**I-Toll System**

Index

1. Abstract

2. Introduction

3. Features

4. Block Diagram & Explanation

5. Flow of working

6. Hardware and software requirements

7. Circuit Diagram

8. Component details

9. Advantages

10. Conclusion

11. References

**Abstract**

Toll Collection in India is a matter of concern as the toll checkpoints are hotbeds of corruption. The purpose of this project is to propose a technique where collection of tolls can be done electronically using image processing technique where we can detect the vehicle and thereby deduct the toll amount. Using this technique we can make an attempt to eradicate corruption from toll checkpoints in India. The current scenario is that the vehicle needs to slow down at the toll collection area in order to pay the amount which leads to traffic problem and sometimes the toll collector doesn’t collect the toll fee from the owner. So the proposed technique will maintain all the records of the toll collected as there is no physical cash involved in this technique

**Introduction**

In this project we examine the image and the respective information will be processing based toll collection system and how to make more efficient and perfect. On any toll booth the vehicle has to stop for paying the toll. We are trying to develop a system that would pay the toll automatically and reduce the queue at the toll booth. In this system camera is used for capturing the image of the vehicle number plate. According to the vehicle the toll would be cut from the customer’s account and then open the gate. Moreover in our system if a vehicle is stolen and an entry is being made in the central database by the police then if the vehicle passes through the toll booth then alarm would buzz which would indicate the operator at the toll booth that the vehicle is a stolen vehicle. For the identification of the vehicles, the information of the vehicles is already stored on the central database. So captured number will be sent to the toll.

**Features**

- Recognition of Number Plate and Type of Vehicle using Image Processing.

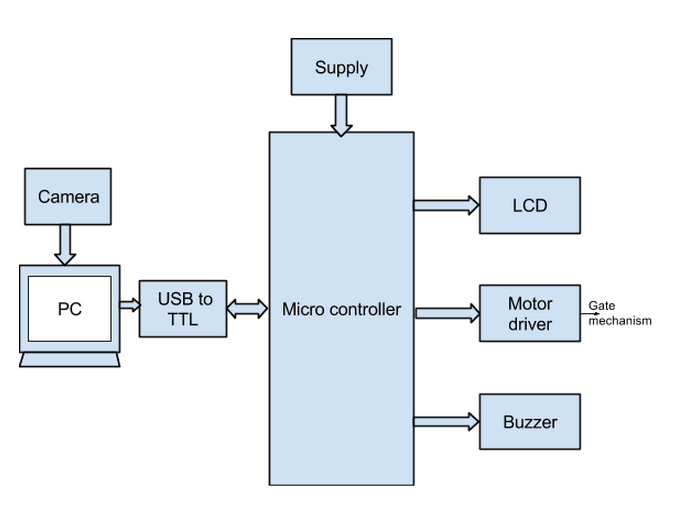
- Based on Feature 1, Stolen Vehicles would be recognized comparing with the entries in the database.

- gate mechanism

- Toll System – The Amount for the vehicle would be determined based on its type and will be displayed on the LCD with number plate or the

determined amount would be deducted from the database entry of that vehicle

**Block Diagram & Explanation**



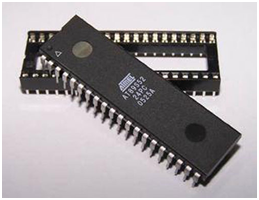
**PC:**

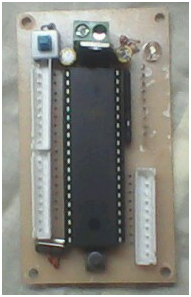
Image processor in our project is a personal computer and image analysis tool is MatLab. So all the features and flow are been incorporated on PC with help of MatLab by using various algorithms. We are also using PC screen as output display as to minimise to hardware complexity at the booth and to save data. Thus an laptop architecture is used where webcam on top of screen is used for project demonstration.

**Microcontroller:**

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K 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 pin out. 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 AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 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 con-tents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.





**FEATURES:**

● Compatible with MCS-51™ Products

● 4K Bytes of In-System Reprogrammable Flash Memory Endurance: 1,000 Write/Erase Cycles

● Fully Static Operation: 0 Hz to 24 MHz

● Three-level Program Memory Lock

● 128 x 8-bit Internal RAM

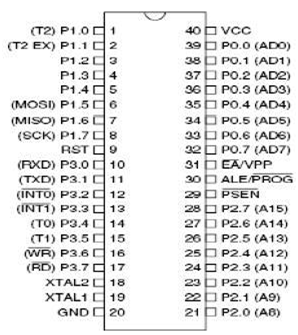
● 32 Programmable I/O Lines

● Two 16-bit Timer/Counters

● Six Interrupt Sources

● Programmable Serial Channel

**PIN DESCRIPTION:**

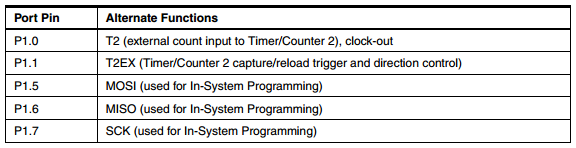


· **VCC**: Supply voltage.

