Mansoura University
Faculty of Engineering
Communication and computers Engineering program.

Project Report on

Automated Car Detection for Security purpose

[Under Supervision of]

Prof. Mohammed Abdel Azeem

Eng. Ahmed Saad Radwan

Eng. Mohammed Reda Abo elfotoh

[Submitted by]

Karem Reda Elbosaty | **Ahmed Abdelkader Abdel Ghaffar**
Amr Mostafa Zaher | **Ahmed Mohamed Elshahawy**
Islam Awad Mohammed | **Ahmed Abdel Aleem Hamed**
Meral Hossam Elsayed | **Abdelrahman Emad Murad**
Eyad Abdallah Eldesoky | **Motaz Mahmoud Eltantawy**
Alaa Essam Abdelghani | **Anas Gamal Elnagar**

بسم الله الرحمن الرحيم

الحمد لله الذي بفضله قد وهبنا العلم وجعله لنا نورا نهتدي به، والصلاة والسلام علي رسول الله نبينا محمد صلي الله عليه وسلم

الشكر والعرفان، بمجهود كل من ساهم معنا في هذا العمل الذي لا يحصي، وكل من ساهم في إنجازه بالشكل الذي يليق به ونود ان نوجه شكرنا خاصه الي

الدكتور: محمد عبد العظيم

استاذ بقسم هندسه الحاسبات والتحكم الالي

عميد كليه الهندسة

المهندس: احمد سعد رضوان

معيد بقسم هندسه الحاسبات والتحكم الالي

كليه الهندسة - جامعه المنصورة

المهندس: محمد رضا أبو الفتوح

معيد بقسم هندسه الحاسبات والتحكم الالي

كليه الهندسة - جامعه المنصورة

وذلك لما بذلوه من جهد على مدي مراحل المشروع وتوجيهنا بالملاحظات والمعلومات التي افادتنا ف اتمام العمل

وكذلك اداره برنامج هندسه الاتصالات والمعلومات لما بذلوه من جهد في مساعدتنا في اتمام المشروع طبقا للجدول الزمني المخطط له

Abstract

Automation – control of systems where the human work in production of goods and services is reduced. One such automation system which has been increased in recent times is the Automatic gate barrier systems. It is a convincing solution to the manual gate barrier method which is employed by the security guards at universities and other higher organizations. Implementation of this system will cause less traffic congestion. This system consists of two parts for automation of gate barrier – one by using RFID technology and the other by car license plate detection. When the vehicle was captured by camera. And then it compares with the database that is made in SQL server. During comparing it uses an edge detection and compares pixel to pixel templates that are already define in the program. And after matching the number plate it requests for the RFID card. For this RF receiver is also interface with it when RFID card is sweep on the RF receiver it detects its ID and match with the ID that is already stored in the database. Gate will open automatically.

.

Table of Contents

List of Figures

List of Table

Chapter 1

1 introduction

1.1 Background

The development of technologies in the today's world has made all the manufacturing fields industry automated. In general automation is defined as technology concerned with performing a process by means of programmed commands combined with automatic feedback control to ensure proper execution of the instructions. The resulting system is capable of operating without human intervention. The development of this technology has become increasingly dependent on the use of computers and computer related technologies. Consequently, automated systems have become increasingly sophisticated and complex. Advanced systems represent a level of capability and performance that surpass in many ways the abilities of humans to accomplish the same activities.

The purpose of this project is to develop an Automatic gate control application which recognizes two important elements from car and driver at entrance gate and take action to let car in or not first we will talk about **license plates from cars.**

PC with camera, catches image frame when a car is detected, the camera takes a picture of the car. The picture is then sent to the Open CV to filter unwanted noise (illumination, blur) etc. leaving the characters of the number plate. The characters can then be identified using algorithm to recognition the characters with number plates coming in various shapes, sizes, fonts and colors, the algorithms need to be refined to increase its efficiency.

Then we will talk about the second element of our project that are very important to the car to get access to enter gate.

RFID (Radio Frequency Identification)

Radio Frequency Identification is being used all over the world from a very long time. Recently an annual growth of 24% was experienced in the RFID implementation. It is used in many different applications which are numerous. Some of the real-world examples where RFID is used are logistics and supply chain management, attendance tracking system, access control, Library systems, laundry management, real time location systems and many more. Quite simply, this technology works on radio waves. RFID mainly consists of three components RFID reader, RFID tag and antenna. The tags receive data and the reader transmits radio waves to the tag. The reader could be a stationary or a handheld reader. The tags can be classified into three types based on the power supply they are active tags, passive and semi-passive tags. These tags and reader contain an antenna which enable them to transmit receive and respond to the signals. They also store information which is used to recognize the item attached. The other benefits of RFID are that the tags have a read-write option unlike in barcodes, i.e.; they can be read or modified only by its authorized users. RFID frequencies range from low frequency to microwave.

Most of the research on system targets hardware that uses programming by Open CV and processing such as microprocessors and microcontroller in our project we use components to interface them together in our project (Open CV, Arduino, camera, mechanical design,).

1.2 Problem Statement

Entry to a gated community or an institution is typically a manual process. Manual processes also often choke main entrances with traffic during peak travel times. Traffic management can be complicated too. Humans are not very accuracy like computer systems so that we are design the project to be more convincingly and safety in entrance gate with a system that prevent (automatically, security and shortest time to do the job).

1.3 Project Aim

Project discuss on automatic gate control system based on vehicle license plate recognition and RFID technology. The aim of this project is to develop and implement an automatic gate control system that will increase convenience and security at entrance of all the important places that require protection and Security. The system will be able to works automatically without need human beings and also the system will be able to recognizes license plates from cars at entrance gate and take an action to let cars enter or not.

1.4 Objective

1.4.1 General

- prevent more security and safety
- saving time
- minimizing numbers of the labor such as (gate guard)

1.4.2 special

- To study similar systems and its characteristics and problems
- To design an Open CV based image processing system that capture the image of plate and perform characters filtering and recognition to extract the text of the plate.
- To design a gate control system using RFID technology
- To implement a database system to compare the extracted text with the plate numbers in the database

1.5 Research question

- How to study and describe three different element (RFID using Arduino, Open CV, database) and how to connect between them using serial communication.
- How to study how an Open CV and RFID can be programmed so as to be able to send a notification alert to each other by serial communication.
- How to develop and build a prototype of the surveillance system based on the Arduino.
- How to design and implement a motion-detecting and tracking system for real-time video analysis.

1.6 Scop of Project

1.6.1 SWOT Analysis

Strength

- Good teamwork
- Enthusiasm for idea from all team members
- The ability to produce new ideas over the duration of the work, as a result of the cooperation of members of the team.

Weakness

- Absence of certain skills
- Time Constraints
- Lack of communication

Opportunity

- Many Resources surrounding to take advantage of.
- Creative freedom
- A new and different idea of the market
- Important field

Threats

- Competitive environment
- Lack of skills may lack to weather work
- Separation of team members after graduation.

1.6.2 Content Scop

This project is focused on developing a surveillance system that contains the RFID section which serves as the key, a motion detection sensor that detects motion and camera that responds speedily by capturing an image and relaying it to an administrator device. The system will require Arduino module. It will, therefore, use these systems together with a suitable program script to accomplish a real-time surveillance system as desired.

1.6.3 Time Scop

This project was based on both theoretical data and methodological data.

Searching Stage

Process	Duration
Research	3 Weeks

Table 1.6-1 Searching State

Hardware Stage

Process	Duration
Component	2 Weeks
Testing	1 Weeks

Table 1.6-2 Hardware Stage

Software Stage

Process	Duration
Coding	4 Weeks
Coding Testing	2 Weeks
Modification	1 Weeks

Table 1.6-3 Software Stage

Testing stage

Process	Duration
Test	1 Weeks

Table 1.6-4 Testing Stage

1.7 Project Model

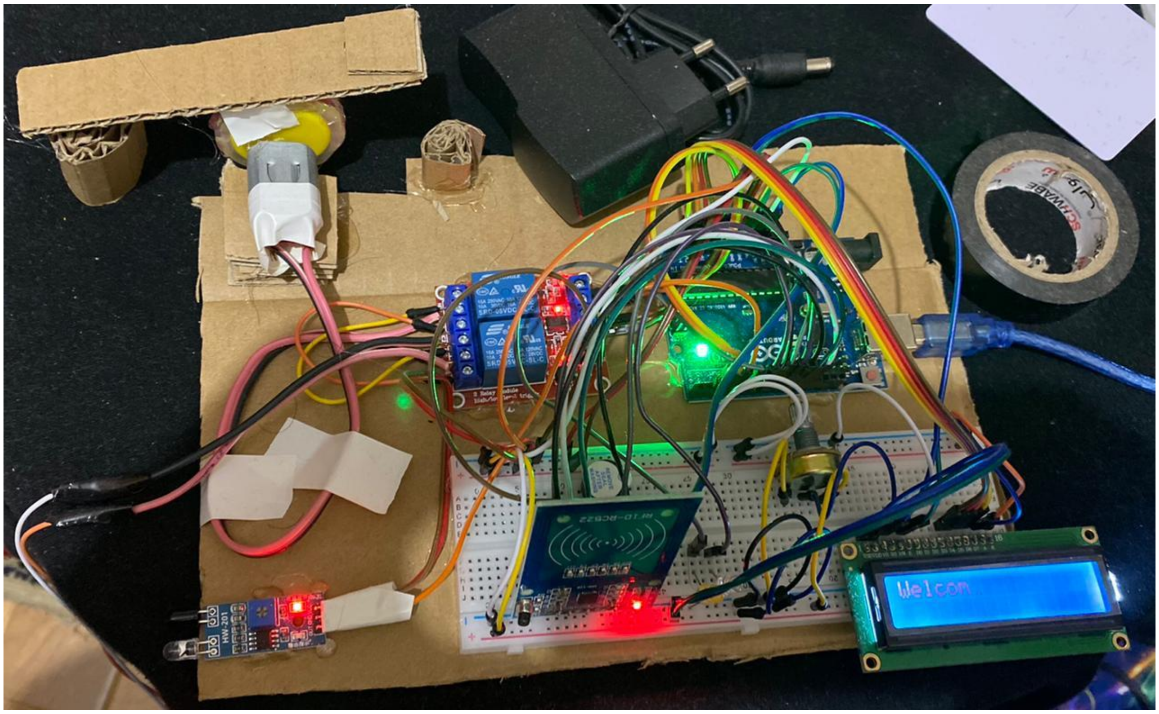


Figure 1.6-1 project model

Chapter 2

2.1 Gate Control System

A gate controller system is primarily used to open the gate, close the gate and keep the gate open if an obstruction is detected. The modern-day gate controllers can be connected to various devices and are very flexible. Previously scenario behind the gate was that the guard, unlock the gate and slid or push it. After the vehicle is moved, the guard has to close the gate and lock it. It is basically a waste of time so we will design an automatic gate control system. In this proposed system, IR Sensor will detect for there is car or not then after detection the process will go on. Camera will capture palate automatic and will check for this on data base and after that will required for RFID if the two elements are matched the gate will open.   The basic aim of Automation concept is to reduce the man power & to increase the accuracy of the system. Figure 2.1-1 shows an automatic gate barrier system.

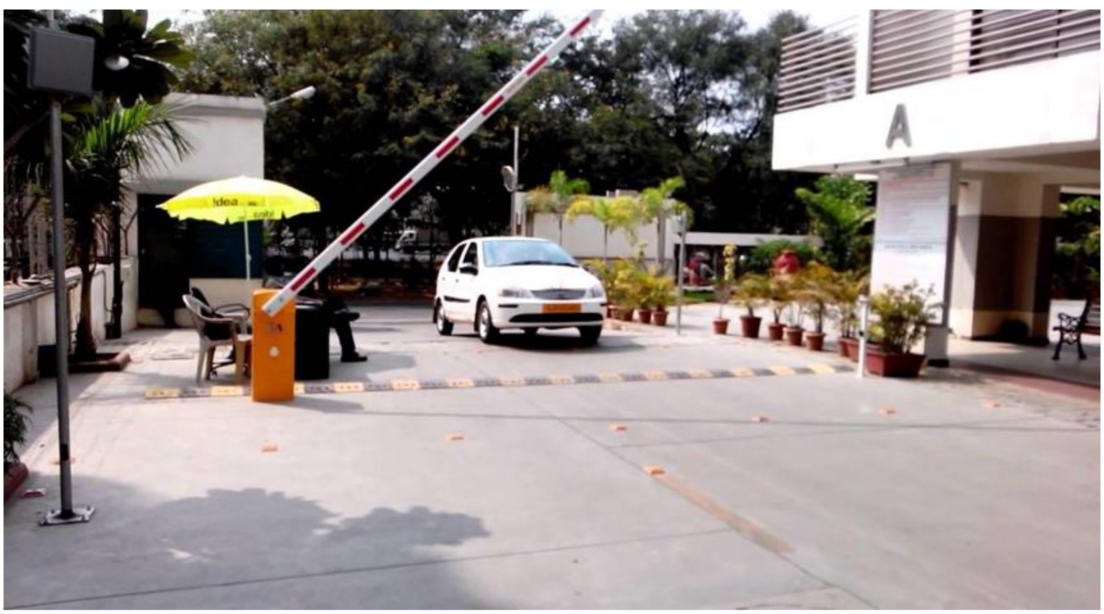


Figure 2.1-1 An automatic barrier system

2.2 System Block Diagram

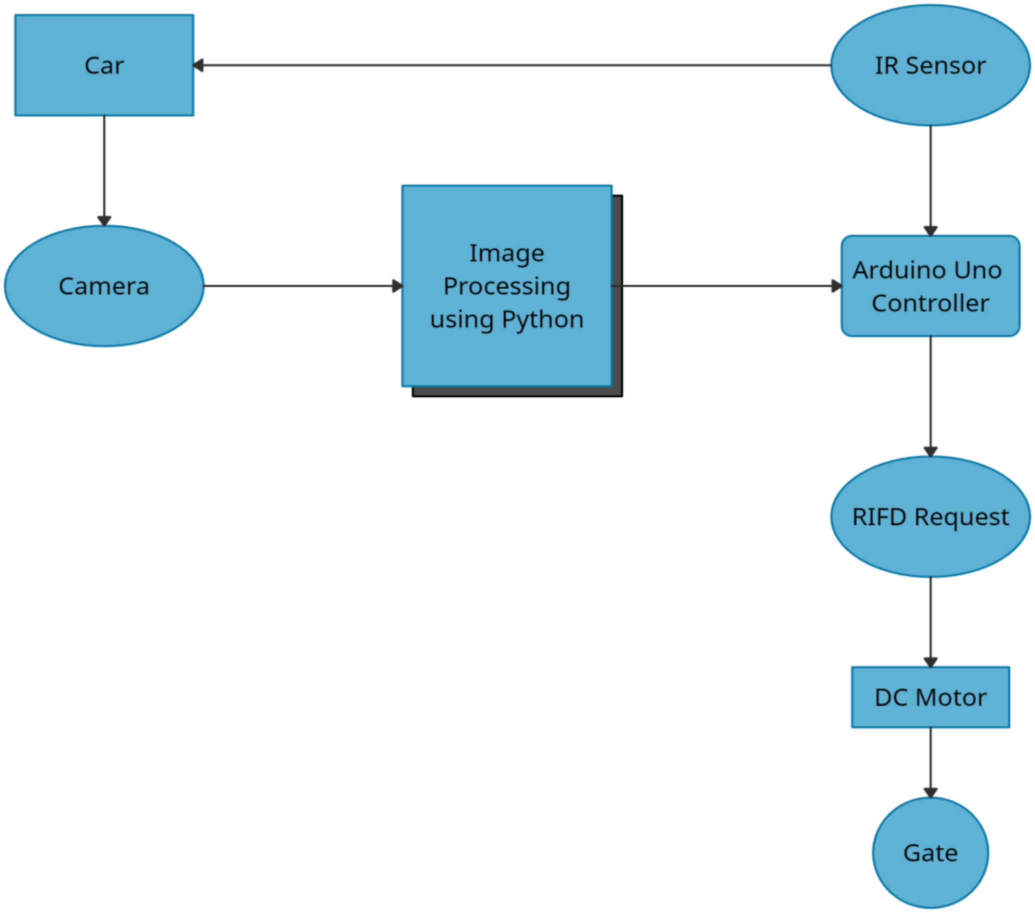


Figure 2.2-1 Block Diagram

2.3 Component

The parking gate system consists of the components listed below.

1. **Controller**

   The user can program the controller for the required specifications upon receiving the input from various input devices connected to a system. As a result, the output is sent to perform the necessary action.

