SMART CARD BASED TOLL GATE AUTOMATED SYSTEM

V.SRIDHAR ¹ M.NAGENDRA ²

¹ Assistant Professor, ECE, Vidya Jyothi Institute of Technology, Hyderabad
² HOD&Associate professor, ECE, Dr.k.v subbareddy institute of technology, Kurnool

varadalasri@gmail.com, nag20209@gmail.com

ABSTRACT: Smart cards are secure tokens that can provide security services to a wide range of applications. Along with other technology advances, smart card technology has changed dramatically as well. However, its communication standards, largely unchanged, do not match with those of mainstream computing, which has limited its success in the Internet age. In our daily life we are seeing toll gate. We are going to pay certain amount to the government in form of tax through this toll gate. We can see this toll gates being placed in some national high ways etc., So in order to pay tax we are normally going to pay in form of cash, but instead of that as the technology is growing we can make use of smart card which is nothing but like a memory card in which we are going to store the details of particular person and certain amount. The main objective of this smart card is to pay the toll gate tax using smart card. Smart card must be recharged with some amount and whenever a person wants to pay the toll gate tax, he needs to insert his smart card and deduct amount using keypad. By using this kind of smart card there is no need to carry the amount in form of cash and so we can have security as well.

Keywords: SMART CARD, KEIL SOFTWARE, MICRO CONTROLLER, EMBEDDED SYSTEMS

I . INTRODUCTION

An embedded system is a special-purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs one or a few predefined tasks, usually with very specific requirements. Since the system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded systems are often mass-produced, benefiting from economies of scale.

Personal digital assistants (PDAs) or handheld computers are generally considered embedded devices because of the nature of their hardware design, even though they are more expandable in software terms. This line of definition continues to blur as devices expand. With the introduction of the OQO Model 2 with the Windows XP operating system and ports such as a USB port — both features usually belong to “general purpose computers”, — the line of nomenclature blurs even more.

Physically, embedded systems ranges from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants.

In terms of complexity embedded systems can range from very simple with a single microcontroller chip, to very complex with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

Fig: 1.1 complex embedded system unit

2 Block diagram:

Fig:2 embedded system unit block diagram
2. Block diagram description:

2.1 Transmitter: transmitter parts consist of 5 parts. They are

- Power supply
- Microcontroller
- Smart card & Smart card reader
- MAX232
- Keypad
- LCD

2.1.1 Power supply:
In this system, we are using 5V power supply for the microcontroller of the Transmitter section as well as the Receiver section. We use rectifiers for converting the A.C. into D.C. and a step-down transformer to step-down the voltage. The full description of the Power supply section is given in this documentation in the following sections i.e., hardware components.

2.1.2 Microcontroller (8051):
In this paper, the microcontroller is playing a major role. Microcontrollers were originally used as components in complicated process-control systems. However, because of their small size and low price, Microcontrollers are now also being used in regulators for individual control loops. In several areas, Microcontrollers are now outperforming their analog counterparts and are cheaper as well.

The purpose of this paper work is to present control theory that is relevant to the analysis and design of Microcontroller systems with an emphasis on basic concepts and ideas. It is assumed that a Microcontroller with reasonable software is available for computations and simulations so that many tedious details can be left to the Microcontroller. The control system design is also carried out up to the stage of implementation in the form of controller programs in assembly language OR in C-Language.

Intel has introduced a family of Microcontrollers called the MCS-51.

2.1.3 MAX232:
The data which we are entering into the hyper terminal editor is available at the COM1 port. Then the data enters into the MAX232 voltage converter via the RS232 cable. The MAX232 converts the voltage levels of the RS232 to the TTL level and then sends to the UART of the microcontroller. So the main duty of the MAX232 is for voltage conversions.

Serial RS-232 (V.24) communication works with voltages (between -15V ... -3V and +3V ... +15V) which are not compatible with today’s computer logic voltages. On the other hand, classic TTL computer logic operates between 0V ... +5V (roughly 0V ... +0.8V referred to as low for binary '0', +2V ... +5V for high binary '1'). Modern low-power logic operates in the range of 0V ... +3.3V or even lower.

So, the maximum RS-232 signal levels are far too high for today's computer logic electronics, and the negative RS-232 voltage can't be grokked at all by the computer logic. Therefore, to receive serial data from an RS-232 interface, the voltage has to be reduced, and the 0 and 1 voltage levels inverted. In the other direction (sending data from some logic over RS-232) the low logic voltage has to be "bumped up", and a negative voltage has to be generated, too.

