MPMCI LAB MINI PROJECT

on

Implementation of **RFID based car parking system** by using

8051 Microcontroller

A Mini Project Report

By

R.Saveri

(1602-19-733-102)

P.Vaishnavi

(1602-19-733-114)

Under the guidance of

Smt. B. Syamala

ASSISTANT PROFESSOR

Department of Computer Science & Engineering

Vasavi College of Engineering.

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad.2

CONTENTS **PAGE NO.**

ACKNOWLEDGEMENT

I would like to express our heartfelt gratitude to Smt. B. Syamala, our project guide, for her valuable guidance and constant support, along with her capable instruction and persistent encouragement.

I am grateful to our Head of Department, Dr. T. Adilakshmi, for her steady support and the provision of every resource required for the completion of this project.

I would like to take this opportunity to thank our Principal, Dr. S. V. RAMANA, as well as the management of the institute, for having designed an excellent learning atmosphere.4

LIST OF FIGURES:

PROJECT ABSTRACT

- RFID based Car Parking System is a simple project that offers an efficient car parking management system with the help of RFID Technology.
- Car parking management in organisations and malls often consists of many tasks like issuing tokens, noting the check-in and check-out time, calculating fare and finally collecting the amount.
- As the number of vehicles are increasing, the problems faced by manual parking management system are also increasing.
- Such problems can be eliminated to some extent by implementing an intelligent parking system where the entry and exit of cars is monitored and payment is made easy with sensor technology.
- This project deals with an interesting manner of security access based car parking system using AT89C51 microcontroller and RFID Technology.
- For this project we are going to use Keil MicroVision for software development environment and Proteus for simulation.

INTRODUCTION TO 8051 MICROCONTROLLER

8051 microcontroller is designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 11.0592 MHz. In case the data is larger than 8 bits then it has to be broken into parts so that the CPU can process conveniently. Most manufacturers have put 4Kbytes of ROM even though the quantity of ROM can be exceeded up to 64 K bytes. The 8051 has been in use in a wide number of devices, mainly because it is easy to integrate into a project or build a device around.

Since the basic layout of a microcontroller includes a CPU, ROM, RAM, etc. the 8051 microcontroller also has a similar layout. The following image shows a brief layout of a typical 8051 Microcontroller. It is a CISC based Microcontroller with Harvard Architecture (separate program and data memory).

8051 Internal Architecture

- 8 – bit CPU with two Registers A (Accumulator) and B.
- Internal ROM of 8K Bytes – It is a flash memory that supports in – system programming.
- Internal RAM of 256 Bytes – The first 128 Bytes of the RAM i.e. 00H to 7FH is again divided in to 4 banks with 8 registers (R0 – R7) in each bank, 16 bit addressable registers and 80 general purpose registers. The higher 128 Bytes of the RAM i.e. 80H to FFH consists of SFRs or Special Function Registers. Using SFRs we can control different peripherals like Timers, Serial Port, all I/O Ports, etc.
- 32 I/O Pins (Input / Output Pins) – Arranged as 4 Ports: P0, P1, P2 and P3.
- 8- bit Stack Pointer (SP) and Processor Status Word (PSW).
- 16 – bit Program Counter (PC) and Data Pointer (DPTR).
- Two 16 – bit Timers / Counters – T0 and T1.
- Control Registers – SCON, PCON, TCON, TMOD, IP and IE.
- Serial Data Transmitter and Receiver for Full – Duplex Operation – SBUF.
- Interrupts: Two External and Three Internal.
- Oscillator and Clock Circuit.7

