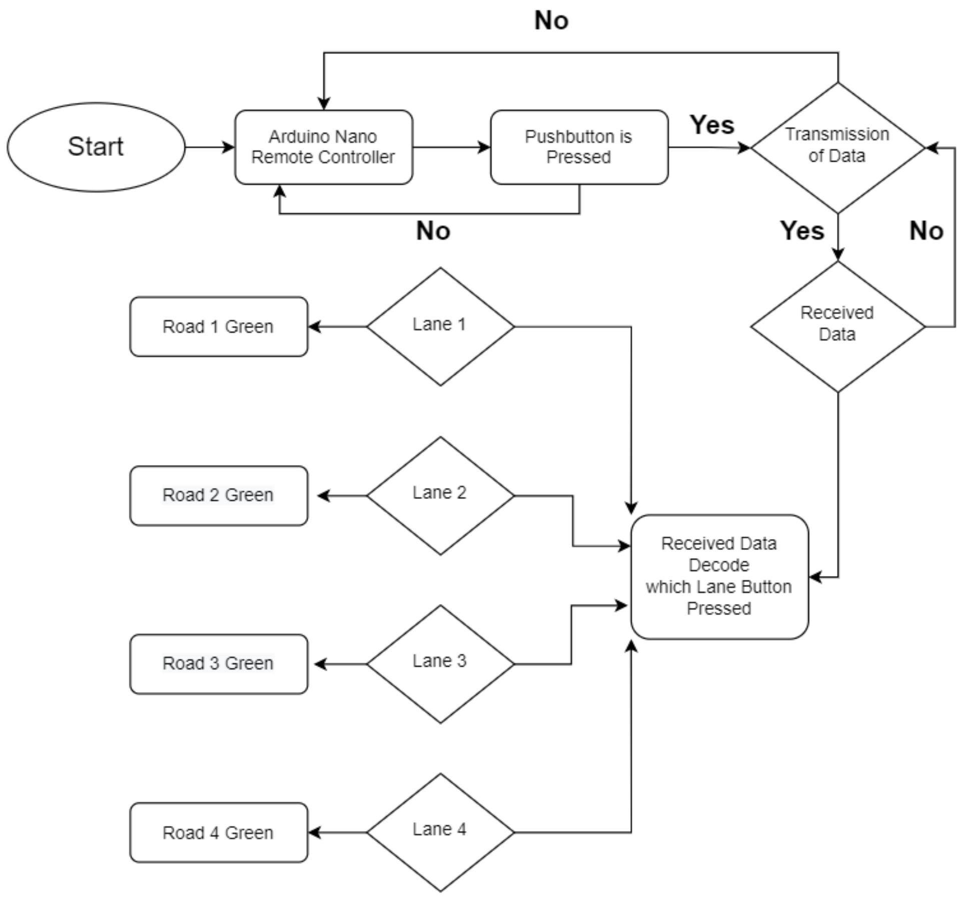


Figure 3.17: Basic Flowchart

Here total five Arduino nano is connected through radio frequency communication system. Each Arduino Nano get the same information as others but which one is get the input on and off data will act upon the traffic light of that lane and rest will be remain same as initial condition.

Chapter 4

Result and Discussion

4.1 Case and Achievement

The traffic light control system used nowadays is not adaptive, it has a fixed time cycle which repeats according to the time attributed. By using the sensor, the presence of heavy traffic in each direction can be seen by the traffic policeman and the traffic light signals will alter subsequently. The Four-way traffic light controller designing with Remote controller is the achievement and we have successfully developed it. The existing Traffic light Management System is not Flexible for various traffic densities and is Controllable by remote so, we are designing a system with Microcontroller instead of 'PLC' system for better Operation of same In this we are designing different Algorithms through which we can control traffic lights According to traffic consumption also we are providing hand held remote operations of traffic signal as per requirement.

In existing TLMS there is PLC system in which there are some drawbacks or limitations which are as follows:

-Timing for all the lanes is same , and if any of lane is highly conjucted or any lane is free then we can not control Traffic signal according to that situation.

-If we modify the existing system according to our needs it will make it too costly.

-The existing system has predefined algorithms to control and it is note controllable with any remote.

So to overcome these limitations we are designing system which is capable to ensure smooth running of Traffic signals as we require.

