PLUG & PLAY (P&P) RFID,
TRAFFIC MONITORING & CONTROL (TM&C)

Submitted By

NC Madiha Hussain Malik

NC Ayesha Binte Ashfaq

Submitted to the Faculty of Computer Science
Military College of Signals, National University of Sciences & Technology,
Rawalpindi in partial fulfillment for the requirements of
B.E. Degree in Computer Software Engineering
April 2006

ABSTRACT

Our project consists of two main phases; i.e. Development of Plug & Play RFID (P&P RFID) software and the development of traffic monitoring and control system (TM&C).

Integrating RFID technology with the complete portfolio of enterprise mobility solutions that enable organizations to capture, move and manage critical information to and from every point of business activity. ‘Plug & Play RFID’ is capable to integrate any RFID system with the organizational structure, thus making the information available to all applications in the ERP for manipulation and decision making. Encoded information can be manipulated and can be retrieved as and when desired.

RFID has become indispensable in a wide range of automated data capture and identification applications, from ticketing and access control to industrial automation. ‘Traffic Monitoring & Control’ is one such application making the best use of RFID technology for the alleviation of congestion in metropolitan cities. It makes use of the data flow from the RFID web for decision making. TM&C is further used for automating the ticketing and tracking process for the first time in the history of the metropolitan city of Islamabad.

Dedicated to our dearest parents, especially our mothers and to all those who devoted their yesterdays for our brighter today.

Being the pioneer in introducing RFID to Pakistan, we dedicate our hard work to the Father of the Nation.

ACKNOWLEDGEMENT

All the praise and glory be to Almighty Allah, who is the source of wisdom and countless blessings of Him be upon his beloved Prophet (PBUH) who showed light of knowledge to the humanity. With their Mercy and Blessings we are able to complete this Project.

We wish to express our deepest obligations to our parents for their sacrifices, prayers, encouragement and inspiration that have always been with us.

For the completion of this project, we are greatly indebted to Dr. Shoaib Khan for his continuous guidance, remarkable suggestions, keen interest, friendly discussions and every possible support. He spared a lot of his precious time in advising and helping us throughout this Project. We think without his support this Project could not have been possible. We also extend our deepest gratitude to Asst. Prof. Arif Raza for always being appreciative of our work and encouraging us during the hardest of times.

We wish to express our deepest gratitude to some GulfNet professionals, especially Mrs. Sobia Yunous and Mr. Tauseef, for providing us with valuable ideas that guided us towards achieving our goals.

We wish to regard special thanks to Mr. Shahzad (London Metropolitan University) and Dr. Arslan (SZABIST) for their immense assistance.

PREFACE

The first real usage of RFID devices predates 1945. During World War II the United Kingdom used RFID devices to distinguish returning British airplanes from inbound German ones[1].

Perhaps the first work exploring RFID is the landmark 1948 paper by Harry Stockman, entitled "**Communication by Means of Reflected Power**"[15] , Stockman predicted that

"...considerable research and development work has to be done before the remaining basic problems in reflected-power communication are solved, and before the field of useful applications is explored."

However, it was only thirty years later that RFID became a reality and its application in different fields became a mainstay in many modern technological developments

Radio frequency identification (RFID) also called *dedicated short range communication (DSRC)*, is a wireless form of automated identification technology that allows for non-contact reading of data, which makes it effective for manufacturing and other hostile environments where bar code labels may not perform well. RFID tags have the potential to streamline and improve inventory management by allowing manufacturers to more efficiently enter and track the flow of goods.

"RFID eliminates the need for line-of-sight reading"

An RFID system consists of an antenna and a transceiver, which read the radio frequency and transfer the information to a processing device, and a transponder, or tag, which is an integrated circuit containing the RF circuitry and information to be transmitted[1].

RFID systems can be used just about anywhere, from clothing tags to missiles to pet tags to food -- anywhere that a unique identification system is needed. The tag can carry information as simple as a pet owners name and address or the cleaning instruction on a sweater to as complex as instructions on how to assemble a car.

RFID (Radio Frequency Identification) is an emerging technology that has found its applications in almost all fields of today's competitive environment. During 2005 and 2006, RFID is set to experience unprecedented growth as the world wakes up to the potential of this technology. Recently RFID aided marines were employed extensively by Americans in their 'WAR AGAINST TERROR' during the attack on Iraq. Marines on the ground in Fallujah had real-time visibility into the location of replenishments for the first time in military history.

Despite the extensive deployment of RFID in almost all fields of today, the systems being developed are too platform specific. As these RFID systems need to talk to disparate ERP (enterprise resource planning) systems, many are worried that investing in an ERP-specific RFID system could be a narrow and expensive path to take.

However, now the worries are no more. We have developed the system that will form a landmark in the field of RFID. The 'Plug & Play RFID' is capable to integrate any RFID system with the organizational structure. P&P integrates the system with the organizational database, thus making the information available to all applications in the ERP for manipulation and decision making. The data on a portable device, the tag, is read in and the database is updated. Multiple tags can be written with information flowing from the organizational system onto the tags. Tags information can be manipulated and can be retrieved as and when desired.

TABLE OF CONTENTS

Chapter 2: P&P RFID

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The RFID systems developed by the vendors around the globe are application specific, thus the need of the day is to develop a system that can integrate with any type of RFID regardless of the application field for which the RFID is meant.

Traffic congestion describes a condition in which vehicle speeds are reduced below normal, increasing drive times, and vehicle queueing occurs. It occurs only when the demand is greater than the roadway's capacity. Thus the need is to develop an algorithm for alleviating congestion in metropolitan cities.

Automating the *ticketing* and *tracking* process so as to minimize human involvement.

1.2. OBJECTIVES

The main objectives are:

Develop a globally accepted solution.

Develop a plug and play software system that is capable of operating with any and all types of RFID systems (following ISO15693 standard) and should integrate the system with the organizational database, thus making the information available to all applications in the ERP for manipulation and decision making.

Multiple tags can be written with information flowing from the organizational system onto the tags. Tags information can be manipulated and can be retrieved as and when desired.

The software should allow the dynamic programming of the tags.

The software should be able to dynamically decode the information read from the tag.

The software should allow organizational database to be RFID enabled.

Develop a scalable solution.

Develop the application that makes best use of the P&P RFID.

Capture events from an RFID reader, refining the RFID reads with filtering, aggregation, grouping and staging to business applications that will execute updates on the refined event data.

Process query requests from external applications against the events stored within the organizational database.

Develop traffic monitoring and control system that collects the run time data about the traffic from the RFID readers and is able to automatically generate the tickets, and to track vehicles at run time.

Algorithm development for the alleviation of congestion on major highways.

The extent of the achievement can be measured by continuous black box and white box testing of the modules developed and the verifications of the result by the personal in the organization.

One of the main personal objectives of pursuing the development in the field of RFID is to provide expertise and information to the students who plan to carry out further development in this field. Moreover, so as not to let Pakistan lag behind in the use and deployment of this promising technology.

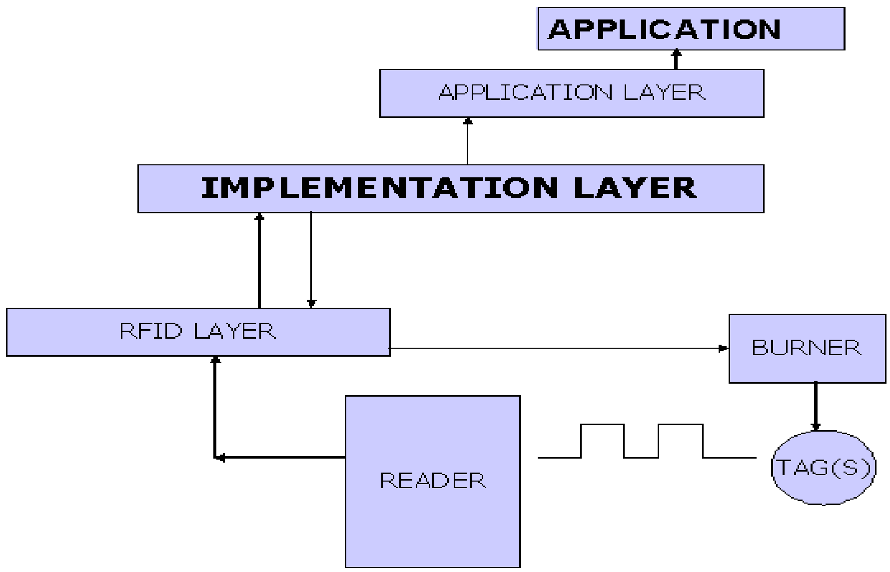


Figure 1.1: Overview of the complete system – Phase-I and Phase-II are explicitly highlighted

1.3. METHODS/TECHNOLOGY TO BE USED

The general methodology that we have used in our project is the software development lifecycle. The lifecycle includes these main phases:

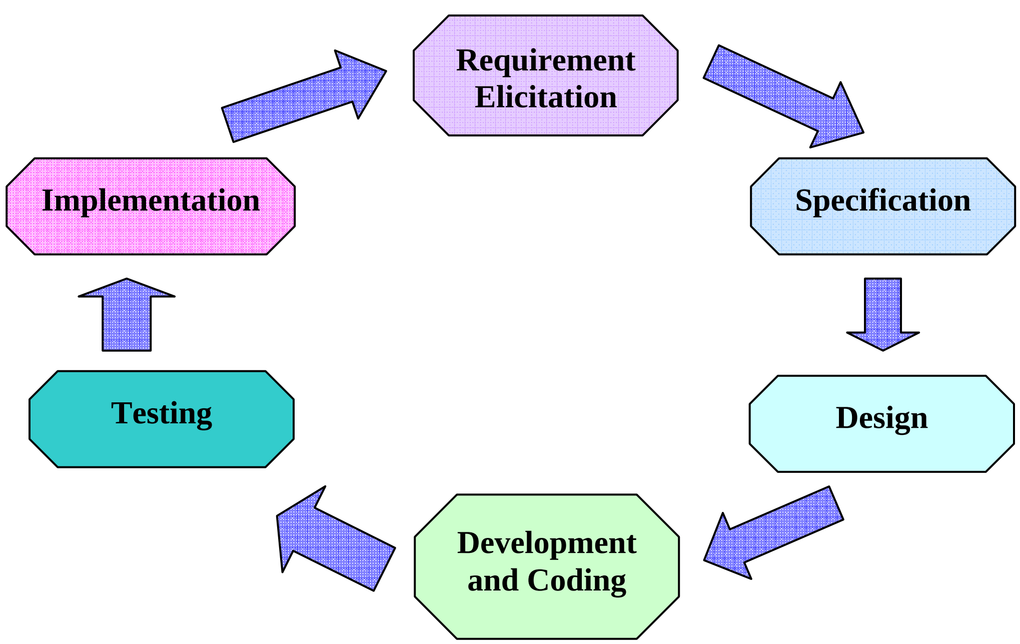


Figure 1.2: Methodology

1.4. PROJECT DELIVERABLES

Project product is the list of all the products or the project deliverables that the project will produce, such as software module, documentation, user guide and reports.
Our project deliverables include:

- The P&P RFID software
- The TM&C software
- Congestion Alleviation Simulation
- User guides
- Project Report

1.5. RESOURCES

1.5.1 Hardware Resources

RightTag Evaluation Kit:

We have purchased the RFID system from RightTag, CA. The specifications of the system are shown below:

The Evaluation Kit contains:

- One RightTag USB Hand-Held RFID Reader/Writer
- 100 RFID Tags
- TagWriter Utility

Table 1.1: Specification

Part	**RFID Readers**
Operating Frequency	13.56 MHz
RF Power	Max 200 mW
Read Range	14 cm with credit card size tag
Antenna bandwidth	1MHz @ -3dB
Antenna Impedance	50 Ohm@ 13.56 MHz
Tag Compatibility	ISO15693, Tag-it
Communication Interface	RS232 or USB
Host Data rate	9600, 19200, 57600 or 115200 N, 8, 1
Operating Temperature	-20°C to +55°C (including self-generated heat)
Storage Temperature	-40°C to +80°C

1.5.2 Software Requirements

Two software environments we will required for development of our project are following:

JAVA

Visual Basic

1.5.3 Software Development

P&P RFID front end has been developed in java, using jdk1.5.

TM&C has been developed in java; the back end is designed in Oracle.

Simulation of the Congestion Alleviation Algorithm has been developed in VB.

2.1 LITERATURE REVIEW

2.1.1 RFID Basics

A basic RFID system consists of three components:

1. An antenna or coil
2. A transceiver (with decoder)
3. A transponder (RF tag) electronically programmed with unique information

2.1.1.1 An Antenna or Coil

The antenna emits radio signals to activate the tag and reads and writes data to it. Antennas are the conduits between the tag and the transceiver, which controls the system's data acquisition and communication. Antennas are available in a variety of shapes and sizes; they can be built into a door frame to receive tag data from persons or things passing through the door, or mounted on an interstate toll booth to monitor traffic passing by on a freeway. The electromagnetic field produced by an antenna can be constantly present when multiple tags are expected continually. If constant interrogation is not required, the field can be activated by a sensor device.

Often the antenna is packaged with the transceiver and decoder to become a reader, which can be configured either as a handheld or a fixed-mount device. The reader emits radio waves in ranges of anywhere from one inch to 100 feet or more, depending upon its power output and the radio frequency used. When an RFID tag passes through the electromagnetic zone, it detects the reader's activation signal. The reader decodes the data encoded in the tag's integrated circuit (silicon chip) and the data is passed to the host computer for processing.

2.1.1.2 Transceiver

The term transceiver is a device that performs, within one chassis, both transmitting and receiving functions that is in a common housing, sometimes designed for portable or mobile use, uses common circuit components for both transmitting and receiving which provides half-duplex operation [14].

2.1.1.3 Transponder

The term **transponder** has the following meanings:

> An automatic device that receives, amplifies, and retransmits a signal on a different frequency.
>
> An automatic device that transmits a predetermined message in response to a predefined received signal.
>
> A receiver-transmitter that will generate a reply signal upon proper electronic interrogation.