Different scenarios can be programmed and can function at the same time.

2. **Input**

There are many types of input devices available to generate a signal when a vehicle is detected. These sensors can also detect any obstacle when the gate is moving and send a signal to the controller to perform necessary actions.

3. **Output**

The output is the motion of the gate which is connected to a motor. The motor should be capable to handle the movement of the gate.

2.4 Input Component

- ➢ Arduino UNO microcontroller board

2.4.1 Arduino UNO microcontroller

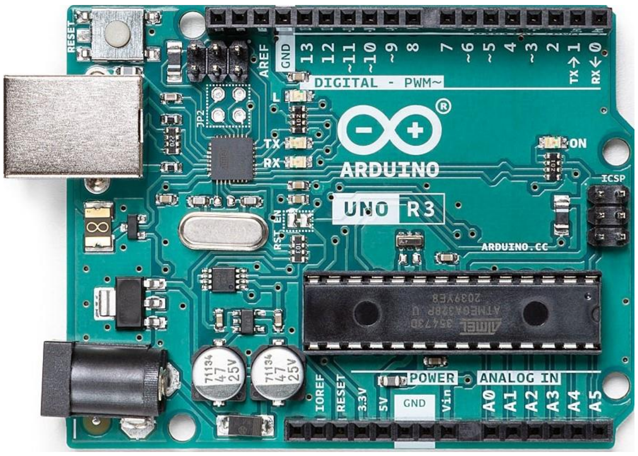


Figure 2.4-1 Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital Input /Output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16MHz ceramic resonator, USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino Boards.

Arduino Feature

Microcontroller	ATmega328
Operating Voltage	5 V
Input Voltage(recommended)	7-12 V
Input Voltage(limits)	6-20 V
Digital I/O Pins	14
Analog Input Pins	6
Dc Current per I/O Pin	40 mA
Dc Current per I/O for 3.3 V Pin	50 mA
Flash Memory	32 KB (A Tmega328) OF WHICH 0.5 KB used by bootloader
SRAM	2 KB (A Tmega328)
EEPROM	1 KB (A Tmega328)
Clock Speed	16 MHz

Table 2.4-1 Arduino Feature

*2.4.2 16*2 LCD*

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 combined with polarizers. Liquid crystals do not emit light directly,[1] instead using a backlight or reflector to produce images in color or monochrome.[2] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and seven-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made from a matrix of small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the color of the backlight, and a character negative LCD will have a black background with the letters being of the same color as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

*Figure 2.4-2 LCD 16*2*

Features

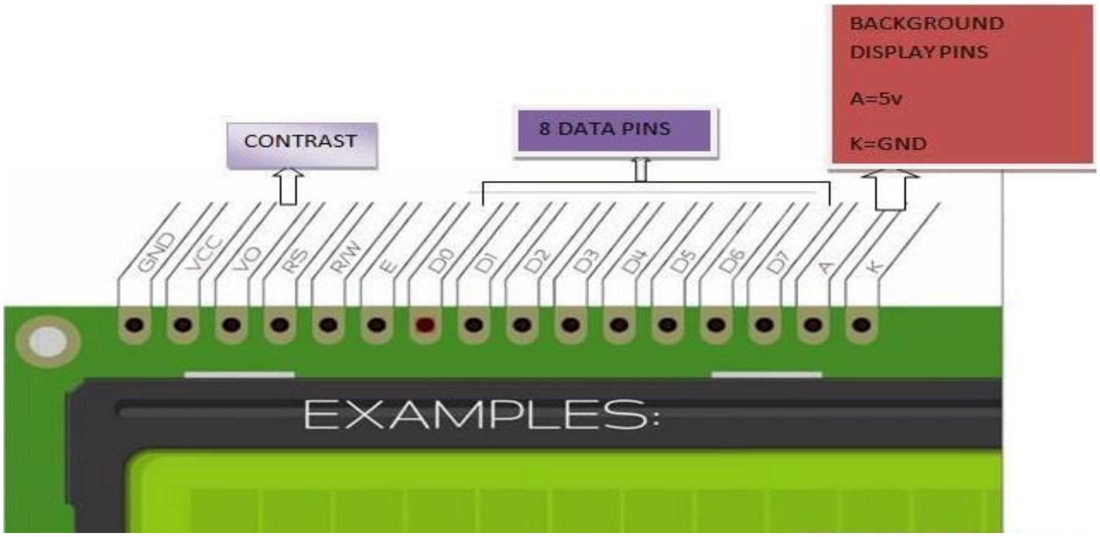
- The operating voltage of this display ranges from 4.7V to 5.3V
- The display bezel is 72 x 25mm
- The operating current is 1mA without a backlight
- PCB size of the module is 80L x 36W x 10H mm
- HD47780 controller
- LED color for backlight is green or blue
- Number of columns – 16
- Number of rows – 2
- Number of LCD pins – 16
- Characters – 32
- It works in 4-bit and 8-bit modes
- Pixel box of each character is 5×8 pixel
- Font size of character is 0.125Width x 0.200height

Electrical Characteristics

Item	Symbol	Condition
Input voltage	VDD	VDD = + 3 v
Supply current	IDD	VDD = + 5 v
Recommended lc driving voltage for normal temperature version module	VDD to v0	-20°c to 70°c
Led forward voltage	VF	25 °c
Led forward current	IF	25 °c
El power supply current	IEL	Vel = 110 vac, 400 HZ

Table 2.4-2 Electrical Characteristics

Interface Pin Function



PIN	N0.	Function
1	VSS	Ground
2	VDD	+ 3 V or + 5 V
3	V0	Contrast adjustment
4	RS	H/L register select signal
5	R/W	H/L read/write signal
6	E	H/L data bus line
7	DB0	H/L data bus line
8	DB1	H/L data bus line
9	DB2	H/L data bus line
10	DB3	H/L data bus line
11	DB4	H/L data bus line
12	DB5	H/L data bus line
13	DB6	H/L data bus line
14	DB7	H/L data bus line
15	A	Power supply for LED (4.2 V)
16	K	Power supply for B/L (0 V)

Table 2.4-3 Lcd Pin Config

2.4.3 13.56 MHZ RFID Reader

A Reader consists of a Radio Frequency module and an antenna which generates high frequency electromagnetic fields. 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.

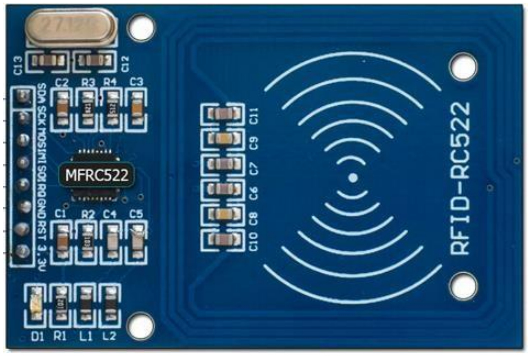


Figure 2.4-3 RIFD Module

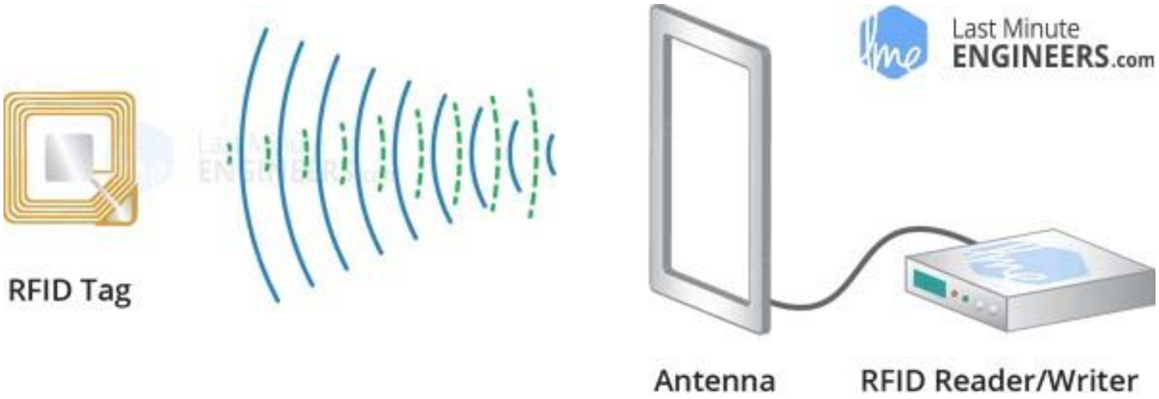


Figure 2.4-4 RIFD Schema

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. The RC522 RFID module based on MFRC522 IC from NXP 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 some sort of secret message in it. RC522 RFID Reader Writer Module with Tag Card and FOB Key Tag 25 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.

Specifications

Name	Parameter
Frequency Range	13.56 MHZ Ism Band
Host Interface	SPI / I2c / UART
Operating Supply Voltage	2.5 V To 3.3 V
Max. Operating Current	13-24ma
Min. Current (Power Down)	10µa
Logic Inputs	5v Tolerant
Read Range	5 Cm

Table 2.4-4 RIFD Specification

Pinout configuration

PIN	Name	Description
1	VCC	Used to Power the module, typically 3.3V is used
2	RST	Reset pin – used to reset or power down the module
3	Ground	Connected to Ground of system
4	IRQ	Interrupt pin – used to wake up the module when a device comes into range
5	MISO/SCL/Tx	MISO pin when used for SPI communication, acts as SCL for I2c and Tx for UART.
6	MOSI	Master out slave in pin for SPI communication
7	SCK	Serial Clock pin – used to provide clock source
8	SS/SDA/Rx	Acts as Serial input (SS) for SPI communication, SDA for IIC and Rx during UART

Table 2.4-5 RFID Pin Config

2.4.4 IR Sensor

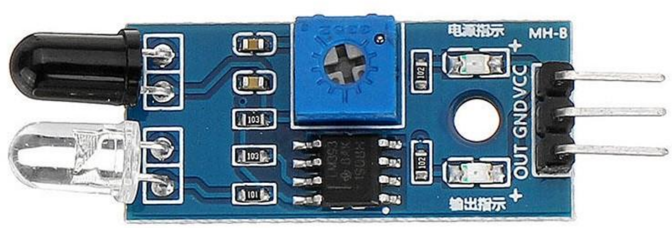


Figure 2.4-5 IR Sensor

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode. Photodiode is sensitive to IR light of the same wavelength which is emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received. There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LEDs of specific wavelength used as infrared sources. The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibers. Optical components are used to focus the infrared radiation or to limit the spectral response.

Types of IR Sensor

There are two types of IR sensors are available and they are,

- Active Infrared Sensor
- Passive Infrared Sensor

Active Infrared Sensor

Active infrared sensors consist of two elements: infrared source and infrared detector. Infrared sources include the LED or infrared laser diode. Infrared detectors include photodiodes or phototransistors. The energy emitted by the infrared source is reflected by an object and falls on the infrared detector.

Passive Infrared Sensor

Passive infrared sensors are basically Infrared detectors. Passive infrared sensors do not use any infrared source and detector. They are of two types: quantum and thermal. Thermal infrared sensors use infrared energy as the source of heat. Thermocouples, pyroelectric detectors and bolometers are the common types of thermal infrared detectors. Quantum type infrared sensors offer higher detection performance. It is faster than thermal type infrared detectors. The photo sensitivity of quantum type detectors is wavelength dependent.

IR Sensor Working Principle

There are different types of infrared transmitters depending on their wavelengths, output power and response time. An IR sensor consists of an IR LED and an IR Photodiode, together they are called as Photo Coupler or Optocoupler.

IR Transmitter or IR LED

Infrared Transmitter is a light emitting diode (LED) which emits infrared radiations called as IR LED’s. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

The picture of an Infrared LED is shown below.

Figure 2.4-6 IR Transmitter

IR Receiver or Photodiode

Infrared receivers or infrared sensors detect the radiation from an IR transmitter. IR receivers come in the form of photodiodes and phototransistors. Infrared Photodiodes are different from normal photo diodes as they detect only infrared radiation. Below image shows the picture of an IR receiver or a photodiode.

Figure 2.4-7 IR Receiver

Different types of IR receivers exist based on the wavelength, voltage, package, etc. When used in an infrared transmitter – receiver combination, the wavelength of the receiver should match with that of the transmitter. The emitter is an IR LED and the detector is an IR photodiode.
The IR photodiode is sensitive to the IR light emitted by an IR LED. The photo-diode's resistance and output voltage change in proportion to the IR light received. This is the underlying working principle of the IR sensor.

When the IR transmitter emits radiation, it reaches the object and some of the radiation reflects back to the IR receiver. Based on the intensity of the

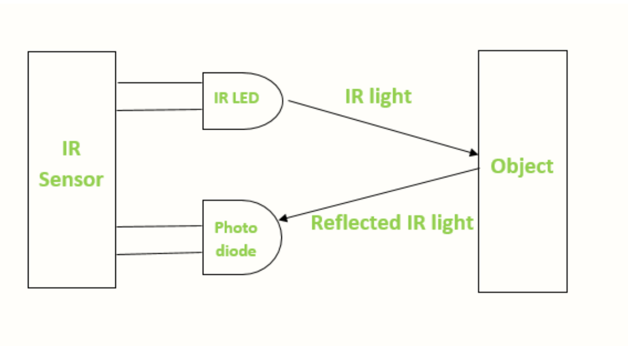


Figure 2.4-8 IR Diagram

reception by the IR receiver, the output of the sensor defines.

specifications

- The operating voltage is 5VDC
- I/O pins – 3.3V & 5V
- Mounting hole
- The range is up to 20 centimeters
- The supply current is 20mA
- The range of sensing is adjustable
- Fixed ambient light sensor

Pinout configuration

Pin	Name	Description
1	VCC	Pin is power supply input
2	GND	Pin is power supply ground
3	OUT	is an active-high o/p

Table 2.4-6 IR pin Config

2.4.5 Dc motor

A DC motor in simple words is a device that converts electrical energy (direct current system) into mechanical energy. It is of vital importance for the industry today, and is equally important for

Figure 2.4-9 DC Motor

engineers to look into the working principle of DC motor in details that has been discussed in this article. In order to understand the operating principle of DC motor we need to first look into its constructional feature.

The very basic construction of a DC motor contains a current carrying armature which is connected to the supply end through commutator segments and brushes. The armature is placed in between north south poles of a permanent or an electromagnet as shown in the diagram above. As soon as we supply direct current in the armature, a mechanical force acts on it due to electromagnetic effect of the magnet. Now to go into the details of the operating principle of DC motor it's important that we have a clear understanding of Fleming's left-hand rule to determine the direction of force acting on the armature conductors of DC motor.

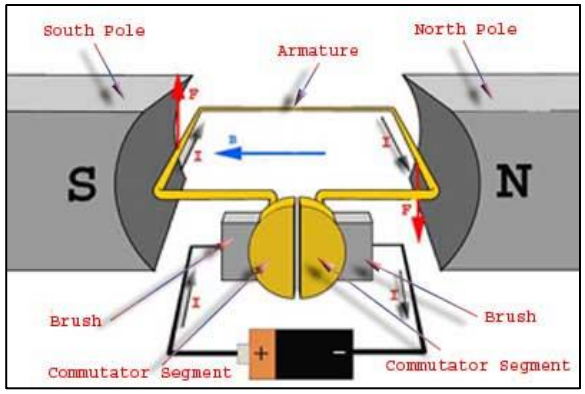


Figure 2.4-10 Construction of DC Motor

If a current carrying conductor is placed in a magnetic field perpendicularly, then the conductor experiences a force in the direction mutually perpendicular to both the direction of field and the current carrying conductor. Fleming's left-hand rule says that if we extend the index finger, middle finger and thumb of our left hand perpendicular to each other, in such a way that the middle finger is along the direction of current in the conductor, and

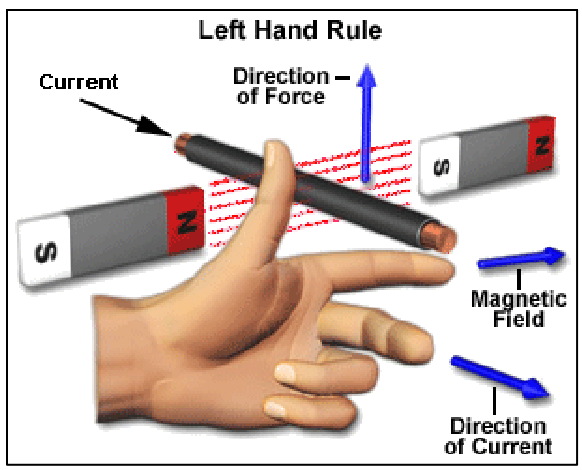


Figure 2.4-11 Left Hand Rule

index finger is along the direction of magnetic field i.e., north to south pole, then thumb indicates the direction of created mechanical force.