· **GND:**  Ground.

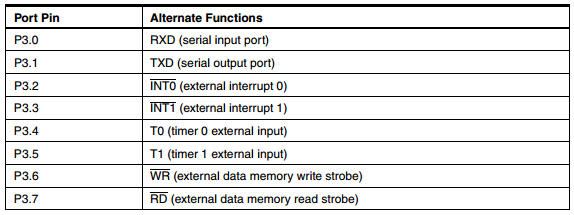
· **Port 0:** Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses 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.

· **Port 1:** Port 1 is an 8-bit bidirectional 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. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table. Port 1 also receives the low-order address bytes during Flash programming and verification.



· **Port 2:** Port 2 is an 8-bit bidirectional 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.

· **Port 3:** Port 3 is an 8-bit bidirectional 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 AT89S52, as shown in the following table.



· **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 DISRTO 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.

· **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.

· **PSEN (Program Store Enable):** PSEN is the read strobe to external program memory. When the AT89S52 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.

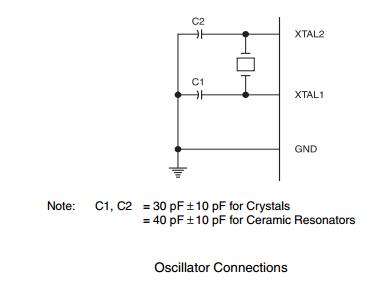
· **EA/VPP External Access Enable**: EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal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

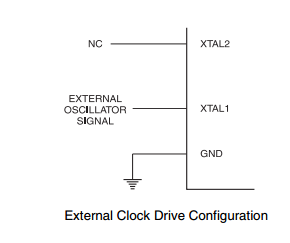
· **XTAL1**: Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

· **XTAL2**: Output from the inverting oscillator amplifier.

**Oscillator Characteristics**:

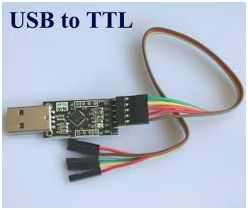
The XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure below. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

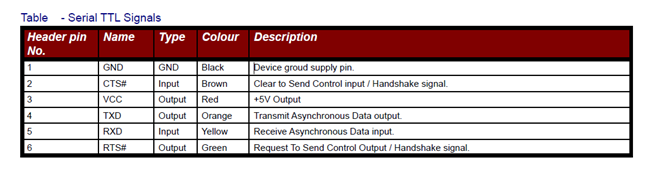




**USB to TTL:**

The cable is easiest way ever to connect to your microcontroller/Raspberry Pi/WiFi router serial console port. Inside the big USB plug is a USB<->Serial conversion chip and at the end of the 36" cable are four wire - red power, black ground, white RX into USB port, and green TX out of the USB port. The power pin provides the 5V @ 500mA direct from the USB port and the RX/TX pins are 3.3V level for interfacing with the most common 3.3V logic level chipsets.



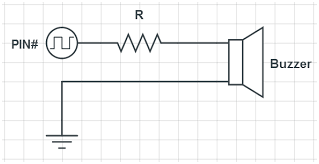


**Buzzer:**

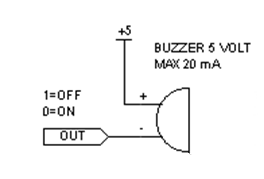
The buzzer produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers can be used alert a user of an event corresponding to a switching action, counter signal or sensor input. They are also used in alarm circuits.



The buzzer produces a same noisy sound irrespective of the voltage variation applied to it. It consists of piezo crystals between two conductors. When a potential is applied across these crystals, they push on one conductor and pull on the other. This, push and pull action, results in a sound wave. Most buzzers produce sound in the range of 2 to 4 kHz.



One lead is connected to the Input and the other lead is connected to Ground.

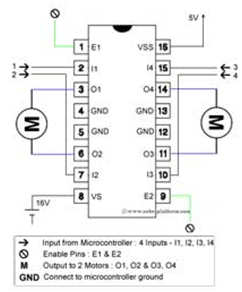


**Motor Driver:**

Since motors require more current then the microcontroller pin can typically generate, you need some type of a switch (Transistors, MOSFET, Relay etc.,) which can accept a small current, amplify it and generate a larger current, which further drives a motor. This entire process is done by what is known as a motor driver.

