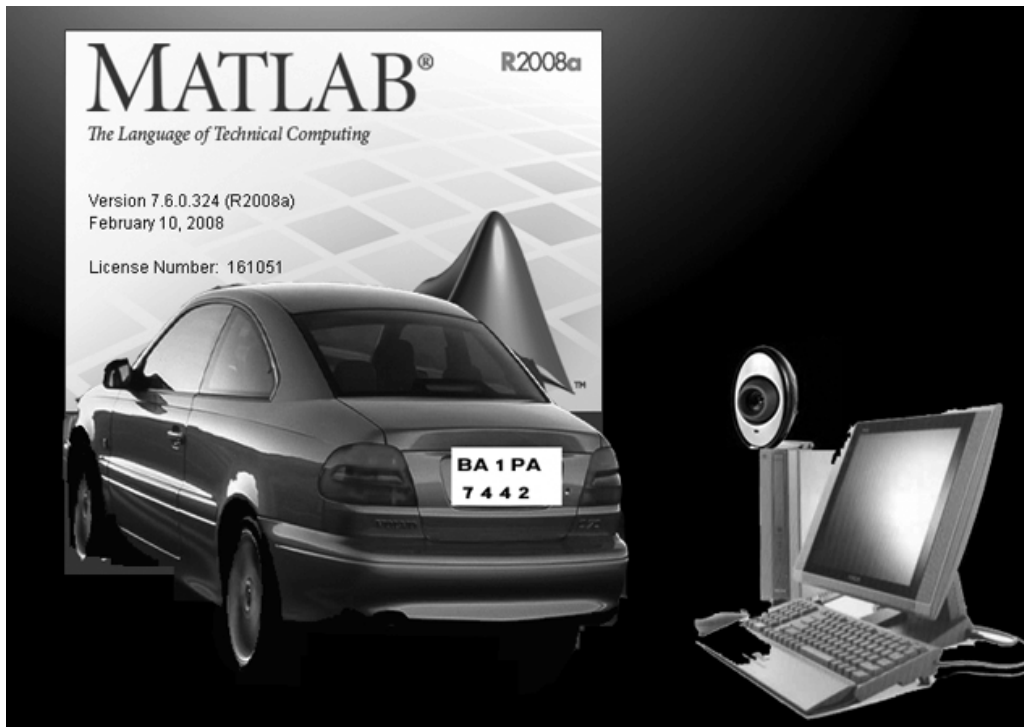


TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS
THAPATHALI, KATHMANDU



Report On:-**ANPR**(**Automatic Number Plate Recognition**) **Using**
ALR(**Automatic Line Tracking Robot**)

Submitted By:

Ashok Basnet (062-DCT-407)
Bishnu Parajuli (062-DCT-410)
Ishor Prasad Rijal (062-DCT-416)
Kabindra Kaji Bajracharya (062-DCT-419)
Kiran Karki (062-DCT-420)
Krishna Bahadur Shrestha (062-DCT-421)
Mohan Pandey (062-DCT-422)
Ram Hari Regmi (062-DCT-430)

Submitted To:

Raju Pandey
Pravin Shakya
(Project Supervisors)
Department Of
Computer
& Electronics
Engineering

PREFACE

It is almost known that the world is nowadays being more dependent in the computer than the human manpower. Today in the developed countries more than 90% of the job is done by automated system and this is due to the accuracy, reliability and versatile nature of the computer. In the developed western countries many big factories, nuclear plants, communication centers all of them are under the control of the computer because that's not the place where human hand can reach up. To add the further mile for this step we have presented an automated number plate recognition using automatic line tracking robot. The project is based on image processing technology using popular software MATLAB.

ACKNOWLEDGEMENT

Initially, we would like to express our sincere gratitude and gratefulness to our professor Mr. Pravin Shakya and Mr. Raju Pandey for providing us with all sorts of basic ideas and techniques essential for carrying out this project work from the very beginning to the end and enabled us to present this dissertation in this form. The teaching staffs also deserve our sincere thanks for sharing their discussion and exchange of ideas.

We are very much grateful to the Tribhuvan University (Institute of Engineering), Thapathali Campus for providing us an enthusiastic support and opportunity. Department of Computer and Electronics also must come in special mention for their unstinting cooperation in completion of this project. We would also like to thank industrial expo committee for providing materials to test and complete this project. We would like to give heartily thanks to our friends who have provided a great help and cooperation for the existence of this output. Our obligation goes to our family and all our friends who assisted us directly and indirectly in completing this study.

Lastly, we would like to extend our sincere gratitude to the known and unknown writers of the books and references that has been taken during the preparation of this project work.

ABSTRACT

The ANPR (Automatic Number Plate Recognition) using ALR (Automatic line Tracking Robot) is a system designed to help in recognition of number plates of vehicles. This system is designed for the purpose of the security and it is a security system. This system is based on the image processing system. This system helps in the functions like detection of the number plates of the vehicles, processing them and using processed data for further processes like storing, allowing vehicle to pass or to reject vehicle. This system also helps to conduct the graphic images of the vehicles which can be further stored in the database in text format reducing size of data to be stored.

List of abbreviation

Abbreviation	Definition
ANPR	Automatic Number Plate Recognition
ALR	Automatic line tracking Robot
OCR	Optical Character Recognition
MATLAB	Matrix Laboratory
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
PC	Personal Computer
LED	Light Emitting Diode
LDR	Light Dependent Diode
DPDT	Double Port Double Through
IC	Integrated Circuit

Table Of Contents:

S/No.	Contents	Page No.
1.	Objectives	1
2.	Introduction	2
	2.1 System Block Diagram	3
	2.2 Image Processing and MATLAB	4
	2.3 Algorithm	5
	2. 4 List of all the functions and headers used in the code	7
	2.5 ANPR Algorithm	9
	2.6 ALR	12
	2.7 Microcontroller	16
	2.8 ISP- Programmer Interface	21
	2.9 Programming needed for the robot	23
	2.10 Parallel Port	26
	2.11 Components Used	30
3.	Problem Domain	36
4.	Requirement	37
5.	Project Scheduling	39
6.	Design	40
7.	Cost Estimation	43
8.	Limitations	44
9.	Conclusion	45
10.	References	46

OBJECTIVES

1. To be familiar with image processing technique for detection of Number plate area.
2. To be familiar with OCR.
3. To be familiar with MATLAB software.
4. To interface hardware using parallel port of the computer.
5. To develop a line tracking robot which can move in a guided track and detect the closed door using IR.
6. To be familiar with Microcontroller and related components for building automatic robot.
7. To know assembly level programming language and implement it in ALR.
8. To develop a security system.

INTRODUCTION

Massive integration of information technologies into all aspects of modern life caused demand for processing vehicles as conceptual resources in information systems. Because a standalone information system without any data has no sense, there was also a need to transform information about vehicles between the reality and information systems. This can be achieved by a human agent, or by special intelligent equipment which is able to recognize vehicles by their number plates in a real environment and reflect it into conceptual resources. Because of this, various recognition techniques have been developed and number plate recognition systems are today used in various traffic and security applications, such as parking, access and border control, or tracking of stolen cars.

In entrance gate, number plates are used to identify the vehicles. When a vehicle enters an input gate, number plate is automatically recognized and stored in database and black-listed number is not given permission. When a vehicle later exits the place through the gate, number plate is recognized again and paired with the first-one stored in the database and it is taken a count. Automatic number plate recognition systems can be used in access control. For example, this technology is used in many companies to grant access only to vehicles of authorized personnel.

In some countries, ANPR systems installed on country borders automatically detect and monitor border crossings. Each vehicle can be registered in a central database and compared to a black list of stolen vehicles. In traffic control, vehicles can be directed to different lanes for a better congestion control in busy urban communications during the rush hours.

Automatic Line Tracking Robot (ALR) is used in this project as a vehicle which contains circuitry for moving in a guided track. It will have mechanism to detect the opened and closed door. It also will have capacity to park in the given parking area.

2.1 System Block Diagram

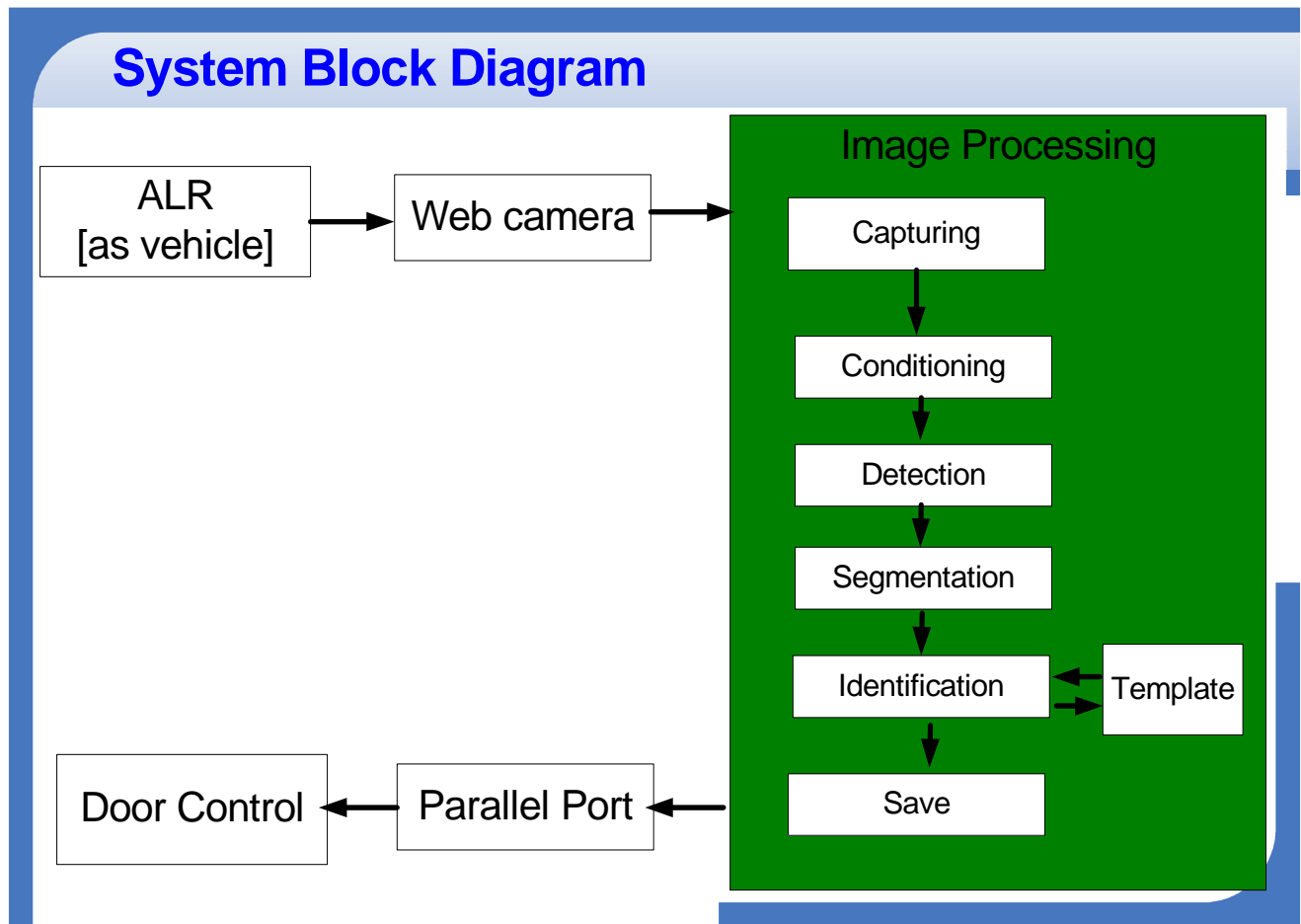


Fig. System Block Diagram

Description:

The system is developed as a security system. The vehicle is a Automatic Line Tracking Robot (ALR) which contains number plate. The webcam is fitted in the door and when ALR comes in front of it, it detects and captures the image of ALR along with the number plate. The computer loaded with MATLAB software processes the image and saves the number plate in text file along with date and time. Then it sends command to parallel port of the computer and with the circuitry along with parallel port, the door is opened. When vehicle enters the gate, it closes the door.