The word transponder, derived from **TRANSmitter/resPONDER,** reveals the function of the device. The tag responds to a transmitted or communicated request for the data it carries, the mode of communication between the reader and the tag being by wireless means across the space or air interface between the two.

In particular, a communications satellite's channels are called transponders, because each is a separate transceiver or repeater.

Another example of transponders is in identification friend or foe systems and air traffic control secondary radar (beacon radar) systems. For general aviation and Commercial aviation transponders used on planes to amplify radar signals making the plane more visible. Most transponders are also able to transmit altitude information and a four digit code known as a transponder code or a "squawk code" to help air traffic controllers in separation.

Sonar transponders operate under water are used to measure distance through water and form the basis of underwater location marking, position tracking and navigation.

2.1.1.3.1 Types of Transponder

Transponders can either be Read Only (RO), Read/Write (R/W), or Digital Signal Transponders (DST).

Read only contains a unique factory code.

Read/Write allows users to reprogram codes many times.

DST offers more memory for system history tracking and additional authorization.

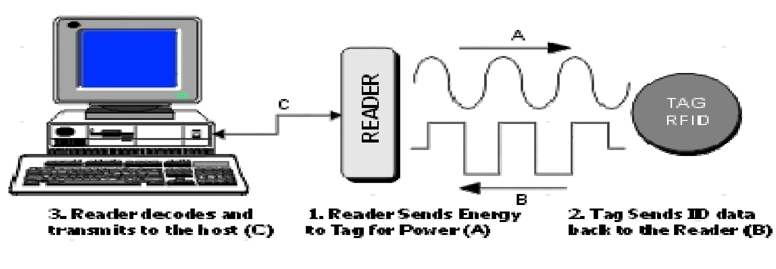


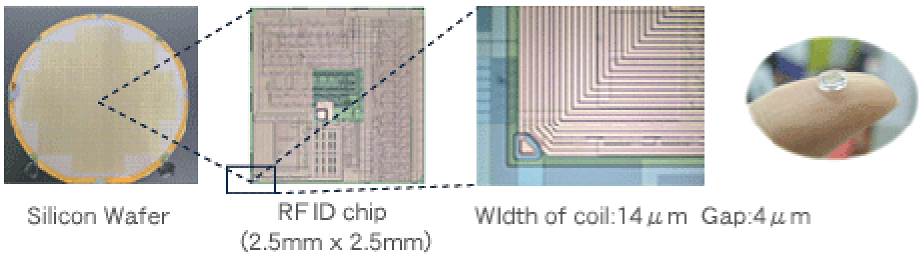
Figure 2.1: Communication between RFID system and Tag

2.1.2 RFID Tags

RFID tags come in a wide variety of shapes and sizes [11]. Animal tracking tags, inserted beneath the skin, can be as small as a pencil lead in diameter and one-half inch in length. Tags can be screw-shaped to identify trees or wooden items, or credit-card shaped for use in access applications. The anti-theft hard plastic tags attached to merchandise in stores are RFID tags. In addition, heavy-duty 5- by 4- by 2-inch rectangular transponders used to track containers or heavy machinery, trucks, and railroad cars for maintenance and tracking applications are RFID tags. RFID tags are categorized as either active or passive.

2.1.2.1 Active RFID tags

They are powered by an internal battery and are typically read/write, i.e., tag data can be rewritten and/or modified [14]. An active tag's memory size varies according to application requirements; some systems operate with up to 1MB of memory. In a typical read/write RFID work-in-process system, a tag might give a machine a set of instructions, and the machine would then report its performance to the tag. This encoded data would then become part of the tagged part's history. The battery-supplied power of an active tag

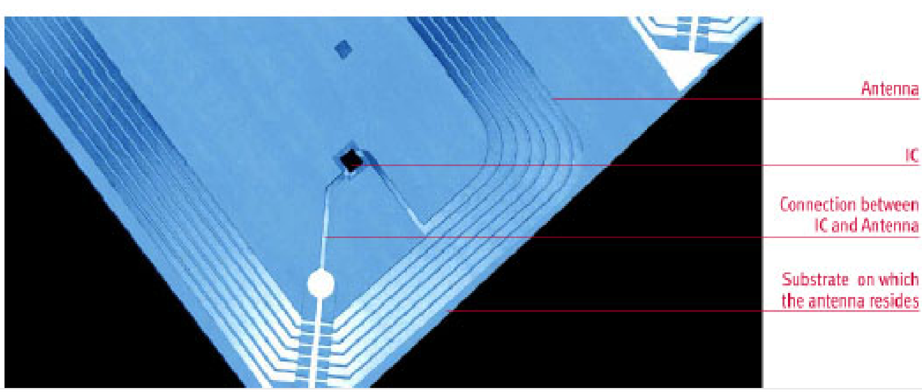


generally gives it a longer read range. The trade off is greater size, greater cost, and a limited operational life (which may yield a maximum of 10 years, depending upon operating temperatures and battery type).

Figure 2.2: Tag Types

2.1.2.2 Passive RFID tags

They operate without a separate external power source and obtain operating power generated from the reader. Passive tags are consequently much lighter than active tags, less expensive, and offer a virtually unlimited operational lifetime. The trade off is that they have shorter read ranges than active tags and require a higher-powered reader. Read-



only tags are typically passive and are programmed with a unique set of data (usually 32 to 128 bits) that cannot be modified. Passive RFID Tag Architecture is shown in the figure [3].

Figure2.3: Tag Architecture

2.1.3 Frequency Ranges

RFID systems are also distinguished by their frequency ranges.

2.1.3.1 Low Frequency System

Low-frequency (30 KHz to 500 KHz) systems have short reading ranges and lower system costs. They are most commonly used in security access, asset tracking, and animal identification applications.

2.1.3.2 High-frequency Systems

High-frequency (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz) systems, offering long read ranges (greater than 90 feet) and high reading speeds, are used for such applications as railroad car tracking and automated toll collection [10]. However, the higher performance of high-frequency RFID systems incurs higher system costs [5].

Table 2.1: RFID systems and their frequency ranges and applications

	Frequency	Distance & Cost	Example Application
LF	125 kHz	Few cm, ¢	Auto-Immobilizer
HF	13.56 MHz	1m, 50¢	Building Access
UHF	915 MHz	7m, 50¢	Supply Chain / retail / CPG
µwave	2.4 GHz	10m, $'s	Traffic Toll

2.1.4 RFID Characteristics

There are six key characteristics of RFID that affect the communication between a tag and reader: Range, Range Adjustment, Propagation, Directionality, Multi-Tag Collection and Memory [9]. Each of these six key characteristics determines the type of communication that takes place between the reader and the tag. These characteristics are described below.

2.1.4.1 Range

Range is defined as the maximum distance for successful Tag-Reader communication. Read range difference will vary and can be very-short, short, or long.

Very Short Range: approx. up to 60cm (2 ft)

Short Range: approx. up to 5 m (16 ft)

Long Range: approx. 100+ m (320+ ft)

2.1.4.2 Range Adjustment

Range Adjustment will also play a role in RFID tag read functionality. Range adjustment is the ability to adjust range and is categorized as very good or poor. Very good range adjustment can be fine-tuned to a specific distance. Tag-Reader communication is guaranteed within the specified range and tag-reader communication outside the range is impossible. Whereas poor range adjustment cannot be adjusted well at all. When there is a signal fall-off pattern or a reflection, tag-reader communication in the physical area is not guaranteed.

2.1.4.3 Propagation

Propagation is the ability to perform tag-reader communication through or around objects and material. With very good propagation, the radio frequency can penetrate through objects allowing successful communication between tag and reader. Plus, very good propagation allows for penetration through water, liquids and human tissue and may even go through metal. Whereas poor propagation works on in line-of-sight and any obstacle such as a wall, people or vehicles between the tag and reader will prevent any successful communication.

2.1.4.4 Directionality

Directionality is the ability to achieve directional RF coverage using directional antennas. There are two types of directionality: Omni-directional and Directional. Omni-directional coverage has similar RF coverage in all directions. With directional coverage, the RF coverage is much stronger in one specific direction.

2.1.4.5 Multi-tag collection

Multi-tag collection is the ability to quickly and reliably collect large number of tags within a designated area.

2.1.4.6 Memory

Memory is key in RFID communication — it determines the read only, read/write, or write once read many capabilities in the tag-reader communication. Some tags have small memory size at 16 bits and others have larger memory with 512 kBytes or more.

2.1.5 Working of RFID System

In a typical RFID system, individual objects are equipped with a small, inexpensive tag. The tag contains a transponder with a digital memory chip that is given a unique electronic product code. The interrogator, an antenna packaged with a transceiver and decoder, emits a signal activating the RFID tag so it can read and write data to it. When an RFID tag passes through the electromagnetic zone, it detects the reader's activation signal. The reader decodes the data encoded in the tag's integrated circuit (silicon chip) and the data is passed to the host computer for processing.

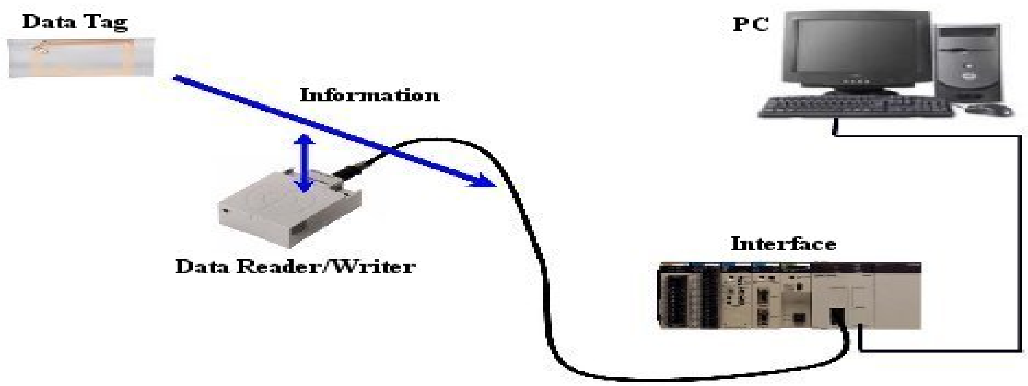


Figure 2.4: Working of RFID systems

2.1.6 Advantages of RFID System

The significant advantage of all types of RFID systems is the non-contact, non-line-of-sight nature of the technology. Tags can be read through a variety of substances such as snow, fog, ice, paint, crusted grime, and other visually and environmentally challenging conditions, where barcodes or other optically read technologies would be useless. RFID tags can also be read in challenging circumstances at remarkable speeds, in most cases responding in less than 100 milliseconds.

The read/write capability of an active RFID system is also a significant advantage in interactive applications such as work-in-process or maintenance tracking. Though it is a costlier technology (compared with barcode) [12].

RFID has become indispensable for a wide range of automated data collection and identification applications that would not be possible otherwise [8].

The purpose of an RFID system is to enable data to be transmitted by a portable device (a tag) which is read by an RFID reader and processed according to the needs of a particular application. The data transmitted by the tag may provide identification or location information, or specifics about the product tagged, such as price, color, date of purchase, etc [4].

RFID offers highly reliable data collection in harsh environments. It eliminates manual data entry that was slow and prone to errors. It inspires new automation solutions by fundamentally changing how processes are managed and how businesses operate. Thus RFID causes a paradigm shift [6].

2.1.6.1 Barcode VS RFID

RFID tags have superior capabilities and benefits over barcode technology [13] in the following ways:

A power source is not required for passive RFID tags, which is a key defining benefit.

Non- "Line of Sight," high-speed, and multiple reads are possible, changing the nature of how this technology can be applied [7].

The RFID Electronic Product Code (EPC) standard extends the UPC standard, by providing for an individual unit to have a specific and unique identity (ID).

RFID tags can have read and write capability for both ID and other data.

2.1.7 Standardization

An RFID system can utilize a few standards. The problem has been that there is no one universally accepted standard. Competing standards have been one of the more difficult issues for RFID, and as a result, most RFID applications have been closed systems.

Standards and specifications may be set at the international, national, industry or trade association level, and individual organizations may term their own specifications as "standard." Many industry standards and specifications set by individual organizations are based on international standards to make implementation and support easier and to provide a wider choice of available products.

Standards can be applied to include the format and content of the codes placed on the tags, the protocols and frequencies that will be used by the tags and readers to transmit the data, the security and tamper-resistance of tags on packaging and freight containers, and applications use. The two largest drivers for RFID today are Wal-Mart and the Department of Defense (DOD). Both have issued mandates for their top suppliers to use RFID technology when shipping products to their distribution centers. They are both looking to accomplish the same thing, but have a slightly different long-term outlook.

2.1.7.1 ISO Standards

The International Organization for Standardization (ISO) is based in Geneva, and its standards carry the weight of law in some countries. All ISO standards are required to be available for use around the world, so users of ISO RFID standards will not have to worry if their systems comply with the different regulations on frequencies and power output for each country where they do business. The ISO is very active in developing RFID standards for supply chain operations and is nearing completion on multiple standards to identify items and different types of logistics containers.

ISO 14443 for **"proximity"** cards and ISO 15693 for **"vicinity"** cards both recommend 13.56 MHz as its carrier frequency. These standards feature a thinner card, higher memory space availability and allow numerous cards in the field to be read almost simultaneously using anti-collision, bit masking and time slot protocols.

- **ISO 14443 proximity cards** offer a maximum range of only a few inches. It is primarily utilized for financial transactions such as automatic fare collection, bankcard activity and high security applications. These applications prefer a very limited range for security.

- **ISO 15693 vicinity cards, or Smart Tags,** offer a maximum usable range of out to 28 inches from a single antenna or as much as 4 feet using multiple antenna elements and a high performance reader system.

2.1.8 RFID Privacy Issues

RFID technology has been subject to controversy. The main concerns relate to privacy, and include:

Whether the purchaser of an item will be aware of the presence of the tag or be able to remove or deactivate it;

Whether the tag can be read at a distance without the knowledge of the individual;

If a tagged item is paid for by credit card or in conjunction with use of a loyalty card, whether it would be possible to tie the unique ID of that item to the identity of the purchaser

The basic privacy concerns associated with an RFID system is the ability of ubiquitous tracking of anybody without consent. And with RFID tags getting smaller and smaller, it is now even possible to hide tags in such a way that the consumer may be unaware of the presence of tags. For example, the tags may be sewn up within garment, or molded within plastic or rubber. To the extent that researchers have already developed tiny coded beads invisible to human eye that can be embedded in inks to tag currency and other documents, or added to substances like automobile paint, explosives, or other products that law enforcement officers or retailers have a strong interest in tracking. Researchers say that the technology should be ready for commercial use in 3-6 years.