2.4.6 Relay

Figure 2.4-12 relay 2 channel

A relay is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof. Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations. The traditional form of a relay uses an electromagnet to close or open the contacts, but other operating principles have been invented, such as in solid-state relays which use semiconductor properties for control without relying on moving parts. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called protective relays. Latching relays require only a single pulse of control power to operate the switch persistently. Another pulse applied to a second set of control terminals, or a pulse with opposite polarity, resets the switch, while repeated pulses of the same kind have no effects. Magnetic latching relays are useful in applications when interrupted power should not affect the circuits that the relay is controlling.

Features

1. **Physical size and pin arrangement**
   If you are choosing a relay for an existing PCB you will need to ensure that its dimensions and pin arrangement are suitable. You should find this information in the supplier's catalogue.

2. **Coil voltage**
   The relay's coil voltage rating and resistance must suit the circuit powering the relay coil. Many relays have a coil rated for a 12V supply but 5V and 24V relays are also readily available. Some relays operate perfectly well with a supply voltage which is a little lower than their rated value.

3. **Coil resistance**
   The circuit must be able to supply the current required by the relay coil. You can use Ohm's law to calculate the current:

   $$\text{Relay Coil Current} = \frac{\text{Supply Voltage}}{\text{Coil Resistance}}$$

   For example: A 12V supply relay with a coil resistance of 400 Ω passes a current of 30mA. This is OK for a 555 timer IC (maximum output current 200mA), but it is too much for most ICs and they will require a transistor to amplify the current.

4. **Switch ratings (voltage and current)**

   The relay's switch contacts must be suitable for the circuit they are to control. You will need to check the voltage and current ratings. Note that the voltage rating is usually higher for AC, for example: "5A at 24V DC or 125V AC".

5. **Switch contact arrangement (SPDT, DPDT etc.)**

   Most relays are SPDT or DPDT which are often described as "single pole changeover" (SPCO) or "double pole changeover" (DPCO). For further information please see the page on switches.

Advantages of relays

- Relays can switch AC and DC, transistors can only switch DC.
- Relays can switch high voltages, transistors cannot.
- Relays are a better choice for switching large currents (> 5A).
- Relays can switch many contacts at once.

Disadvantages of relays

- Relays are bulkier than transistors for switching small currents.
- Relays cannot switch rapidly (except reed relays), transistors can switch many times per second.
- Relays use more power due to the current flowing through their coil.
- Relays require more current than many chips can provide, so a low power transistor may be needed to switch the current for the relay's coil.

Dual-Channel Relay Module Specifications

- Supply voltage – 3.75V to 6V
- Trigger current – 5mA
- Current when relay is active - ~70mA (single), ~140mA (both)
- Relay maximum contact voltage – 250VAC, 30VDC
- Relay maximum current – 10A

Pinout configuration

PIN	Name	Description
1	JD-V_{CC}	Input for isolated power supply for relay coils
2	V_{CC}	Input for directly powering the relay coils
3	GND	Connected to Ground of system
4	GND	Connected to Ground of system
5	IN1	Input to activate the first relay
6	IN2	Input to activate the second relay
7	VCC	V_{CC} to power the optocouplers, coil drivers, and associated circuitry

Table 2.4-7 Relay Pin Config

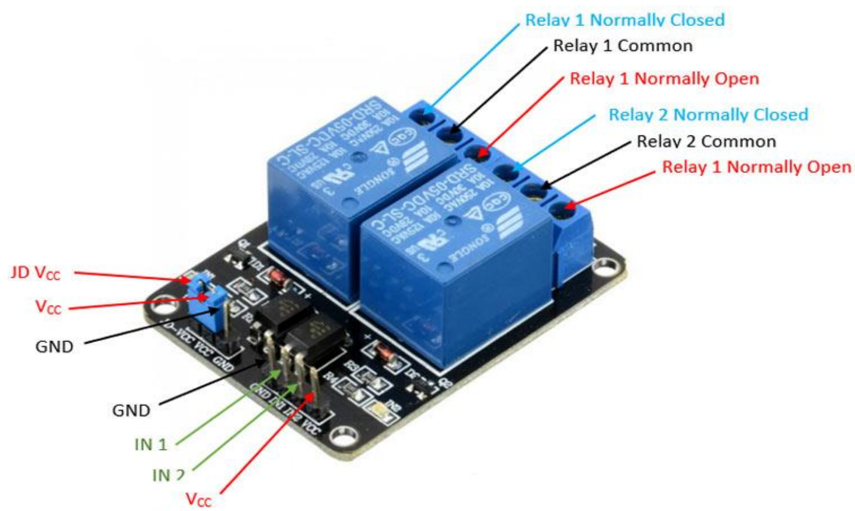


Figure 2.4-13 relay pin config

2.4.7 USB Camera

A webcam is a video camera that feeds or streams its image in real time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks via systems such as the internet, and emailed as an attachment. When sent to a remote location, the video stream may be saved, viewed or on sent there. Unlike an IP camera (which connects using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, or similar cable, or built into computer hardware, such as laptops. The term "webcam" (a clipped compound) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet. Some of them, for example, those used as online traffic cameras, are expensive, rugged professional video cameras. The USB webcam used in this project is capture license plat using python.

Chapter 3

3 Literature Review

3.1 Background

Automatic license plate recognition (ALPR) plays an important role in numerous applications such as unattended parking lots security control of restricted areas traffic law enforcement congestion pricing and automatic toll collection. Due to different working environments, LPR techniques vary from application to application. Most previous works have in some way restricted their working conditions, such as limiting them to indoor scenes, stationary backgrounds fixed illumination, prescribed driveways limited vehicle speeds or designated ranges of the distance between camera and vehicle. The aim of this study is to lessen many of these restrictions. Of the various working conditions, outdoor scenes and non-stationary backgrounds may be the two factors that most influence the quality of scene images acquired and in turn the complexity of the techniques needed. In an outdoor environment, illumination not only changes slowly as daytime progresses, but may change rapidly due to changing weather conditions and passing objects (e.g., cars, airplanes, clouds, and overpasses). In addition, pointable cameras create dynamic scenes when they move, pan or zoom. A dynamic scene image may contain multiple license plates or no license plate at all. Moreover, when they do appear in an image, license plates may have arbitrary sizes, orientations and positions. And, if complex backgrounds are involved, detecting license plates can become quite a challenge. ALPR systems generally comprises of a camera, software to compare the transformed license plate characters to databases in the system and a user interface to display the images captured with results of transformation. A license plate recognition system generally works in four main parts namely image acquisition, license plate detection, character segmentation and character recognition. Vehicles in each country have a unique license number, which is written on its license plate. This number distinguishes one vehicle from the other, which is useful especially when both are of same make and model. An automated system can be implemented to identify the license plate of a vehicle and extract the characters from the region containing a license plate. The license plate number can be used to retrieve more information about the vehicle and its owner, which can be used for further processing. Such an automated system should be small in size, portable and be able to process data at sufficient rate.

3.2 Automatic Plate Recognition

Automatic plate recognition system transcends the barriers of human efforts and also saves precious time by effectively recognizing the R.P. and providing useful information about the vehicle. Since, in practice vehicles belonging to different places in the world have varied characteristics of their R.P.'s in accordance to government norms of the place; Automatic Vehicle Registration Plate Recognition AVRPR has been profusely worked on both by individuals as well as institutions round the world. Most available algorithms for plate localization are based either on edge detection or are morphological approaches. In edge extraction approaches are proposed. These are based on detecting R.P. boundaries. A modification of conventional approaches to yield a cleaner result by application of heuristic approach is given by Dubey has suggested grayscale morphology to do the same. These algorithms while are excellent for the set of images considered by the author, they tend to fail miserably when implemented on alien vehicular plates. To implement them successfully on such vehicles, significant alterations in coding is required. Radon transform is ubiquitously accepted as correction for skew and rotation. The processing to make R.P. characters more prominent involve diverse algorithms such as horizontal and vertical histograms, mean shift, spatial transforms. To find characters in localized R.P., approaches used are of fuzzy logic, neural networks are more accurate but are also more sensitive even to non-R.P. characters present on the plate image. Also, algorithms based on Artificial Neural Networks (ANN) are found to take more time to run owing to their complexity. Character segmentation is followed by optical character recognition for detection of segmented characters. The features are extracted by skeletons, template matching or neural networks.

3.2.1 Automatic Number Plate Recognition using Python

In the text found on the vehicle plates is detected from the input image and this requires the localization of number plate area in order to identify the characters present on it. In this work, simple color conversion edge detection and removal of noise with the application of median filter as one of the operators is attempted. This work presents an approach using simple but efficient morphological operations, filtering and finding connected components for localization of Indian number plates. It proposes the identification of stolen cars. The algorithm has been tested on 10 samples and is found to extract both alphabets and numbers from vehicle license plates images with an accuracy of 90% for four-wheeler license plates, the results of this work is shown in figure

3.2.2 Automatic Vehicle Identification by Plate Recognition

Automatic Vehicle Identification (AVI) proposed in has many applications in traffic systems (highway electronic toll collection, red light violation enforcement, border and customs checkpoints, etc.). License Plate Recognition is an effective form of AVI systems. In this study, a smart and simple algorithm is presented for vehicle’s license plate recognition system. The proposed algorithm consists of three major parts: Extraction of plate region, segmentation of characters and recognition of plate characters. For extracting the plate region, edge detection algorithms and smearing algorithms are used. In segmentation part, smearing algorithms, filtering and some morphological algorithms are used. And finally statistical based template matching is used for recognition of plate characters. The performance of the proposed algorithm has been tested on real images. Based on the experimental results, we noted that our algorithm shows superior performance in car license plate recognition. Experiments have been performed to test the proposed system and to measure the accuracy of the system. The system is designed in Python 6.5 for recognition of license plates. The images for the input to the system are colored images with the size of 1200x1600. The test images were taken under various illumination conditions.

3.2.3 Tracking number plate from vehicle using Python

The work in states the following, tracking of the number plate from the vehicle is an important task, which demands intelligent solution. In this document, extraction and Reorganization of number plate from vehicles image has been done using Python. It is assumed that images of the vehicle have been captured from Digital Camera. Alphanumeric Characters on plate has been Extracted and recognized using template images of alphanumeric characters. The system works with 99% accuracy when images are captured from fixed distance and captured from the center position. Vehicle should be stationery and image are captured from fixed angle parallel to horizon. Car number plate should be according to 1989 motor vehicle limited. vehicle number plates do not follow a standard language, font or size. Due to the variations in representation of number plates, vehicle number plate extraction, segmentation and recognition are crucial. This demonstration considers vehicle number plates which can contain English characters and numbers only. The system works satisfactorily for wide variation of condition and different type of vehicle plates. The system is implemented and executed in Python OpenCV and performance is tested on genuine images.

3.2.4 Open CV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeit era, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in

Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers. OpenCV (Open-Source Computer Vision Library) is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

3.3 Information- Radio Frequency Identification Technology

Radio frequency identification has become significant part in our day to day lives, like supply chain management, attendance tracking, race timing, library system, laundry management and many more. This technology provides security, increases effectiveness and accessibility. This technology is mainly used to communicate in digital information for stationary and mobile objects. There are many applications of RFID, out of which some are prevention of vehicles theft, automatic toll collection, traffic control, automatic gates at buildings or organizations like airports or universities, entry and exit of vehicles in gated societies, dispensing goods and objects which are to be supervised in supply chain. Radio frequencies ranging from 100 kHz to 10 GHz are being used, for complex devices like readers or interrogators which are more capable and are connected to a central server.

The RFID system mainly works by its four main components they are:

1. Reader or interrogator
2. Antenna
3. RFID tags (Active or Passive)
4. A host system

3.3.1 RFID Reader

The most important part of the RFID system is the reader, which is also known as interrogator. This reader contains a RF module which acts as a trans-receiver. These transmit RF power by creating a frequency with the help of an oscillator, this signal is amplified to send data to the tag. The read range of the reader is determined by EIRP (Equivalent Isotropic Radiated

Power). A microprocessor is also used to memorize and store the data. The information regarding the configuration, frequency and the EIRP all are stored in the reader’s memory. Information regarding the tags such as Tag Identification number is also stored in the memory. In the case where the connection between the hardware and host computer is lost while the reader keeps on reading the tags the information from the tags can be stored in this memory space which can be later on read by the software.

3.3.2 Reader Antenna

The antenna and the readers work together to read tags. They convert electric current to electromagnetic wave. The antenna works based on the polarization. There are two types of polarization that the antenna works on: linear polarization and circular polarization. Figure 3.3-2 and Figure 3.3-1 show linear polarization and circular polarization respectively. Linearly polarized antennas have long ranges and radiate electric current with high levels of electric power. These are sensitive tag antennas i.e.; it depends on the angle of the tag and its position. In circularly polarized antennas the radiated power is divided into axes, hence, it can read the tag data in all possible directions, this is used in applications where the tag is not fixed. The read range of such antennas is small.

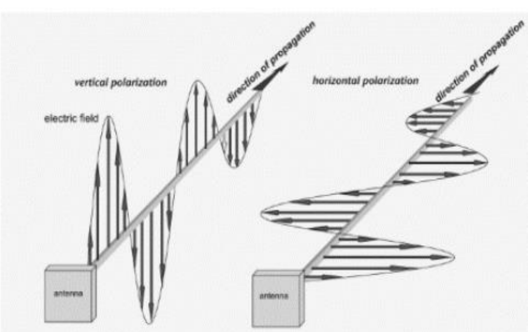


Figure 3.3-2 Linear polarization

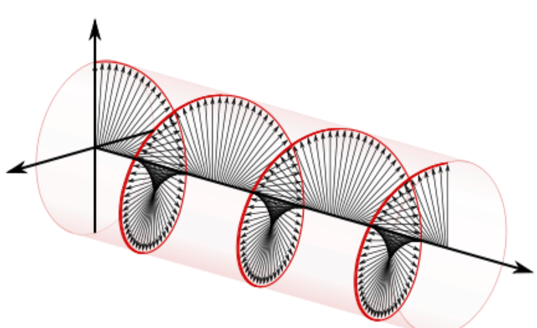


Figure 3.3-1 Circular polarization

3.3.3 RFID tags

An RFID tag, or transponder, consists of a chip and an antenna. The chip stores unique identification number- Electronic Product Code (EPC). The antenna, which is connected to the microchip, transmits data from the chip to the reader. Ordinarily, a larger antenna indicates a longer read range. The main component in the RFID tag is the RF antenna, it functions in two ways – first, it receives information from the reader's antenna and second it transmits back the information to the reader antenna. Figure 3.3-3 shows the basic architecture of the RFID tag. The tag has its own memory block, the size of the memory may vary from tag to tag. Active tags use larger memory space due to its more power. There are three types of memories in read only (RO), read-write memory (RW) and write once read many (WORM). The read-only memory is used when the information needs to be secured, such that no other person can over-write the data in the tag. The read-write option allows to read as well as write information on memory.

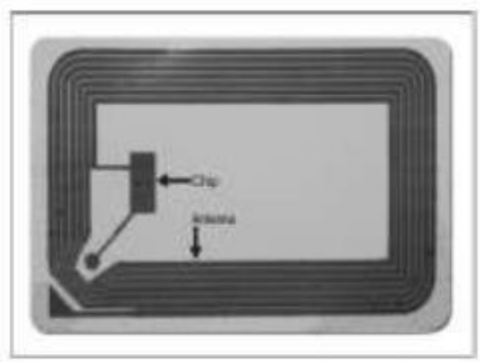


Figure 3.3-3 Basic tag architecture

3.3.4 RFID tags classification

Active tags

Active tags need battery power to power up their circuit to send signals. They are also useful in the applications where there is more than one sensor attached to the tag as they can support large memory space henceforth, enabling the tag to store and transmit data. Compared to the passive tags these have longer range and are more expensive. The usage of battery to power the tag decreases the life span of the gadget, though with the current power it might run as long as 10 years.6

Passive tags

These tags require no internal power, instead they use the electromagnetic radiation that is transmitted from the reader. The antennas in the passive tags use magnetic coupling for data transfer from tag to reader. There are two ways of coupling: inductive coupling for near field and backscatter method for far field.