2.2 Image Processing and MATLAB

2.2.1 Matlab

Matlab is an abbreviation of Matrix Laboratory. It is a popular Mathematical Programming Environment used extensively in Education as well as in Industry. The trick behind Matlab is that everything is represented in the form of arrays or matrices. Mathematical Operations starting from simple algebra to complex calculus may be conveniently carried out using this environment. The main use of Matlab in Software Development is Algorithm Design and Development. Code developed in Matlab can be converted into C, C++ or Visual C++. Additionally Matlab may be called as ActiveX Object from still higher level languages like Visual Basic, etc.

2.2.2 Image Prcessing

Image Processing generally involves extraction of useful information from an image. This useful information may be the dimensions of an engineering component, size of diagnosed tumor, or even a 3D view of an unborn baby. The main areas of application of Image Processing are Bio-Medical, Engineering, Quality Control, Face Detection, Traffic Control etc.

Block Diagram of Image Processing Steps

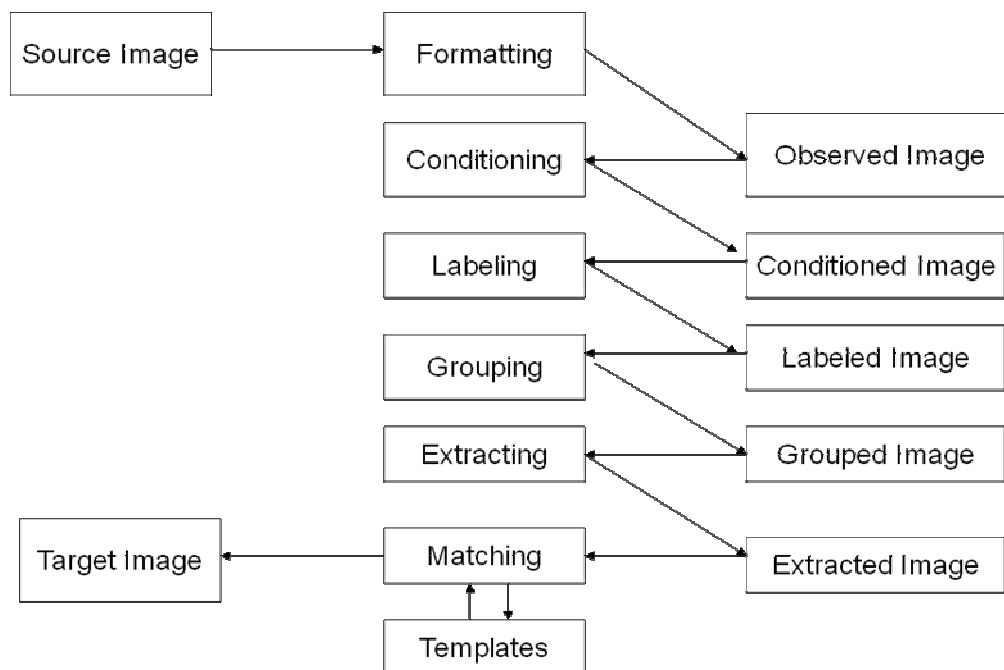


Fig. Image processing Steps

2.3 Algorithm

2.3.1 Algorithm for ANPR system

1. Input image from webcam.
2. Convert image into binary.
3. Detect number plate area.
4. Segmentation.
5. Number identification.
6. Save to file in given format.

2.3.2 Input Image from file.

1. Capture image from webcam..
2. Store the captured image into a image file for further processing.

2.3.3 Convert image into binary.

1. Identify the intensity of the image.
If image intensity = high
 Reduce intensity
 Else if intensity = low
 Increase intensity
 Else
 No change.
2. Convert image into grayscale.
3. Calculate appropriate threshold value for the image.
4. Convert the image into binary image using the calculated threshold.

2.3.4 Detecting Number plate area

1. Fill small holes including numbers of Number plate so that number plate area will be large to isolate from figure.
2. Determine width and height of the image.
3. Scan each pixel of line counting number of white pixels in the following system,
 If number of 'white' pixels $< x$; pixels become 'black'
 Else; no change
 If number of 'white' pixels $> y$; pixels become 'black'
 Else; no change
 The value of x and y may be changed according the image intensity and plate area.
4. Use the step no. 3 for both horizontal and vertical direction.
5. Check number of possible areas.
 If number of areas > 1
 Select suitable area
6. Logically AND with binary image obtained at "Convert image into binary" algorithm.
7. Crop the required area.

2.3.5 Segmentation

1. Filter the noise level present in the image.
2. Clip the plate area in such a way that only numbers of plate area extracted.
3. Separate each character from the plate.

2.3.6 Number identification

1. Create the template file from the stored template images.
2. Resize image obtained from segmentation to the size of template.
3. Compare each character with the templates.
4. Store the best matched character.

2.3.7 Save to file in given format

1. Open a text file in write mode.
2. Store the character obtained from the number identification process to text file in given format.
3. Close the file.

2.4 List of all the functions and headers used in the code

2.4.1 Inbuilt functions of Matlab used

clc: Clear Command Window

clear - Remove items from workspace, freeing up system memory.

Clear all - removes debugging breakpoints in M-files and reinitializes persistent variables.

Imread - Read image from graphics file.

Rgb2gray - Convert RGB image or colormap to grayscale.

Graythresh - Global image threshold using Otsu's method.

Im2bw - Convert image to binary image, based on threshold. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black).

Imfill - Fill image regions and holes.

Double - Convert to double precision for arithmetic calculation.

Size - returns the sizes of each dimension of an image in matrix a and b.

Mat2cell - Divide matrix into cell array of matrices.

Save - Save workspace variables to disk.

Medfilt2 - Performs median filtering of the matrix A using the default 3-by-3 neighborhood.

Fopen - Open file, or obtain information about open files.

Bwlabel - returns in num the number of connected objects found in BW.

Bwareaopen - Morphologically open binary image (remove small objects).

Max - Largest elements in array.

Min - Smallest elements in array.

Edge - Find edges in grayscale image.

Find - Find indices and values of nonzero elements.

Zeros - Create array of all zeros.

Islogical - Determine whether input is logical array.

Resize - Resize image.

Load - Load workspace variables from disk.

Corr2 - 2-D correlation coefficient.

Fprintf - Write text to device.

Isempty - Determine whether array is empty.

Fclose - Close one or more open files.

Imshow - Display image.

For Webcam:-

videoinput - Specify the adaptor name and resolution.

preview - View webcam interface and video demo the webcam at runtime.

getsnapshot - Store the image of that instant into the variable data in a matrix.

For Parallel Port:-

daqhwinfo,digitalio,addline - for initializing the parallel port.

putvalue - puts the value to line number of the DIO configured port.

getvalue - will get the input value from parallel port,

2.4.2 User Defined functions of Matlab used

Main () – This is the starting function from which all other required functions are called and loads image to be processed.

convertBw() - Converts the RGB image into black and white using appropriate threshold value.

hSmearing()- It counts number of connected white pixels in Horizontal direction and converts the pixels into black if the number of white pixels doesn't lie within the range.

vSmearing()- It counts number of connected white pixels in Vertical direction and converts the pixels into black if the number of white pixels doesn't lie within the range.

Clip() – Clips required part of the image

createTemplates() - Create a template file from the stored template files in the given size so that it can be compared with the resized number plate characters.

Ocr() – It identifies the numbers extracted from the number plate and saves the number or character into text file.

Identify() - Identifies extra disturbances and lines.

sameDim() – Resizes the extracted characters into the size as of templates.

readLetter() - Computes the correlation between template and input image.

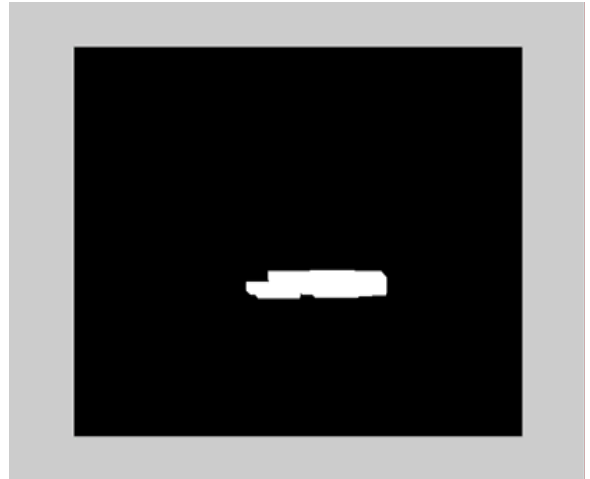
2.5 ANPR Algorithm

2.5.1 Plate Area Extraction:

Plate region extraction is the first stage in this algorithm. Image captured from the camera is first converted to the binary image consisting of only 1's and 0's (only black and white). by thresh holding the pixel values of 0 (black) for all pixels in the input image with luminance less than threshold value and 1 (white) for all other pixels. Captured image (original image) and diarized image are shown in Figure 1(a) and 1(b) respectively.



(a) Captured image



(a) Plate region



(b) Binarized image



(b) Image involving only plate

Fig. 1(a) Original Image, (b) Binarized Image

Fig. 2 (a) Plate region, (b) Image involving only plate

The binarized image is then processed using some methods. To find the plate region, firstly smearing algorithm is used. Smearing is a method for the extraction of text areas on a mixed image. With the smearing algorithm, the image is processed along vertical and horizontal runs (scan-lines). If the number of white pixels is less than a desired threshold or greater than any other desired threshold, white pixels are converted to black. In this system, threshold values are selected as 10 and 100 for both horizontal and vertical smearing.

If number of 'white' pixels < 10 ; pixels become 'black'.

Else ; no change

If number of 'white' pixels > 100 ; pixels become 'black'

Else ; no change

After smearing, a morphological operation, dilation, is applied to the image for specifying the plate location. However, there may be more than one candidate region for plate location. To find the exact region and eliminate the other regions, some criteria tests are applied to the image by smearing and filtering operation. The processed image after this stage is as shown in Figure 2(a) and image involving only plate is shown in Figure 2(b).

After obtaining plate location, region involving only plate is cut giving the plate as shown in Figure 3.

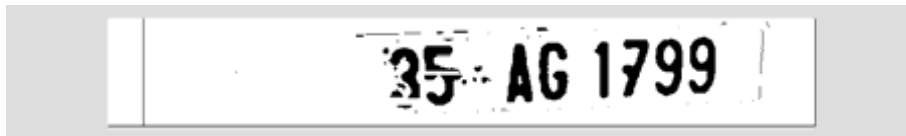


Fig. 3 Plate Image

2.5.2 SEGMENTATION:

In the segmentation of plate characters, license plate is segmented into its constituent parts obtaining the characters individually. Firstly, image is filtered for enhancing the image. and removing the noises and unwanted spots. Then dilation operation is applied to the image for separating the characters from each other if the characters are close to each other. After this operation, horizontal and vertical smearing are applied for finding the character regions. The result of this segmentation is in Figure 4.



Fig. 4 Locations of plate characters

The next step is to cut the plate characters. It is done by finding starting and end points of characters in horizontal direction. The individual characters cut from the plate are as follows in Figure 5.

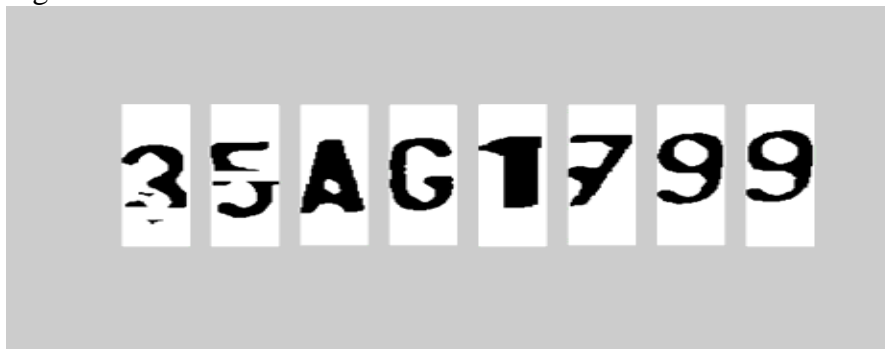


Fig. 5 Individual characters

2.5.3 CHARACTER RECOGNITION:

Before recognition algorithm, the characters are normalized. Normalization is to refine the characters into a block containing no extra white spaces (pixels) in all the four sides of the characters. Then each character is fit to equal size as shown in Figure 6.