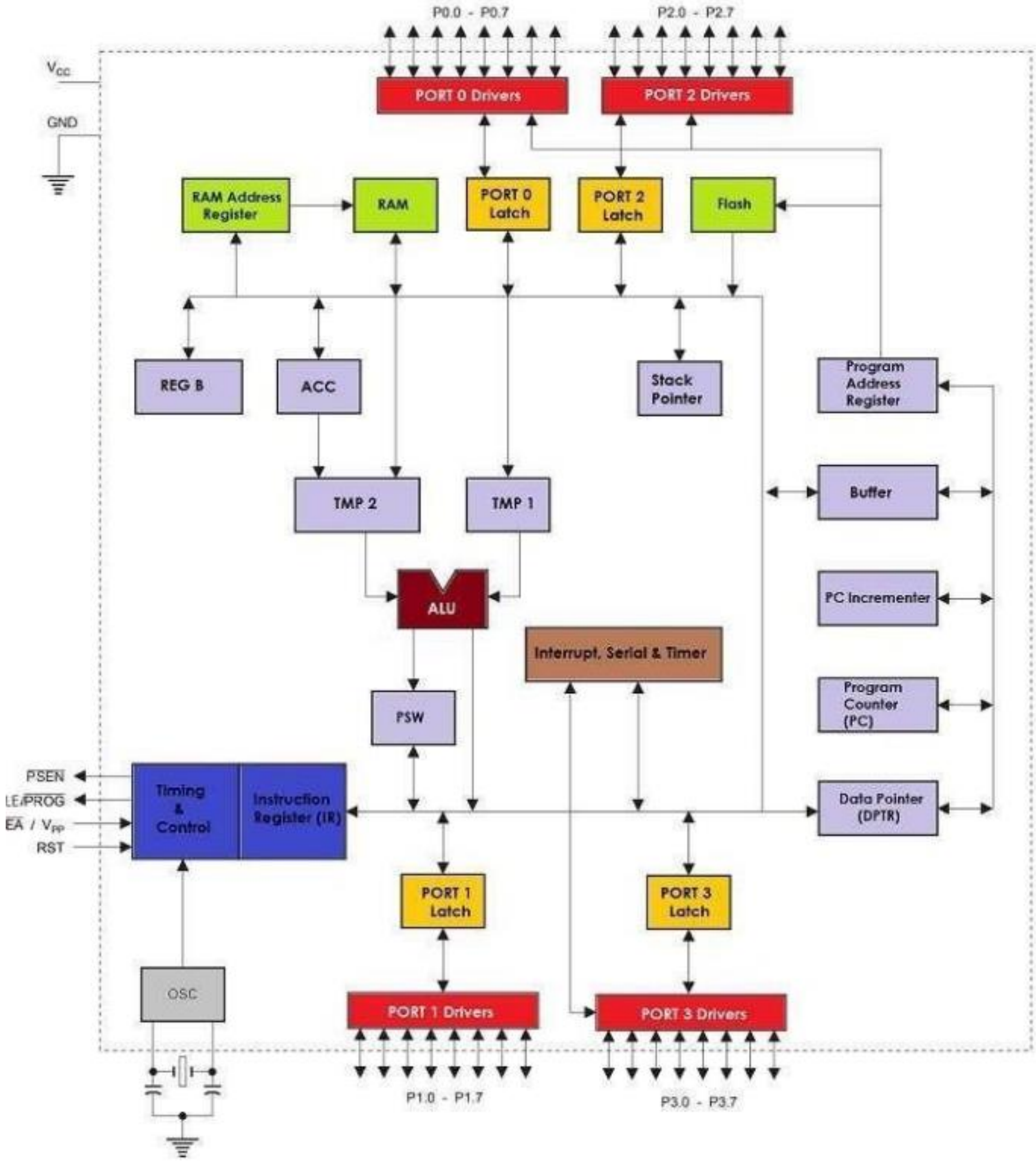


Figure 1: Layout of 8051 Microcontroller8

SOFTWARE REQUIREMENT

Keil MicroVision5

For this project, we have used Keil MicroVision5 as the software development environment. Keil MDK is the complete software development environment for a wide range of Arm Cortex-M based microcontroller devices. MDK includes the µVision IDE and debugger, Arm C/C++ compiler, and essential middleware components.

The µVision IDE combines project management, run-time environment, build facilities, source code editing, and program debugging in a single powerful environment. µVision is easy-to-use and accelerates your embedded software development. µVision supports multiple screens and allows you to create individual window layouts anywhere on the visual surface.

The Debugger provides a single environment in which you may test, verify, and optimize your application code. The debugger includes traditional features like simple and complex breakpoints, watch windows, and execution control and provides full visibility to device peripherals.

Proteus 8

And for simulation, we have used Proteus 8 Professional (Schematic Capture). It allows us to develop hardware implementation of the project by providing a wide range of components to build circuits in workspace and thereby allows us to attach “hex” code to the simulation and run the simulation at real-time.

Proteus 8 Professional is a software which can be used to draw schematics, PCB layout, code and even simulate the schematic. It is developed by Labcenter Electronic Ltd.

Features of Proteus:

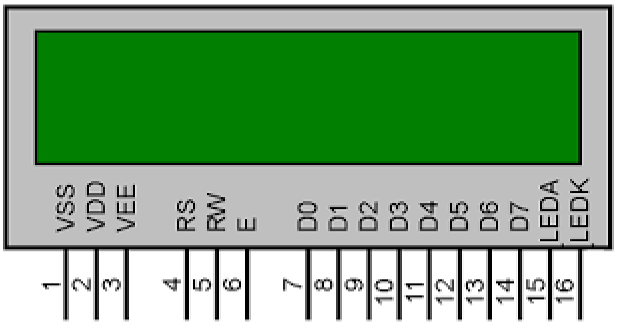
- Schematic Capture – One can easily draw the schematic using proteus. We can select the devices by clicking ‘Pick Devices’ button and select the desired component.
- Simulation – Many components like microcontrollers can be simulated in proteus.
- Designing PCB
- 3D Visualization9

About LCD

- Liquid Crystal Display is very commonly used electronic display module and having a wide range of applications such as calculators, laptops, mobile phones etc.
- This is a 16-pin device, constituting 8 data pins and 3 command pins. It has two registers, namely the command register and data register, used to store commands and data respectively. They can be selected using the command pin RS. When RS = 0 the command register is selected and when RS = 1 the data register is selected. With respect to the data pins, when the LCD is used in 8-bit mode, all 8 data inputs are used.