2.1.8.1 Methods of Ensuring Privacy

With all these privacy concerns, there is bound to be some effort to thwart such attempt at privacy and maintain the popularity of RFIDs. Researches at various places have yielded the following methods of avoiding above-mentioned attacks.

- **RSA Blocker Tags:**

  These tags are similar in size and appearance to RFID tags helps in maintaining the privacy of consumer by “spamming” any reader that attempts to scan tags without the right authorization, thus confusing the reader to believe that there are many tags in its proximity.

- **Kill Switches:**

  Newer RFID tags are being shipped with a “Kill Switch”, which will allow the RFID tags to be disabled. Thus a consumer will be given an option of disabling the RFID tag before leaving the store, thus avoiding the possibility of stealth tracking and profiling.

2.2 REQUIREMENT ELICITATION

Despite the extensive deployment of RFID in almost all fields of today, the systems being developed are too platform specific. As these RFID systems need to talk to disparate ERP (enterprise resource planning) systems, many are worried that investing in an ERP-specific RFID system could be a narrow and expensive path to take.

Majority of the RFID systems are application specific. If you buy an RFID system then its software comes along with it, so if you are running number of applications on your central server that are using RFID systems, then this implies that you have number of standalone applications running on your server, so our aim is to replace all these applications with a single generalized software that is able to connect with any and all types of RFID system, gets the tagged information and route the information to its relevant application.

Moreover, the software should be able to bridge up the gap between the RFID systems and the organizational database applications.

2.3 REQUIREMENT ANALYSIS

2.3.1 Generic Tool

So a generic software is required that is able to connect with any and all types of RFID systems. Actually, there is no single standard that is followed globally. Although many standard making bodies like ISO and IEC have developed multiple standards but there is not a single globally followed standard. However, IOS15693 is a standard that is followed in UK and USA, so if any vendor wants his products to sell in these countries then he has to follow ISO15693. So we have picked this standard and decided to develop the system that is able to connect with any and all types of RFID systems that follow ISO15693 standard.

There are two types of communication. One is from reader to tag communication and another is from PC to reader communication. Reader to tag communication follows ISO15693 standard, but the PC to reader communication varies from vendor to vendor. So the main task is to standardize this communication thus by allowing the PC to connect and integrate with all types of RFID systems.

2.3.2 Bridging Software

Second main requirement is to bridge up the gap between the RFID systems and the organizational database, thus making the information available to all applications in the ERP for manipulation and decision making. Once the data is read in, its validity must be checked, then it should be decoded, then it is routed to its particular application.

2.3.3 Support for Commands and Options

Third main requirement is that software should support all ISO15693 commands to date. There are number of commands that are defined in ISO15693 standard, and the software should be able to execute all those commands.

Each command has specific options that can be set. These options let the user execute the commands as per his/her requirements. So the fourth main requirement is to provide specific options for every command.

2.3.4 Read Modes

There are two types of read modes available, i.e. continuous mode and on click mode. Both these modes should be supported by the software. In continuous mode, the data should be read continuously and at the back end the database should be updated.

2.3.5 Writing Modes

Another requirement is that single as well as multiple tags can be written with information flowing from the organizational system onto the tags. Tags information can be manipulated and can be retrieved as and when desired.

2.3.6 Platform Independent

The organization with which we have developed the system develops ERP solutions. So it has number of clients and the major requirement is that the software should be platform independent so that it is capable of running on any platform.

2.3.7 Support for Database Tools

The database tools like Oracle and Access should be supported. As Oracle is used in most of the organizations so support of this tool is provided to make the software more generic.

2.4 DESIGN

2.4.1 Basic Design

P&P acts as a middleware between RFID system and the applications in the ERP. Two API are needed, one API is needed for communication between P&P and RFID system and another API facilitates the communication between P&P and applications in the ERP. The basic design is shown in figure 2.5.

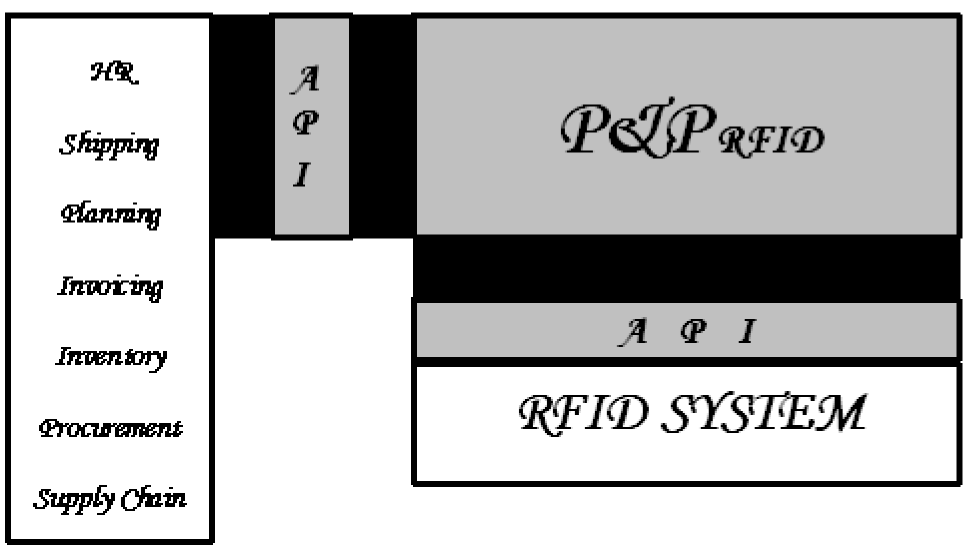


Figure 2.5: P&P acting as a middleware between the RFID hardware & the applications in the ERP

P&P RFID gets the information from the RFID system. The DB at the back end is updated by the P&P based on the information available. The data from the database is available to the applications in the ERP for decision making. Moreover, ERP can communicate with the RFID system through the P&P RFID layer. This dataflow is shown in figure 2.6.

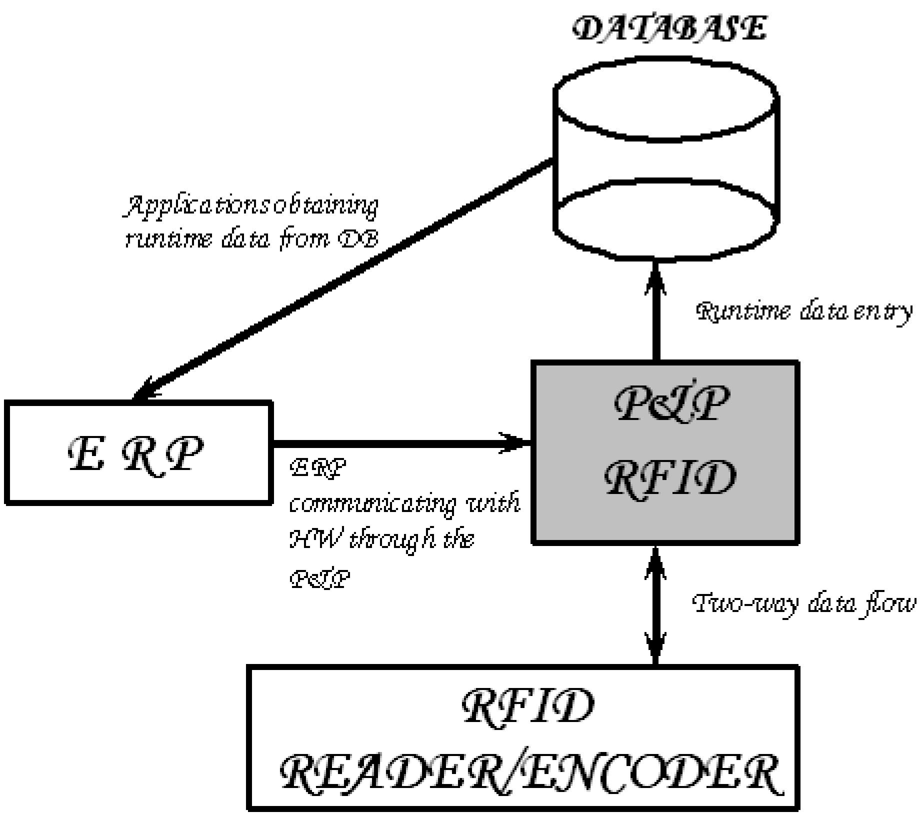


Figure 2.6: P&P offering duplex communication

2.4.2 Main Phases

The P&P RFID software design was divided into following phases.

<u>Phase I</u> Setting connectivity with the RFID reader.

<u>Phase II</u> Configuring the Organizational Database with RFID System

<u>Phase III</u> Defining the Encoding Scheme.

<u>Phase IV</u> Dynamic Encoding of the tags/Smart Labels.

<u>Phase V</u> Dynamic Decoding the information read from the tag.

<u>Phase VI</u> Supporting Advanced Commands.

<u>Phase VII</u> Allowing Continuous Mode read operation.

<u>Phase VIII</u> Advanced Options

<u>Phase IX</u> Generic Tool

<u>Phase X</u> GUI setup.

We have developed P&P RFID software that will be an independent plug and play system that supports RFID systems from different vendors. It can be employed by the organization for dynamically enabling the database application to be RFID configured.

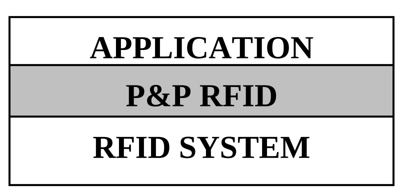


Figure 2.7: P&P RFID acting as a middleware

Moreover, the tagged information read in by the reader will be dynamically decoded. The decoded information will then be placed into the internal organizational database. For the first time the IL software is run, it will integrate the Organizational Database with the

RFID system, thus making it RFID enabled. From then on, any information read in by the reader will be dynamically placed in the Organizational Database.

Phase I   Setting connectivity with the RFID reader

Depending on the underlying hardware, functionality is also provided to set:

COM Port

Once a com port is selected, the system tests the port for reader presence. If the reader is found, the connection is established across which then data is sent and received. If however, the reader is not found on the port specified, the user is prompted to select another port and the same procedure is repeated.

The auto-detect option is also provided, which let the system to find whether the reader is connected to the PC or not.

Baud Rate

The baud rate can also be set by the user, as per user requirements. If it is required to send and receive data at a higher rate, higher baud rate can be selected and so on.

In order to ensure the correctness of data read or written, CRC checking is also provided to the user.

CRC ON

With each manipulation, data is verified and the user is informed.

CRC OFF

No data checking is done. Any error that might occur is not detected.

Phase II   Configuring the Organizational Database with RFID System

The software allows connection to the database. Supported database types are Oracle and Access. Once connected, values are retrieved and stored as tags are written and read respectively. The values are thus available to applications instantaneously.

Phase III Defining the Encoding Scheme

The data is stored on the tag for future reference. So some encoding scheme need to be defined that will assist in retrieving the information from the RFID tag. As we are integrating the RFID system with the organizational database, so it is required that whenever any tag is read or written, the database should be updated. So we have defined an encoding scheme that identifies the relevant table, column and row that is to be updated. During writing process, the data is written onto the tag according to the encoding scheme, and during reading process; the data is read and then decoded based on the encoding scheme.

Phase IV Dynamic Encoding of the tags/Smart Labels

Encoding or writing process is supported in our software. Data can be written on to each individual tag. Data selection methodology includes:

Single Write

Select a single value from the database. It is written onto a single tag.

Multiple Write

Multiple fields are selected by the user from the database. A single value is taken and written on to a tag followed by selection of the next specified value and so on. Hence multiple tags are written in turn.

Phase V Dynamic Decoding the information read from the tag

It includes reading the data on the tags. This is handled by the air interface protocol which constitutes part-2 of ISO15693 standard. Once the tag is read and data is available to the P&P software, then the data is processed and subsequently decoded. The relevant table, column and row are identified and the database is updated.

Phase VI Supporting Advanced Commands

Data fields

The protocol format is raw binary. Every REQUEST / RESPONSE starts with a 0x02 character, which indicates the start of data packet.

The possible fields for requests and responses are:

1. 0x02 (mandatory)
2. Flags (1 byte, mandatory)
3. Reader address (1 byte, optional)
4. Command (1 byte, mandatory)
5. Length of data n (1 byte, mandatory)
6. Data (n bytes, n may also be 0)
7. CRC (2 bytes, optional)

Flags

The Flags byte indicates the presence of Reader address byte (ie if the command is addressed to a specific reader or any reader listening to the command), the presence of CRC bytes.

Binary format of Flags byte: 00000CBA

Bit A: set, if CRC bytes are present in the request / response.

Bit B: set, if the command is addressed to a specific reader (ie address byte is present in the request / response).

Bit C: Error flag, currently only the Reader can set this flag to indicate that something went wrong when processing the request from PC.

Other bits are RFU and should be set to 0.

Command byte

Command byte selects the purpose of the message and will determine the layout of data field in the request / response. A response by the Reader is always with the same command byte than it was in the request.

CRC

If the CRC is present in the request, it will also be included in the response. The CRC is calculated on all the bytes in the request / response excluding the 0x02 character at the beginning and the CRC itself. The 16-bit CRC is written MSB first into the message.

Get Tag Info

Information particular to the tag, as for example its Manufacturer ID, User Memory, Status of the blocks and so on. This command can be used to get the TAG's parameters.

Request: 0x02 0x00 0x22 0xnn 0xXX (Tag ID if in addressed mode)
Response: 0x02 0x00 0x22 0x06 0xYY (DSFID) (AFI) (Memory structure) (IC reference)

0xXX is the **information byte**.
Bits of it have the following meaning: HGFEDCBA

Bits FEDCBA: Tag type.

Bit G: If set, the transmitter remains on after the command is executed.

Bit H: If set, then the command is meant for a specific tag and the ID of this tag follows this byte (length depends on the tag's protocol).