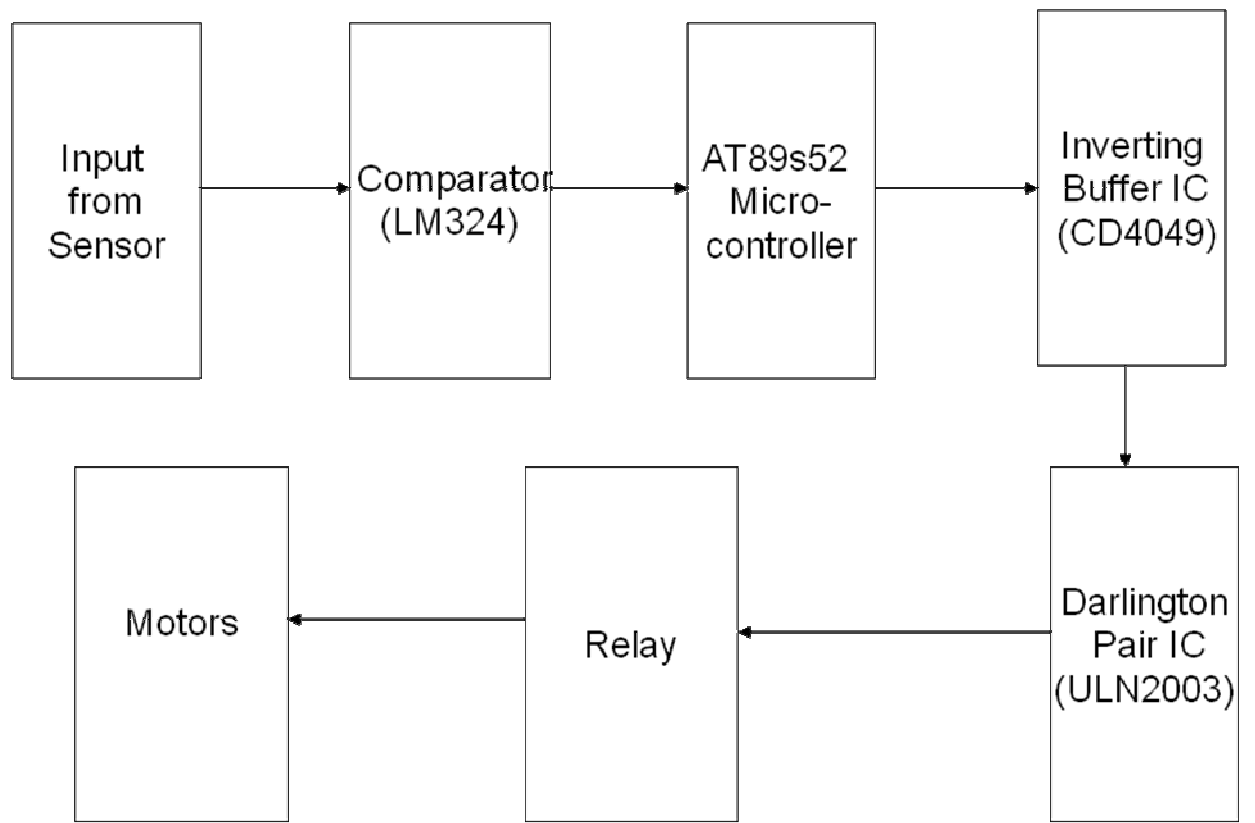


Fig. 6 Equal-sized characters

Fitting approach is necessary for template matching. For matching the characters with the database, input images must be equal-sized with the database characters. Here the characters are fit to 36×18 . The extracted characters cut from plate and the characters on database are now equal-sized. The next step is template matching. Template matching is an effective algorithm for recognition of characters. The character image is compared with the ones in the database and the best similarity is measured.

2.6 ALR (Automatic Line Tracking Robot)

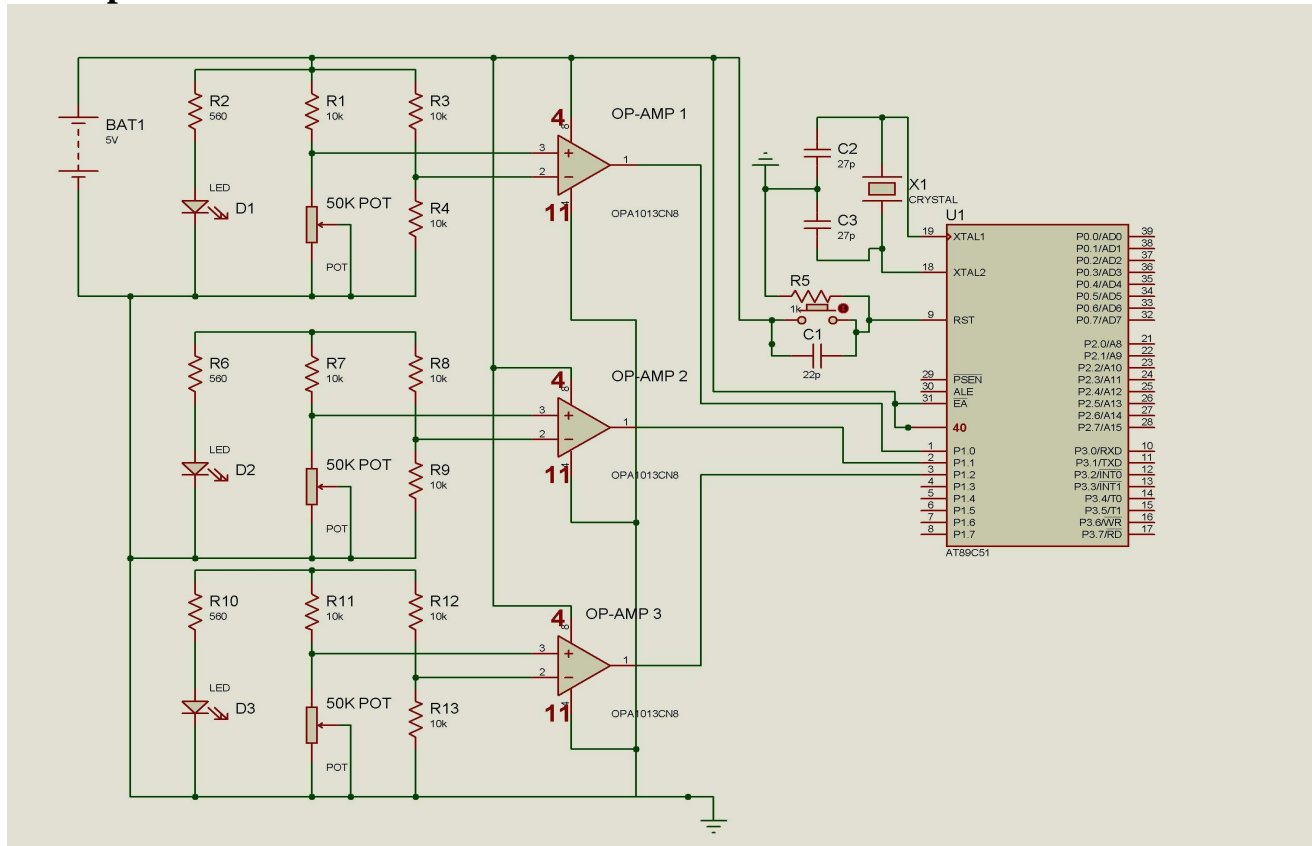
2.6.1 Block Diagram of ALR



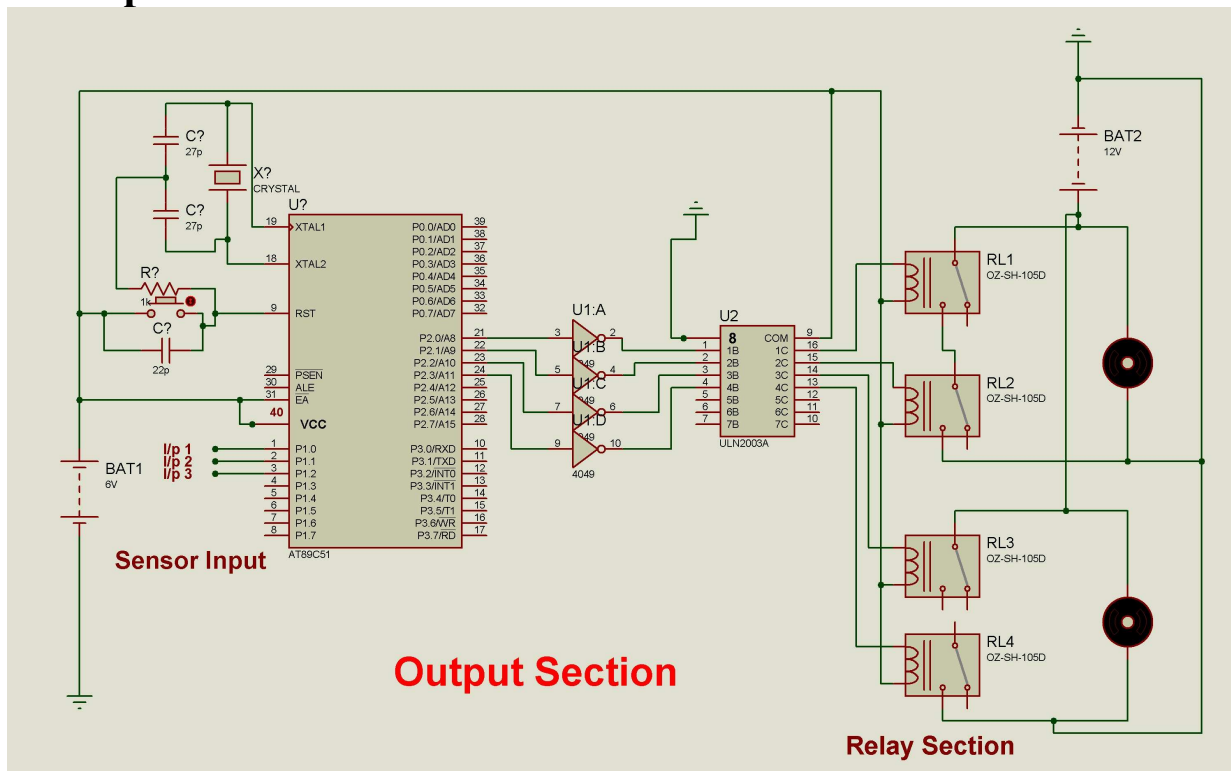
Block Diagram Of Line Tracking Robot.

2.6.2 Circuit Diagram:

Input Section:-



Output Section:-



Explanation:

The operation of the automatic line tracking robot is simply discussed below. As per the block diagram for the automatic line tracking robot the initial block of the system is sensor section. The sensor section consists of LDR and LED and a resistor for the protection of the led. As the line tracking robot tracks its path on a line i.e. of a white line. As the LDR is a special type of resistor having the property of providing a high resistance when there is no light and at the presence of light it provides a low resistance. Thus in the sensor when the led is on and reflects its light towards ldr and the ldr provides a low resistance path to the voltage in this way a simple tracking can be performed. After that the voltage comparator section, here a op-amp as a comparator is applied. It consists of two input terminal one is positive and other is negative, in the negative terminal (inverting input) a constant voltage is applied by doing a voltage divider by two 10 kohms resistors. A constant 3v is applied at the inverting input. At the positive (non-inverting) terminal the voltage through the ldr is passed. Thus when there is light sensed by the ldr or low resistance path is provided by ldr or a high voltage input at the non-inverting. Then the comparator provides a high output, that is high input to the microcontroller. In the microcontroller we have to initially define the input and output ports. Now as per the programming we will get the outputs at the output ports. At the outputs of microcontroller we applied the inverting buffer which helps to remove the load affects between the microcontroller and further output circuits. Normally a cd4049 is used as a buffer. It acts as a isolator circuit. Then a power buffer amplifier i.e. uln2003 is applied to raise the current strength which helps to drive the relays. This also acts as the buffer circuit. This provides a ground path to the relay terminal to drive a motors. Relays switching section contains the h-bridge which helps to drive the single motor in both forward and reverse condition. Relay is a switching component having the NC and NO contacts. The motor is driven through relays. In this way a normal line tracking robot is operated.