For this we are using microcontroller i.e. Atmega 2560 Arduino and UNO which have multiple input/output pins through which we can easily control traffic system.
In addition with microcontroller we are using RF modules to make it wireless and to control Traffic light system with remote,also we have included a RTC module to collect real time data to control traffic signal according to different timeline.
To design different algorithm we have used embedded c programming in microcontroller.

4.2 Problem

As the problem description states, develop a software/hardware automation prototype for managing the peak hours at the traffic signals. For the problem mentioned above our idea proposes hardware of a secured wireless remote control of traffic light management system (TLMS). Generally, traffic signal lights are placed in open space in about 50 meter radius and controlled by a traffic light controller which is currently based on PLC. Our idea is to replace the PLC with our cost effective TLMS which can be controlled from a wireless remote control during the peak hours.

In our idea, we are going to propose a radio frequency connection between the remote and the TLMS. Two firmware will be designed for the complete system, one will be wireless remote control side and another will be TLMS side. The remote will have a button for the user input and process the input using a microcontroller and send commands to TLMS through the RF transmitter. The TLMS will consist of a microcontroller, RF receiver, RTC and Relay board. For TLMS firmware we will design different algorithms to control traffic light signals during normal operational hours, morning peak hours, evening peak hours and remote priority based control. Beside these all features our system will have the solar based power backup so that the system will never go out of running operation at any time of the day.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Traffic jam is obstructing for trade and commerce also waste valuable time. The main reason of traffic jam can be not maintain traffic rules, faulty traffic signaling systems. Illegal parking is another reason for traffic jam. Cars, trucks and other vehicles are parked almost everywhere. Faulty traffic signaling systems, inadequate manpower and narrow road spaces and overtaking tendency of drivers create pro-longed traffic congestion and intensify sufferings of commuters keeping people motionless as well as creating suffocating condition in the streets. Also there are bus terminals not authorized by the traffic department and drivers do not go by traffic rules. VIP protocol maintaining is another reason for frequent traffic jams in the streets and divider problem in the city's different important roads also causes congestion. So if we want to overcome this problem we must install a modern traffic controller system also grow up the tendency of maintaining traffic rules. The reason of taking Traffic Controller as a project to reduce that problem, hopefully this is a good effort.

The work that we have done is that we fully give the controls to the traffic policeman. So when any lane has high traffic then that particular lane will go green by the remote controller and this is done in radio frequency transmission. The traffic lights are all connected wirelessly through radio frequency with the remote controller and all this communication is due to nRF24L01 Module which is both transmitter and receiver that is transceiver. The cost of this is also reduced to low level as compared to existing PLC based traffic light controlling system. The chances of errors are also reduced to lowest level.

5.2 Future Scope

The future scope of this is we can implemented this project on solar based electricity connection that is it can connects to the solar light. So that it doesn't require any electricity from the transmission line and it can run 24*7 without and power cut.Solar power is used to provide the power to the solar lights. So this project is very useful to the government to save the power.

The solar panel is solar photovoltaic modules use solar cells to convert light from the sun into electricity. Now-a-days, instead of using the power from the supply line for various operations, most of them are going for solar energy source, as it is cheapest. They trap the solar energy and they are using it for several applications. One such type of application is solar based light control system. This is the cheapest method and moreover we are attaining our desired target.Solar panel consists of number of silicon cells, when sun light falls on this panel it generate the voltage signals then these voltage signals given to charging circuit. Depends on the panel board size the generated voltage amount is increased.

In charging circuit the voltage signal from the board is gathered together and stored in the battery. Then the battery voltage is given to microcontroller and traffic lights. Here the microcontroller is the flash type re-programmable microcontroller in which we have already programmed depends upon the density in the road. The microcontroller output is given to driver circuit. The driver circuit used to turn ON and turn OFF the traffic lights