0xYY contains the following bits: 0000DCBA

Bit A: Set, if DSFID field (1 byte) is present.

Bit B: Set, if AFI field (1 byte) is present.

Bit C: Set, if information about memory structure (2 bytes) is present. The first byte represents the total number of memory blocks in the TAG (the number is 1 less than the actual number of blocks, so that 1...256 blocks may be present). The second byte shows the block size (the number is also 1 less than the actual number of blocks).

Bit D: Set, if IC reference byte is present

Get Security Status

Provides information about the security status of the tag, if the security is critical data cannot be updated.

Request: 0x02 0x00 0x27 0xnn 0xXX (Tag ID if in addressed mode) 0xYY 0xZZ

Response: 0x02 0x00 0x27 0xmm (mm bytes of security status)

0xXX is the information byte.

0xYY is the number of first block.

0xZZ is the number of blocks.

There is 1 byte for each requested block in the data area of the response: 0x00 shows, that the block is unlocked, 0x01 means locked.

Maximum number of blocks that can be asked using a single request is 64.

Inventory

Inventory all the tags passing through the reader field. All active tags in return provide their UID.

This command can be used to inventory tag(s).
Request: 0x02 0x00 0x29 0x02 0xXX 0xYY
Response: 0x02 0x00 0x29 0xmm (TAG ID)

0xXX is the information byte. Note that the MSB of this byte does NOT select between addressed / non-addressed mode, but selects the read mode: single or multiple. If set, the reader detects all tags in the field. Also note that if multiple read is selected, many responses can be expected.

Write AFI

AFI (Application Family Identifier) specifies the application field of the information on the tag. If the tagged data belongs to a particular type of application, it can be specified at write time.
This command can be used to write the TAG's AFI field.

Request: 0x02 0x00 0x23 0xnn 0xXX (Tag ID if in addressed mode) 0xYY
Response: 0x02 0x00 0x23 0x00
0xXX is the information byte.
0xYY is the new AFI byte.

Lock AFI

Lock the field of AFI so that it cannot be altered later. Once locked the field cannot be unlocked
This command can be used to permanently lock the AFI field.

Request: 0x02 0x00 0x24 0xnn 0xXX (Tag ID if in addressed mode)

Response: 0x02 0x00 0x24 0x00

0xXX is the information byte.

Write DSFID

DSFID (Data Storage Format Identifier) has a separate field on the tag. It can also be specified by the user.

This command can be used to write the TAG's DSFID field.

Request: 0x02 0x00 0x25 0xnn 0xXX (Tag ID if in addressed mode) 0xYY

Response: 0x02 0x00 0x25 0x00

0xXX is the information byte.

0xYY is the new DSFID byte.

Lock DSFID

Lock the field of DSFID so that it cannot be altered later. Once locked the field cannot be unlocked

This command can be used to permanently lock the DSFID field.

Request: 0x02 0x00 0x26 0xnn 0xXX (Tag ID if in addressed mode)

Response: 0x02 0x00 0x26 0x00

0xXX is the information byte.

Lock Block

Lock a particular block on the tag. Each block contains four bytes.

This command can be used to permanently lock a block of memory.

Request: 0x02 0x00 0x20 0xnn 0xXX (Tag ID if in addressed mode) 0xYY

Response: 0x02 0x00 0x20 0x00

0xXX is the information byte.
0xYY is the number of block that will be locked.

Write TAG

This command can be used to write the TAG's memory.

Request: 0x02 0x00 0x20 0xnn 0xXX (Tag ID if in addressed mode) 0xYYYY 0xZZ (data to be written into the tag's memory)
Response: 0x02 0x00 0x20 0x00
0xXX is the information byte.
0xYYYY is the number of first byte that will be written.
0xZZ is the number of bytes to be written into the tag's memory.

Read Data

This command can be used to read data from tag.

Request: 0x02 0x00 0x2A 0xnn 0xXX (Tag ID if in addressed mode) 0xYYYY 0xZZ
Response: 0x02 0x00 0x2A 0xZZ (ZZ bytes of data)
0xXX is the information byte.
0xYYYY is the number of first byte that will be read.
0xZZ is the number of bytes to read.

Phase VII Allowing Continuous Mode read operation

The mode for the tag reading can also be set. The options are:

Continuous

When the reader is set in continuous mode, it continues to read each tag as they pass through the reader spectrum. If the continuous mode is selected, all other commands are disabled. The Set Mode command is used to set the mode of reading the tag. This command can be used to set the working mode for the Reader.

Request: 0x02 0x00 0x05 0x01 0xXX

XX is the new mode (1 byte). Bits of it have the following meaning: 0GFEDCBA
Bit A: Send TAG ID automatically.
Bit B: Send TAG Data automatically.
Bit C: Continuous read mode - if set, the Reader is in continuous read mode and sends all TAG IDs / Data whenever a TAG comes present in the Reader's antenna field.
Bit D: Disable button.
Bit E: If set, the reader identifies all tags in the field when the pushbutton is pressed. If cleared, only the first tag will be identified.
Bit F: If set, the transmitter remains ON after pressing the pushbutton, otherwise it will be switched OFF.
Bit G: If set, the reader will send tag ID in protocol format (bits A and B have no meaning in such case).
Other bits are RFU and should be set to 0.

On Click
Allows the tags to be read only when specified. Any command can be executed.

Phase VIII   Advanced Options

Each command has to it specific options that can be set. These options let the user execute the commands as per his/her requirements.

Tag Type
Specify the tag type the user is operating. Each tag type has its own structure and requirements; hence it must be handled carefully to prevent data loss.

Tag ID

Specified if the user wants to manipulate a particular tag. If other tags are present in the reader field, they are not disturbed.

Leave Transmitter On

Once a command is executed, the transmitter should be left on for further executions.

Multiple Inventory

Allows the reader to perform multiple inventory incase more than one tags are present in the reader spectrum.

AFI

Specify the AFI for the tag to be read. Only the tags with, matching AFI fields are read.

DSFID

Specify the DSFID for the tag to be read. Only the tags with, matching DSFID fields are manipulated.

Phase IX Generic Tool

There are basically two types of communication. One is from PC to reader and another is from reader to PC.

From reader to tag communication, standard ISO 15693 is followed, but the PC to reader communication format varied for different vendors. Each vendor specifies their own command codes. We have standardized this communication by defining the command codes.

Phase X GUI setup

GUI design and development is done, keeping in view the requirements of the user. User friendly menus and screens have been designed. Each command is shown with all the relevant options.

2.5 DEVELOPMENT

2.5.1 Front End Development

Front end of the application is developed in java. The java environment is used to fulfill the objective of platform independence. The jdk1.5 is used, and the programming environment used is JCreator Pro version.

For every command in ISO15693, separate class is written. So there are ten different classes, each specific to different command. Namely the classes written are:

1. ReadData.java
2. WriteData.java
3. GetTagInfo.java
4. GetSecurityStatus.java
5. Inventory.java
6. LockBlock.java
7. WriteAFI.java
8. LockAFI.java
9. WriteDSFID.java
10. LockDSFID.java

For connectivity purpose three classes are written. These classes establish the connectivity with the reader. They first find the RFID reader on all ports of the computer, and once found then communicate with the Reader.

GeneralParameters.java: This class sets basic parameters for connectivity purpose.

SerialCommunication.java: This class opens and closes the input and output streams with the RFID Reader.

ReaderFound.java: This class finds the reader on all ports of the computer.

PacketExchange.java: For sending and retrieving the information from or to the tag, commands are sent to the reader, and the responses are handled by the software. PacketExchange class is written that deals with the exchange of packets to and from the reader.

There are two types of reading modes, i.e. continuous and on click mode. For supporting the Continuous Mode operation, two classes are written. i.e.

SetContinuousMode.java

UnSetMode.java

Set ContinuousMode.java class is written to convert the Read mode to continuous mode. And UnSetMode.java is written to convert the Read mode to on click read mode. Once the reader is set to read in continuous mode then all the other commands are disabled.

There are two types of mode available for writing as well. User can specify one value at a time, or can specify multiple values at a time, which will be written to the tags. The values that are written to the tag are selected from the database. The classes available for specifying the values for the tag writing are:

Driver_DB.java

DB_Multiple.java

MultipleSelectionTest.java

Inter.java

Whenever, any tag is written the relevant portion of the database needs to be updated. Moreover, when any tag is read, either by on click mode or in continuous mode then the database at the back end needs to be updated. For this purpose following class is written.

updateDB.java

Two database tools are supported in our software, i.e. Oracle and Access. For establishing the connection with these databases, a separate class, first.java, is written.

first.java: This class first authenticates the access to the database and then establishes the connection that will be used for accessing the database. Unless the login is verified the user cannot proceed further.

For setting up of the GUI, number of classes are written. These classes include:

GUI_Frame.java

ashimadi.java

spinner.java

ProgressBar.java

They set the user interface of the software. The GUI is designed in such a way that it provides the user with the list of supported commands. Along with each command, list of available options are also specified.

Tag Type

User is provided with the option to specify the tag type of the tag. In this way tag can be specified for tags of a particular type.

Tag ID

User can specify the tag id of the particular tag, in this way the command is executed for that particular tag, and only its information can be manipulated and other tags in the field of the reader are not disturbed.

Leave Transmitter On

User is provided with the option to leave the transmitter on for further execution.

Multiple Inventory

User is provided with the option to inventory multiple tags. In this way tag ids of all the tags are returned, that are currently in the field of the reader.

AFI

User is provided with the option to specify the AFI for the tag to be read. Only the tags with, matching AFI fields are read.

DSFID

User can specify the DSFID for the tag to be read. Only the tags with, matching DSFID fields are manipulated.

In order to provide interface to the external application to access the software and thus by allowing the external application to read and write data to the tags, two classes are written. These classes are:

ReadData_API.java

WriteData_API.java

These classes provide the interface in the forms of the methods that can be called by any external application that wants to use the functionality of the software to integrate with the RFID system.

2.5.2 Back End Development

At the Back end we have used both Oracle and MS Access. Both the DB engines are supported to make the system more generic in terms of DB settings as well.

2.5.3 User Manual

The user manual is written to assist the user in using the software. The user manual covers the instructions for installation as well. Trouble shooting is included in the user manual to assist the user to tackle the problems himself. User Manual of the software is attached in appendix A.

2.6 TESTING & VALIDATION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Our software is thoroughly tested based on the specification of the user. The testing process that we have followed consists of following phases:

Defining Test Data

Firstly we define what will be the input to the test process. Tests are conducted for both valid and invalid inputs. We have divided the input data in number of partitions and one sample input data is selected from each partition.

Defining Test Cases

Input to test the system and the predicted output from these inputs if the system operates according to its specification is called Test Case. We have defined test cases for checking the application functionality against different user interactions.

Conducting Tests

Once the test data and the test cases are ready, then the system is thoroughly tested. The aim of conducting tests is top highlight the system defects.

Comparing outcome of the process with the predicted result.

Once the outcome of the test process is available, then that is compared with the predicted result of conducting that test. The difference gives the deviation of actual performance from the desired performance.

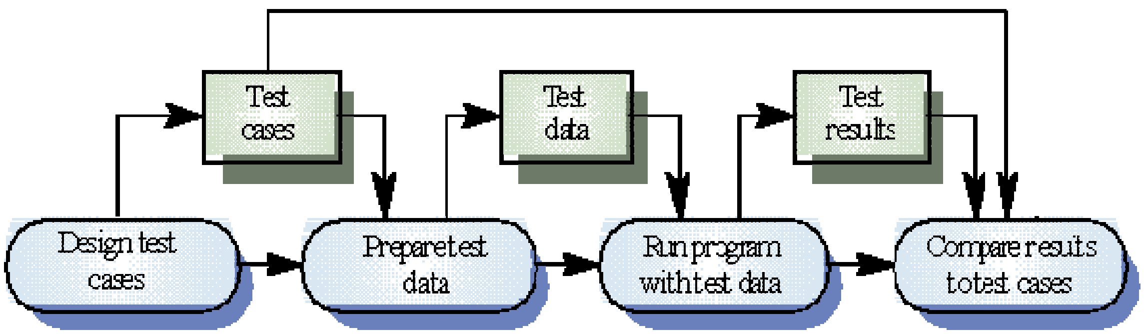


Figure 2.8: Test Data and Test Cases

This process is recursive and cyclic. Testing is done until the reasonable performance measure is achieved. The order in which we perform the testing process is as follows:

Component/Unit Testing

As the modules are developed, they are tested side by side, to remove maximum errors at initial stages. We have developed separate class for every ISO15693 command. So each class is tested as an individual entity.

Integration Testing

Once all the modules are developed, then they are integrated to form a complete system. The errors that might have occurred during the integration process are removed by conducting integration testing. The GUI is developed that acts as an interface to execute different commands, so the thorough testing of the GUI is carried out to ensure error handling at each stage of user interaction.

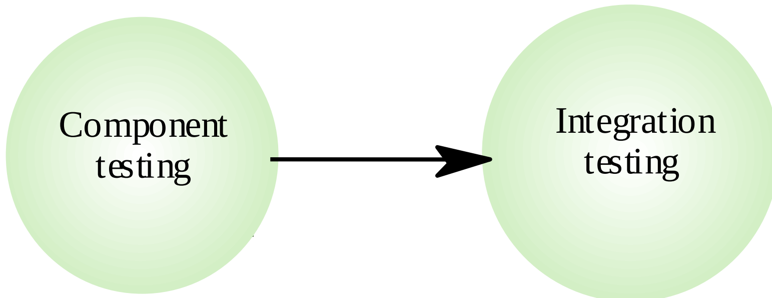


Figure2.9: Unit to Integration Testing

The testing of the Software was divided into two major testing phases:

Black Box Testing

It is an approach to testing where the program is considered as a ‘black-box’. The program test cases are based on the system specification.

White Box Testing

In white box testing, test cases are derived from the program structure. Knowledge of the program is used to identify additional test cases. Objective is to exercise all program statements

3.1 LITERATURE REVIEW

3.1.1 RFID Applications

RFID has become indispensable in a wide range of automated data capture and identification applications, from ticketing and access control to industrial automation. The fact that the RFID tag needs only to be located within the EMF field of the reader, not necessarily in a direct line-of-sight as required to read a barcode, enhances its functionality for item tracking over barcodes that need direct line-of-sight to the reader. RFID has the potential to streamline and improve inventory management by allowing manufacturers to more efficiently enter and track the flow of goods.