Operation of Sensor Section:

A line sensor in its simplest form is a sensor capable of detecting a contrast between adjacent surfaces, such as difference in color, roughness, or magnetic properties, for example. The simplest would be detecting a difference in color, for example black and white surfaces. Using simple optoelectronics, such as LEDs and LDRs, color contrast can easily be detected. LEDs emitter/detectors or photo-transistors/LDRs are inexpensive and are easy to interface to a microcontroller. In addition, standard red LEDs work well too.

When light shines on a white surface, most of the incoming light is reflected away from the surface. In contrast, most of the incoming light is absorbed if the surface is black. Therefore, by shining light on a surface and having a sensor to detect the amount of light that is reflected, a contrast between black and white surfaces can be detected.

Figure shows an illustration of the basics just covered.

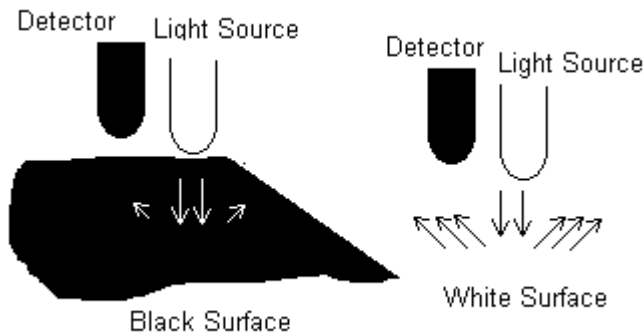


Figure 1. Light reflecting off a white and black surface.
More light is reflected from the white surface compared to the black surface.

Using what we know about light, and black and white surfaces, the objective of tracking a line is simple can be achieved using the appropriate sensors. The drive configuration for the robot is assumed to be differential, i.e., like the tracks of an army tank vehicle. From the figure, the three pairs of sensors are used to keep the robot on the line as it moves. Each sensor output is monitored to determine the location of the tape relative to the robot. The main objective of the robot is to position itself such that the tape line falls between the two extreme sensors. If the tape line ever ventures past these two extreme sensors, then the robot corrects by turning in the appropriate direction to maintain tracking. Two different types of light sensors set up in the configuration shown below will be used for line tracking.

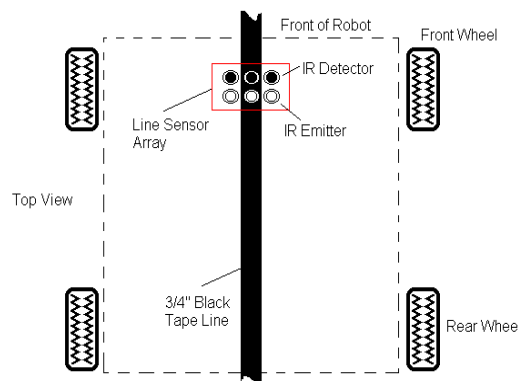


Figure 2. Top view of a robot using three pairs of emitter and detector sensors for line following.

2.7 Microcontroller AT89S52:

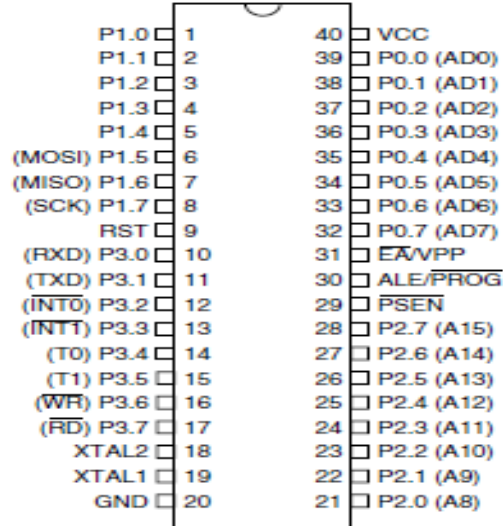
2.7.1. Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of In-System Programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

2.7.2 Features

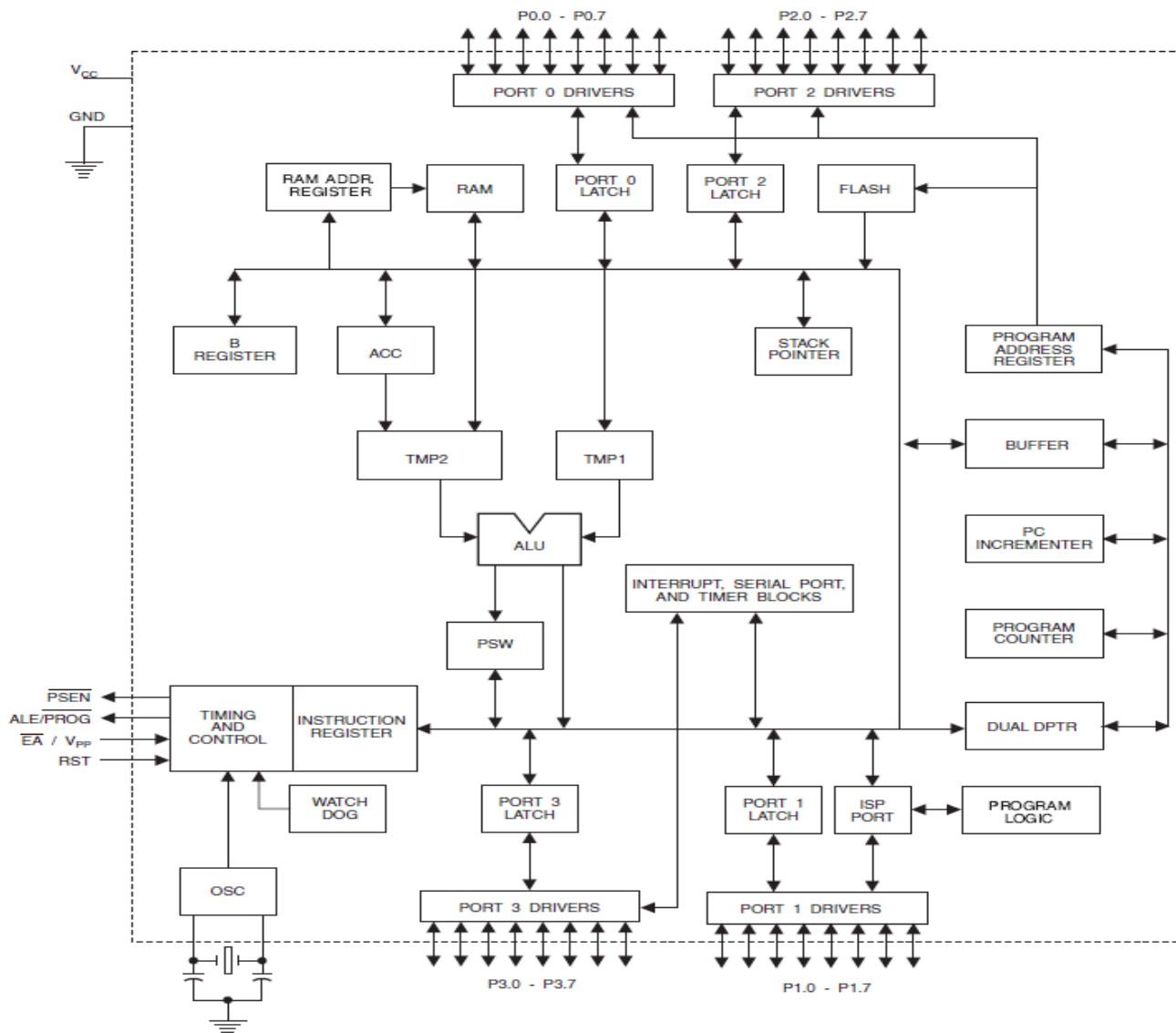
- Compatible with MCS®-51 Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
 - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Dual Data Pointer • Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

40-lead PDIP



2.7.3 Block Diagram

Block Diagram



2.7.4. Pin Description

- i. **VCC** Supply voltage (all packages except 42-PDIP).
- ii. **GND** Ground (all packages except 42-PDIP; for 42-PDIP GND connects only the logic core and the embedded program memory).
- iii. **VDD** Supply voltage for the 42-PDIP which connects only the logic core and the embedded program memory.
- iv. **PWRVDD** Supply voltage for the 42-PDIP which connects only the I/O Pad Drivers. The application board **MUST** connect both VDD and PWRVDD to the board supply voltage.
- v. **PWRGND** Ground for the 42-PDIP which connects only the I/O Pad Drivers. PWRGND and GND are weakly connected through the common silicon substrate, but not through any metal link. The application board **MUST** connect both GND and PWRGND to the board ground.
- vi. **Port 0** Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**
- vii. **Port 1** Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

viii. **Port 2** Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

ix. **Port 3** Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are

externally being pulled low will source current (IIL) because of the pull-ups. Port 3 receives some control signals for Flash programming and verification. Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

x. RST Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DIS-RTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

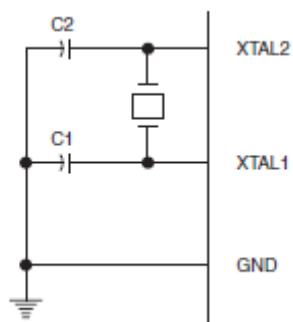
xi. ALE/PROG Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

xii. PSEN Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

xiii. EA/VPP External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

xiv. XTAL1 Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

xv. XTAL2 Output from the inverting oscillator amplifier



Operating Temperature -55°C to +125°C

Storage Temperature -65°C to +150°C

Voltage on Any Pin
with Respect to Ground -1.0V to +7.0V

Maximum Operating Voltage 6.6V

DC Output Current..... 15.0 mA

2.8 ISP- Programmer Interface

To burn the Microcontroller with appropriate HEX program file, we use ISP Flash Programmer Circuit and ISP PGM software application.

This ISP Programmer can be used either for in-system programming or as a stand-alone spi programmer for Atmel ISP programmable devices. The programming interface is compatible to STK200 ISP programmer hardware so the users of STK200 can also use the software which can program both the 8051 and AVR series devices.

2.8.1 Hardware

Figure below shows the circuit diagram of the in-system programmer interface, the power to the interface is provided by the target system. The 74HCT541 ic isolate and buffer the parallel port signals. It is necessary to use the HCT type ic in order to make sure the programmer should also work with 3V type parallel port.

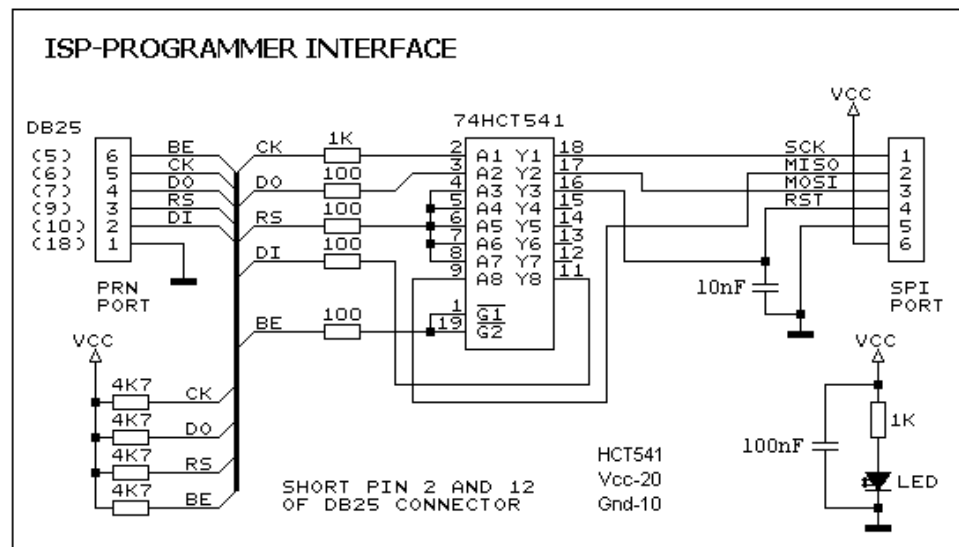


Figure: Circuit Diagram of the ISP Programmer Interface

The main screen view of the program is shown in figure below:

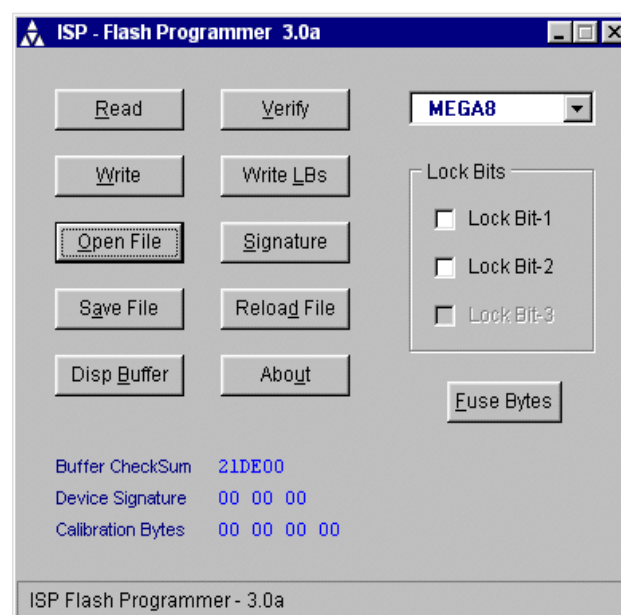


Figure: Main screen of the program ISP-Pgm Ver 3.0a

Software uses the default parallel port i.e. LPT1.