LCD Interfacing:

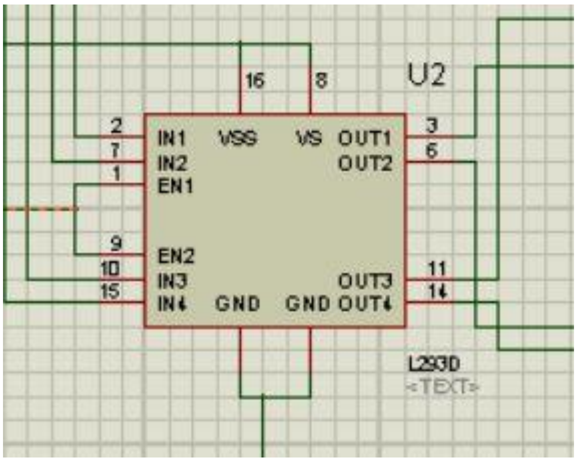
- The circuit diagram given above shows how to interface a 16×2 LCD module with AT89S1 microcontroller.
- P1.0 to P1.7 pins of the microcontroller is connected to the DB0 to DB7 pins of the module respectively and through this route the data goes to the LCD module.
- P3.3, P3.4 and P3.5 are connected to the E, R/W, RS pins of the microcontroller andthrough this route the control signals are transferred to the LCD module.10



About L239D

- L239D is a typical motor driver or motor driver IC which allows DC motor to drive on either direction.

- L239D is a 14 pin IC which can control a set of 2 DC motors simultaneously in any direction. It means that you can control 2 DC motors with a single L293D IC.



RFID AND IR SENSORS

- There will be an RFID Tag attached to every vehicle. As soon as the vehicle passes through the entry gate the RFID Reader reads the tag and gets the Unique ID and then logs an entry into the database and upon exit another RFID Reader reads the tag and deducts money from the users.

- An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment.

SIMULATION USING PROTEUS 8 :

The microcontroller used for this simulation is AT89C51 (ATMEL) microcontroller.

COMPONENTS:

AT89C51 Microcontroller - 8051 microcontroller

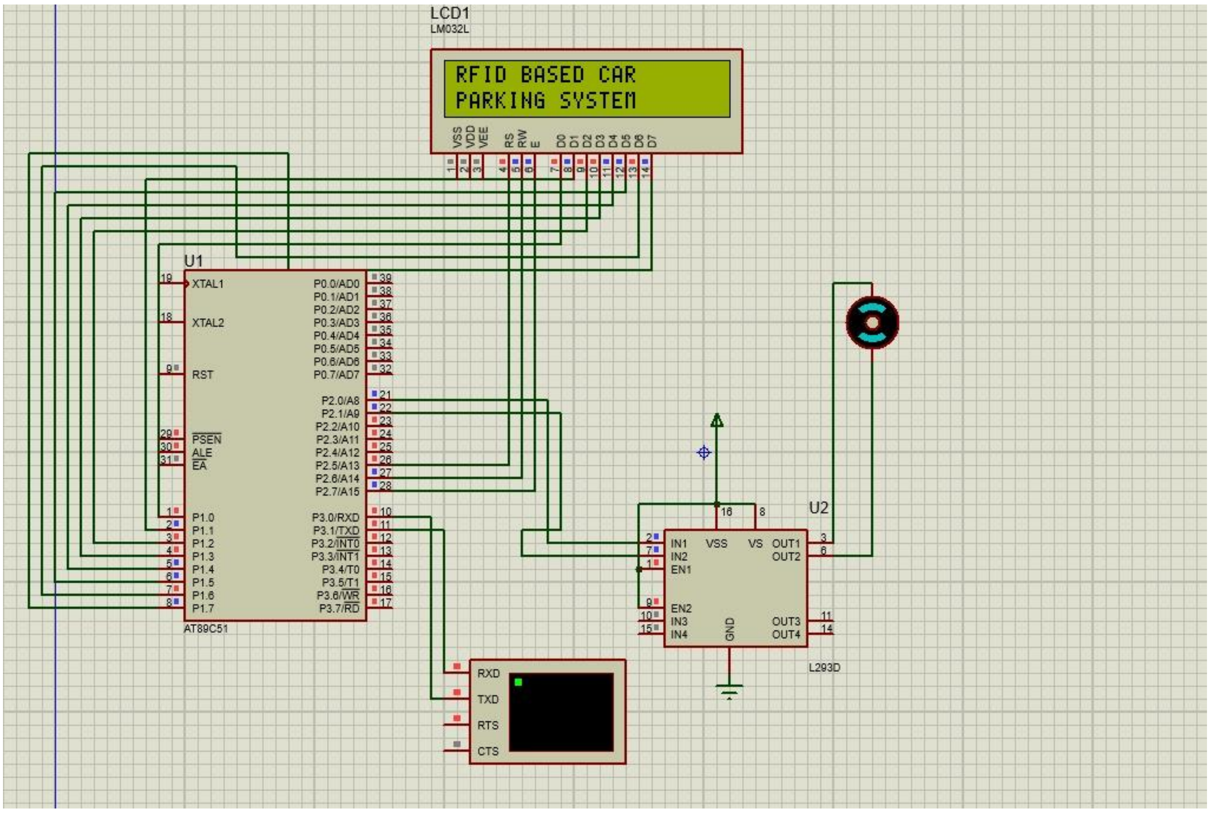
2) LCD 16 X 2 - PC.0 to PC.7 connected to d0 -d7 of LCD