Table 1:

<table>
<thead>
<tr>
<th>RS-232</th>
<th>TTL</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15V ... -3V</td>
<td>+2V ... +5V</td>
<td>1</td>
</tr>
<tr>
<td>+3V ... +15V</td>
<td>0V ... +0.8V</td>
<td>0</td>
</tr>
</tbody>
</table>
2.1.4 Smart Card Reader Section: This section consists of a Smart card reader. The Smart card reader is a device which reads the smart card being inserted into it. The data stored on the smart card which resembles the information of a person is read by the smart card reader. The card reader communicates with microcontroller using serial communication. The card reader is interfaced to microcontroller using MAX 232, a serial driver.

Smart Card Readers are also known as Card Programmers (because they can write to a card), card terminals, card acceptance device (CAD) or an interface device (IFD). When the smart card and the card reader come into contact, each identifies itself to the other by sending and receiving information. If the messages exchanged do not match, no further processing takes place.

2.1.4(a): Smart Card Reader Working

Smart Card Readers are also known as card programmers (because they can write to a card), card terminals, card acceptance device (CAD) or an interface device (IFD). There is a slight difference between the card reader and the terminal. The term 'reader' is generally used to describe a unit that interfaces with a PC for the majority of its processing requirements. In contrast, a 'terminal' is a self-contained processing device.

The reader provides a path for your application to send and receive commands from the card. There are many types of readers available, such as serial, PC Card, and standard keyboard models. Unfortunately, the ISO group was unable to provide a standard for communicating with the readers so there is no one-size-fits-all approach to smart card communication.

Each manufacturer provides a different protocol for communication with the reader.

- First you have to communicate with the reader.
- Second, the reader communicates with the card, acting as the intermediary before sending the data to the card.
- Third, communication with a smart card is based on the APDU format. The card will process the data and return it to the reader, which will then return the data to its originating source.

2.1.4(b): The following classes are used for communicating with the reader:

- ISO command classes for communicating with 7816 protocol
- Classes for communicating with the reader
- Classes for converting data to a manufacturer-specific format
- An application for testing and using the cards for an intended and specific purpose

2.1.4(c): Communicating with a Smart Card Reader

The reader provides a path for your application to send and receive commands from the card. There are many types of readers available, such as serial, PC Card, and standard keyboard models. Unfortunately, the ISO group was unable to provide a standard for communicating with the readers so there is no one-size-fits-all approach to smart card communication.

Each manufacturer provides a different protocol for communication with the reader.

- First you have to communicate with the reader.
- Second, the reader communicates with the card, acting as the intermediary before sending the data to the card.
- Third, communication with a smart card is based on the APDU format. The card will process the data and return it to the reader, which will then return the data to its originating source.

In this paper the Smart Card used is of the type Contact type cards. Basically this type of Smart Cards got SIM like Structure Embedded on a Plastic card for Physical Structure and Strength. There exist different types of SIM structures according to the type of Application, Memory and features involved in the Smart Card. Some of them are shown below.
2.1.4 Types of SIM Structures

These Contact type Smart cards have a contact area, comprising several gold-plated contact pads, that is about 1cm square. When inserted into a reader, the chip makes contact with electrical connectors that can read information from the chip and write information back.

2.1.4(d): Electrical signals description

2.1.5 Keypad Section:
This section consists of a Linear Keypad. This keypad is used to enter the amount to deduct from smart card. The keypad is interfaced to microcontroller which continuously scans the keypad.

2.1.5 (a): LINEAR KEYPAD
This section basically consists of a Linear Keypad. Basically a Keypad can be classified into 2 categories. One is Linear Keypad and the other is Matrix keypad.

1. Matrix Keypad.
2. Linear Keypad.

2.1.5 (b): Matrix Keypad:
This Keypad got keys arranged in the form of Rows and Columns. That is why the name Matrix Keypad.

According to this keypad, In order to find the key being pressed the keypad need to be scanned by making rows as i/p and columns as output or vice versa. This Keypad is used in places where one needs to connect more no. of keys with less no. of data lines.

2.1.5 (c): Linear Keypad:
This Keypad got ‘n’ no. of keys connected to ‘n’ data lines of microcontroller. This Keypad is used in places where one needs to connect less no. of keys.

In this paper, Linear Keypad is used with 3 switches being connected because the no. of switches is less (less than 8).