Following are the main features of this software,

- ☐ Read and write the Intel Hex file
- ☐ Read signature, lock and fuse bits
- ☐ Clear and Fill memory buffer
- ☐ Verify with memory buffer
- ☐ Reload current Hex file
- ☐ Display buffer checksum
- ☐ Program selected lock bits & fuses
- ☐ Auto detection of hardware

The software helps to load the appropriate HEX file and burn it to the microcontroller chip. The chip then can be used in the required system i.e. in our case Line Tracking Robot.

2.9 Programming needed for the robot:-

The programming for Automatic Line Tracking robot is done in KEIL programming language which uses C syntax to produce hex file that is ready to be kept in the Microcontroller(AT89s52). The ISP PGM burner software is used that uses a ISP Flash programmer circuit to burn the Microcontroller.

The programming section is divided into two parts:-

define.h:-header file containg all the necessary functions needed for the operation of robot.

main.c :- main C file that contains main function from which functions are called.

define.h

```
#include<regx51.h>                                //Header file needed for the 8051 family of microcontroller

//motor control
#define LF 0xFD
#define RF 0xFE
#define LR 0xFB
#define RR 0xF7
#define rturn 0xF6
#define lturn 0xF9
#define bothF 0xFC
#define bothR 0xF3
#define stop 0xFF
#define output P2

//input pins
sbit ip1 = P1^7;
sbit ip2 = P1^0;
sbit ip3 = P1^1;
sbit irIp = P3^0;
sbit irOp = P1^5;

void portInitialize()
void delay(unsigned char);
void lineTrack(unsigned char maxCount);
void lTurn();
void rTurn();
void RobotStop();

void portInitialize()
{
    P1=0x00;
    P2=0xFF;
}

void delay(unsigned char value)
{
    unsigned int a;
    for(a = 0; a < value; a++)
```

```

{
    TMOD =0x01;
    TH0=0x3C;
    TL0=0xB0;
    TF0=0;
    TR0=1;
    while(TF0==0);
    TR0=0;
}
}

void lineTrack(unsigned char maxCount)
{
    unsigned char count;
    count = 0;
    do
    {
        if(ip1==0 && ip2==1 && ip3==0)
            output=bothF;
        else if(ip1==0 && ip2==0 && ip3==1)
            output=LF;
        else if(ip1==1 && ip2==0 && ip3==0)
            output=RF;
        else if(ip1==0 && ip2==0 && ip3==0)
            output=stop;
        else if(ip1==0 && ip2==1 && ip3==1)
            output=LF;
        else if(ip1==1 && ip2==1 && ip3==0)
            output=RF;
        else if(ip1==1 && ip2==1 && ip3==1)
        {
            output=bothF;
            ++count;
            if (count == maxCount)
            {
                delay(5); //
                break;
            }
        }
        delay(10);
    }
    }while(1);
}

```

```

void lTurn()
{
    output = lturn;
    delay(5);
    while(ip2==0)
        output = lturn;
}

```

```
}
```

```
void rTurn()
```

```
{
```

```
    output = rturn;
```

```
    delay(5);
```

```
    while(ip2==0)
```

```
        output = rturn;
```

```
}
```

```
void RobotStop()
```

```
{
```

```
    while(1)
```

```
        output = stop;
```

```
}
```

main.c

```
#include<define.h>    //include the define.h header file
```

```
void main()
```

```
{
```

```
    portInitialize();    //Initialize the input and output ports
```

```
    lineTrack(2);
```

```
    delay(200);    //delay(1)= 50ms
```

```
    rTurn();    //Turn Right
```

```
    lineTrack(2);
```

```
    lTurn();    //Turn Right
```

```
    lineTrack(2);
```

```
    lTurn();
```

```
    RobotStop();    //Stop the Robot
```

```
}
```

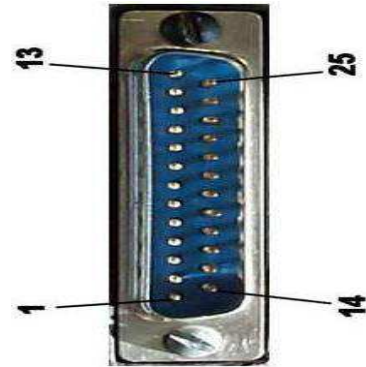
2.10 Parallel Port

2.10.1 Introduction

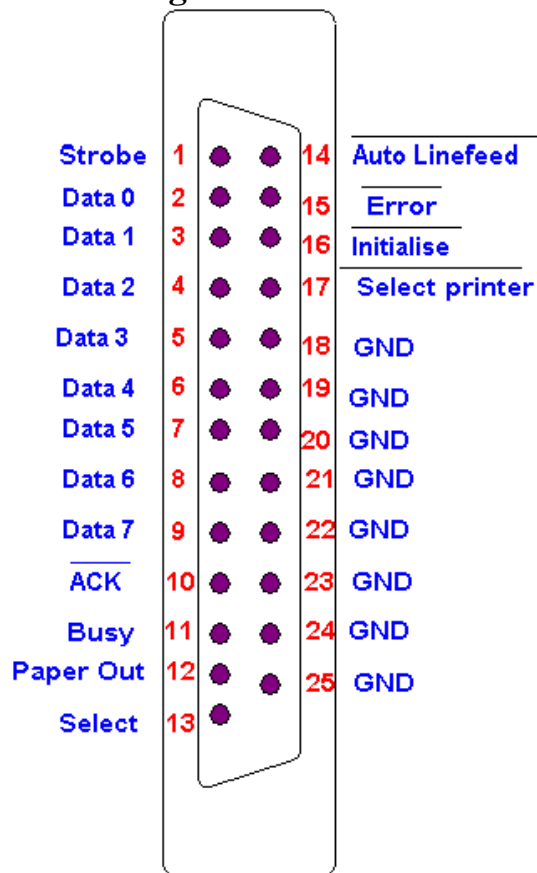
Parallel port is the most easiest and efficient way to communicate computer with external hardware and control devices.

Parallel communication requires as much wires as the no. of bits in a word for its transmission.

Parallel port is generally a 25 pin female connector with which a printer is usually attached.



2.10.2 Pin Configuration of Parallel Port



Pin Configuration of Parallel Port

2.10.3 Pin Assignment

Pin No	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

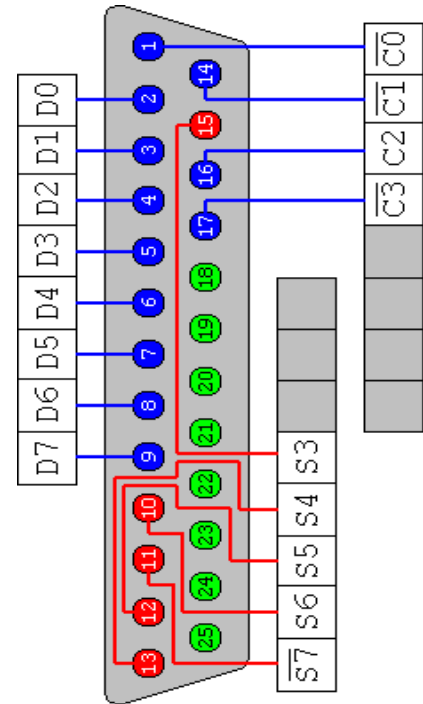
Fig: Pin assignment of Parallel Port

2.10.4 Parallel port registers

The Data, Control and status lines are connected to there corresponding registers inside the computer. So by manipulating these registers in program , one can easily read or write to parallel port with programming languages like 'C' and BASIC.

1. Data register
2. Status registers
3. Control register

Data register is connected to Data lines, Control register is connected to control lines and Status register is connected to Status lines. So what ever you write to these registers , will appear in corresponding lines as voltages, Of course, you can measure it with a multimeter. And What ever you give to Parallel port as voltages can be read from these registers. For example , if we write '1' to Data register , the line Data0 will be driven to +5v. Just like this ,we can programmatically turn on and off any of the data lines and Control lines.



2.10.5 Interfacing System using Parallel Port

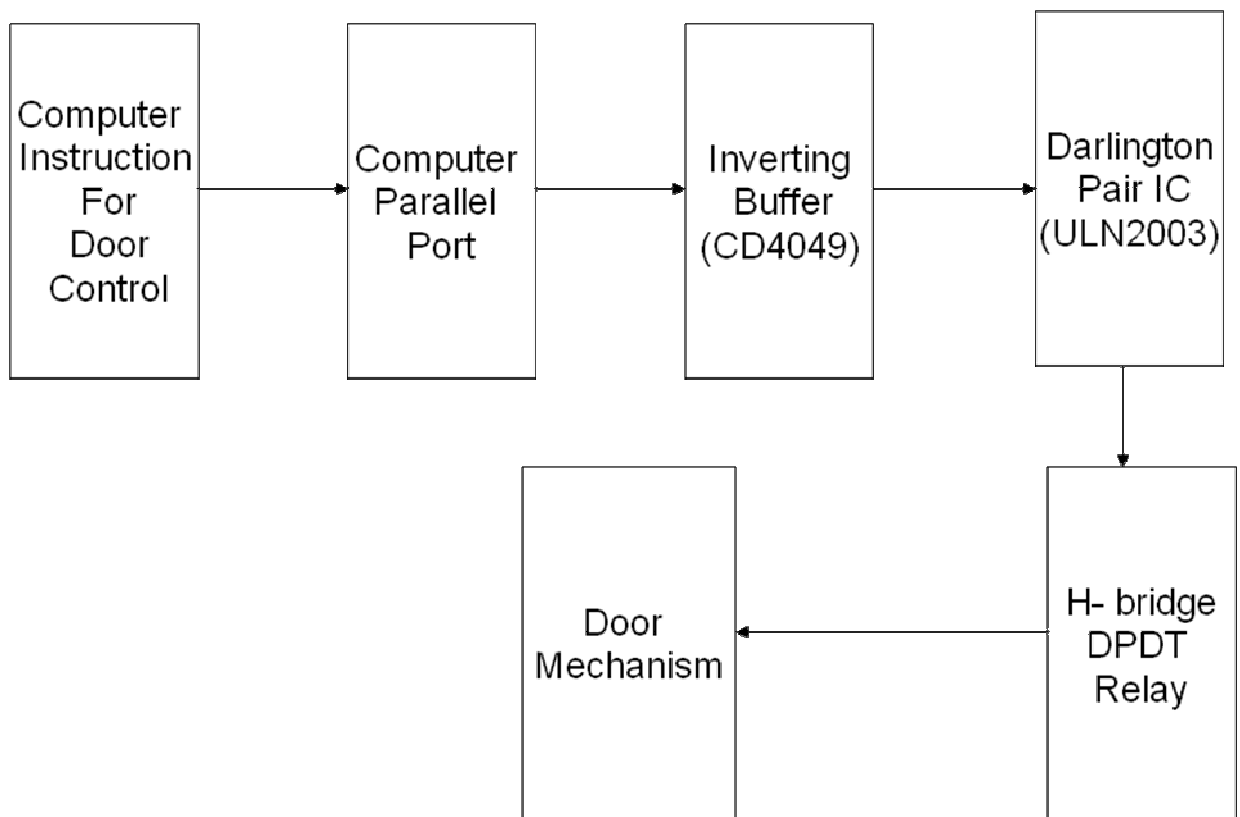


Fig:- Block Diagram to Control Door

Working Principle:-

After the number plate is processed and the number plate with its respective date and time is saved, the command is send to parallel port by the MATLAB software. The instruction for either opening or closing door is given and the parallel port output pin (data pins from 2-9) is set to high. The circuitry for controlling the door consists of Inverting Buffer (CD4049) IC which isolates the circuit from outer interface and inverts the output. Since the current provided by PC's parallel port is too low for driving the relays, we need a darlington pair IC (ULN2003) to amplify the current. H-bridge of DPDT relay is made to control the direction of single motor of door in both forward and backward direction. The door mechanism opens or closes the door according to the motion of motor.