Semi-passive tags

As the name suggests, these tags use battery to power up their circuit and also use incoming radio waves for transmitting information. Hence, they are called as semi-passive tags. Table 3.3-1 Characteristics of different types of tags is summarized table of the characteristics of the three tags (passive, active and semi-active)

Characteristics of different types of tags

Attribute	Passive Tag	Active Tag	Semi (Ac/Pas) Tag
Internet Power Source	No	Yes	Yes
Response Distance	Short	Very Long	Long
Weight	Light	Less Light	Less Light
Cost	Cheap	Expensive	Less Expensive
Life Cycle	Long	Short	Long

Table 3.3-1 Characteristics of different types of tags

Inductive Coupling: In this the reader transmits only short alternating magnetic current through antenna. The passive tags use this current to power up the circuit and as well to send signals. The magnetic field produced generates voltage which powers up the micro circuit in the tag. The signal is sent to the reader's antenna via the tag antenna as varying voltage, this is detected by the reader's antenna and decodes it. Hence the tag takes the advantage of the magnetic field produced by the reader and each time the tag uses the magnetic field there will be a drop in the voltage. This type of coupling is observed for the RFID readers which operate at the frequency of 123 to 135 KHZ and 13.56 MHz Figure 3.3-4 Inductive coupling and load modulation in near field shows a clear picture of inductive coupling and load

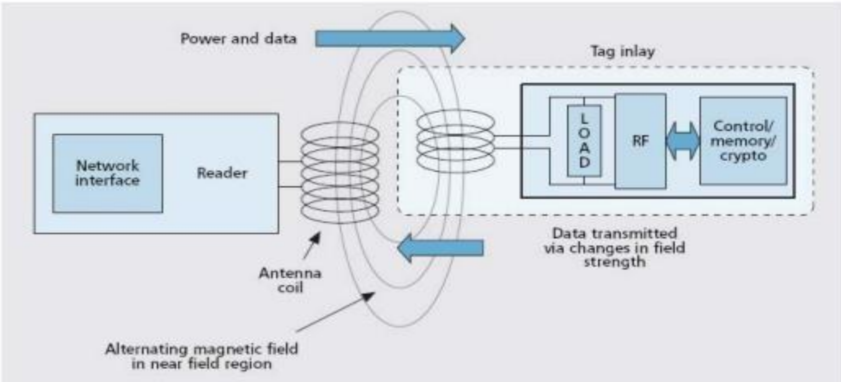


Figure 3.3-4 Inductive coupling and load modulation in near field

modulation

Electromagnetic Coupling: Electromagnetic Coupling in far-field is also known as backscatter modulation. In this the antenna of the reader transmits long range electromagnetic waves as shown in the Figure 3.3-5. The tag uses electromagnetic coupling to activate power from the signals in order to power the circuit of the tag. Using the principles of back-scattering modulation, the communication is performed. The incoming waves are partially reflected back by changing the load resistance which is connected parallel to the antenna to causing modulation in the amplitude of the electromagnetic waves. The received waves are detected by the antenna and decoded. As the rate of attenuation for electromagnetic far-field is 1/R2 and that of short range is 1/R6, hence Backscatter modulation provides longer range communication. They can be found mostly in passive tags which operate at frequency of 868MHz and high frequency.

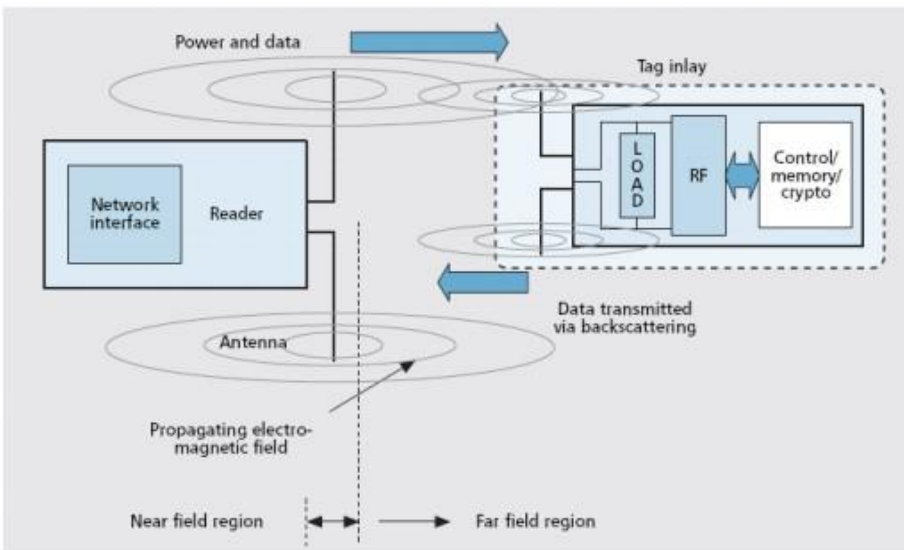


Figure 3.3-5 Electromagnetic coupling and backscatter modulation in far field

3.3.5 RFID types

According to the frequency the RFID systems are broadly classified into 3 types – low frequency, high frequency and ultra-high frequency. Table 3.3-2 shows frequency and ranges of RF waves. In general, low frequency passive tags have range of approximately 30cm, high frequency passive tags have range of 1m and ultra-high frequency passive tags range from 3m-5m. Where greater range is needed like in applications of railways, active tags are used which can range to about 100m, but they require internal power supply.

Frequency and their respective ranges

Frequency	**Typical Read Range**
Low-Frequency (LF) 124 KHZ – 135khz	Up To Half a Meter
High-Frequency (HF) 13.56mhz	Up to 1m
Ultrahigh-Frequency (UHF) 860mhz – 960mhz	4 – 5 Meters
Microwave 2.45ghz	Up to 1m

Table 3.3-2 Frequency and their respective ranges

Summary of RFID range and its applications

Frequency band	Characteristics	Typical application
Low 100-500 KHZ	Short to medium range inexpensive low reading speed	Access control animal identification inventory control car immobilizer
Intermediate 10-15 KHZ	Short to medium range potentially inexpensive medium reading speed	Access control smart cards library control
High 850-950mhz,2.4-5.8mhz	Long read range, high reading speed line of sight required expensive	Railway vehicle monitoring tool collection systems pallet and container tracking vehicle tracking

Table 3.3-3 Summary of RFID range and its applications

3.4 SQL Server

SQL Server is a relational database management system, or RDBMS, developed and marketed by Microsoft.

Similar to other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL server is tied to Transact-SQL, or T-SQL, the Microsoft's implementation of SQL that adds a set of proprietary programming constructs. SQL Server works exclusively on Windows environment for more than 20 years. In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.

3.4.1 SQL Server Architecture

The following diagram illustrates the architecture of the SQL Server:

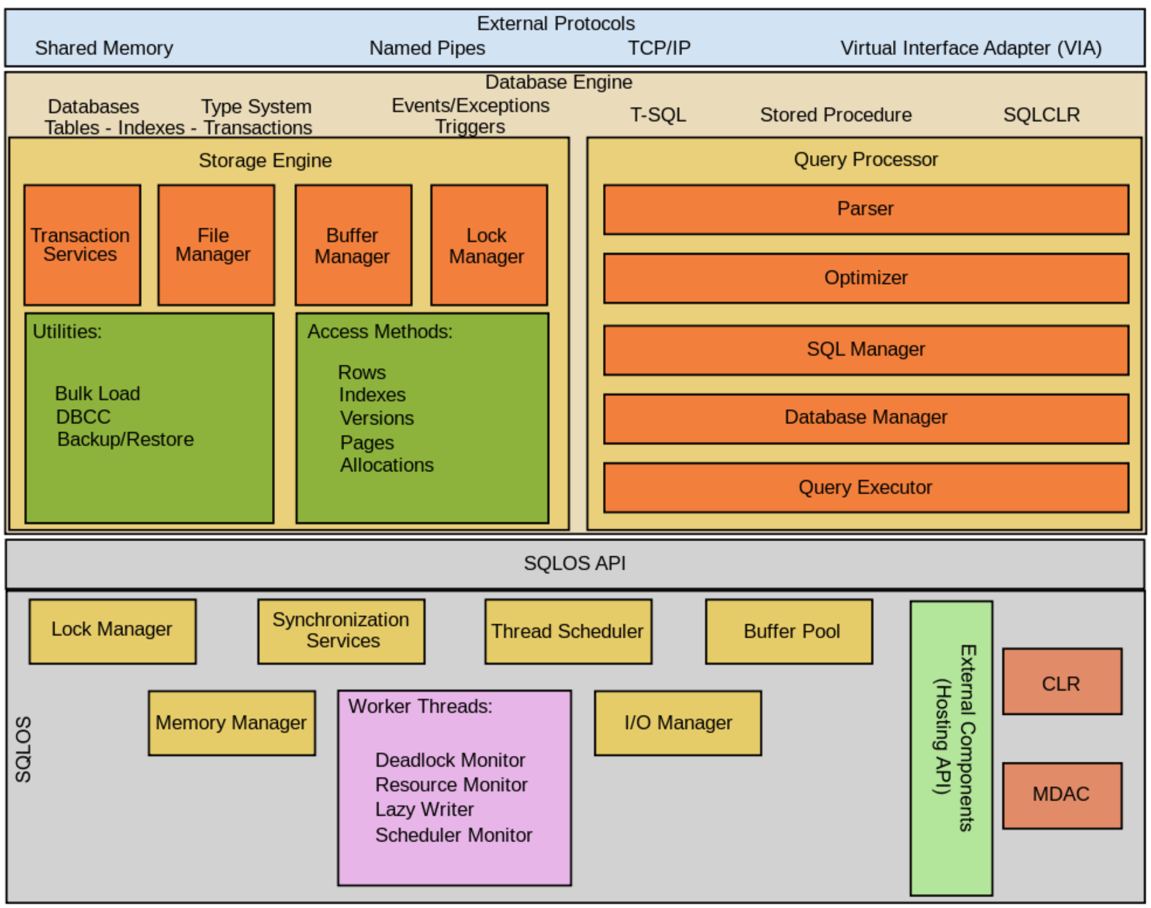


Figure 3.4-1 SQL Server Architecture

3.4.2 SQL Server consists of two main components

- Database Engine
- SQLOS

Database Engine

The core component of the SQL Server is the Database Engine. The Database Engine consists of a relational engine that processes queries and a storage engine that manages database files, pages, pages, index, etc. The database objects such as stored procedures, views, and triggers are also created and executed by the Database Engine.

Relational Engine

The Relational Engine contains the components that determine the best way to execute a query. The relational engine is also known as the query processor. The relational engine requests data from the storage engine based on the input query and processed the results. Some tasks of the relational engine include querying processing, memory management, thread and task management, buffer management, and distributed query processing. Storage Engine The storage engine is in charge of storage and retrieval of data from the storage systems such as disks and SAN.

SQLOS

Under the relational engine and storage engine is the SQL Server Operating System or SQLOS. SQLOS provides many operating system services such as memory and I/O management. Other services include exception handling and synchronization services.

3.4.3 SQL Server Services and Tools

Microsoft provides both data management and business intelligence (BI) tools and services together with SQL Server. For data management, SQL Server includes SQL Server Integration Services (SSIS), SQL Server Data Quality Services, and SQL Server Master Data Services. To develop databases, SQL Server provides SQL Server Data tools; and to manage, deploy, and monitor databases SQL Server has SQL Server Management Studio (SSMS). For data analysis, SQL Server offers SQL Server Analysis Services (SSAS). SQL Server Reporting Services (SSRS) provides reports and visualization of data. The Machine Learning Services technology appeared first in SQL Server 2016 which was renamed from the R Services.

3.4.4 SQL Server Editions

SQL Server has four primary editions that have different bundled services and tools. Two editions are available free of charge:

SQL Server Developer edition for use in database development and testing.

SQL Server Expression for small databases with the size up to 10 GB of disk storage capacity.

For larger and more critical applications, SQL Server offers the Enterprise edition that includes all SQL server’s features.

SQL Server Standard Edition has partial feature sets of the Enterprise Edition and limits on the Server regarding the numbers of processor core and memory that can be configured.

For the detailed information on the SQL Editions, check it out the available Server 2017 Editions.

In this tutorial, you have a brief overview of the SQL Servers including its architecture, services, tools, and editions

Chapter 4

4 Methodology and flow chart

4.1 System Overview

The proposed system in this thesis will be implemented into two essential parts. First part is the IR sensor that are connect to Arduino, IR sensor will detect there is a car or not, if the car detected the IR sensor will send a message to Arduino and Arduino will send message to computer (python) through serial monitor that will immediately capturing car license plate and check the result in SQL server and compare it, if the plate are matched to the stored data base it will send message to Arduino through serial monitor and will ask for RFID permission to get the full authorization for car, if the card have the access the gate will open automatically, if the card doesn't have the access the gate will not open.