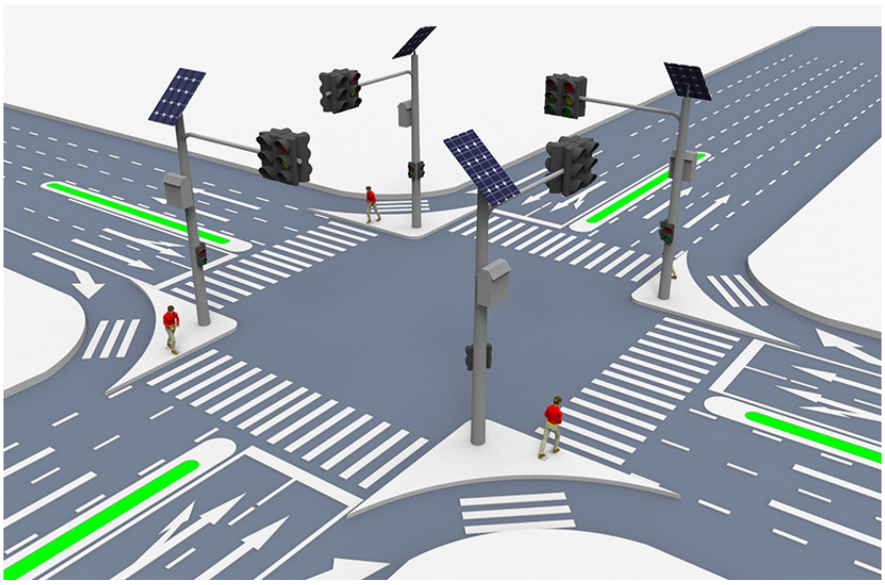


Figure 5.1: Solar power based traffic lights

References

[1] Malik Tubaishat, Yi Shang and Hongchi Shi Department of Computer Science University of Missouri – Columbia; Adaptive Traffic Light Control with Wireless Sensor Networks.https://www.researchgate.net/publication/228613082

[2] Marco Wiering, Jelle van Veenen, Jilles Vreeken, Arne Koopman. Intelligent Traffic Light Control. Institute of Information and Computing Sciences, Utrecht University. https://www.researchgate.net/publication/2942266

[3] Michael Osigbemeh, Michael Onuu, Olumuyiwa Asaolu, Design and development of an improved traffic light control system using hybrid lighting system. journal of traffic and transportation engineering (english edition) 2017.

[4] N.M.Z.Hashim, A.S.Jaafar, N.A. Ali, L.Salahuddin, N.R.Mohamad,"Traffic Light Control System for Emergency Vehicles Using Radio Frequency",IOSR Journal of Engineering, Vol. 3, Issue 7 (July. 2013)

[5] Michael R. Smith, Paul J. Davidson and Henry L. Pfister, "Emergency Vehicle Warning and Traffic Control System", United States Patent, October 4th, 1998.

[6] [Thorpe and Andersson, 1996] Thorpe, T. L. and Andersson, C. (1996). Traffic light control using sarsa with three state representations. Technical report, IBM corporation.

[7] Y. N. Udoakah, and I. G. Okure, dept. of electrical/electronics computer engr', university of uyo, uyo, akwa ibom state. Nigeria. Design and implementation of a density-based traffic light control with surveillance system, Vol. 36, No. 4, October 2017, pp. 1239 – 124. Print ISSN: 0331-8443, Electronic ISSN: 2467-8821

[8] Sajid M. Sheikh, Lebone Powder Ibo Ngebani, a smart microprocessor-based four way stop road traffic controller, University of Botswana, Department of Electrical Engineering, Gaborone, Botswana. International Journal of Electrical and Electronics Engineering (IJEEE) ISSN(P): 2278-9944; ISSN(E): 2278-9952 Vol. 7, Issue 4, Jun – Jul 2018, 9-22

[9] Mohamed S. Shehata, Jun Cai, Wael Maged Badawy, Tyson W. Burr, Muzamil S. Pervez,Robert J. Johannesson, and Ahmad Radmanesh, —Video-Based Automatic Incident Detection for Smart Roads: The Outdoor Environmental Challenges Regarding False Alarms||, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 9, NO. 2, JUNE 2008.

[10] Jifeng Xu, ”The design and research of intelligent traffic signal control system”, Beijing Technology and Business University, 2006.

[11] Pengjie Zeng, Xiaoliang Wang, Hao Li, Frank Jiang, Robin Doss, ”A Scheme of Intelligent Traffic Light System Based on Distributed Security Architecture of Blockchain Technology”, IEEE Access, vol.8, pp.33644-33657, 2020.