3) L239D

4) DC Motor

5) Virtual terminal

CIRCUIT DIAGRAM:



SAMPLE CODE

```
#include<reg51.h>
sbit doorp=P2^0;
sbit doorn=P2^1;
sbit rs=P2^5;
sbit rw=P2^6;
sbit en=P2^7;
void lcddat(unsigned char);
void lcdcmd(unsigned char);
void lcddis(unsigned char *);
void lcd_init();
void serial_init();
void check();
void delay();
void mdelay();
unsigned char rfid[12],v1;
void main()
{
doorp=doorn=0;
serial_init();
lcd_init();
lcddis("RFID BASED CAR");
lcdcmd(0xc0);
lcddis("PARKING SYSTEM");
mdelay();
lcdcmd(0x01);
while(1)
        {
        lcdcmd(0x01);
        lcddis("SWIPE YOUR CARD");
        for(v1=0;v1<12;v1++)
        {
        while(RI==0);
        rfid[v1]=SBUF;
        RI=0;
        SBUF=rfid[v1];
        while(TI==0);
        TI=0;
        }
        check();
        }
}
void check(){
//1A3465B8935- SHARMA SLOT,3245FR6786A- MURTHY SLOT
//234R56Y7129-KHAN SLOT ,145678HA341- REDDY SLOT
if(rfid[0]=='1'
&&rfid[1]=='A'&&rfid[2]=='3'&&rfid[3]=='4'&&rfid[4]=='6'&&rfid[5]=='5'&&rfid[6]=='B'
&&rfid[7]=='8'&&rfid[8]=='9'&&rfid[9]=='3'&&rfid[10]=='5'){
lcdcmd(0x01);
lcddis("ASOCIATION MEMBER");
lcdcmd(0xc0);
lcddis("SHARMA-SLOT 101");
delay();
delay();
lcdcmd(0x01);
doorp=1;doorn=0;
lcddis("DOOR OPENING");
lcdcmd(0x01);
lcddis("ALLLOW INSIDE");
mdelay();
doorp=0;doorn=0;
```

```
delay();
lcdcmd(0x01);
doorp=0;doorn=1;
lcddis("DOOR CLOSING");
mdelay();
doorp=0;
doorn=0;
 }
else
if(rfid[0]=='3'&&rfid[1]=='2'&&rfid[2]=='4'&&rfid[3]=='5'&&rfid[4]=='F'&&rfid[5]=='R'&
&rfid[6]=='6'&&rfid[7]=='7'&&rfid[8]=='8'&&rfid[9]=='6'&&rfid[10]=='A')
{
lcdcmd(0x01);
lcddis("ASOCIATION MEMBER");
lcdcmd(0xc0);
lcddis("MURTHY-SLOT 105");
delay();
delay();
lcdcmd(0x01);
doorp=1;doorn=0;
lcddis("DOOR OPENING");
lcdcmd(0x01);
lcddis("ALLLOW INSIDE");
mdelay();
doorp=0;doorn=0;
delay();
lcdcmd(0x01);
doorp=0;doorn=1;
lcddis("DOOR CLOSING");
mdelay();
doorp=0;doorn=0;
```