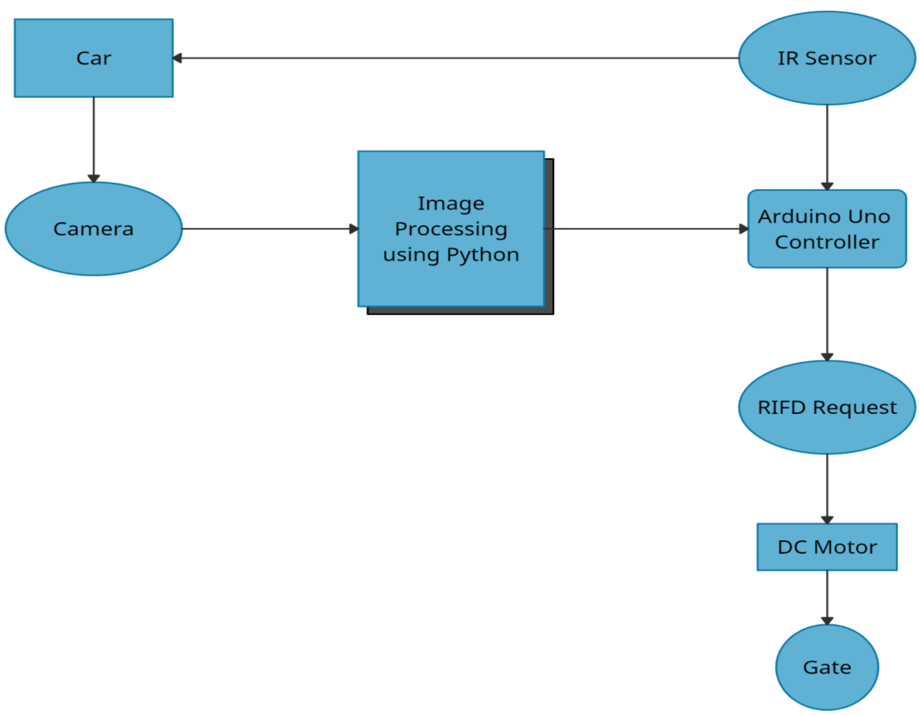


Figure 4.1-1 Design Diagram

4.2 General Flow chart

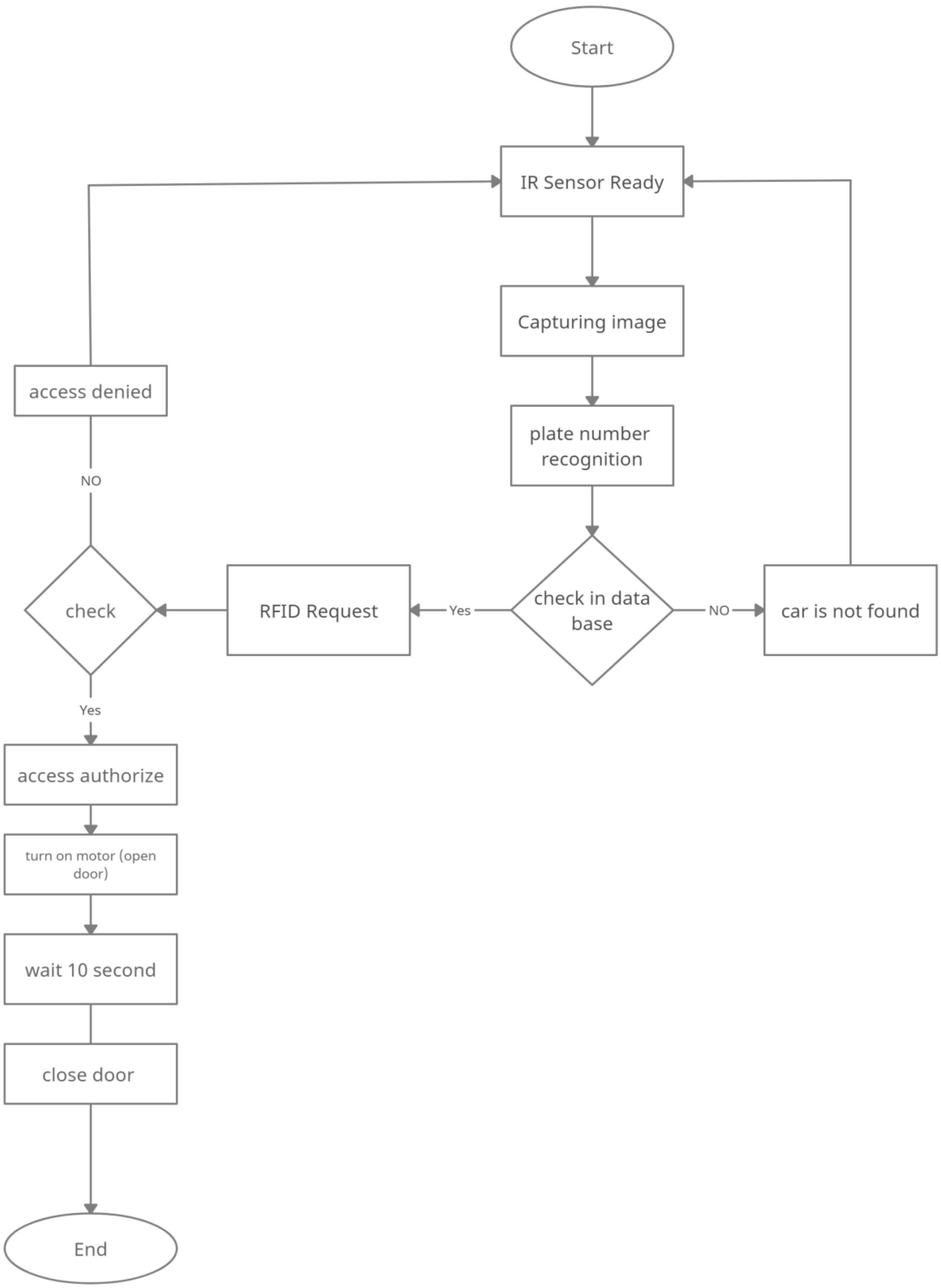


Figure 4.2-1 General Flow chart

4.3 Schematic Design

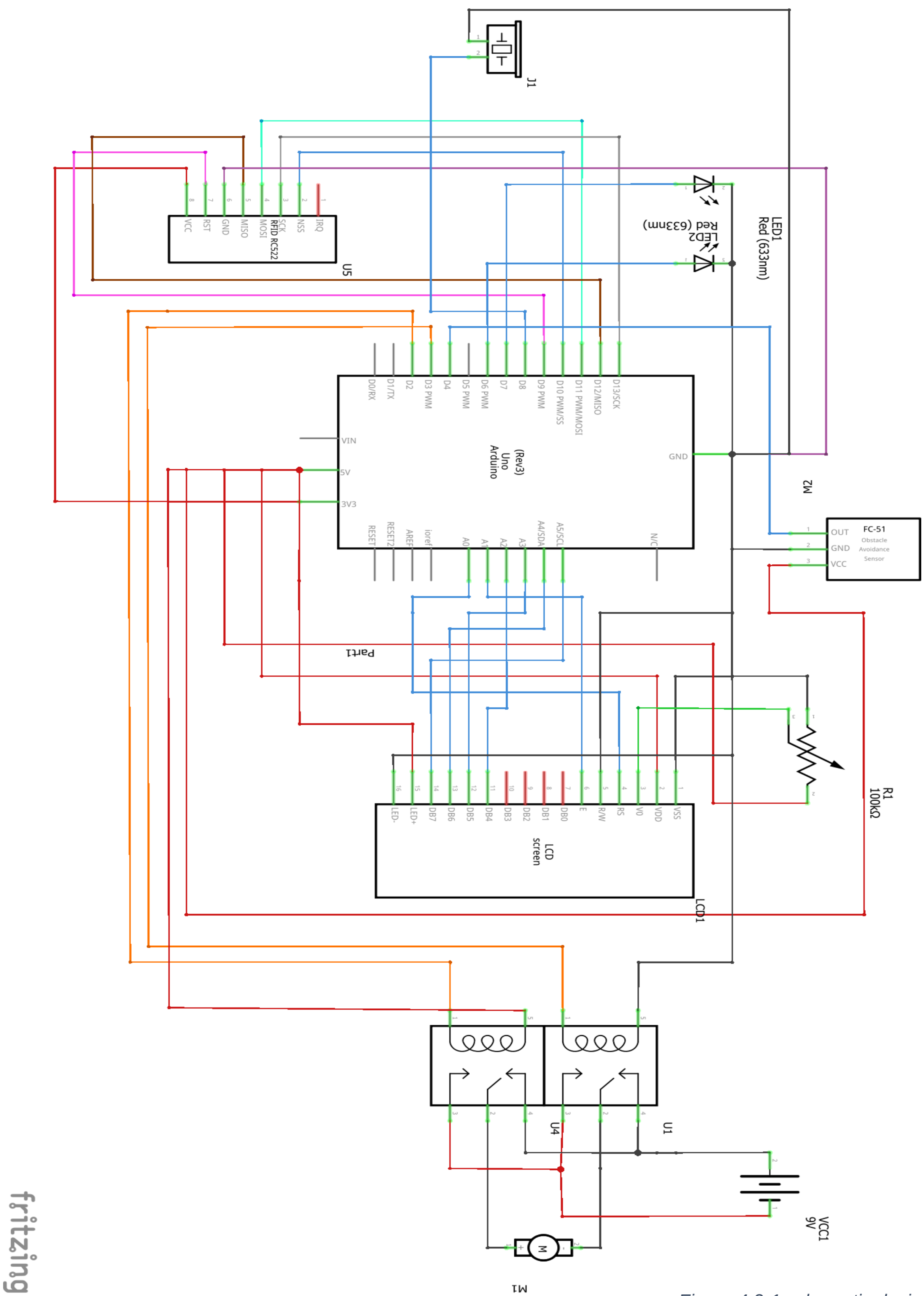


Figure 4.2-1 schematic design

4.4 Pin connection

4.4.1 RFID Module RC522

Arduino will read RFID Tag and Display to the Serial Monitor

RFID RC522 and Arduino Uno Pin Configuration

RFID RC522	Arduino Uno
SS/SDA	D10
SCK	D13
MOSI	D11
MISO	D12
RQ	Not Connected
GND	GND
RST	D9
3.3V	3.3V

Table 4.4-1 RFID with Arduino

We connected LCD to the Analog Pins.

Analog Pins can be act as a Digital Pins,

But Digital Pins CAN NOT be act as an Analog Pins.

4.4.2 LCD (16x2) Connections

VSS pin	to GND
VCC pin	to 5V
VEE pin	to GND
RS pin	to Analog pin A0
R/W pin	to ground
Enable pin	to Analog pin A1
D4 pin	to Analog pin 5
D5 pin	to Analog pin 4
D6 pin	to Analog pin 3
D7 pin	to Analog pin 2
LED+ pin	to POWER
LED- pin	to GND

Table 4.4-2 lcd pin

We are connecting "Contrast pin to GND for Maximum Brightness “You can also connect it to 10K Pot if you wish to change the Brightness Of LCD.

motor Digital Pin 2, 3

Red pin to Digital pin 6

Green pin to Digital pin

Buzzer to Digital pin 8

4.5 Hardware simulation

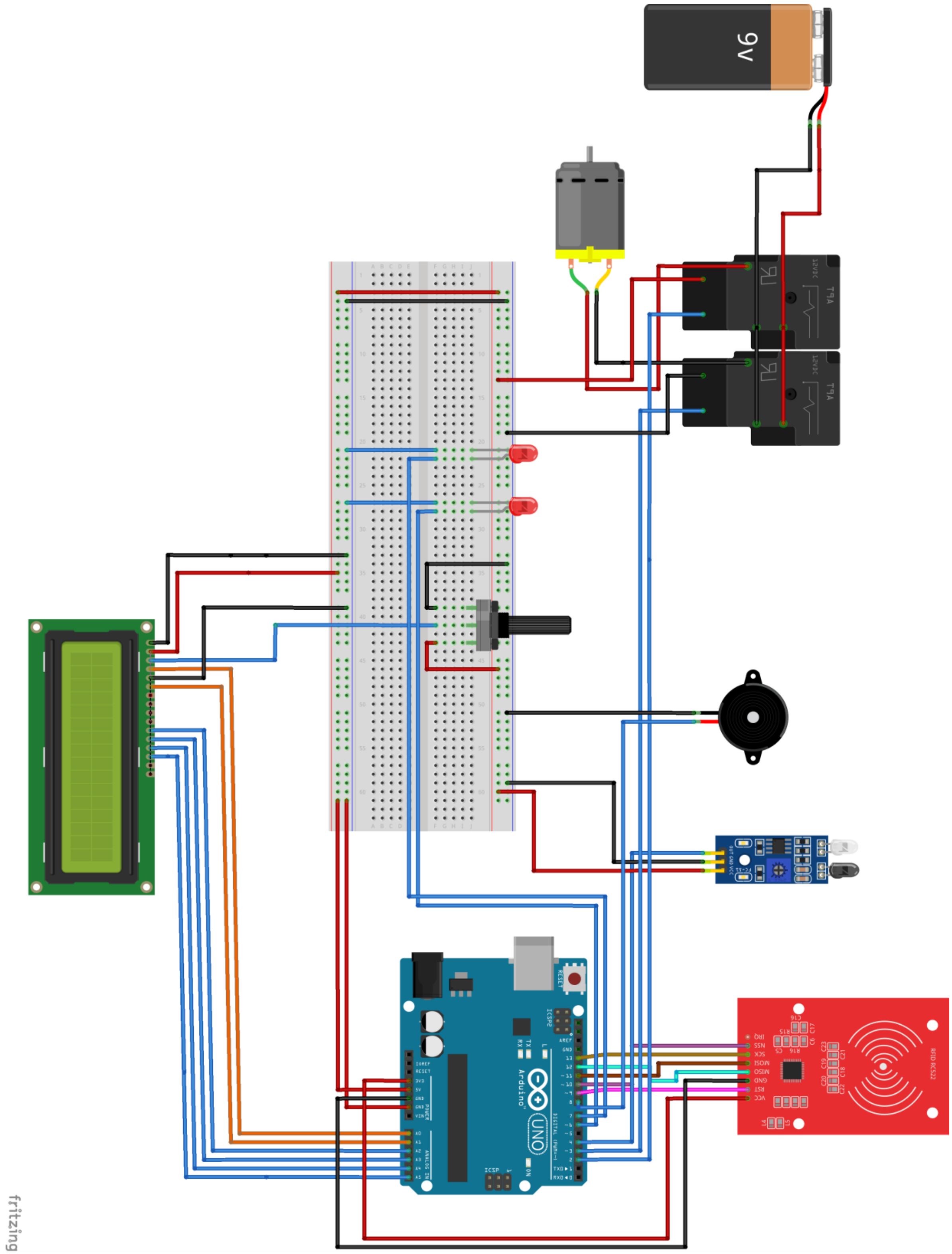


Figure 4.4-1 hardware simulation

4.6 PROJECT IMPLEMENTATION

4.6.1 Vehicle Detection

The first stage of this system is detecting the vehicle when it arrives the tollgate. This detection can be done by using the sensors. Here we are using IR sensors to detect the vehicle. It generates the active LOW output when it detects the vehicle. This sensor is connected to Arduino then whenever it detects the vehicle the sensor generates the signal to start the process and send 0 to computer to capture license plate automatic. We are already programmed the controller whenever sensors send the acknowledgement signal.

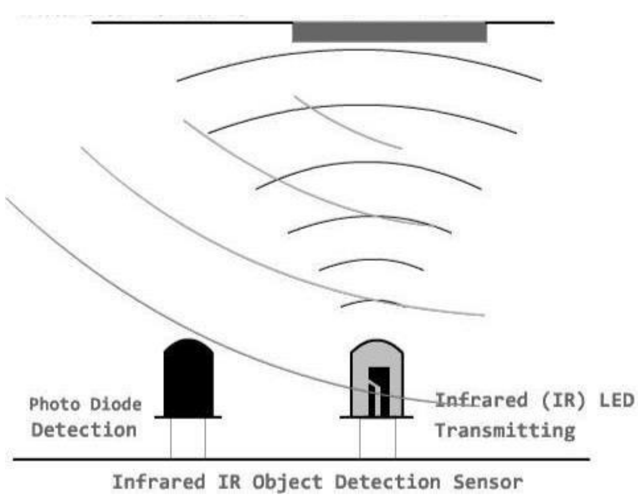


Figure 4.6-1 IR object detection

4.6.2 Image Capturing

After the successful detection of the vehicle arriving the controller sends the signal to the camera module then it captures the image of the vehicle and sends the captured image to process the image to extract the number plate from the captured image.

Figure 4.6-2 Original image captured

4.6.3 Gray scale conversion

Convert into gray image for processing, because color information doesn't help us identify important edges or another feature. A grayscale image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel.

Figure 4.6-3 Grey scale converted image

4.6.4 Number plate detection

Computer Vision is a field of data science that is unlike any other. Instead of dealing with datasets they deal with images. In the same way that Neural Networks try to process data like the human brain, Computer Vision attempts to see, identify, and process images in the way that human vision does. So, object Detection is a computer technology related to computer vision, image processing, and deep learning that deals with detecting instances of objects in images and videos. It is widely used in computer vision tasks such as image annotation, vehicle counting, activity recognition, face detection, face recognition, Plate detection, video object co-segmentation. It is also used in tracking objects.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Initially, the algorithm needs a lot of positive images (images of plate number) and negative images (images without plate number) to train the classifier. Then we need to extract features from it. These features on the image makes it easy to find out the edges or the lines in the image, or to pick areas where there is a sudden change in the intensities of the pixels.

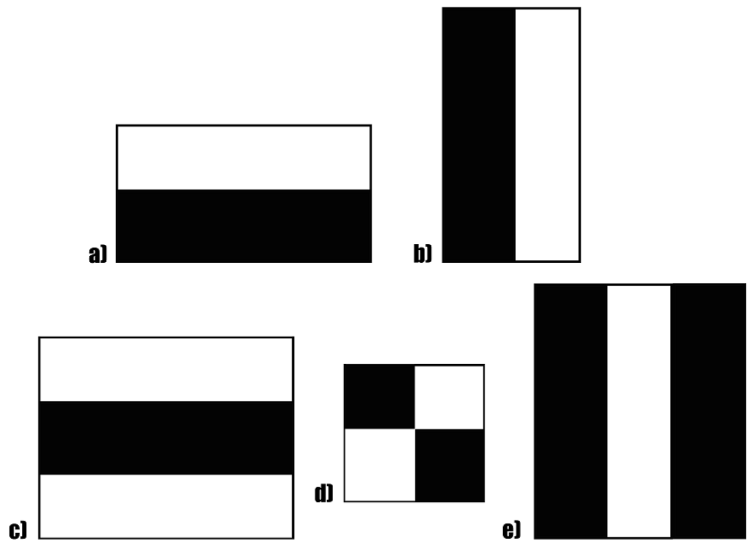


Figure 4.6-4 object detection

All possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels. So how do we select the best features out of 160000+ features? It is achieved by Ad boost. We apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the plates to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the plate and non-plate images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated.

Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

In an image, most of the image is non-face region. So, it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot, and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

Figure 4.6-5 haar cascade

After we apply Cascade Classification on the image, if a license plate is detected it will return the x, y location of where it was found in the image and the width and height of the plate. Now that we have the points in the image that show where the license plate is we can draw on the image, a rectangle around it. Using OpenCV, (a Computer Vision library) it has a function where we can draw rectangles on a given image, all it needs is the points of where it should be drawn. So, from the x, y, width, and height outputted from Cascade Classifier we can change this image.

Figure 4.6-6 cropping plate

4.6.5 Characters Segmentation

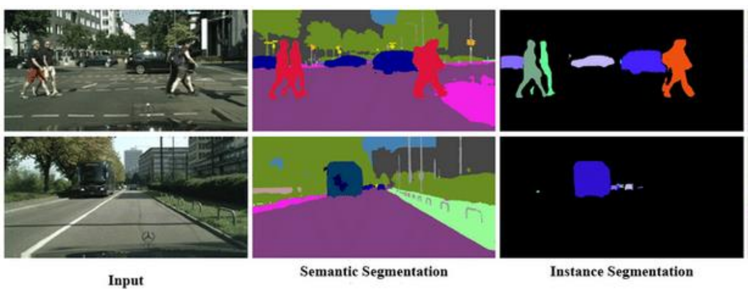


Figure 4.6-7 character segmentation

There are many methods for characters segmentation and recognition, including advanced and complex deep learning algorithms. However, we will use a simpler approach. In order to read the license plate, it will take two stages. The first stage is to segment the characters, and the second stage is to recognize those characters. Image segmentation is an image processing task in which the image is segmented or partitioned into multiple regions such that the pixels in the same region share common characteristics.