Radio frequency identification (RFID) has been used in managing inventory, electronic access control, security systems, automatic identification of cars on toll roads, electronic article surveillance ("EAS"), etc. By utilizing the advantages of radio frequency, RFID tags will work under more hostile environmental conditions than optical bar-code labels since RFID tags are capable of being read through non-metallic substances such as paint, water, dirt, dust, human bodies, concrete, and even through the tagged item itself. RFID is established in a wide range of applications.

3.1.1.1 Animal Identification

Low-frequency RFID tags are commonly used for animal identification, beer keg tracking, and automobile key-and-lock, anti-theft systems. Pets are often embedded with small chips so that they may be returned to their owners if lost. In the United States, two RFID frequencies are used: 125 kHz (the original standard) and 134.5 kHz, the international standard.

3.1.1.2 Book Tracking

High-frequency RFID tags are used in library book [25] or bookstore tracking, pallet tracking, building access control, airline baggage tracking, and apparel item tracking. High-frequency tags are widely used in identification badges, replacing earlier magnetic stripe cards. These badges need only be held within a certain distance of the reader to authenticate the holder.

3.1.1.3 Container Tracking

UHF RFID tags are commonly used commercially in pallet and container tracking, and truck and trailer tracking in shipping yards.

3.1.1.4 Electronic Toll Collection

The latest and most prominent use of RF tags is in highway toll collection applications where they operate as E-Z Pass, FastPass, or similar names. They allow vehicles to pass through toll collection stations without stopping to speed up the collection process on toll bridges and toll roads.Some toll booths, such as California's FasTrak and Illinois' I-Pass system, use RFID tags for electronic toll collection. The tags are read as vehicles pass;

the information is used to debit the toll from a prepaid account. The system helps to speed traffic through toll plazas.

3.1.1.5 Animal tagging, Identify Fraud

Implantable RFID "chips", originally designed for animal tagging are being used and contemplated for humans as well. Applied Digital Solutions proposes their chip's "unique under-the-skin format" as a solution to identity fraud, secure building access, computer access, storage of medical records, anti-kidnapping initiatives and a variety of law-enforcement applications. Combined with sensors to monitor body functions, the Digital Angel device could provide monitoring for patients. The Baja Beach Club in Barcelona, Spain uses an implantable Verichip to identify their VIP customers, who in turn use it to pay for drinks. The Mexico City police department has implanted approximately 170 of their police officers with the Verichip, to allow access to police databases and possibly track them in case of kidnapping.

3.1.1.6 RFID Aided Marines in Iraq

Marines on the ground in Fallujah had real-time visibility into the location of replenishments for the first time in military history [23].

Feb. 21, 2005—It would be an exaggeration to say radio frequency identification technology was the U.S. Marines Corps' secret weapon during the battle for Fallujah in Iraq late last year. But the technology did give the Marines real-time visibility into the location of en route replenishments during the battle. It was the first time battalion commanders on the ground could see and control the flow of replenishments, according to Col. Mark Nixon, head of the Marine Corps' Logistics Vision and Strategy Center.

"War fighters have to be able to influence the distribution pipeline, and RFID will enable them to do that," he said, speaking at the U.S. Department of Defense's recent RFID Summit for Industry. "It's still in its infancy."

Figure 3.1: U.S. Marine Corps shipping containers with active RFID tags

Nixon said that when the Marine Corps went to Iraq, some units had active tags not just on pallets, but on vehicles. RFID readers were set up at a distribution center in Kuwait, at the Iraq-Kuwaiti border and at checkpoints along the main arteries into Iraq. When trucks passed the readers, the location of the goods they were carrying was updated in the U.S. Department of Defense's In Transit Visibility network database. That enabled commanders on the ground to see the precise location of replenishments needed to sustain operations.

<u>3.1.1.7 Automatic Retail Fueling System</u>

Figure 3.2: Automatic Retail Fueling System

<u>3.1.1.8 RFID Takes Attendance</u>

Here's how the InClass system works [24]: A unique 15-digit ID number is written to each tag and associated with the name of the student to whom it is issued. As the students pass through the reader-generated interrogation field under a doorway, the reader sends the tags' unique ID numbers to a central server. InCom has developed a software program, installed on the server, that collects the tag data and uploads a list of present, absent and tardy (based on when they enter the classroom) students to a PDA that is issued to the teacher. The upload is done wirelessly over an 802.11b Wi-Fi protocol. The teacher can then perform a visual check on the InClass-generated attendance list by scanning the room to reconcile what the list says with what she sees in the classroom. Once confirmed, the list is submitted wirelessly via the same PDA to school administrators, who are required to file attendance records to a state board of education.

Figure 3.3: Automatic Attendance System

3.1.1.9 Electronic Resources Cataloger [16]

Radio frequency identification (RFID) tagging is an alternative to bar coding for physical access to library materials. The technology is rather new for libraries although it has been used in retail and warehousing applications since the 1980s[17]. It promises to streamline operations by enabling faster self-checkout and self-returns, improving shelf management and inventory control, and providing better theft protection. Security can be handled in two ways. Security gates can query the ILS to determine its security status or the tag may contain a security bit which would be turned on and off by circulation or self-check reader stations. Security gates can then detect whether or not the item has been properly checked out of the library. When users return items, the security bit is re-set and the item record in the ILS is automatically updated. In some RFID solutions a return receipt can be generated. At this point, materials can be roughly sorted into bins by the return equipment. Inventory wands provide a finer detail of sorting. This tool can be used to put books into shelf-ready order. It can also be taken to the stacks to detect out-of-place items.

Richard Boss provides a cost/benefit comparison of RFID systems[18]. The first benefit is faster circulation operations. The tags can be read regardless of how the item is placed on the circulation reader and multiple items can be read simultaneously. This could reduce long line- ups at busy circulation desks. The second benefit is increased self-check. Users do not have to carefully line up bar-codes for a successful transaction. The units have a simpler interface which can make it easier for patrons to serve themselves. Circulation staff can be free to perform other public service work. The third benefit mentioned by Boss is reliable security. Vendors claim that the theft detection rate of RFID is high, although there is not yet any evidence that it is better than electro-magnetic systems. At least those RFID systems which can query the ILS can indicate what precisely what items are moving out of the library. This is great for replacing stolen items. The fourth benefit is faster inventorying.

3.1.2 Road Traffic Engineering

Traffic engineering is a branch of civil engineering that uses engineering techniques to achieve the safe and efficient movement of people and goods. It focuses mainly on research and construction of the immobile infrastructure necessary for this movement, such as roads, railway tracks, bridges, traffic signs and traffic lights[19].

Increasingly however, instead of building additional infrastructure, dynamic elements are also introduced into road traffic management. These use sensors to measure traffic flows and automatic, interconnected guidance systems (for example traffic signs which open a lane in different directions depending on the time of day) to manage traffic especially in peak hours.

The relationship between lane flow (Q) (vehicles per hour) maximum speed (V) (kilometers per hour) and density (K) (vehicles per kilometer) is $Q = KV$. Observation on limited access facilities suggests that up to a maximum flow, speed does not decline while density increases, but above a critical threshold, increased density reduces speed, and beyond a further threshold, increased density reduces flow as well.

Therefore, managing traffic density by limiting the rate that vehicles enter the highway during peak periods can keep both speeds and lane flows at bottlenecks high[20]. Ramp meters, signals on entrance ramps that control the rate at which vehicles are allowed to enter the mainline facility, provide this function (at the expense of increased delay for those waiting at the ramps).

3.1.2.1 Hierarchy of roads

Roads have been distinguished by function probably since their beginning, and the hierarchy of roads was firmly established by the time of the Roman Empire.

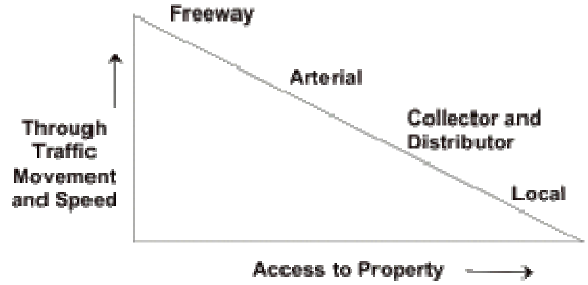


Figure 3.4: Hierarchy of Roads

3.1.2.1.1 Freeways

At the top of the hierarchy are limited access roads freeways or motorways, including most toll roads. These roads provide largely uninterrupted travel, often using partial or full access control, and are designed for high speeds. Some freeways have collector/distributor roads or even more extensive local lanes (also known as collector/distributor lanes) which further reduce the number of access ramps that directly interface with the freeway, rather the freeway periodically interfaces with these parallel roadways, which themselves have multiple on and off-ramps. These allow the freeway to operate with less friction at an even higher speed and with higher flow.

3.1.2.1.2 Arterials

In general arterials are major through roads that are expected to carry large volumes of traffic. Arterials are often divided into major and minor arterials, and rural and urban arterials.

In some places there are large divided roads with few or no driveways that cannot be called freeways because they have occasional at-grade intersections or they are just too short. Such roads are usually classified as arterials.

3.1.2.1.3 Collectors

Collectors collect traffic from local roads and distribute it to arterials. Traffic using a collector is usually going or coming from somewhere nearby.

3.1.2.1.4 Local roads

At the bottom of the hierarchy are local roads. These roads have the lowest speed limit, and carry low volumes of traffic.

3.1.2.2 Traffic Congession

Traffic congestion describes a condition in which vehicle speeds are reduced below normal, increasing drive times, and vehicle queueing occurs. It occurs only when the demand is greater than the roadway's capacity (or when capacity is less than the demand). These conditions are also more frustrating for drivers (see road rage), and automobile accidents may be more frequent. However, congestion reduces speed which is a primary factor contributing to accidents. So, traffic congestion can reduce the number and severity of accidents. Furthermore, vehicles burn unnecessary fuel when stuck at idle. A period of extreme traffic congestion is known as a **traffic jam.** Traffic engineers sometimes apply the rules of Fluid Dynamics to traffic flow, likening it to the flow of a fluid in a pipe.

Figure 3.5: Traffic Jam

Economist Anthony Downs, in his books *Stuck in Traffic* (1992) and *Still Stuck in Traffic* (2004), offers a dissenting view: rush hour traffic congestion is inevitable because of the benefits of having a relatively standard work day. In a market economy, goods can be allocated either by pricing (ability to pay) or by queueing (first-come first-serve); congestion is an example of the latter. Instead of the traffic engineer's solution of making a "pipe" large enough to accommodate the total demand for peak-hour vehicle travel, either by widening roadways or increasing "flow pressure" via automated highway systems, Downs advocates greater use of road pricing to reduce congestion, in turn plowing the revenues generated therefrom into public transportation projects.

The Texas Transportation Institute estimates that in 2000 the 75 largest metropolitan areas experienced 3.6 billion vehicle-hours of delay, resulting in 5.7 billion US gallons (21.6 billion liters) in wasted fuel and $67.5 billion in lost productivity. Traffic congestion is increasing in major cities, and delays are becoming more frequent in smaller cities and rural areas.

Figure 3.6: The Santa Monica Freeway in Los Angeles is famous for traffic congestion.

3.1.2.2.1 Attempts to alleviate traffic congestion

Figure 3.7: Traffic jam because of road accident, Algarve, Portugal

Improvements of junctions

Building new roads and widening of existing ones (this practice is criticised by proponents of the induced demand hypothesis)

Road pricing, or tolls, such as the London Congestion Charge, a fee levied on vehicle drivers entering the centre of the city

Cheaper and better public transport offered by the local or provincial government, see Public Transport Funding

Setting of school opening times to avoid problems associated with the school run

High-occupancy vehicle lanes or "carpool lanes"

Quotas on the number of vehicles on the road. Examples are the fairly successful "cap and trade" method used in Singapore and number plate restrictions on certain days of the week, as practiced in Athens and Metro Manila.

Traffic management and prevention of accidents

Doing away with left turn at crossroads with traffic lights, which reduces the time needed to go through such crossroads.

3.1.2.3 Traffic flow

The mathematical study of **traffic flow**, and in particular vehicular traffic flow, is done with the aim to get a better understanding of these phenomena and to assist in prevention of traffic congestion problems. The first attempts to give a mathematical theory of traffic flow dated back to the 50s, but to this day we still do not have a satisfactory and general theory to be applied in real flow conditions.

This is because traffic phenomena are complex and nonlinear, depending on the interactions of a large number of vehicles. Moreover, vehicles do not interact simply following the laws of mechanics, but also due to the reactions of human drivers. In particular, they show phenomena of cluster formation and backward propagating shocks of vehicle density. Fluctuations in measured quantities (e.g. mean velocity of vehicles) are often huge, leading to a difficult understanding of experiments [21].

Thus, the modelling of traffic flow is one of the most challenging themes of mathematical physics.

Scientists approach the problem in mainly three ways, correspnding to the three main scales of observation in physics.

microscopic scale: you write for every vehicle a model, i.e. an equation, that is usually an ODE.

macroscopic scale: in analogy with fluid dynamics models, you write a system of (PDE) balance laws for some gross quantities of interest, e.g density, mean velocity, flux.

mesoscopic (kinetic) scale: you define a function $f(t,x,V)$ which expresses the probability of having a vehicle at time t in position x which runs with velocity V. This function, following methods of statistical mechanics, can be computed solving an integro-differential equation, like the Boltzmann Equation[22].

3.1.2.4 Traffic light

Figure 3.8: Traffic Signal

A traffic light or traffic signal is a signalling device positioned at a road intersection or pedestrian crossing to indicate when it is safe to drive, ride or walk, using a universal color code.

Traffic lights can have several additional lights for filter turns or bus lanes. This one also shows the distinctive red + amber combination.

Traffic lights for normal vehicles or pedestrians always have two main lights, a red light that means stop and a green (or sometimes white for pedestrians) light that means go.