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC, Dual H-bridge Motor Driver integrated circuit (IC). The l293d can drive small and quiet big motors as well.





The circuit shown above is the most basic implementation of L293D IC. There are 16 pins sticking out of this IC and we have to understand the functionality of each pin before implementing this in a circuit

1. Pin1 and Pin9 are "Enable" pins. They should be connected to +5V for the drivers to function. If they pulled low (GND), then the outputs will be turned off regardless of the input states, stopping the motors. If you have two spare pins in your microcontroller, connect these pins to the microcontroller, or just connect them to regulated positive 5 Volts.

2. Pin4, Pin5, Pin12 and Pin13 are ground pins which should ideally be connected to microcontroller's ground.

3. Pin2, Pin7, Pin10 and Pin15 are logic input pins. These are control pins which should be connected to microcontroller pins. Pin2 and Pin7 control the first motor (left); Pin10 and Pin15 control the second motor(right).

4. Pin3, Pin6, Pin11, and Pin14 are output pins. Tie Pin3 and Pin6 to the first motor, Pin11 and Pin14 to second motor

5. Pin16 powers the IC and it should be connected to regulated +5Volts

6. Pin8 powers the two motors and should be connected to positive lead of a secondary battery. As per the datasheet, supply voltage can be as high as 36 Volts.

LOGIC TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin 1** | **Pin 2** | **Pin 7** | **Function** |
| **High** | **High** | **Low** | **Turn Anti-clockwise (Reverse)** |
| **High** | **Low** | **High** | **Turn clockwise (Forward)** |
| **High** | **High** | **High** | **Stop** |
| **High** | **Low** | **Low** | **Stop** |
| **Low** | **X** | **X** | **Stop** |

High ~+5V, Low ~0V, X=Either high or low (don't care)

**LCD(Liquid Crystal Display):**

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments](http://www.engineersgarage.com/content/seven-segment-display) and other multi segment [LED](http://www.engineersgarage.com/content/led)s. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even [custom characters](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](http://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so on.



**Flow of working**

* Camera is used to capture vehicle size and also number plate of that vehicle. In our project we are using two different types of the vehicle.
* Camera will capture the number plate of the vehicle, It will send to the PC.
* MATLAB software is used for the image processing. MATLAB code is used to recognized the number plate and also check that vehicle is stolen or not.
* We will create a database of the stolen vehicle. If current captured number plate matched with stolen number plate database so it will conclude that vehicle is stolen and gate mechanism will never be open in that case and buzzer will turn on.
* To control the gate mechanism we are using the DC motor which is connected to the Uc through motor driver IC.
* If vehicle is not stolen type then camera will capture the vehicle image. The size of the vehicle will be decided by the MATLAB code.
* Toll charges depends on the size of the vehicle that we will mentioned in MATLAB code, according to that PC will send signal to the Uc using USB to TTL.
* LCD is used to display the toll charges which is depend on the size of the vehicle and also determined amount would be deducted from the database entry of that vehicle