4.6.6 thresholding

One of the most used techniques for the analysis of the images is that of the thresholding, the application of a threshold along a particular scale of values, to filter in some way an image. One of these techniques is for example the one that converts any image in grayscale (or color) in a totally black and white image. Often this is very useful for recognizing the regular shapes, contours within an image, or even to delimit and divide zones inside, to then be used in a different way in the subsequent processing

Figure 4.6-8 thresholding

In global thresholding, we used an arbitrary chosen value as a threshold. In contrast, Otsu's method avoids having to choose a value and determines it automatically.

Consider an image with only two distinct image values (bimodal image), where the histogram would only consist of two peaks. A good threshold would be in the middle of those two values. Similarly, Otsu's method determines an optimal global threshold value from the image histogram.

In order to do so, the threshold function is used, where THRESH_OTSU is passed as an extra flag. The threshold value can be chosen arbitrary. The algorithm then finds the optimal threshold value which is returned as the first output.

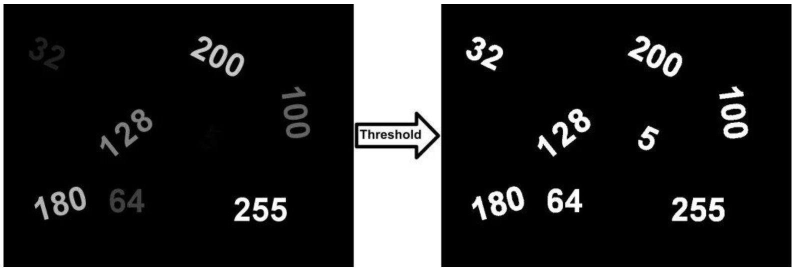


Figure 4.6-9 num

We should process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

4.6.7 morphological dilation and erosion

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image and makes objects more visible and fills in small holes in objects, while erosion removes pixels on object boundaries and removes small objects so that only substantive objects remain. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image.

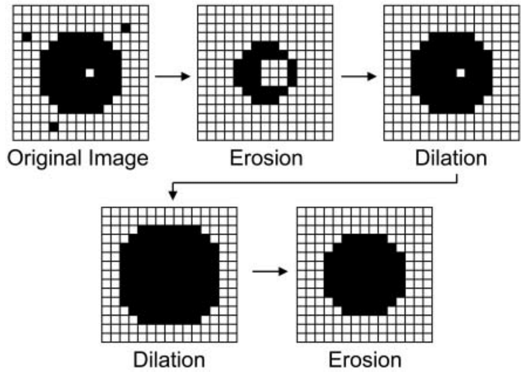


Figure 4.6-10 morphological dilation and erosion

4.6.8 Find contours

The next step is to find the contours that joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. It stores the (x, y) coordinates of the boundary of a shape. by this contour approximation method. So, you pass CHAIN_APPROX_NONE, all the boundary points are stored. Or CHAIN_APPROX_SIMPLE. Which removes all redundant points and compresses the contour, thereby saving memory.

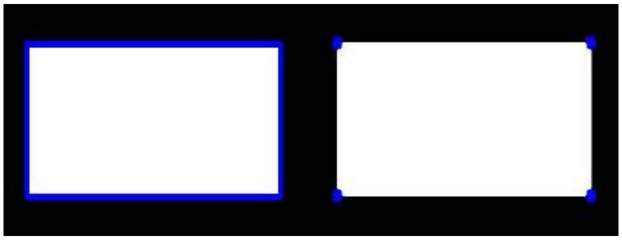


Figure 4.6-11 find couture

There are more things we can do with the contour. We can find the centroid of an image or calculate the area of a boundary field with the help of the notion called image moment. period in common usage. But in physics terminology, a moment is the product of the distance and another physical quantity meaning how a physical quantity is distributed or located. So, in computer vision, Image moment is how image pixel intensities are distributed according to their location. It's a weighted average of image pixel intensities and we can get the centroid or spatial information from the image moment.

To get the area of the contours, we can implement the function contourArea(). Then draw a convex contour or rectangular contour line of a figure. we draw a straight rectangular shape. With the outer contour line, we'll draw the rectangle around the object. The function boundingRect() returns the 4 points of the bounding box. After that you will get all the characters separately.

4.6.9 plate number Extraction

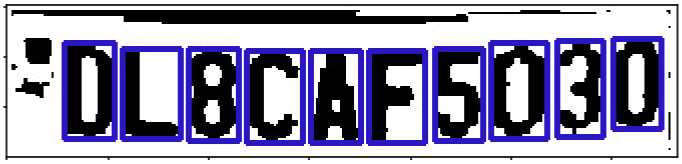


Figure 4.6-12 plate extraction

The binary images of extracted characters. The characters may be unsorted but sorts the character according to the position of their bounding rectangle from the left boundary of the plate.

We use recognition process of license plate characters using Convolutional Neural Network (CNN). Each of the 36 characters is trained using these algorithms in different environments. Accuracy is extracted from the training and testing data.

Figure 4.6-13 out put

4.6.10 Number plate extraction with SQL server

After extraction number plate as output, it will check automatically in SQL server in a valid table that we are create in data base if the number plate are their already, the first factor of our security will be active then it will send to Arduino to ask for second factor to get gate open if the number plate are different in that valid table it will send to Arduino the car is not found in system and don’t have the authorization

ER Diagram for database

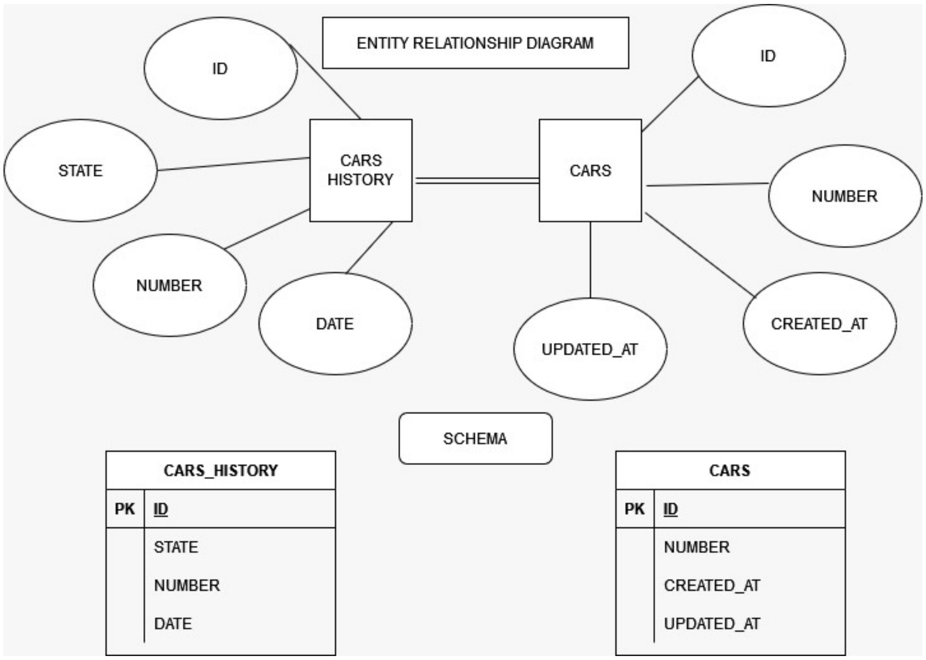


Figure 4.6-14 ER diagram

Valid table

DESKTOP-29F41A4.c...tes - dbo.valid222

	car_numbers
▶	W668AHK
	P688CC
	222RXP
	JUX703
	CA8902
*	*NULL*

Figure 4.6-15 valid table

History table

DESKTOP-29F41A4.c...tes - dbo.logs222

	car_id	car_number	date
	1	P688CC	6/2/22
	2	222RXP	7/2/22
	3	JUX703	8/2/22
*	NULL	NULL	NULL

Figure 4.6-16 history table

4.6.11 Controlling Hardware

After we check the number plate this step is the second factor to check if this car driver hold the right one or not, the process will go on that way the computer will send one to serial port that the check are correct then the lcd monitor will show that to place your RIFD card to scan it, if the RFID are the correct one that have access to open the gate the lcd monitor will show that access grenade then the gate will open automatically, it will wait 10 sec and will close again to get these process again , if the RFID card are wrong the lcd will show that access denied

Chapter 5

5 Software

5.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

5.1.1 FUNCTIONAL REQUIREMENTS

Functional requirement refers to the functionalities that are applicable to a system. The functional requirements of automatic license plate recognition system are stated below.

The system must be able to:

- Load videos from the system.
- Extract frames from the video.
- Localize license plate region from the frames.
- Segment characters from the localized plate.
- Recognize the segmented characters and display it on the terminal.

5.1.2 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a system must behave or how is the system's behavior. This also specifies how the system's quality characteristics or quality attributes in order to put this constraint upon the specific system behavior, the qualities goals of the designed system should go in these:

Execution qualities

- Functionality
- Security
- Usability
- Effectiveness & Efficiency

Evolution qualities

- Availability
- Reliability
- Manageability

5.1.3 USER REQUIREMENTS

1. The ANPR system should captures the videos of vehicle number plates from moving vehicles in both day and night time conditions.
2. Quick detection of the vehicles' number plates.
3. Providing an operator control of the observed by the ANPR system object, including the supervising the ANPR live video-stream of the controlled road, lane, etc. and visualization of the found number plates

5.1.4 SYSTEM REQUIREMENTS

The following is the suggested minimum system configuration to run the ANPR

software:

- Intel i3 CPU or higher
- 4GB RAM or more
- Windows 7 (64 bit), Windows 8 (64 bit), Windows 10 (64 bit)

5.2 SOFTWARE REQUIREMENTS

5.2.1 PYTHON VERSION 3.10.0

Python is a high level, interpreted and general-purpose dynamic programming language that focuses on code readability. It has fewer steps when compared to Java and C. It was founded in 1991 by developer Guido Van Rossum. It is used in many organizations as it supports multiple programming paradigms. It also performs automatic memory management.

Advantages of Python

The diverse application of the Python language is a result of the combination of features which give this language an edge over others. Some of the benefits of programming in

Python includes

1. **Presence of Third-Party Modules**

The Python Package Index (PYPI) contains numerous third-party modules that make Python capable of interacting with most of the other languages and platforms.

2. **Extensive Support Libraries**

Automatic license plate recognition for real time videos Software requirements Python provides a large standard library which includes areas like internet protocols, string operations, web services tools and operating system interfaces. Many high use programming tasks have already been scripted into the standard library which reduces length of code to be written significantly.

3. **Open Source and Community Development**

Python language is developed under an OSI-approved open-source license, which makes it free to use and distribute, including for commercial purposes. Further, its development is driven by the community which collaborates for its code through hosting conferences and mailing lists, and provides for its numerous modules.

4. **Learning Ease and Support Available**

Python offers excellent readability and uncluttered simple-to-learn syntax which helps beginners to utilize this programming language. The code style guidelines, PEP 8, provide a set of rules to facilitate the formatting of code. Additionally, the wide base of users and active developers has resulted in a rich internet resource bank to encourage development and the continued adoption of the language.

5. User– friendly Data Structures

Python has built-in list and dictionary data structures which can be used to construct fast runtime data structures. Further, Python also provides the option of dynamic high-level data typing which reduces the length of support code that is needed.

6. Productivity and Speed

Python has clean object-oriented design, provides enhanced process control capabilities, and possesses strong integration and text processing capabilities and its own unit testing framework, all of which contribute to the increase in its speed and productivity. Python is considered a viable option for building complex multi-protocol network applications. As can be seen from the above-mentioned points, Python offers a number of advantages for software development. As upgrading of the language continues, its loyalist base could grow as well. Automatic license plate recognition for real time videos Software requirements

5.2.2 Python File

The standard Python installer already associates the .py extension with a file type (Python. File) and gives that file type an open command that runs the interpreter. This is enough to make scripts executable from the command prompt as ‘foo.py’. If you’d rather be able to execute the script by simple typing ‘foo’ with no extension you need to add .py to the PATHEXT environment variable.

5.2.3 Creating and Running Script File

To create scripts files, you need to use a text editor. You can open the python editor in two

ways:

- Using the command prompt
- Using the IDE

If you are using the command prompt, type edit in the command prompt. This will open the editor. You can directly type edit and then the filename (with .py extension).

5.2.4 USE CASE DIAGRAM

Actor

- User

Use Case

- Capture video and acquire image
- Verify Vehicle
- Identify Number

Precondition

- A camera is placed at 4-5 m away from the vehicle to get the clear view of the number plate.
- Videos are captured and stored in a repository.

Post condition

- The license plate numbers are recognized and displayed on the terminal

5.2.5 Python code

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
from datetime import datetime
import tensorflow as tf
import pyodbc
import serial
import time


def find_contours(dimensions, img) :

    # Find all contours in the image
    cntrs, _ = cv2.findContours(img.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

    # Sorting largest 15 contours contours according size for license
plate
    cntrs = sorted(cntrs, key=cv2.contourArea, reverse=True)[:15]

    # Retrieve potential dimensions
    lower_width = dimensions[0]
    upper_width = dimensions[1]
    lower_height = dimensions[2]
    upper_height = dimensions[3]

    x_cntr_list = []
    img_res = []

    for cntr in cntrs :
        #detects contour in binary image and returns the coordinates of
rectangle enclosing it
        x, y, w, h = cv2.boundingRect(cntr)

        #checking the dimensions of the contour to filter out the
characters by contour's size
```

```
        if w > lower_width and w < upper_width and h > lower_height and h
< upper_height :
            x_cntr_list.append(x) #stores the x coordinate of the
character's contour, to used later for indexing the contours

            # Create image using numpy 24*44
            char_copy = np.zeros((44,24))

            # Extracting each character using the enclosing rectangle's
coordinates.
            char = img[y:y+h, x:x+w]

            char = cv2.resize(char, (20, 40))

            # Show image plate
            cv2.rectangle(img, (x,y), (w+x, y+h), (50,21,200), 2)
            plt.imshow(img)


            # Make result formatted for classification: invert colors
            char = cv2.subtract(255, char)

            # Resize the image to 24x44 with black border
            char_copy[2:42, 2:22] = char
            char_copy[0:2, :] = 0
            char_copy[:, 0:2] = 0
            char_copy[42:44, :] = 0
            char_copy[:, 22:24] = 0

            #List that stores the character's binary image
            img_res.append(char)

    plt.show()
    #arbitrary function that stores sorted list of character indeces
    indices = sorted(range(len(x_cntr_list)), key=lambda k:
x_cntr_list[k])
    img_res_copy = []
    for idx in indices:
        # stores character images according to their index
        img_res_copy.append(img_res[idx])
    img_res = np.array(img_res_copy)

    return img_res
```

```python
def segment_characters(image) :

    # Preprocess cropped license plate image using Otsu's Binarization
thresholding
    _, img_binary_lp = cv2.threshold(image, 0, 255,
cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    # Morphological Dilation and Erosion to remove the noise
    img_binary_lp = cv2.erode(img_binary_lp, (3,3))
    img_binary_lp = cv2.dilate(img_binary_lp, (3,3))

    LP_WIDTH = img_binary_lp.shape[0]
    LP_HEIGHT = img_binary_lp.shape[1]


    # Estimations of character contours sizes of cropped license plates
    dimensions = [LP_WIDTH/6,
                        LP_WIDTH/2,
                        LP_HEIGHT/10,
                        2*LP_HEIGHT/3]
    plt.imshow(img_binary_lp)
    plt.show()

    # Get contours within cropped license plate
    char_list = find_contours(dimensions, img_binary_lp)

    return char_list

def fix_dimension(img):
    new_img = np.zeros((28,28,3))
    for i in range(3):
        new_img[:,:,i] = img
    return new_img

#database connection
conn = pyodbc.connect('Driver={SQL Server};'
                'Server=DESKTOP-SHEAU6H\SQLEXPRESS;'
                'Database=carsDB;'
                'Trusted_Connection=yes;')

cursor = conn.cursor()
arduino=serial.Serial('com5',9600)
time.sleep(2)
count = 0
countS = 0
# CNN loading model Traning to extract the right number
```

```python
model = tf.keras.models.load_model('Training_CNN' )

# Capture Live Video from Camera.
cap = cv2.VideoCapture(0)

# Set the width and height for the frame.
cap.set(3,900)
cap.set(4,900)

# Check if camera opened successfully.
if (cap.isOpened()== False):
  print("Error opening Camera.")

while True:
    countS = 0
    # Capture frame-by-frame.
    success,img = cap.read()
    ##############################