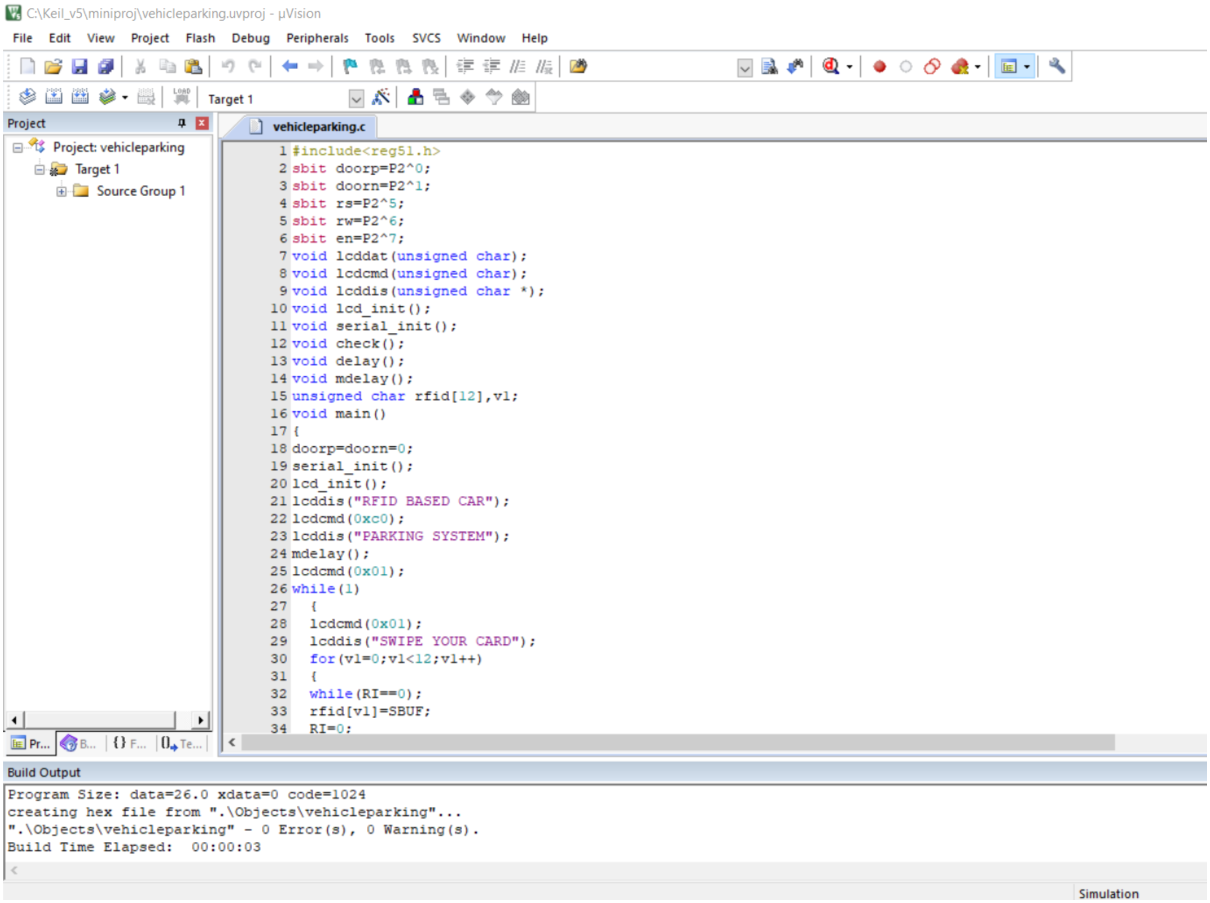
```
 }

 else if(rfid[0]=='2'&&rfid[1]=='3'&&rfid[2]=='4'&&rfid[3]=='R'&&rfid[4]=='5'
&&rfid[5]=='6'&&rfid[6]=='Y'&&rfid[7]=='7'&&rfid[8]=='1'&&rfid[9]=='2'&&rfid[10]=='9
')
{
lcdcmd(0x01);
lcddis("ASOCIATION MEMBER");
lcdcmd(0xc0);
lcddis("KHAN -SLOT 405");
delay();
delay();
lcdcmd(0x01);
doorp=1;doorn=0;
lcddis("DOOR OPENING");
lcdcmd(0x01);
lcddis("ALLLOW INSIDE");
mdelay();
doorp=0;doorn=0;
delay();
lcdcmd(0x01);
doorp=0;doorn=1;
lcddis("DOOR CLOSING");
mdelay();
doorp=0;doorn=0;
 }
else if(rfid[0]=='1'&&rfid[1]=='4'&&rfid[2]=='5'&&rfid[3]=='6'&&rfid[4]=='7'
&&rfid[5]=='8'&&rfid[6]=='H'&&rfid[7]=='A'&&rfid[8]=='3'&&rfid[9]=='4'&&rfid[10]=='
E')
{
lcdcmd(0x01);
```

```
lcddis("ASOCIATION MEMBER");
lcdcmd(0xc0);
lcddis("REDDY -SLOT 103");
delay();
delay();
lcdcmd(0x01);
doorp=1;doorn=0;
lcddis("DOOR OPENING");
lcdcmd(0x01);
lcddis("ALLLOW INSIDE");
mdelay();
doorp=0;doorn=0;
delay();
lcdcmd(0x01);
doorp=0;doorn=1;
lcddis("DOOR CLOSING");
mdelay();
doorp=0;doorn=0;
 }
else
{
lcdcmd(0x01);
lcddis("OUTSIDE PERSON");
lcdcmd(0xc0);
lcddis("NO SLOT FOR YOU");
mdelay();
}
}
void lcd_init()
{
```

```
lcdcmd(0x38);
lcdcmd(0x01);
lcdcmd(0x10);
lcdcmd(0x0c);
lcdcmd(0x80);
}
void lcdcmd(unsigned char val)
{
P1=val;
rs=0;
rw=0;
en=1;
delay();
en=0;
}
void lcddat(unsigned char val)
{
P1=val;
rs=1;
rw=0;
en=1;
delay();
en=0;
}
void delay()
{
unsigned int v5;
for(v5=0;v5<6000;v5++);
}
void lcddis(unsigned char *s)
```

```
{
unsigned char w;
for(w=0;s[w]!='\0';w++)
{
lcddat(s[w]);
}
}
void serial_init()
{
SCON=0X50;
TMOD=0x20;
TH1=-3;
TR1=1;
}
void mdelay()
{
unsigned int v6,v7;
for(v6=0;v6<2;v6++)
{
for(v7=0;v7<60000;v7++);
}
}
```