Usually, the red light contains some orange in its hue, and the green light contains some blue, to provide some support for people with red-green color blindness. In most countries there is also a yellow (or amber) light, which when on and not flashing means stop if able to do so safely. In some systems, a flashing amber means that a motorist may go ahead with care if the road is clear, giving way to pedestrians and to other road vehicles that may have priority. A flashing red essentially means the same as a regular stop sign. There may be additional lights (usually a green arrow or "filter") to authorize turns.

Traffic lights for special vehicles (such as buses or trams) may use other systems, such as vertical vs. horizontal bars of white light.

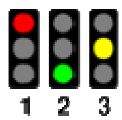


Figure 3.9: 3-state traffic light (as used in the United States)

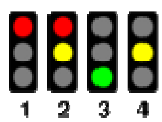


Figure 3.10: 4-state lights warn traffic that it will shortly be free to move

In most countries, the sequence is red (stop), green (go), amber (prepare to stop).

In most countries, amber officially means 'stop' (unless it would cause an accident to do so) but in practice, is treated as 'prepare to stop'. In the UK, Hong Kong (but not mainland China), Germany, Poland, and Iceland, among others, the sequence includes red + amber together before green, which helps draw attention to the impending change to green, to allow drivers to prepare to move off.

Depending on the jurisdiction, traffic may turn after stopping on a red (right in right-driving countries; left in left-driving countries), provided they yield to pedestrians and other vehicles. In some jurisdictions which generally forbid this, a green arrow sign next to the traffic light indicates that it is allowed at a particular intersection. Conversely, jurisdictions which generally allow this might forbid it at a particular intersection with a "no turn on red" sign, or might put a green arrow to indicate specifically when a right turn is allowed without having to yield to pedestrians (this is usually when traffic from the perpendicular street is making a left turn onto one's street and thus no pedestrians are allowed in the intersection anyway). Some jurisdictions allow turning on red in the opposite direction (left in right-driving countries; right in left-driving countries) from a one-way road onto another one-way road; some of these even allow these turns from a two-way road onto a one-way road. Also differing is whether a red arrow prohibits turns; some jurisdictions require a "no turn on red" sign in these cases. A study in the State of Illinois concluded that allowing drivers to proceed straight on red after stopping, at specially posted T-intersections where the intersecting road went only left, was dangerous. Proceeding straight on red at T-intersections where the intersecting road went only left was once legal in Mainland China with right-hand traffic provided that such movement would not interfere with other traffic, but when the Road Traffic Safety Law of the People's Republic of China took effect on 1 May 2004, such movement was outlawed.

3.1.2.4.1 Control and coordination

Traffic signals must be instructed when to change phase. They can also be coordinated so that the phase changes called for occur in some relationship with nearby signals.

Traffic signal phase changes are based on one of three systems: pre-timed, semi-actuated, and fully-actuated. The simplest control system uses a timer; each phase of the signal lasts for a specific duration before the next phase occurs; this pattern repeats itself regardless of traffic. Many older traffic light installations still use timers; timer-based signals are effective in one way grids where it is often possible to coordinate the traffic lights to the posted speed limit.

More sophisticated control systems use electronic sensor loops buried in the pavement to detect the presence of traffic waiting at the light, and thus can avoid giving the green light to an empty road while motorists on a different route are stopped. A timer is frequently used as a backup in case the sensors fail; an additional problem with sensor-based systems is that they may fail to detect vehicles such as motorcycles or bicycles and cause them to wait forever (or at least until a detectable vehicle also comes to wait for the light). The sensor loops typically work in the same fashion as metal detectors; small vehicles or those with low metal content may fail to be detected. Attempts are often made to place traffic signals on a coordinated system so that drivers encounter long strings of green lights. The distinction between coordinated signals and synchronized signals is very important. Synchronized signals all change at the same time and are only used in special instances or in older systems. Coordinated systems are controlled from a master controller and are set up so lights "cascade" in sequence so platoons of vehicles can proceed through a continuous series of green lights.

3.1.2.4.2 Red light cameras

In some areas, a device usually called a red light camera has come into recent use. A camera is connected to the triggering mechanism for the corresponding traffic light, which is targeted to photograph any vehicle which crosses against the light. The driver or owner (depending on local laws) of a vehicle so photographed can then be fined for violating traffic laws. Such cameras have evoked controversy on a number of fronts: in some jurisdictions, the fine cannot be contested, and is therefore seen by some as a violation of due process. Opposition has also stemmed from the practice of paying commissions to the companies which process the photographs from these cameras, as this is seen as an incentive to falsify images. Some have accused municipalities of purposely shortening the yellow-light intervals on intersections equipped with cameras in order to generate more fines. The presence of a red light camera is sometimes, but not always, indicated by a sign some distance before the intersection. Many red light cameras also face the front of vehicles they are used to catch running red lights; therefore, it is possible for vehicles either registered in states that do not use or require front license plates (or vehicles illegally without front plates from states that do require them) to escape being caught.

3.1.3 RMI Basics

Remote method invocation is the object-oriented equivalent of remote method calls, which allows programmers to build network applications using a programming construct similar to the local procedure call, providing a convenient abstraction for both interprocess communication and event synchronization. In this model, a process invokes the methods in an object, which may reside in a remote host. As with RPC, arguments may be passed with the invocation.

An Overview of RMI Applications

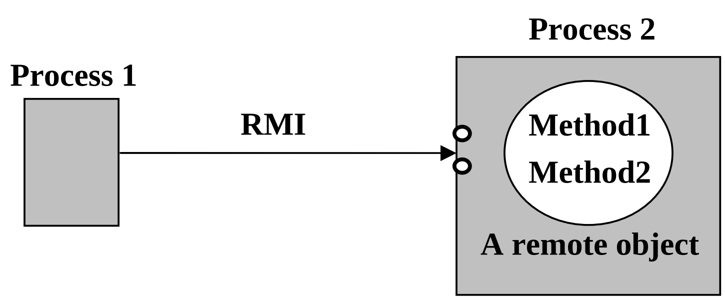


Figure 3.11: The Remote Method Call Paradigm

3.1.3.1 Server

Creates some remote objects

Makes references to them accessible

Waits for clients to invoke methods on these remote objects.

3.1.3.2 Client

Gets a remote reference to one or more remote objects in the server

Invokes methods on them.

3.2 DESIGN

3.2.1 Overall Design

The knowledgebase has been designed for minimizing the response time and to maximize the throughput and the operational speed so as to update the DB swiftly. Once the data is read in by the readers deployed at the crossings, the data is sent to the central server which then updates the DB so that up-to-date information is available at all times for manipulation and decision making. During the KB design the ‘Client-Server Architecture’ is followed, the basic reason being that such an architecture supports the development and execution of database applications.

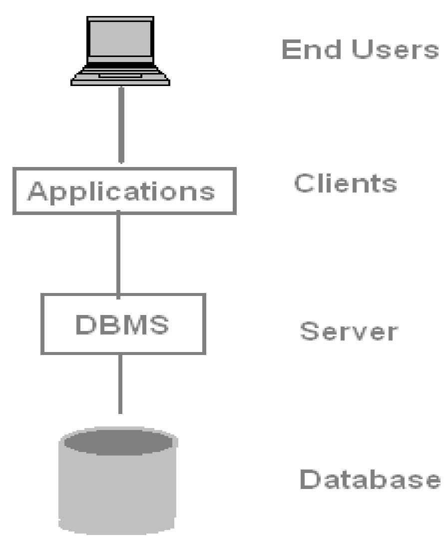


Figure 3.12: Basic Design

As the reader machines are connected to the main server in a communication network, thus the architecture can be more appropriately represented as follows

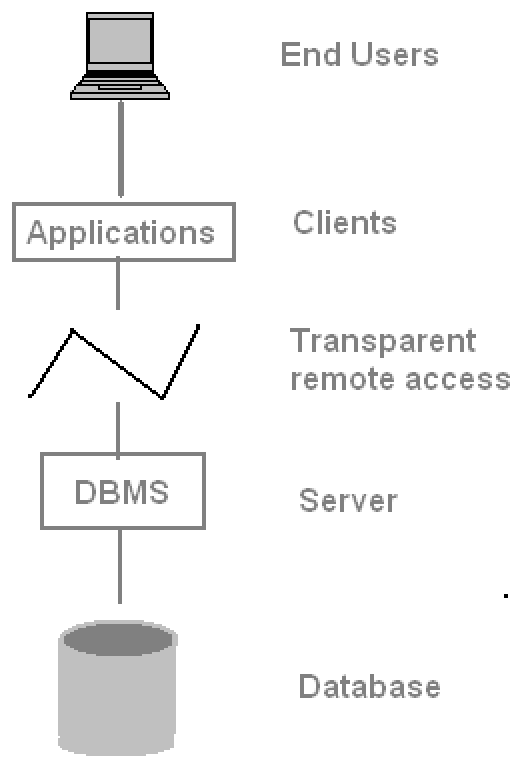


Figure 3.13: Basic Design

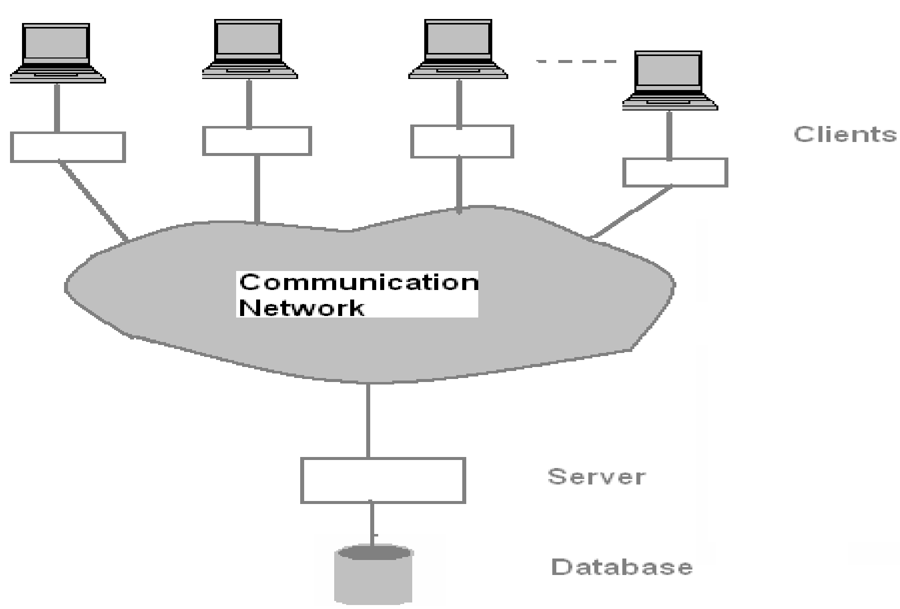


Figure 3.14: Overall Design

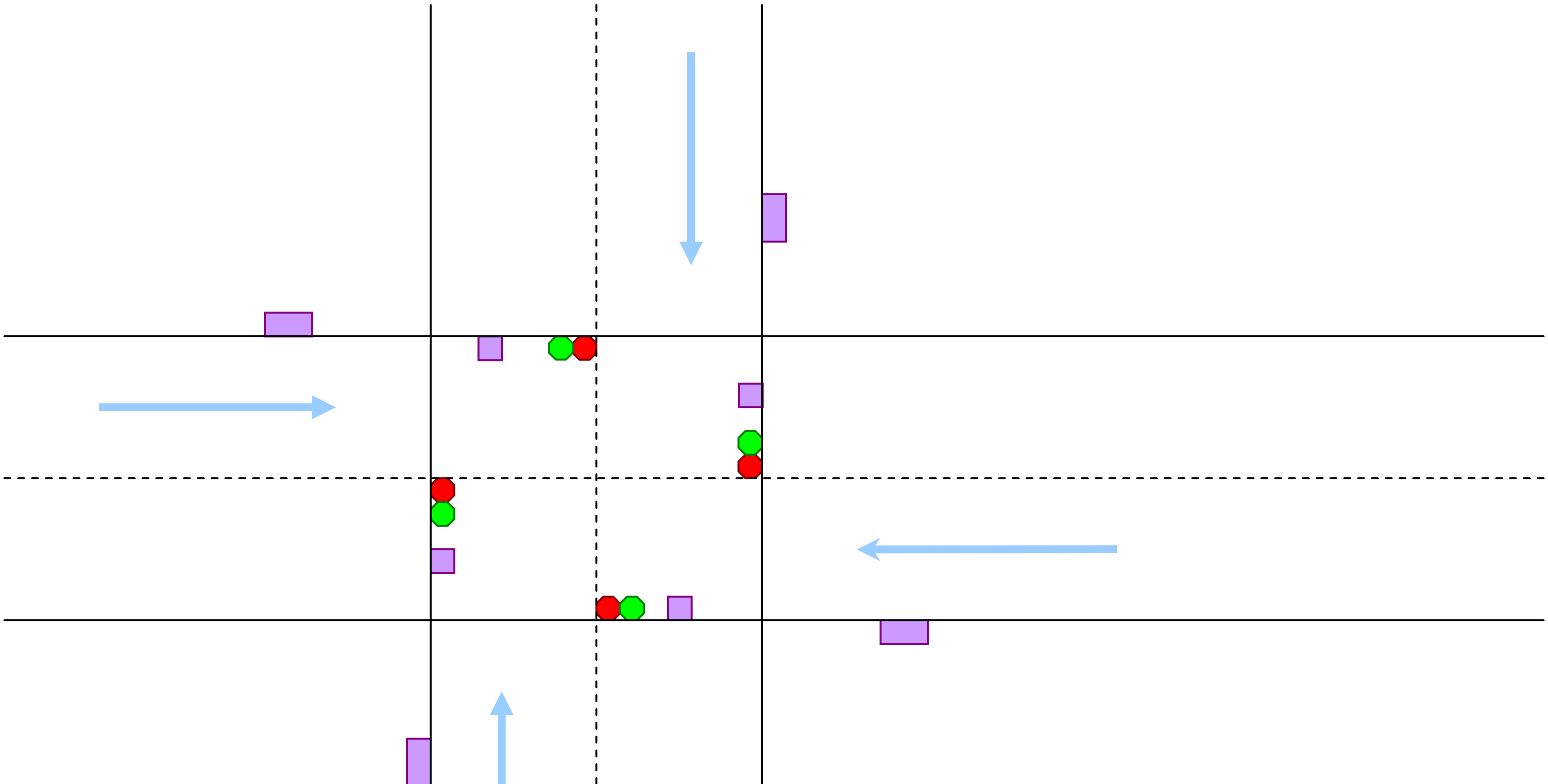
Thus several different client machines will be able to access the same server machine.. Thus a single database is shared among across several distinct client systems.