2.11 Components Used

System we will go over the operation of some of the devices used in it.

i. Diode



Diode is a two terminal semiconductor device made of two sections called n and p side. A terminal is connected to each of these two sides. If a positive voltage of more than a few tenths of a volt is applied to the p side with respect to the n side, the diode conducts current very easily. This is called forward-bias. On the other hand, if the p side is negative with respect to the n side, the diode conducts very little current and is called reverse-biased.

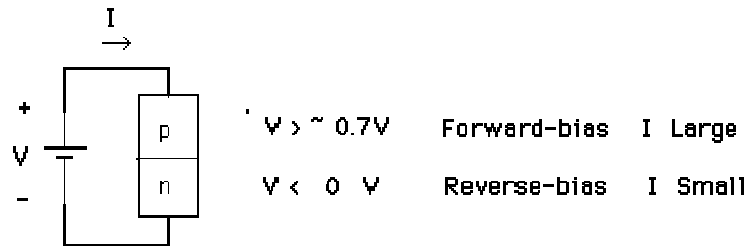


Figure 1

ii. Light Emitting Diode



Certain diodes, under the condition of forward-bias, emit light. These could be of different colors. In our case it is infrared.

iii. Transistor

A transistor is a three-terminal device that is made of three sections called **emitter**, **base** and **collector**. A transistor is either npn or pnp.

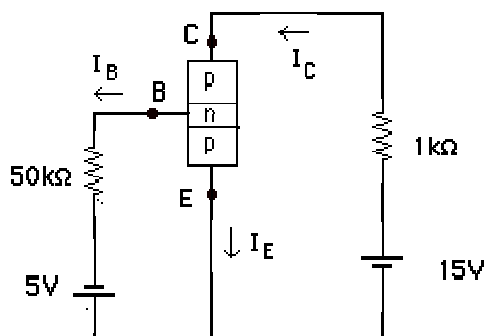
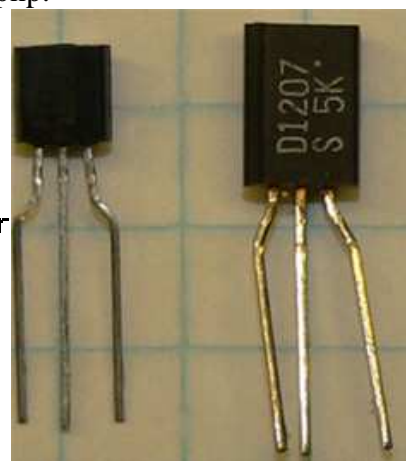


Figure 2

C: Collector
E: Emitter
B: Base



A small base current, I_B , is used to control a much larger collector current I_C . A small signal (small variations) superimposed on the I_B results in similar but larger variations in I_C . This is called amplification.

iv. Phototransistor

A phototransistor is a transistor that is sensitive to light. When light shines on it, the collector current will greatly increase, while in the absence of light, I_c is very small. We can view the effect of light as controlling the effective resistance between the collector and emitter. In the presence of light this resistance is low, resulting in a large collector current, while in the absence of light the opposite happens. The phototransistor is functioning as a switch; no connection is made to the base.

v. Potentiometer

This device is essentially a variable resistor. It has three terminals. Figure 3 shows the example of a $10k\Omega$ potentiometer. The contact C can be moved along the $10k\Omega$ resistor from A to B. The resistance between A and B is fixed ($10k\Omega$ in this case). The resistance between A and C can be changed from 0 to $10k\Omega$ by moving the pointer from A to B. As this happens, the resistance between C and B changes from $10k\Omega$ to 0.

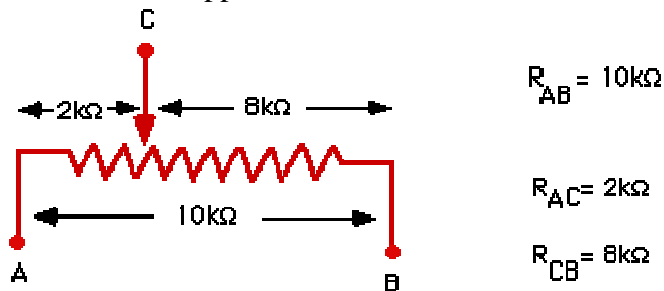
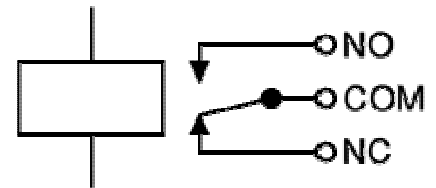


Figure 3

vi. Relays

A relay is an **electrically operated switch**. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are **double throw (changeover)** switches.



Circuit symbol for a relay

vii. DC motor:-

After relay is operated then the supply is fed to the dc motor which makes the dc motor to operate and the motor as it is connected to the wheel through the shaft rotates to move from one place to another either front, back, left or right. Here all the control mechanism is controlled through the computer.

viii. Capacitors:-

A capacitor is a physical device which is capable of storing energy by virtue of a voltage existing across it. The voltage applied across the capacitor sets up an electric field within it and the energy is stored in the electric field. A capacitor is basically meant to store electrons, and release them whenever required. Capacitance is a measure of capacitor's ability to store charge and is measured in Farads (F). Farad, the unit of capacitance is very large, so microfarad or nanofarad is usually used.



ix. Resistors

Resistors are components that have a predetermined resistance. Resistance determines how much current will flow through a component. Resistors are used to control voltages and currents. A very high resistance allows very little current to flow. The unit of resistor is ohm

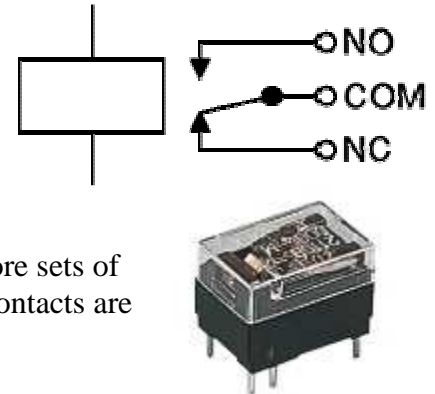
Circuit symbol: 

x. Relays

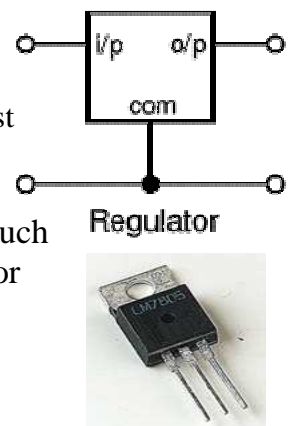
A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches.

Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit. There is no electrical connection inside the relay between the two circuits; the link is magnetic and mechanical.

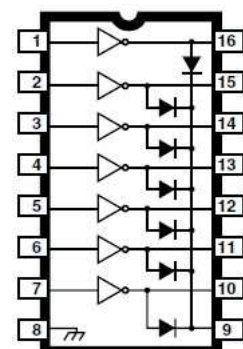
Relays are usually SPDT or DPDT but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available

**xi. Regulating ICs**

Regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection'). Many of the fixed voltage regulator ICs has 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. They include a hole for attaching a heatsink if necessary

**xii. Darlington pair IC (ULN2003):**

Ideally suited for interfacing between low-level logic circuitry and multiple peripheral power loads, the Series ULN20xxA/L high-voltage, high-current Darlington arrays feature continuous load current ratings to 500 mA for each of the seven drivers. At an appropriate duty cycle depending on ambient temperature and number of drivers turned ON simultaneously, typical power loads totaling over 230 W (350 mA x 7, 95 V) can be controlled. Typical loads include relays, solenoids, stepping motors, magnetic print hammers, multiplexed LED and incandescent displays, and heaters. All devices feature open-collector outputs with integral clamp



diodes.

xiii. Non-inverting Buffer (CD4049):

The CD4049UBC and CD4050BC hex buffers are monolithic complementary MOS (CMOS) integrated circuits constructed with N- and P-channel enhancement mode transistors. These devices feature logic level conversion using only one supply voltage (VDD). The input signal high level (VIH) can exceed the VDD supply voltage when these devices are used for logic level conversions. These devices are intended for use as hex buffers, CMOS to DTL/ TTL converters, or as CMOS current drivers, and at VDD = 5.0V, they can drive directly two DTL/TTL loads over the full operating temperature range.

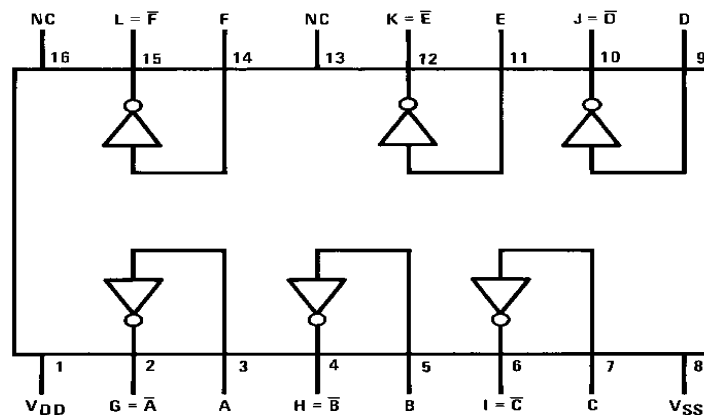


Fig. Pin configuration of CD4049

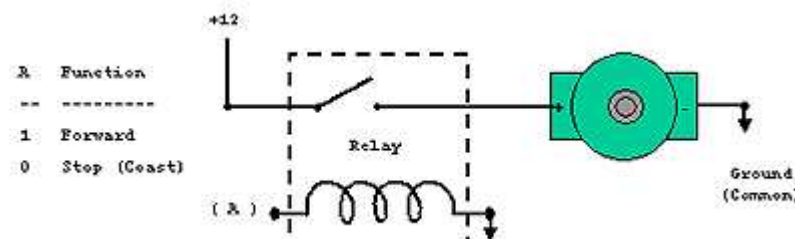
xiv. Brief H-Bridge Theory of Operation

How do we make a motor turn?

We take a battery; hook the positive side to one side of we DC motor. Then we connect the negative side of the battery to the other motor lead. The motor spins forward. If we swap the battery leads the motor spins in reverse.

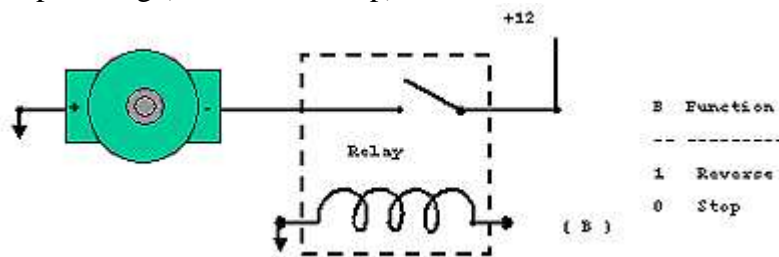
Ok, that's basic. Now lets say we want a Micro Controller Unit (MCU) to control the motor, how would we do it? Well, for starters we get a device that would act like a solid state switch, a transistor, and hook it up the motor.