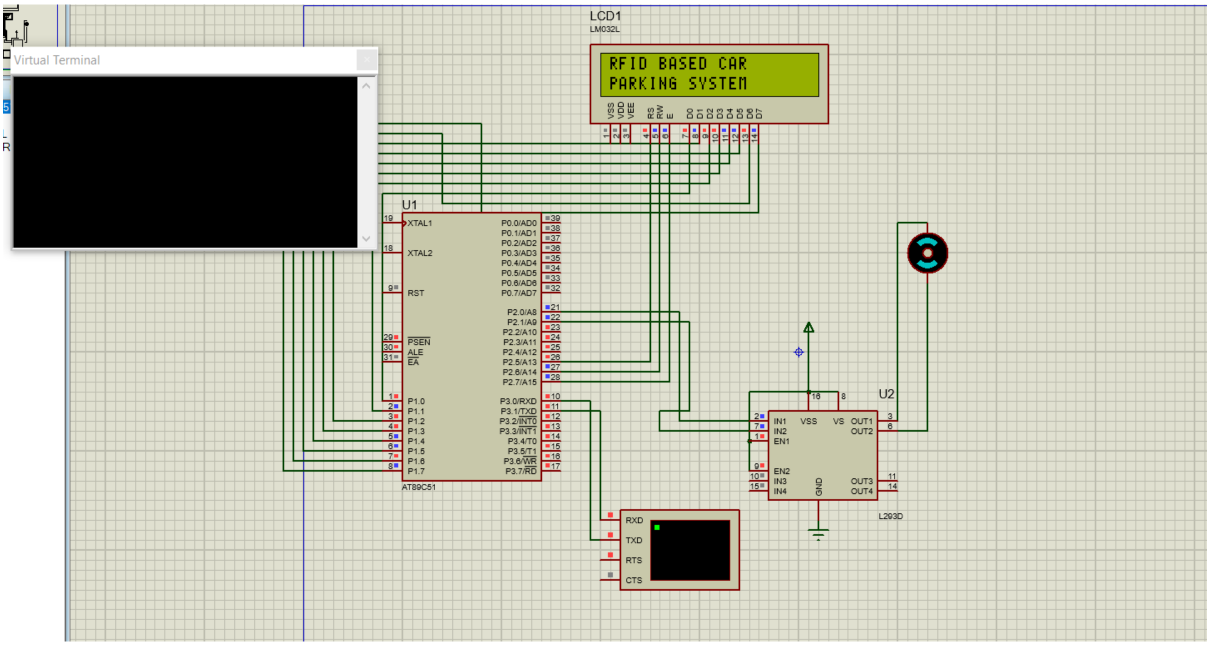
Creation of HEX file:



```c
#include<reg51.h>
sbit doorp=P2^0;
sbit doorn=P2^1;
sbit rs=P2^5;
sbit rw=P2^6;
sbit en=P2^7;
void lcddat(unsigned char);
void lcdcmd(unsigned char);
void lcddis(unsigned char *);
void lcd_init();
void serial_init();
void check();
void delay();
void mdelay();
unsigned char rfid[12],vl;
void main()
{
doorp=doorn=0;
serial_init();
lcd_init();
lcddis("RFID BASED CAR");
lcdcmd(0xc0);
lcddis("PARKING SYSTEM");
mdelay();
lcdcmd(0x01);
while(1)
  {
  lcdcmd(0x01);
  lcddis("SWIPE YOUR CARD");
  for(vl=0;vl<12;vl++)
  {
  while(RI==0);
  rfid[vl]=SBUF;
  RI=0;
```