    # Show date_time in frame during live capture
    #date_time = str(datetime.now().strftime("Date %d-%m-%Y Time
%I:%M:%S:%f"))
    #cv2.putText(img, date_time, (10, 80), cv2.FONT_ITALIC, 1, (0, 105,
255), thickness=2)
    #################################

    # Convert Color Frame to Gray-Scale.
    gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Load Haar-Cascade file for plate-number detection.
    PlateCascade = cv2.CascadeClassifier("haarcascade_plate_number.xml")
    # Apply the plate-number detection on gray-image.
    numberPlates = PlateCascade.detectMultiScale(gray_image, scaleFactor=
1.6, minNeighbors= 7)

    # Draw Rectangle around the detected plate
    for (x, y, w, h) in numberPlates:
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 255), 2 )
        # Crop the detected plate from original frame.
        imgRoi = img[y:y+h, x:x+w]
        # Convert detected Frame to Gray-Scale.
        gray_Roi = cv2.cvtColor(imgRoi, cv2.COLOR_BGR2GRAY)
        plt.imshow(imgRoi)
        # Display the Gray cropped plate
        cv2.imshow("ROI", gray_Roi)
```

```
        char = segment_characters(gray_Roi)

        dic = {}
        characters = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
        for i,c in enumerate(characters):
            dic[i] = c

        output = []
        for i,ch in enumerate(char):
            img_ = cv2.resize(ch, (28,28))
            img = fix_dimension(img_)
            #preparing image for the model
            img = img.reshape(1,28,28,3)

            y = model.predict(img)
            # Return the highest expected probability of character
            character = dic[np.argmax(y[0])]
            # Storing the result in a list
            output.append(character)

        # Convert list to string
        plate_number = ''.join(output)
        print('plate_number',plate_number)

        cursor.execute("select * from cars where number ='" + plate_number
+ "'")
        carexist = len(cursor.fetchall())
        print(carexist)
        if (carexist == 1):
            arduino.write("1".encode())
            print ("on")
        else:
            count = count + 1
            if(count == 10):
                count = 0
                arduino.write("0".encode())
                print ("off")


        #return present time
        now_time = datetime.now().strftime("%I:%M:%S:%f")
        print('time : ',now_time)

        cv2.waitKey(2000)
```

```python
    # Display the resulting frame
    try:
        cv2.imshow("video",img)
    except:
        pass

    # Press Escape-ESC- on keyboard to stop recording
    key = cv2.waitKey(1)
    if key == 27:
        break

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
serial.close();
```

5.3 Arduino

5.3.1 Working Principle

- Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.
- Additionally. The Arduino IDE uses a simplified version of C++, making it easier to learn the program.
- Setup the Arduino IDE on our computer and prepare the board to receive the program via USB Cable.

5.3.2 Setting It Up

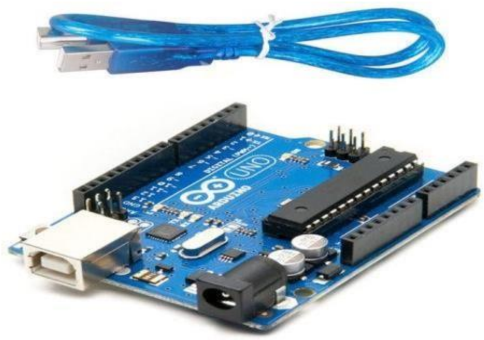


Figure 5.3-1 Arduino

Step 1 - First you must have your Arduino board and a USB Cable [5]

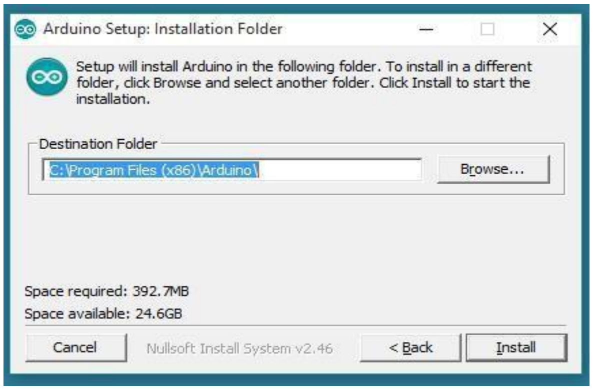


Figure 5.3-2 setup

Step 2 – Download Arduino IDE Software

Step 3 - Power up your board

- The USB Connection to the Computer is considered as Power Supply.
- Connect the Arduino Board to your Computer using the USB Cable.
- The Green Power LED should Glow.

Step 4 - Launch Arduino IDE

After your Arduino IDE Software is downloaded, you need to unzip the folder, inside the folder you can find the Application Icon with an infinity

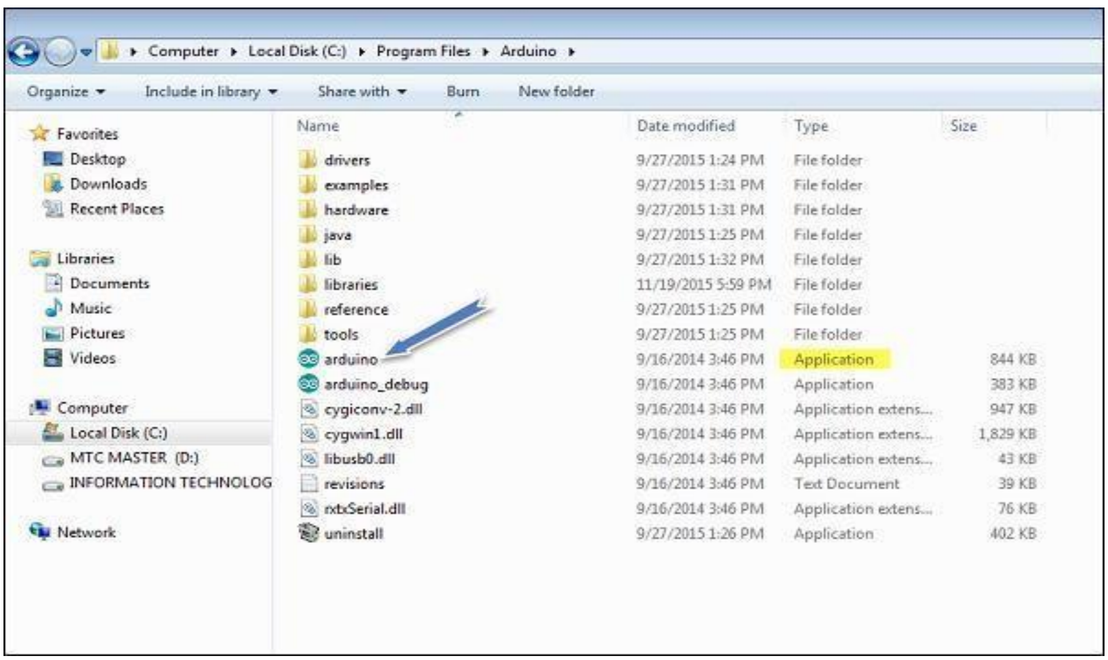


Figure 5.3-3 setup 2

label. Double click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options

- ➢ Create a new project.
- ➢ Open an existing project example.

To create a new project, select File → new.

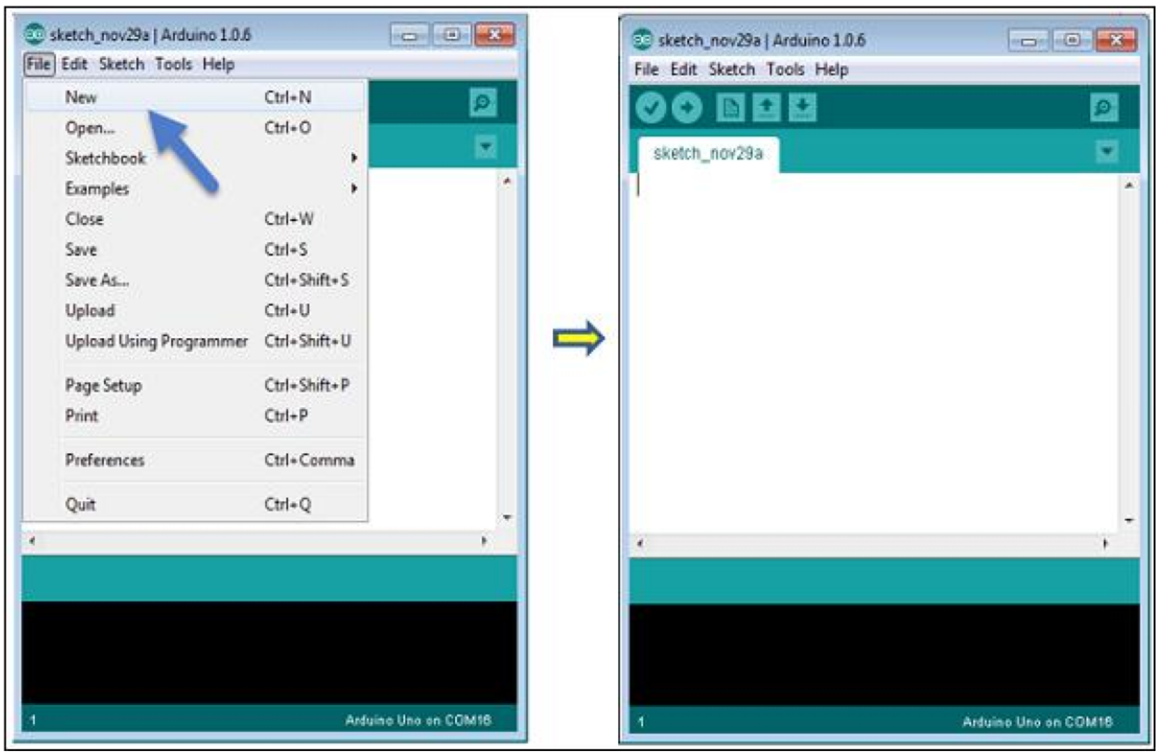


Figure 5.3-4 create new project

To open an existing project example, select File → Example → Basics →

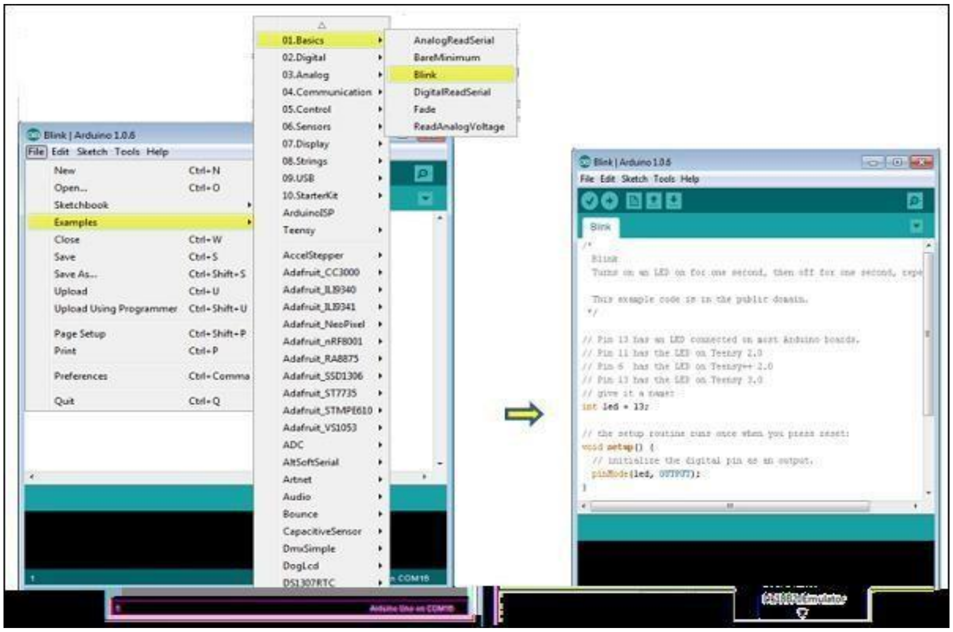


Figure 5.3-5 open exiting file

Blink.

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board

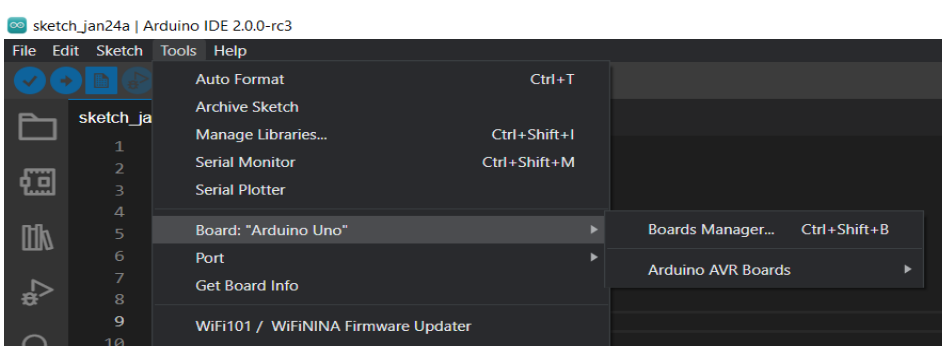


Figure 5.3-6 port connection

connected to your computer. Go to Tools → Board and select your board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to 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 Arduino board and re-open the menu, the entry that disappears should be of the Arduino board.

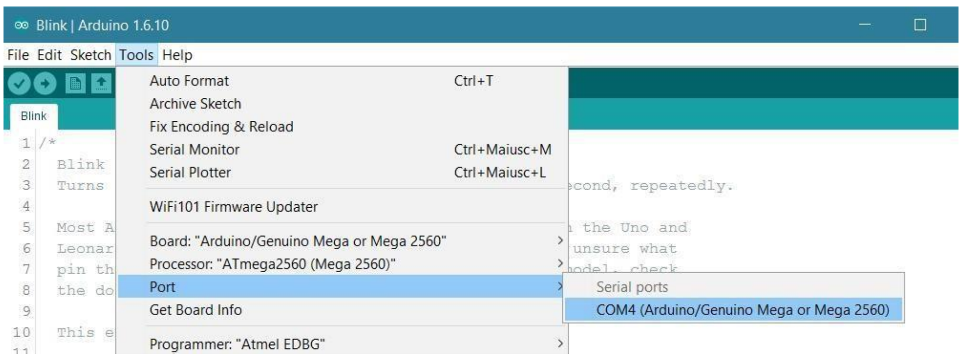


Figure 5.3-7 select your serial

Reconnect the board and select that serial port.

Step 8 – install mfrc522 library from (sketch)

1- include library 2-manage library

3- search for mfrc522

After that we will get this code to know RFID card serial in hex

5.3.3 RFID serial code

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         9          // Configurable, see typical pin layout
 above
#define SS_PIN          10         // Configurable, see typical pin layout
 above

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance

void setup() {
  Serial.begin(9600);   // Initialize serial communications with the PC
  while (!Serial);    // Do nothing if no serial port is opened (added for
 Arduinos based on ATMEGA32U4)
  SPI.begin();      // Init SPI bus
  mfrc522.PCD_Init();   // Init MFRC522
  delay(4);       // Optional delay. Some board do need more time after in
it to be ready, see Readme
  mfrc522.PCD_DumpVersionToSerial();  // Show details of PCD -
 MFRC522 Card Reader details
  Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."))
;
}

void loop() {
  // Reset the loop if no new card present on the sensor/reader. This save
s the entire process when idle.
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  // Dump debug info about the card; PICC_HaltA() is automatically called
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

5.3.4 Arduino Code

```
/*
 *  RFID Module RC522
 *  Simple Project:-
 Arduino will read RFID Tag and Display To the Serial Monitor!
 *
 *  RFID RC522 and Arduino Uno Pin Configuration
 *
 *  RFID RC522          Arduino Uno
 *  SS/SDA              D10
 *  SCK                 D13
 *  MOSI                D11
 *  MISO                D12
 *  IRQ                 Not Connected
 *  GND                 GND
 *  RST                 D9
 *  3.3V                3.3V
 *
 *  //Note: We connected LCD to the Analog Pins.
 *  Remember, Analog Pins can be act as a Digital Pins,
 *  But Digital Pins CAN NOT be act as a Analog Pins.
 *
 *  LCD(16x2) Connections
 *
 *  VSS pin to GND
 *  VCC pin to 5V
 *  //We are connecting "Contrast pin to GND for Maximum Brightness"
 *  //You can also connect it to 10K Pot if you wish to change the Brightn
ess Of LCD.
 *  VEE pin to GND
 *  RS pin to Analog pin A0
 *  R/W pin to ground
 *  Enable pin to Analog pin A1
 *  D4 pin to Analog pin 5
 *  D5 pin to Analog pin 4
 *  D6 pin to Analog pin 3
 *  D7 pin to Analog pin 2
 *  LED+ pin to POWER
 *  LED- pin to GND
 *
 *  motor Digital Pin 2 , 3
 *
 *  RGB LED
 *  From RGB LED, We need Only Red and Green Color,
```