NOTE: If we connect up these relay circuits, remember to put a diode across the coil of the relay. This will keep the spike voltage (back EMF), coming out of the coil of the relay, from getting into the MCU and damaging it. The anode, which is the arrow side of the diode, should connect to ground. The bar, which is the Cathode side of the diode, should connect to the coil where the MCU connects to the relay.



If we connect this circuit to a small hobby motor we can control the motor with a processor (MCU, etc.) Applying a logical one, (+12 Volts in our example) to point A

causes the motor to turn forward. Applying a logical zero, (ground) causes the motor to stop turning (to coast and stop).



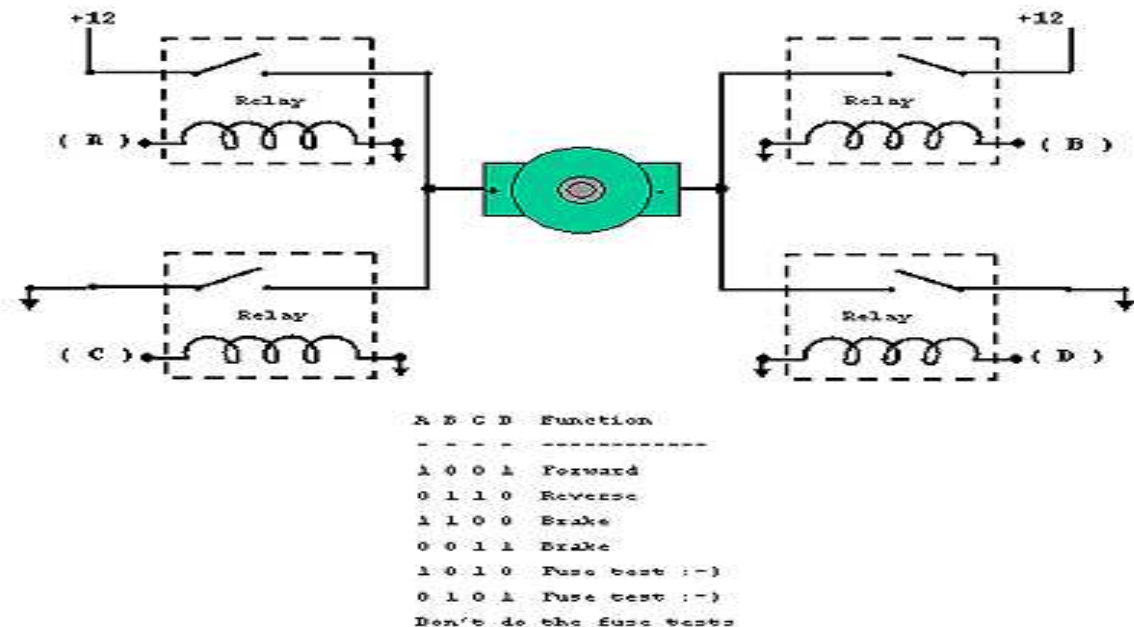
Hook the motor up in this fashion and the circuit turns the motor in reverse when we apply a logical one (+12Volts) to point B. Apply a logical zero, which is usually a ground, causes the motor to stop spinning.

If we hook up these circuits we can only get the motor to stop or turn in one direction, forward for the first circuit or reverse for the second circuit.

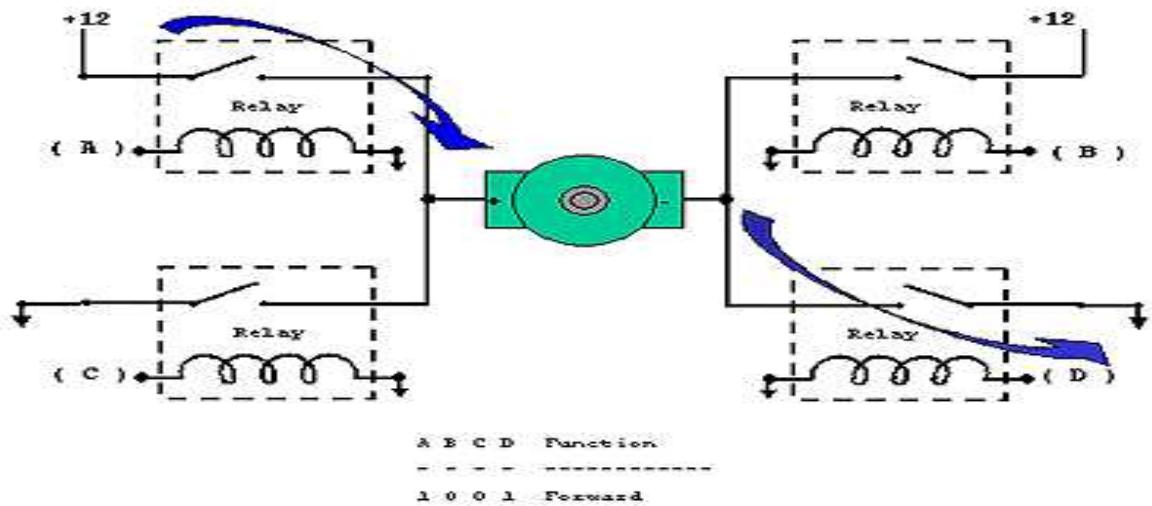
Motor Speed

We can also pulse the motor control line, (A or B) on and off. This powers the motor in short burst and gets varying degrees of torque, which usually translates into variable motor speed.

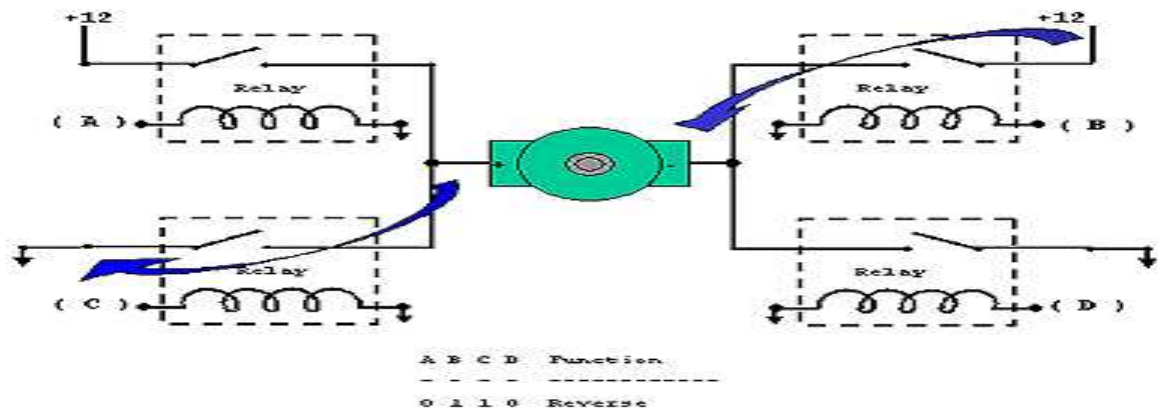
But if we want to be able to control the motor in both forward and reverse with a processor, we will need more circuitry. We will need an H-Bridge. Notice the "H"-looking configuration in the next graphic. Relays configured in this fashion make an H-Bridge. The "high side drivers" are the relays that control the positive voltage to the motor. This is called sourcing current. The "low side drivers" are the relays that control the negative voltage to sink current to the motor. "Sinking current" is the term for connecting the circuit to the negative side of the power supply, which is usually ground.



So, we turn on the upper left and lower right circuits, and power flows through the motor forward, i.e.: 1 to A, 0 to B, 0 to C, and 1 to D.



Then for reverse we turn on the upper right and lower left circuits and power flows through the motor in reverse, i.e.: 0 to A, 1 to B, 1 to C, and 0 to D.



CAUTION: We should be careful not to turn on both circuits on one side or the other, or we have a direct short which will destroy we circuit; Example: A and C or B and D both high (logical 1).

PROBLEM DOMAIN

- Weather condition
- Lighting condition
- License plate placement in the picture
- Vehicle movement
- Mechanical plate damages

REQUIREMENTS

Software Requirements:-

1. Windows Operating System
2. MATLAB R2008a
3. Webcam Driver
4. IspPgm (Burner Software)
5. Assembler

Hardware Requirements

1. Pentium 4 computer with at least 512MB RAM and 40 GB hard disk recommended 1GB RAM.
2. Robot (As a Vehicle with number plate)
3. Webcam
4. Door
5. 8051 s-series Burner

A list of hardware components for the project:-

S.No.	Component Name	No. of Components
1.	Web Cam	1
2.	Parallel Port Connector	1
3.	8051 Series Burner	1
4.	AT89s52Microcontroller	1
5.	40 pin IC holder	1
6.	White LEDs	3
7.	LDR	3
8.	Voltage Comparator IC(LM324)	1
9.	Resistors, Variable Resistors	As needed
10.	Crystal Oscillator	1
11.	Inverting Buffer(4049)	2
12.	Darlington IC	1
13.	Relays	4
14.	PCB	1
15.	Capacitors	3
16.	Connectors, jumper wires	As needed
17.	Regulator ICs	2
18.	16 pin IC holder	3
19.	Push on and push to on Switch	2
20.	LCD display	1
21.	Solder Iron	1
22.	Soldering wires	
23.	DC Power Source	
24.	DC motors	3
25.	Mechanical components	Help from Mechanical

PROJECT SCHEDULING

PERT CHART

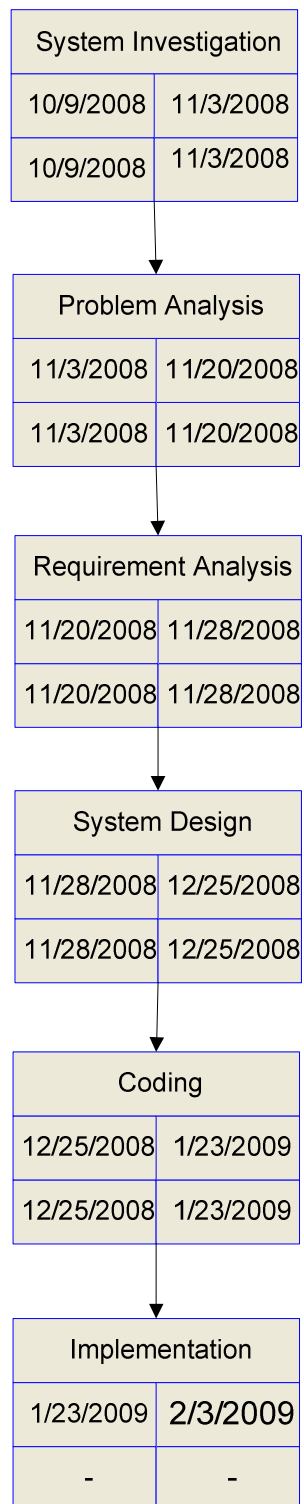
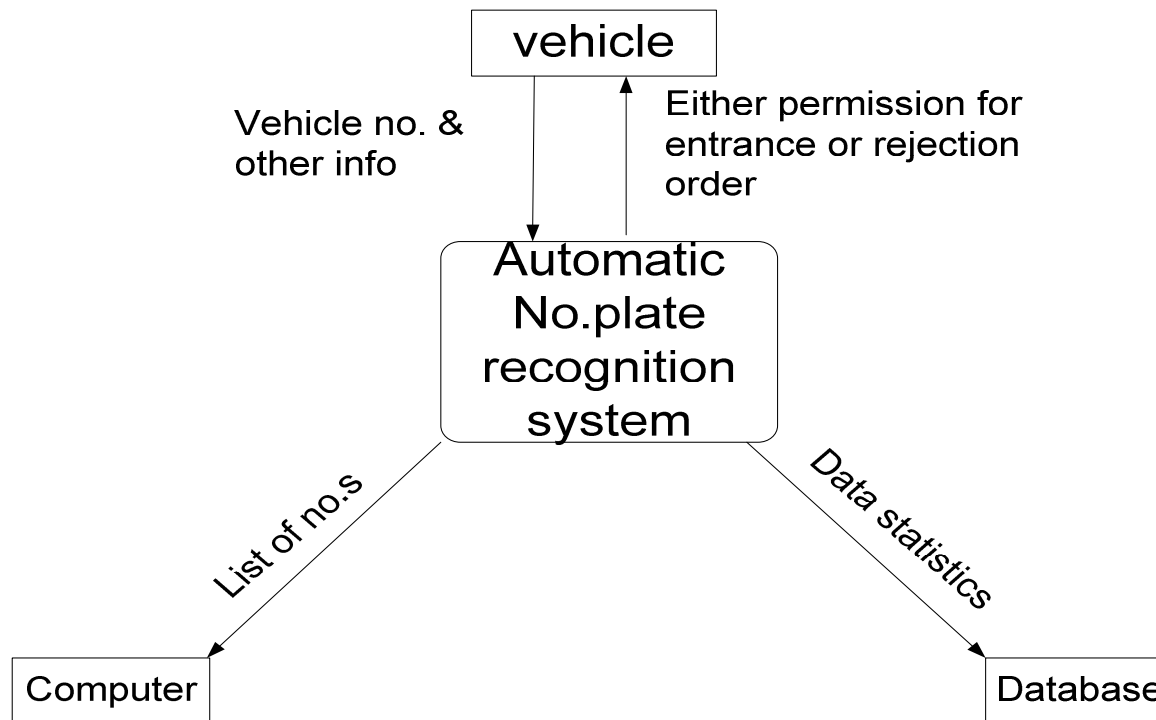