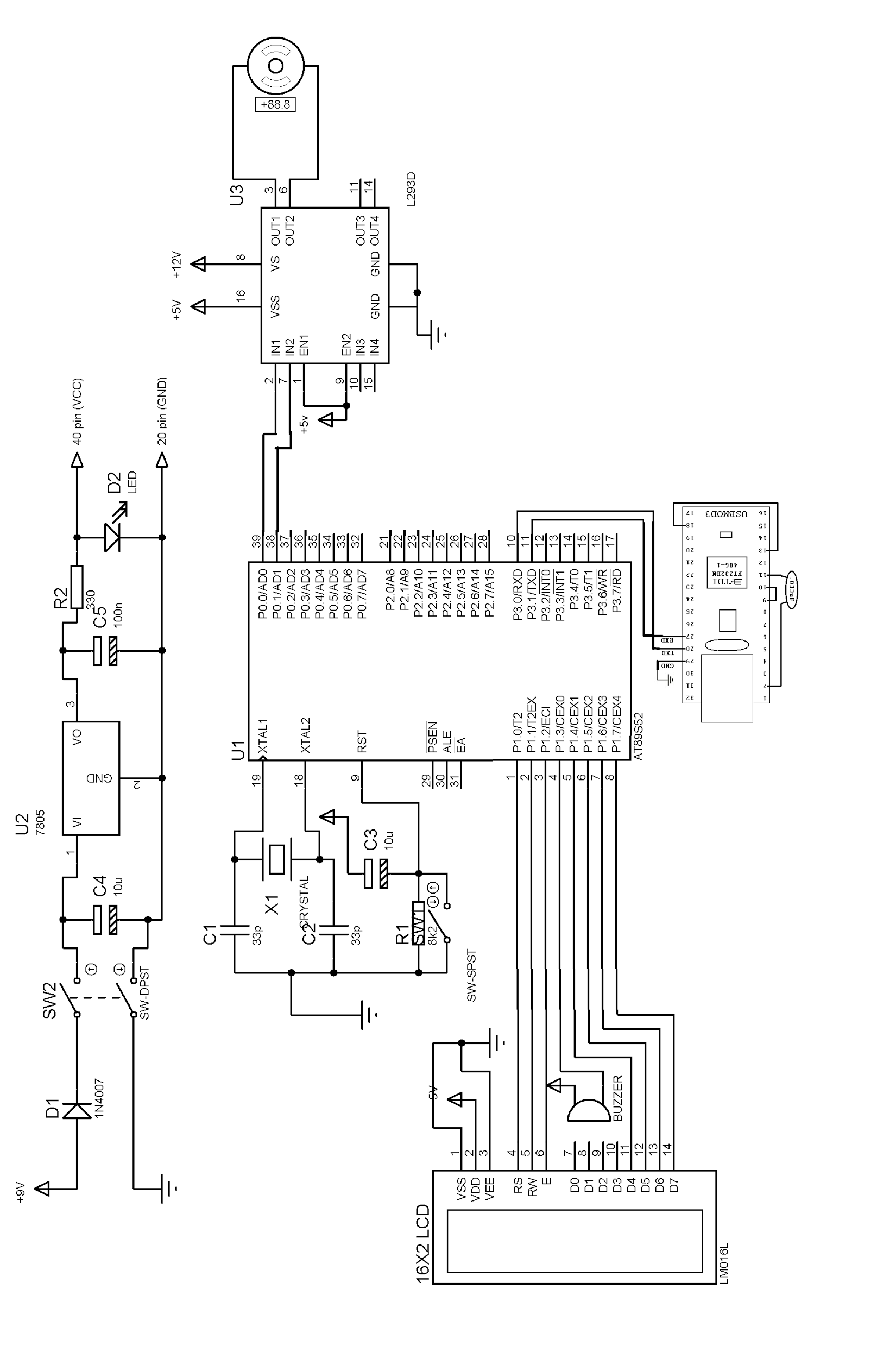
**Hardware requirements:**

* Microcontroller AT89S52
* LCD 16\*2
* Buzzer
* Motor Driver L293D
* DC Motor for gate mechanism

**Software requirements:**

* Keil for assembly language coding in microcontroller
* ucflash for program burning in IC
* Diptrace to make the PCB (printed circuit board) Layout
* Visual Basic for database and GUI

**Circuit Diagram**



**Component details**

Refer the attached files

**Advantages**

* Reduces Traffic near the toll checkpoint areas.
* Reduces the overhead of collecting physical cash from commuters.
* Reduces fraudulent behaviour at toll checkpoints.
* Implementation cost is less.

**Conclusion**

In this project, we have discussed the image processing technique to implement the automated toll collection in order to reduce congestion and fraudulent behaviour at the toll checkpoints. Further we have discussed the working of the automated deduction of the toll from the vehicle owner's bank account. The proposed system will help in reduce the human intervention at the toll collection areas. The purpose of this project was to improvise the toll collection as well as implement a system for people in order to avoid physical cash.

**Reference**

[1] IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 12, NO. 3, SEPTEMBER 2011 An Algorithm for License Plate Recognition Applied to Intelligent Transportation System. Ying Wen, Yue Lu, Member, IEEE, Jingqi Yan, Zhenyu Zhou, Karen M. von Deneen, and Pengfei Shi, Senior Member, IEEE

[2] International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 3, March 2014 Study of Different Electronic toll Collection Systems and Proposed toll Snapping and Processing System. Apurva Hemant Kulkarni M.E Department of Computer Science & Engineering, G.H.R.I.E.M jalgaon North Maharashtra university,INDIA

[3] Prathamesh Kulkarni (Student Member, IEEE), Ashish Khatri, Prateek Banga, Kushal Shah, “Automatic Number Plate Recognition (ANPR) System for Indian conditions”, IEEE Transactions, 2009.

[[4] International Journal of Scientific Engineering Research Volume 3, Issue 8, August -2012 1. Still Image Recognition Of License Plate System. N.Kanagaraj1, G.Baskaran2,S.Saravanan3,A.Ramachandran4 1,2,3,4 Assistant professor, Srinivasan Engineering College, Perambalur,Tamilnadu, India](http://hello-engineers.blogspot.in/2009/10/scada-technology-paper-presentation.html)