Generally, in Linear Keypads one end of the switch is connected to Microcontroller (Configured as i/p) and other end of the switch is connected to the common ground. So whenever a key of Linear Keypad is pressed the logic on the microcontroller pin will go LOW. Here in this project, a linear keypad is used with switches connected in a serial manner. Linear keypad is used in this project because it takes less no. of port pins. The Linear Keypad with 4 Keys is shown below.

2.1.6 LCD Display Section:
This section is basically meant to show up the status of the project. This project makes use of Liquid Crystal Display to display / prompt for necessary information.

2.1.6 (a) Introduction to LCD:
In recent years the LCD is finding widespread use replacing LED s (seven-segment LED or other multi segment LED s). This is due to the following reasons:

1. The declining prices of LCD s.
2. The ability to display numbers, characters and graphics. This is in contrast to LED s, which are limited to numbers and a few characters.
3. Incorporation of a refreshing controller into the LCD, there by relieving the CPU of the task of refreshing the LCD. In the contrast, the LED must be refreshed by the CPU to keep displaying the data.
4. Ease of programming for characters and graphics.
3. Schematic diagram:
4. Simulation results:

1. Click on the Keil uVision Icon on Desktop
2. The following fig will appear

![Fig:4.1](image1.png)

3. Click on the Project menu from the title bar
4. Then Click on New Project

![Fig:4.2](image2.png)

5. Save the Project by typing suitable project name with no extension in your own folder sited in either C:\ or D:\

![Fig:4.3](image3.png)

6. Then Click on save button above.
7. Select the component for your paper. i.e. Atmel……
8. Click on the + Symbol beside of Atmel

![Fig:4.4](image4.png)
9. Select AT89C51 as shown below

10. Then Click on “OK”

11. The Following fig will appear

12. Then Click either YES or NO……..mostly “NO”

13. Now your project is ready to USE

14. Now double click on the Target1, you would get another option “Source group 1” as shown in next page.

15. Click on the file option from menu bar and select “new”
16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.

![Fig:4.9](image)

17. Now start writing program in either in "C" or "ASM"

18. For a program written in Assembly, then save it with extension " .asm" and for "C" based program save it with extension " .C"

![Fig:4.10](image)

19. Now right click on Source group 1 and click on “Add files to Group Source”

![Fig:4.11](image)

20. Now you will get another window, on which by default “C” files will appear.

![Fig:4.12](image)

21. Now select as per your file extension given while saving the file
22. Click only one time on option “ADD”

23. Now Press function key F7 to compile. Any error will appear if so happen.

24. If the file contains no error, then press Control+F5 simultaneously.

25. The new window is as follows

Fig: 4.13

26. Then Click “OK”

27. Now Click on the Peripherals from menu bar, and check your required port as shown in fig below

Fig: 4.15

28. Drag the port a side and click in the program file.

Fig: 4.16

29. Now keep Pressing function key “F11” slowly and observe.

30. You are running your program successfully
5. Conclusion

This paper “SMART CARD BASED TOLL GATE AUTOMATED SYSTEM” has been successfully designed and tested. Integrating features of all the hardware components used have developed it. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit.

6. acknowledgments:

This paper is dedicated to my mother V. Neelamma and my father V. Lakshmaiah.

VIII References:
1. 8051-MICROCONTROLLER AND EMBEDDED SYSTEM ali .Mohd.Mazidi
2. WWW.MITEL.DATABOOK.COM
3. WWW.ATMEL.DATABOOK.COM
4. WWW.FRANKLIN.COM
5. WWW.KEIL.COM

Authors Biography:

V.SRIDHAR working as Assistant professor in ECE department at Vidya Jyothi Institute of Technology, Hyderabad from 2009. completed M.Tech with Specialization Wireless and mobile communication systems from vardhaman college of engineering (AUTONOMOUS)JNTU, Hyderabad in 2011. He has completed M.Sc (IT) from Nagarjuna University, guntur, AndhraPradesh. completed Electronics and telecommunication engineering from vidya jyothi institute of technology, JNTU Hyderabad in 2007. His areas of research interests include Wireless and Mobile communications, Digital signal processing, image processing, Telecommunications, communication systems, signal processing. He is Lifetime Member of ISTE and IETE, IAE NG AND SDIW C. He is reviewer of SDIWC, IJARCET, and SSRGJ-IJCTT.

G.NAGENDRA is working as an HOD & Associate professor in ECE department at DR.k.v.subbareddy institute of technology, Kurnool. He completed M.TECH (communications and signal processing) from S.K University, Ananthapur. He completed B.TECH degree in ECE from JNTUH. His areas of research interests include digital signal processing, embedded systems, mobile communications and Artificial neural networks.