3.2.2 Road Network Design

For the verification of the software, TM&C, the following road network has been used. It employs two crossings. Traffic Monitoring and control software is completely scalable and can be used for any road network.

Readers have been deployed at the crossings and along road sides so as to assist the operations of ticketing and tracking. Furthermore, the reader deployment is critical in the congestion alleviation algorithm implementation. The algorithm relies heavily on the locations the readers are deployed. Thus a complete reader deployment map has also been presented later in the report.

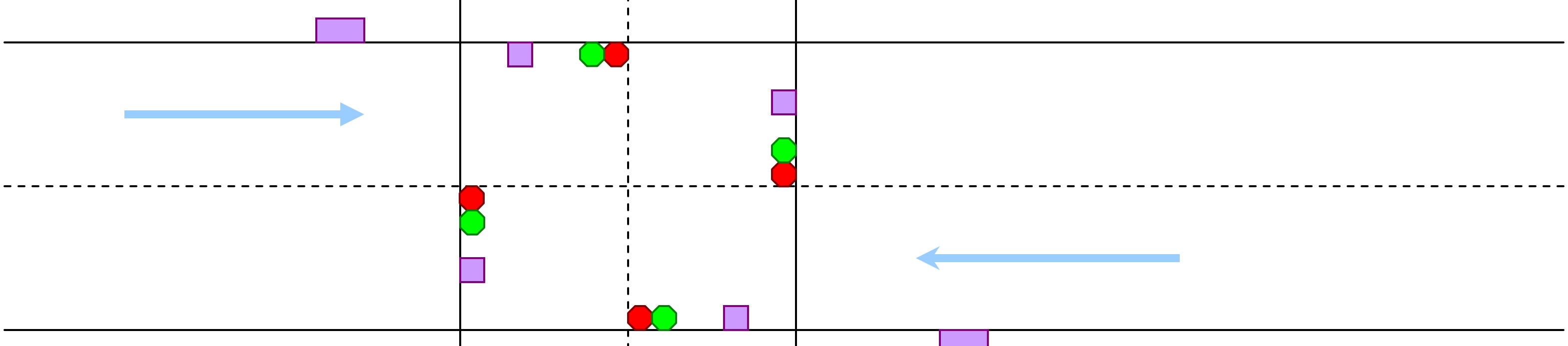
The map is an integral part of the software which forms a landmark in the automation of traffic control operations.



3.2.3 DB Design

The foundation of modern database technology is without question the relational model; it is that foundation that makes the field of science. The relational model is today the primary data model for commercial data-processing applications. It has attained its primary position because of its simplicity. It provides a very simple yet powerful way of representing data.

A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a relationship among a set of values.



Since a table is a collection of such relationships, there is a close correspondence between the concept of a table and the mathematical concept of a relation, from which the relational data model takes its name.

For TM&C following basic DB design was outlined:

3.2.3.1 User Information

We need to have a record of the user and the car that he owns. This is required when generating tickets. Thus whenever some speed violation is noted, we need to identify the individual against whom to charge the amount. In addition to the individual's name, his address and phone number need also be available.

3.2.3.2 City Information

For tracking purposes, we need to have a city map. This can further be made available to the user so that information can be displayed graphically. Thus for the very purpose, we need to have information in the knowledgebase that is so comprehensive that it can be utilised for generating a correct map of the city. Furthermore, in order to track a car, the locations the car has passed through need also be noted and correctly presented graphically when required.

3.2.3.3 Traffic Flow Information

Information regarding the flow of traffic need also be placed at the back end for later use. As the cars flow across the reader spectrum, the information flows fro clients to servers, which are placed in an web, and the information is directed to a backend knowledge base. Once this information becomes part of the DB, it can be utilized for ticketing and tracking.

This makes a triangular DB design

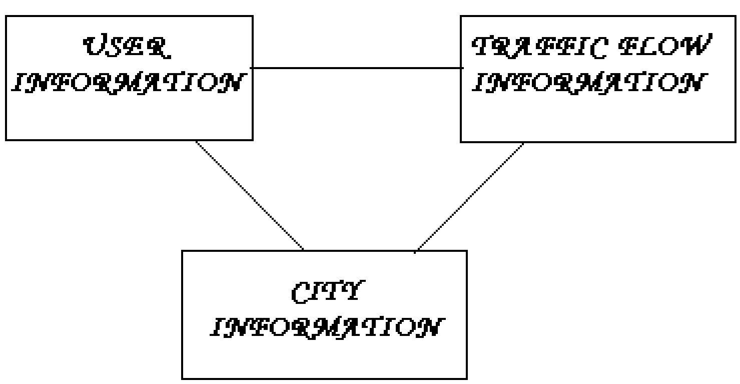


Figure 3.16: DB Design

3.2.4 Readers Deployment

Readers are deployed at the crossings and at the road sides. The readers assist us in the following tasks:

- Congestion alleviation
- Ticketing
- Tracking

Thus the readers are places at locations that are critical to the application so as to help it in the execution of the tasks mentioned above.

The reader placements are shown in purple in the figure below:

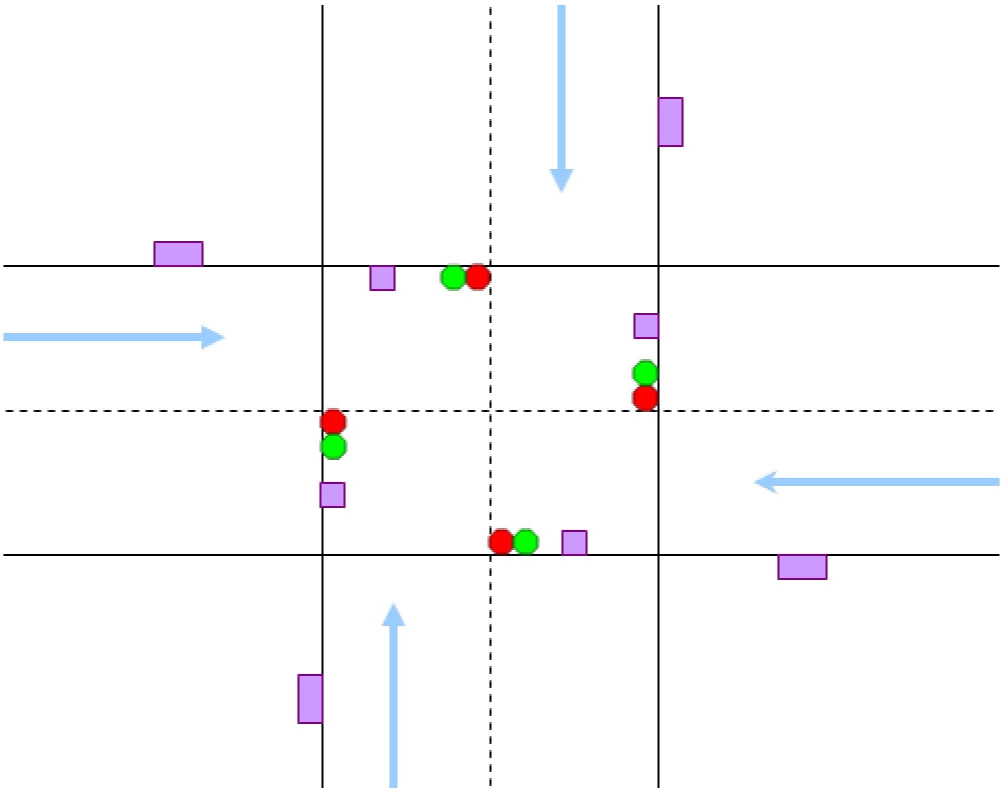


Figure 3.17: Readers Deployment

3.2.5 Distributed System Design

3.2.5.1 RMI

Distributed object applications need to

Locate remote objects

Communicate with remote objects

Load class byte codes for objects that are passed around

A distributed application built using Java RMI is made up of interfaces and classes. The interfaces define methods, and the classes implement the methods defined in the interfaces and, perhaps, define additional methods as well. In a distributed application some of the implementations are assumed to reside in different virtual machines. Objects that have methods that can be called across virtual machines are *remote objects*

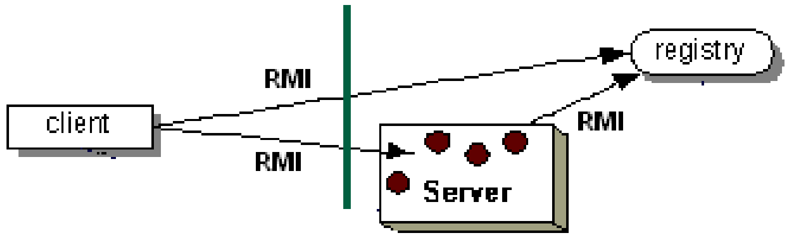


Figure 3.18: RMI Design

3.2.5.1.1 Stub

RMI passes a remote stub for a remote object
The stub acts as the local representative, or proxy, for the remote object
and basically *is*, to the caller, the remote reference
The caller invokes a method on the local stub, which is responsible for carrying out the method call on the remote object.

3.2.6 Client Side

3.2.6.1 Basic Features

3.2.6.1.1 Reader Settings

Each reader deployed has a unique identification number assigned to it, corresponding to which remote method is invoked at the server side. Thus the first step in the operation of the client side module is to correctly identify the reader connected to the system.

3.2.6.1.2 Mode Settings

Once the reader connected to the system is identified, then the mode of operation of the reader is set. This is important because the reader has to be turned on in continuous mode so as for it to start operating correctly.

3.2.6.1.3 GUI Design

Maximum efforts have been made to design the graphical user interface as simple and easy to user as possible. It provides two options to the user, reader settings, and mode settings; which are only on and off. It has a text area that allows the user to see the progress as the reader operates.

3.2.7 Central Server

3.2.7.1 Basic Features

3.2.7.1.1 DB Connection

Before the client can start to operate, connection is established with the central database at the server side. Once the server is bound and the DB connection established, the client is run. As the tagged information is read in by the reader, the data is decoded and transmitted to the main server, which makes the relevant updations in the DB.

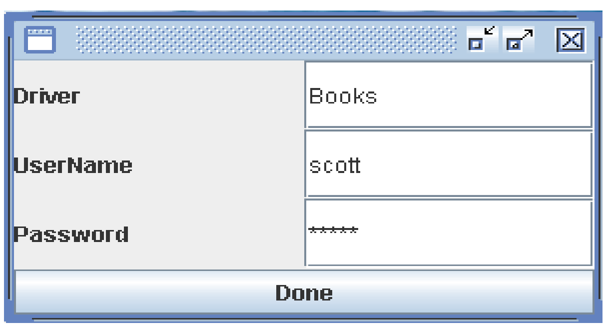


Figure 3.19: DB Login

3.2.7.1.2 Map Generation

Once the connection is established with the DB, map is generated using the information at the backend. Unless the map generation phase is carried out, ticketing and tracking operations cannot be formed. This if due to the fact that tracking is represented in pictoral form as well, and for that map is required, thus it must be generated.

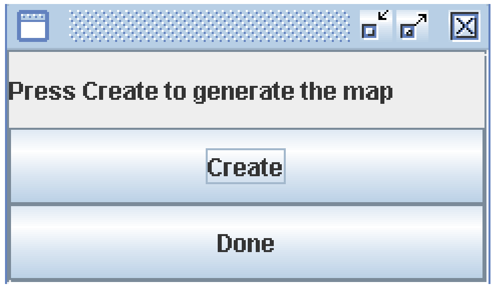


Figure 3.20: Map Generation

3.2.7.1.3 DB Updations

Once connection is established with the DB, the information flowing in from the clients is routed to the appropriate part of the DB. This information is further utilized for ticketing and tracking operations.

3.2.7.1.4 Ticketing

To date this operation of ticketing is being performed around the globe manually, where the speed of the vehicles is checked using speed-checking devices. If violated, the individual is approached by uniformed men and he is made to pay the amount as per the charge.

However, TM&C has automated this ticketing procedure. Once data flows in to the server, the speed of the moving vehicle is checked against the speed limit set for the particular road. If violated, a ticket is generated automatically that specified the speed violation, the car and the owner to whom the car belongs.

If any speed violation is noted, it saved in the form of a report as per user requirements. If however, no speed violation is noticed, it is reported to the admin as such.

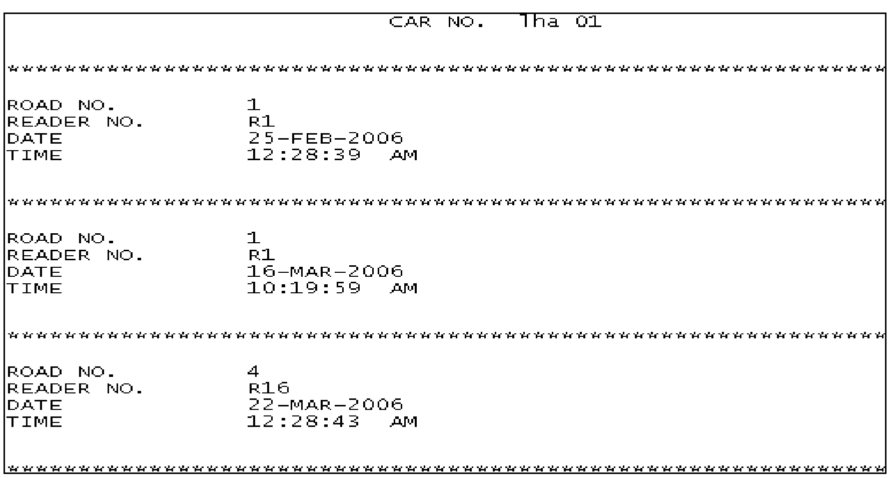
Figure 3.21: No Violation Noticed

3.2.7.1.5 Tracking

Tracking literally means ‘to know where we are’. Thus this operation involves identifying where a car has been at specific dates. This helps tnot only in the field of security but ensuring legal movement of the cars as well.