Fig. Pert Chat for the Automatic number plate recognition using automatic line tracking robot

DESIGN

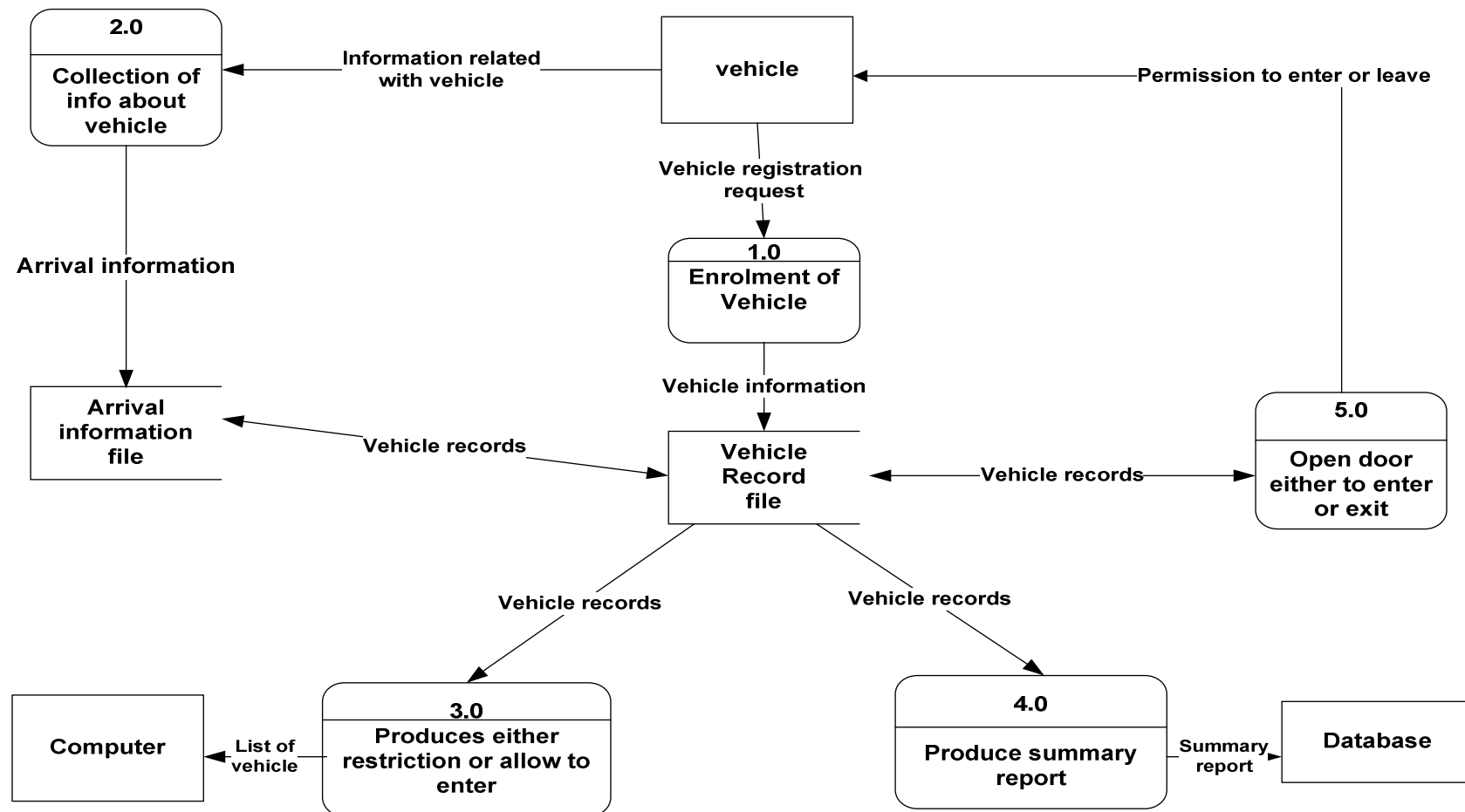
1 . Data Flow Diagrams(DFD)

Level 0 DFD of ANPR



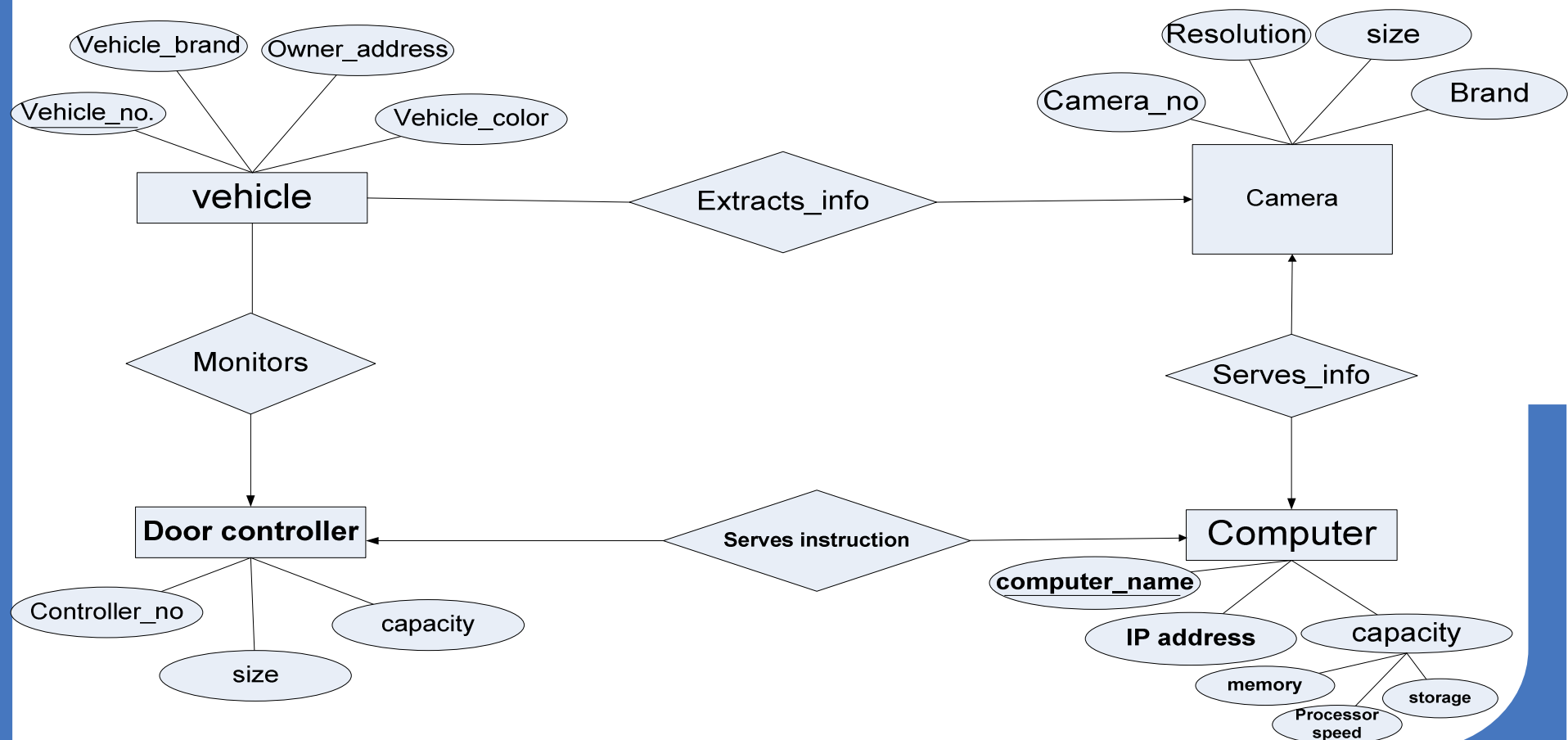
b. Level – 1 DFD

Level 1 DFD of ANPR



2.Entity Relationship Diagram

Entity Relationship Diagram



COST ESTIMATION

- Effort Cost= Rs.150000
- Software requirements=Rs.2500
- Hardware Requirements=Rs.6000
- Internet Surfing= Rs.2000
- Reference materials bought=Rs.5000
- Transportation Cost=Rs.1000
- Communication Cost=Rs.1000

Total cost = Rs. 1,67,500/-

LIMITATIONS

- Recognizes only specified font (Times New Roman) number plate only.
- Can't clearly recognize or distinguish character like B and 8, 2 and Z , 0 and O etc.
- The distance of image captured is also limited.
- Lighting Conditions greatly effect the image processing.

CONCLUSION

The project “ANPR based security system using ALR” was developed as III year II part project of Diploma In Computer Engineering, Thapathali Campus. The project was designed keeping in mind the automation of the number plate detection system for security reason that could replace the current system of manual entry.

This project was a success in recording the number plate of a vehicle although it has got it's own limitation of image processing and other hardware requirements. From this project we learn about image processing and OCR(Optical Character Reader),hardware interface and automatic line tracking robot(ALR). The automatic line tracking robot was developed for the demonstration of actual system and to have the knowledge of robots. The mechanical parts of the vehicle were designed by Robotics and Automation Club, Thapathali Campus. The basic focus of the project was in the image processing using MATLAB. The MATLAB v7.6 was used as programming tool which provides many features related to image processing. The parallel port interfacing was done to control the door mechanism for giving entry to the vehicle. This project implements above processes to accomplish a task of “ANPR based Security System with ALR”. Practical implementation of the project range from small use such as keeping record of incoming and outgoing vehicles from the parking area to a vast implementation such as security system. The project was completed within 6 months time period.

REFERENCES

1. D.G. Bailey, D. Irecki, B.K. Lim and L. Yang “Test bed for number plate recognition applications”, Proceedings of First IEEE International Workshop on Electronic design, Test and Applications (DELTA’02), IEEE Computer Society, 2002.
2. R.A. Lotufo, A.D. Morgan, and AS. Johnson, 1990, “Automatic Number-Plate Recognition,” Proceedings of the IEE Colloquium on Image analysis for Transport Applications, V01.035, pp.6/1-6/6, February 16, 1990.
3. A.S. Johnson, B.M. Bird, 1990, “Number-plateMatching for Automatic Vehicle Identification,” IEE Colloquium on Electronic Image and Image Processing in Security and Forensic, Aprl, 1990.
4. M.M.M. Fahmy, 1994, “Automatic Number-plate Recognition : Neural Network Approach,” Proceedings of VNIS’94 Vehicle Navigation and Information System Conference, 3 1 Aug-2 Sept, 1994
5. J.A.G. Nijhuis, M.H. Ter Brugge, K.A. Helmholt, J.P.W. Pluim, L. Spaanenburg, R.S. Venema, M.A. Westenberg, 1995, “Car License Plate
6. Automatic Vehicle Identification by Plate Recognition
Author - Serkan Ozbay, and Ergun Ercelebi
7. ALGORITHMIC AND MATHEMATICAL PRINCIPLES OF AUTOMATIC
NUMBER PLATE RECOGNITION SYSTEMS.
Author - ONDREJ MRTINSKY
8. <http://www.matpic.com>