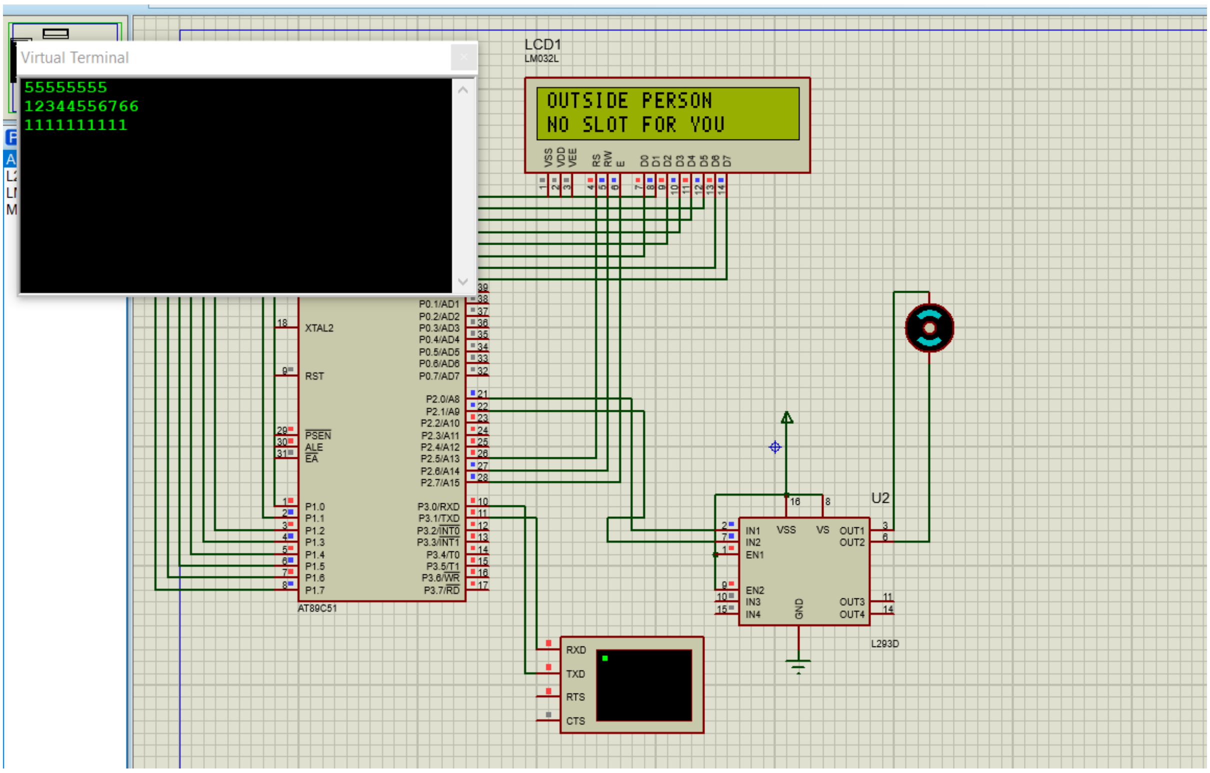
```
Program Size: data=26.0 xdata=0 code=1024
creating hex file from ".\Objects\vehicleparking"...
".\Objects\vehicleparking" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:03
```