```
 *  So we are not connecting Blue pin of the RGB.
 *
 *  Red pin to Digital pin 6
 *  Green pin to Digital pin 7
 *
 *  Buzzer to Digital pin 8
 *
 *
 *  Note:- RFID uses SPI Protocol to transfer the information.
 *  I took the Melody(for the Buzzer) from,
 *  File -> Examples -> Digital -> ToneMelody
 */

//Include sections
#include <MFRC522.h>

#include <MFRC522Extended.h>

#include <deprecated.h>

#include <require_cpp11.h>

#include <SPI.h>

#include <LiquidCrystal.h>

#include "pitches.h"

//Define Component to Arduino Pins
#define SS_PIN 10
#define RST_PIN 9

#define Red_LED 6
#define Green_LED 7

#define Buzzer 8
int relay1 = 2;
int relay2 = 3;
int datafromUser = 0;
int value = 0;

//initialize the library with the numbers of the interface pins
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);

MFRC522 rfid(SS_PIN, RST_PIN);
```

```
//Unique ID of RFID Tag, which you want to give access.
int My_RFID_Tag[5] = {
  0xE9,
  0x53,
  0xEA,
  0xB2
};

//variable to hold your Access_card
boolean My_Card = false;

// notes in the melody, taken from:
//File -> Examples -> Digital -> ToneMelody
int melody[] = {
  NOTE_C4,
  NOTE_G3,
  NOTE_G3,
  NOTE_A3,
  NOTE_G3,
  0,
  NOTE_B3,
  NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4,
  8,
  8,
  4,
  4,
  4,
  4,
  4
};

void setup() {
  // put your setup code here, to run once:
  //set the pins as an input/output
  pinMode(Red_LED, OUTPUT);
  pinMode(Green_LED, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(4, INPUT);
  //motor Connnected to pin Digital Pin 2,3
```

```
  pinMode(relay1, OUTPUT); // set pin as output for relay 1
  pinMode(relay2, OUTPUT); // set pin as output for relay 2

  // keep the motor off by keeping both HIGH
  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);

  Serial.begin(9600); // initialize serial monitor with 9600 baud
  //Initialise the LCD to 16x2 Character Format
  lcd.begin(16, 2);
  //Initialise motor and RFID
  SPI.begin();
  rfid.PCD_Init();
}

void loop() {
  My_Card = true;
  value = digitalRead(4);
  Serial.println(value);
  if (digitalRead(4) == LOW)
  {
    lcd.clear();
    lcd.print("Car Detected ");
    delay(500);
    if (Serial.available() > 0) {
      datafromUser = Serial.read();

      if (datafromUser == '1') {
        datafromUser = 0;
        My_Card = true;
        lcd.clear();
        lcd.print("please place your ");
        lcd.setCursor(0, 1);
        lcd.print("card here. ");
        while(My_Card)
        {
            //Check if any RFID Tags Detected or not?
            if (rfid.PICC_IsNewCardPresent()) {

              //if RFID Tag is detected, check for the Unique ID,
              //and print it on the Serial Window
              if (rfid.PICC_ReadCardSerial())
              {
                    lcd.clear();
```

```
                    lcd.print("Reading Card");
                    delay(2000);

                    //Compare this RFID Tag Unique ID with your My_RFID_Ta
g's Unique ID
                    for (int i = 0; i < 5; i++) {
                      //if any one Unique ID Digit is not matching,
                      //then make My_Card = false and come out from loop
                      //No need to check all the digit!
                      if (My_RFID_Tag[i] != rfid.uid.uidByte[i]) {
                        My_Card = false;
                      }
                    }
                    Serial.println();
                    delay(1000);

                    //If RFID Tag is My_Card then give access to enter int
o gate
                    //else dont open the gate.
                    if (My_Card) {
                      Serial.println("Authorized access ");
                      lcd.clear();
                      lcd.print("Authorized access ");
                      lcd.setCursor(0, 1);
                      lcd.print("welcome");
                      delay(2000);

                      //Turn on the Green LED as an indication of permissi
on is given
                      //to access the gate.
                      digitalWrite(Green_LED, HIGH);

                      //Buzzer Config, taken from:
                      //File -> Examples -> Digital -> ToneMelody
                      // iterate over the notes of the melody:
                      int i = 0;
                      while (i < 2) {
                        for (int thisNote = 0; thisNote < 12; thisNote++)
{
                          // to calculate the note duration, take one seco
nd
                          // divided by the note type.
                          //e.g. quarter note = 1000 / 4, eighth note = 10
00/8, etc.
```

```
                              int noteDuration = 1000 / noteDurations[thisNote
];
                              tone(8, melody[thisNote], noteDuration);
                              // to distinguish the notes, set a minimum time
between them.
                              // the note's duration + 30% seems to work well:
                              int pauseBetweenNotes = noteDuration * 1.30;
                              delay(pauseBetweenNotes);
                              // stop the tone playing:
                              noTone(8);
                            }
                            i = i + 1;
                            delay(500);
                          }
                          delay(1000);

                          //Now, Open the gate with the help of Motor

                          delay(200);
                          lcd.clear();
                          lcd.print("Gate is Open");
                          lcd.setCursor(0, 1);
                          lcd.print("Now!");
                          delay(3000);
                          digitalWrite(relay1, LOW); // turn relay 1 ON
                          digitalWrite(relay2, HIGH); // turn relay 2 OFF
                          Serial.println("Rotating in CCW");
                          delay(200); // wait for 3 seconds

                          lcd.clear();

                          // stop the motor
                          digitalWrite(relay1, HIGH); // turn relay 1 OFF
                          digitalWrite(relay2, HIGH); // turn relay 2 OFF
                          Serial.println("Stopped");
                          delay(2000); // stop for 2 seconds
                          for (int i = 0; i < 63; i++)  // This is where I wan
t to flush any remaing data
                          {
                            if (Serial.available() > 0)
                            {
                              byte discard = Serial.read();
                            }
                          }
                          //Give 10 Sec delay to enter the gate
```

```
                    //After that gate will again closed!
                    for (int i = 10; i > 0; i--) {
                      lcd.print("warning! Gate");
                      lcd.setCursor(0, 1);
                      lcd.print("will close in ");
                      lcd.print(i);
                      lcd.print(" Sec");
                      delay(1000);
                      lcd.clear();
                    }

                    //Now,gate is closed and Green LED is Turned-Off.

                    digitalWrite(Green_LED, LOW);
                    delay(200);
                    lcd.clear();
                    lcd.print("Gate is Close");
                    lcd.setCursor(0, 1);
                    lcd.print("Now!");
                    delay(3000);
                    digitalWrite(relay1, HIGH); // turn relay 1 OFF
                    digitalWrite(relay2, LOW); // turn relay 2 ON
                    Serial.println("Rotating in CW");
                    delay(200); // wait for 3 seconds

                    digitalWrite(relay1, HIGH); // turn relay 1 OFF
                    digitalWrite(relay2, HIGH); // turn relay 2 OFF
                    Serial.println("Stopped");
                    My_Card = false;
                    delay(2000); // stop for 2 seconds

                  }

                  // If RFID Tag is not My_Card then
                  // Do not open the gate
                  //Turn-
On Red LED and Buzzer as an indication of Warning:
                  //Somebody else is trying to enter rong id .
                  else
                  {
                    Serial.println("acsses denied");
                    lcd.clear();
                    lcd.print("Card isNOT FOUND!");
                    lcd.setCursor(0, 1);
                    lcd.print("acsses denied");
```

```
                  for (int i = 0; i < 7; i++) {
                    digitalWrite(Buzzer, HIGH);
                    digitalWrite(Red_LED, HIGH);
                    delay(500);
                    digitalWrite(Buzzer, LOW);
                    digitalWrite(Red_LED, LOW);
                    delay(500);
                  }
                  delay(1000);
                }
            }
          }
      }

    } else if (datafromUser == '0') {

      lcd.clear();
      lcd.print("Car doesn't have");
      lcd.setCursor(0, 1);
      lcd.print("entry permition");
      delay(9000);
    }

  }
} else {
  My_Card = false;

  lcd.clear();
  lcd.print("Welcom");
  //Put RFID Reader into Halt, untill it not detects any RFID Tag.
  rfid.PICC_HaltA();
}
}
```

5.3.5 Data base code

```sql
CREATE TABLE cars(
  id int primary key,
  number varchar(30) ,
  created_at datetime DEFAULT NULL,
  updated_at datetime DEFAULT NULL
)

-- Dumping data for table `cars`
--

INSERT INTO cars (id, number, created_at, updated_at) VALUES
(1, 'W668AHK', '2022-01-25 16:54:13', '2022-01-25 16:54:13'),
(2, 'P688CC', '2022-01-25 16:54:13', '2022-01-25 16:54:13'),
(3, '222RXP', '2022-01-25 16:54:13', '2022-01-25 16:54:13'),
(4, 'JUX703', '2022-01-25 16:54:13', '2022-01-25 16:54:13'),
(5, 'CA8902', '2022-01-25 16:54:13', '2022-01-25 16:54:13'),
(6, 'JXF600', '2022-01-25 16:54:13', '2022-01-25 16:54:13'),
(7, 'VSZ422', '2022-01-25 16:54:13', '2022-01-25 16:54:13')

CREATE TABLE cars_history (
  id int primary key,
  number varchar(30) DEFAULT NULL,
  date datetime DEFAULT NULL,
 state varchar(20)
)
```

Chapter 6

6 The Result

6.1 Hardware model

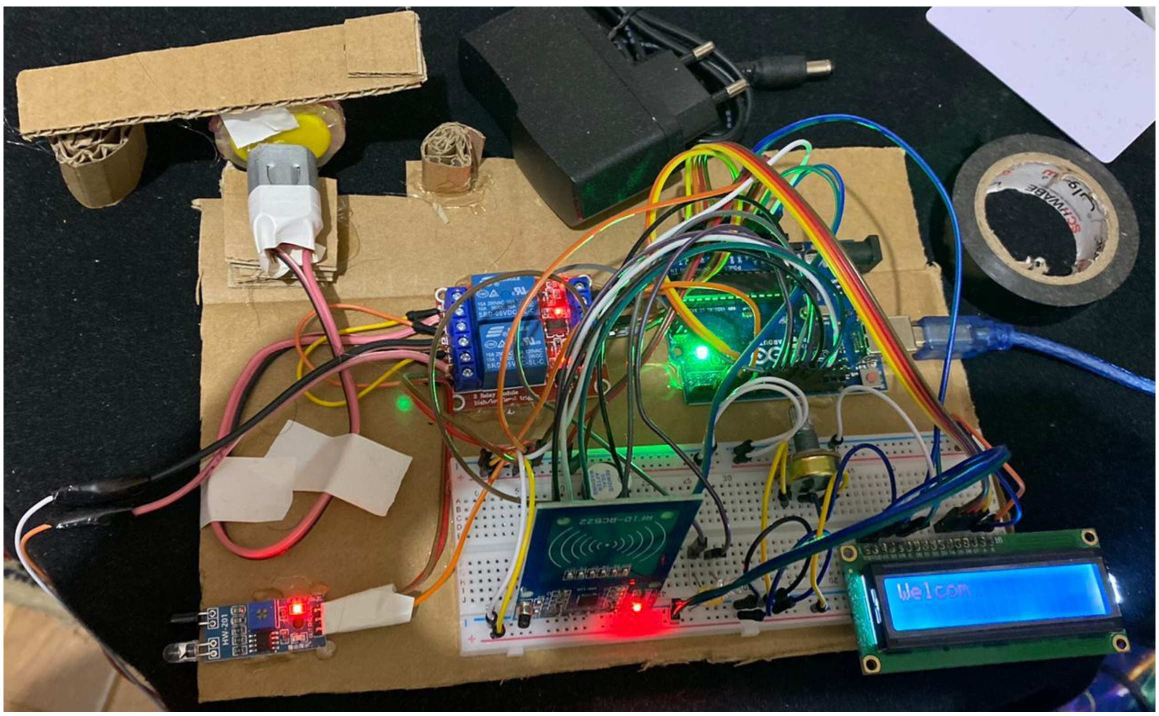


Figure 6.1-1 Hardware model

Figure 6.1-2

6.2 License plate extraction

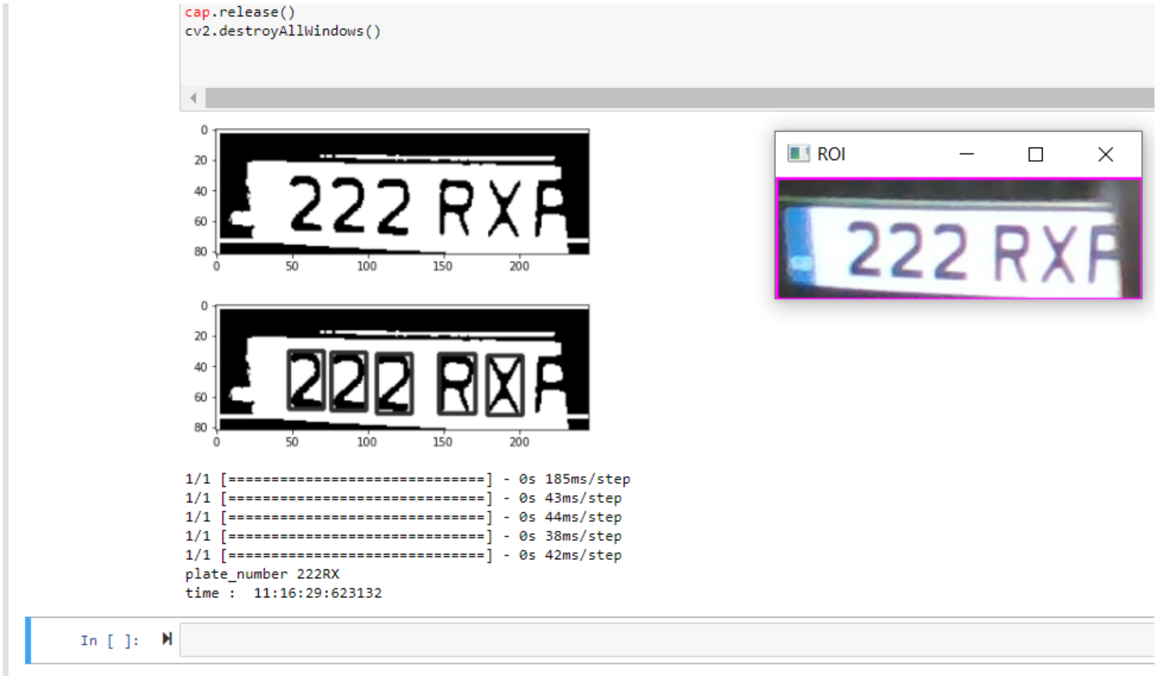


Figure 6.2-1 license plate extraction

Chapter 7

7 Conclusion and recommendation

In this study, for the automation of gate barrier, two methods have been proposed. One is RFID Technology and the other is by the use of machine learning, detection of car license plate. The objectives specified have been achieved. The read range 3-5m of detection of passive RFID tag is achieved by the use of UHF RFID reader and its proper implementation. The reader is interfaced with the Arduino board and programmed well such that when the vehicle with valid tag enters the read range of the reader the gate is opened, the RF waves of the reader can be able to detect the tag beyond 5m. The Reader tag identification rate is 100%. This system is designed in reference with the gates. To this system a camera was added, which was fixed to computer, for the purpose of detection car license plate, which were trained using CNN model and Open CV. The accuracy of detection is 82.35%, due to the non-availability of car license dataset, but it can be improved if more license plate is available for training the data more efficiently. This system has been interfaced with the Arduino board such that both systems can run, which means if vehicle enters, he gains access to the gate and if a car has the authorization, then it will ask for RFID to get in. Another problem has been solved, with the use of IR sensor, the gate
was programmed in such a way that it cannot use this process till the IR sensor detect there is a car or not. If a car is there then the process run automatically

The main recommendations for this system, it can be designed for security purpose. By keeping a data logger, the entry and exit times of the RFID tagged vehicle can be stored in the computer. For car license plate, when the car placed in front of the camera is detected, ensuring the camera captures the image of the car and saves in the data base and the vehicle entry time is also saved.