OUTPUTS SCREENSHOTS



RFID BASED CAR
PARKING SYSTEM

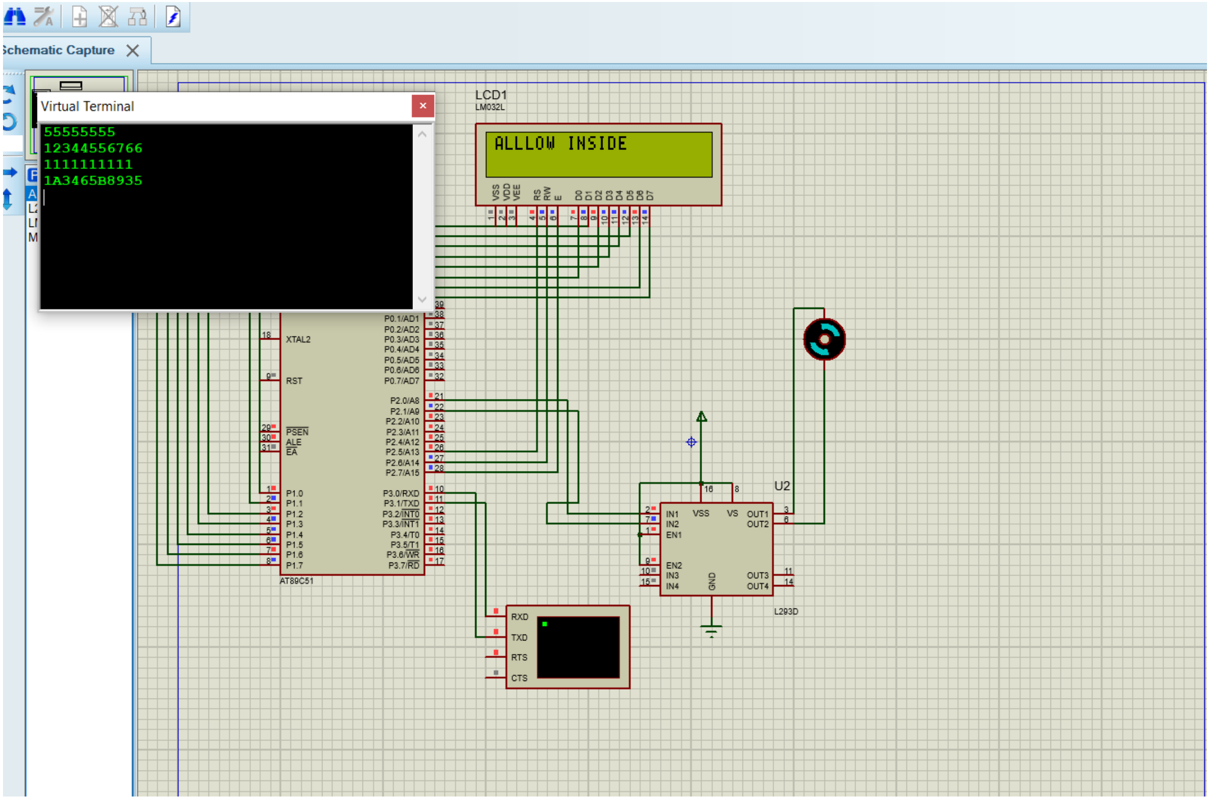
Display



```
55555555
12344556766
1111111111
```

OUTSIDE PERSON
NO SLOT FOR YOU

Entering number which is not association member’s



Virtual Terminal

```
55555555
12344556766
1111111111
1A3465B8935
```

```
ALLLOW INSIDE
```

Schematic Capture
LCD1
LM032L
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
XTAL2
RST
PSEN
ALE
EA
P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6 P1.7
P0.1/AD1 P0.2/AD2 P0.3/AD3 P0.4/AD4 P0.5/AD5 P0.6/AD6 P0.7/AD7
P2.0/A8 P2.1/A9 P2.2/A10 P2.3/A11 P2.4/A12 P2.5/A13 P2.6/A14 P2.7/A15
P3.0/RXD P3.1/TXD P3.2/INT0 P3.3/INT1 P3.4/T0 P3.5/T1 P3.6/WR P3.7/RD
AT89C51
U2
IN1 IN2 EN1 EN2 IN3 IN4
VSS VS GND
OUT1 OUT2 OUT3 OUT4
L293D
RXD
TXD
RTS
CTS

Entering number of association member

CONCLUSION

Now days in many public places such as malls, multiplex systems, hospitals, offices, market areas there is a crucial problem of car parking. The car-parking area has many lanes/slots for car parking. So to park a car one has to look for all the lanes. Moreover, this involves a lot of manual labour and investment. So, there is a need to develop an automated parking system that indicates directly the availability of vacant parking slots in any lane right at the entrance. It involves a system including infrared transmitter- receiver pain in each lane and a display outside the car parking gate. So the person desirous to park his vehicle is well informed about the status of availability of parking slot.

REFERENCES

- http://www.keil.com/ - KEIL MICROVISION
- https://electronicswork.wordpress.com/2012/10/19/microcontroller-8051- piano-in-keil-and-proteus-simulator/
- https://www.youtube.com/watch?v=oWr37z8LIrg
- https://www.electronicshub.org/8051-microcontroller-introduction/
- https://www.youtube.com/watch?v=UfTTas8HjVE
- https://www.labcenter.com/ - PROTEUS2