3.2.7.1.6 Report generation

In tracking operation, reports can be generated on demand. Moreover for a single entry multiple reports can be requested. The movement of the vehicle is represented in the form of locations it has passed through and saved. This is extremely useful information that can be further used to identify and monitor the movement of goods and people from one place to another.



```
CAR NO.   Tha 01
*****************************************************************
ROAD NO.          1
READER NO.        R1
DATE              25-FEB-2006
TIME              12:28:39  AM
*****************************************************************
ROAD NO.          1
READER NO.        R1
DATE              16-MAR-2006
TIME              10:19:59  AM
*****************************************************************
ROAD NO.          4
READER NO.        R16
DATE              22-MAR-2006
TIME              12:28:43  AM
*****************************************************************
```

Figure 3.22: Ticket Report

3.3 DEVELOPMENT

3.3.1 Front End

The front end of TM&C is developed in java. This gives the software the advantages of usability, extensibility and platform independence.

3.3.1.1 Server Software

The server side software comprises the following classes:

GUI
Task
Graphic
Tracking
Compute
DB_Settings
generateMap
Table_display
Compute Engine

GUI

It is the main class that sets up the graphical user interface. Creates the objects of other classes and thus calls their functions.

Graphic, Table_display, Tracking

These classes are responsible for the generation of maps. There are two types of maps being generated. One is the map generated when the server software is first run. This is the basic map of the city roads. Once it is in place, and the user tracks a particular car, a map is generated showing the locations the car has crossed, in red. The very information is also represented in tabular form as well.

DB_Settings

This class deals with the database settings so as to ensure that the knowledgebase is being updated correctly as the data flows in from the client systems. The class is further being utilized for ticketing and tracking where manipulations are made for the generation of tickets and for tracking a particular carid.

ComputeEngine, Compute Task

The sole purpose of these classes is to establish communication between the server and the client systems. Once the connection is established, data can be correctly transmitted between the server and the client systems. The communication is established based on remote method invocation where the server stub is copies into the client workstation so that they can communicate on a standard platform. The client calls the server stub located on the client machine, which in turn calls the server stub located on the server machine. They can communicate because both stubs belong to the same server. Once the information reaches the server, it can be user as and per requirements.

3.3.1.2 Client Software

The client side software comprises of following classes:

GUI
Task
SetMode
Compute
ReadData

ComputePi

GUI

This is the main class setting up the graphical user interface, and creates the objects of other classes thus in turn calling their functions.

SetMode, ReadData

This is used to make the reader operate in continuous mode. Once in continuous mode, the reader sets up an electromagnetic spectrum in circular around itself for as long as the user desires. During this duration whenever a tag passes through the spectrum, it is identified and the information relating to it is passes to the server. The mode can be set on or off.

ComputePi, Compute, Task

The sole purpose of these classes is to establish communication between the itself and the server system. Once the connection is established, data can be correctly transmitted between the client and the server. The communication is established based on remote method invocation where the server stub is copies into the client workstation so that they can communicate on a standard platform. The client calls the server stub located on the client machine, which in turn calls the server stub located on the server machine. They can communicate because both stubs belong to the same server. Thus in this way the client can pass its information to the server.

3.3.2 Back End

The backend can be in

Oracle

Access

The DB designed for the 'Traffic Monitoring & Control' application has the following relations:

RoadInfo
RoadReaderInfo
ReaderInfo
ChockInfo
TrafficInfo
CarInfo

3.3.2.1 Database Schema

RoadInfo

RoadInfo-schema = (roadno, sp_x, sp_y, ep_x, ep_y, noonreaders)

Roadno	*Sp_x*	*Sp_y*	*Ep_x*	*Ep_y*	*noofreaders*

roadno:
Every road in the network is assigned a unique number, so as differentiate one from the other.

sp_x:
The x-coordinate of the starting point of the road. Helps in map generation a at the tracking stage.

sp_y:
The y-coordinate of the starting point of the road. Helps in map generation a at the tracking stage.

ep_x:

The x-coordinate of the ending point of the road. Helps in map generation a at the tracking stage.

ep_y:

The y-coordinate of the ending point of the road. Helps in map generation a at the tracking stage.

noofreaders:

Specifies the number of readers installed on the particular road.

RoadReaderInfo

RoadReaderInfo-schema = (roadno, R1, -----, R16)

Roadno	*R!*	-	-	*R16*

roadno:

Specifies the unique identification number of the road.

R1 – R16:

A total of 16 deployed readers are considered. So we specify where a particular reader is deployed in the specified road or not.

ReaderInfo

ReaderInfo-schema = (readerno, x, y, ticket, roadno, speed_limit)

Readerno	*X*	*Y*	*Ticket*	*Roadno*	*Speed_limit*

readerno:

Each reader is assigned a unique identification number. This is so called the readerno.

x:

The x-coordinate of the location of a reader

y:

The y-coordinate of the location of the reader

ticket:

Specifies whether ticketing is valid for the specified road or not.

roadno:

Each road is assigned a unique identification number for differentiating it from other roads in the network.

speed_limit:

Specifies the speed limit for the particular road. Used at a later stage for the generation of the tickets.

ChockInfo

ChockInfo-schema = (chockno, RS, RN, RE, RW)

Chockno	*RS*	*RN*	*RE*	*RW*

chockno:

Each crossing is given a unique identification number. This is used for map generation.

RS, RN, RE, RW:

Specified the readers installed towards north, south, east, west.

TrafficInfo

TrafficInfo-schema = (readerno, carid, time_passing, date_passing, am_pm)

Readerno	*Carid*	*Time_passing*	*Date_passing*	*Am_Pm*

readerno:

Unique identification number of the reader.

carid:

The id of the car passes across the particular reader.

time_passing:

Time at which the car has passed.

date_passing:

The date at which the car has passed the reader at the specified time.

am_pm

Specifying the time as am or pm.

CarInfo

CarInfo-schema = (carid, carno, owner_name, address, phone_no)

Carid	*Carno*	*Ownwe_name*	*Address*	*Phone_no*

carid:

ID of the car.This is the id of the tag places on the car.

carnumber:

This corresponds to the car number such as IDL 01

ownername, address, phoneno:

corresponds to information specific to the owner of the car.

3.3.3 User Manual

User Manual of TM&C is attached in appendix B.

3.4 TESTING/VALIDATION

Software testing is one element of a broader topic that is often referred to *'Verification and Validation'* (V&V). Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to the different set of activities that ensures that the software that has been built is traceable to customer requirements. Boehm states this another way:

Verification: "Are we building the product Right?"

Validation: "Are we building the right product?"

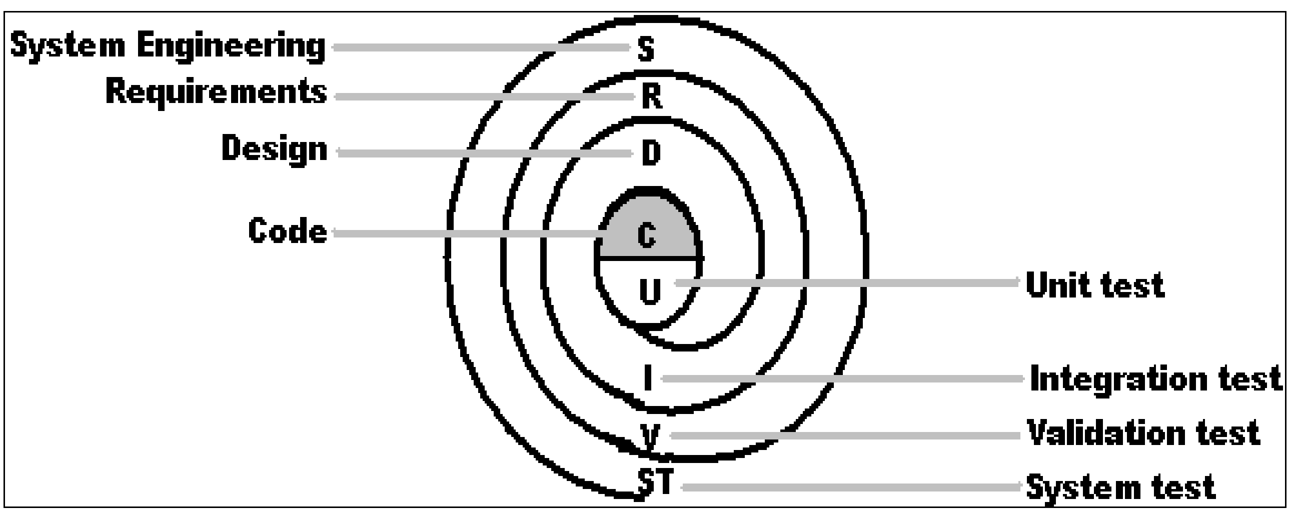


Figure 3.23: Testing Process

3.4.1 Unit Testing

Unit testing begins at the vortex of the spinal and concentrates on each unit of the software as implemented in source code. In TM&C each component as developed is individually tested so as to check for possible errors that could occur.

3.4.2 Integration Testing

In integration testing focus is on design and the construction of software architecture. We have structured the classes as per the user requirement so the extensibility of the software is guaranteed. The system is constructed and tested ensuring conformity with the basic objectives of the software testing strategy. The design is fully based on user specification.

3.4.3 Validation Testing / Black Box Testing

In this, the requirements established as part of software requirements analysis, are validated against the software that has been constructed. TM&C has been fully validated as per meeting user requirement. It provides final assurance that the software meets all functional, behavioural, and performance requirements. Black box testing techniques have been used exclusively during validation.

3.4.4 System Testing

In software testing the software and other system elements are tested as a whole. TM&C has been tested by implementing it in a real time scenario. It has been verified that all elements mesh properly and that the overall system function/performance is achieved.

4.1 ABSTRACT

The paper presents a novel idea as to how to improve the traffic congestion scenario in today's metropolitan cities around the globe. The inevitable use of RFID (Radio Frequency Identification) technology has become evident in almost all fields especially in traffic control system and the paper advocates the very idea. The system reacts in real time and address the issue of over-crowded city roads. Normally, the traffic signals have pre-programmed intervals, i.e. no significance is given to the actual traffic flow. But by the planned deployment of RFID systems coupled with the effective database design, it is now possible to control the traffic signals at runtime. Thus, synchronizing fast traffic flow and preventing unnecessary congestion.

4.1.1 Keywords

Congestion, RFID, Traffic queuing, traffic control system, AVI.

4.2 INTRODUCTION

Traffic related problems such as delays, congestion, and pollution are increasing at an alarming rate, especially congestion. Traffic congestion describes a condition in which vehicle speeds are reduced below normal, increasing drive times, and vehicle queueing occurs.

"Americans were spending over 1.6 billion hours in traffic jams each year, burning up 4 percent in fuel. 1991 projections estimated that by the year 2005, those numbers would rise to 8.1 billion hours lost and 20 percent of fuel consumed (Gose, 1993) [26]".

The use of IT could serve as the engine for acquiring, storing and distributing traffic information in such as way as to improve the congested conditions on the nation's highways while improving highway safety.

Normally, the traffic signal is programmed in a manner that it keeps on changing irrespective of the actual traffic flowing in that particular road.

In a 1996 report by Turner [28], a variety of techniques for travel time data collection were discussed, along with the advantages and disadvantages of each. These data collection techniques included electronic distance measuring instruments (DMI's), License plates matching, Cellular phone tracking, Automatic vehicle location (AVL), Automatic vehicle identification (AVI) and Video imaging. Turner specifically noted that travel time information was of particular importance for applications including congestion measurement and real-time travel information.

The research community's interest in developing reliable and accurate surveillance systems is a primary motivation for the evaluation of San Antonio's AVI system [28].

Well known, Automated Vehicle Identification (AVI) is the process and technology by which individual vehicles are uniquely identified as they pass specific points in the road network, without any action being required of the driver or wayside observer [26] [36].

AVI technology utilizes Radio Frequency Identification (RFID) to identify vehicles equipped with transponder tags. RF energy is broadcast into traffic lanes from overhead antennas [35] [37]. This energy is then reflected by the windshield-mounted AVI tags, and a modified signal containing the tag's unique ID is sent back to the antenna and transmitted to a central computer system. This process is characterized as one “reading”. Upon passing another antenna, the vehicle's tag is again identified. The central computer system processes all tag reads recorded by the AVI system and matches like tag ID's to calculate antenna-to-antenna travel times of tag-equipped vehicles [26].

The study reveals that the RFID systems can form a landmark in the field of real time traffic data collection, so we have used RFID technology in our design.

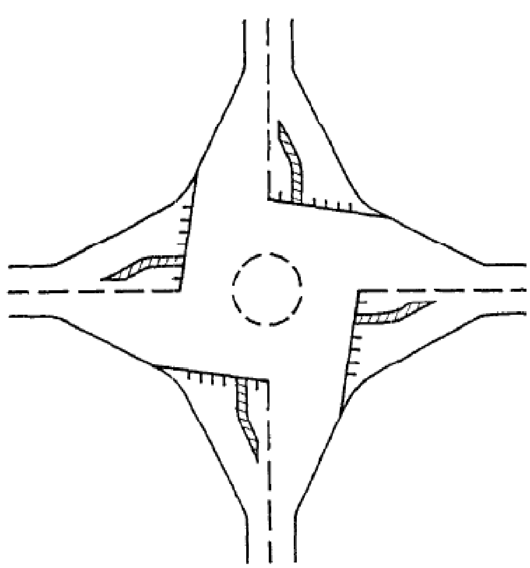
Each vehicle carries a RFID tag, tiny microchips with radio transmitters that can be attached to plastic or paper tags or cards. The chips store data about the tagged object (in our case, about the vehicle) and act as transponders that relay information to an RFID reader. In short, RFID is ideal for use in traffic control systems [29] [30].

RFID technology has played a role in London's novel traffic abatement program. U.K. officials tackle a widespread problem with innovative solutions like radio frequency identification and a charging scheme in London's city center [27].

Some congestion alleviation algorithms have already been developed, which try to reduce the congestion on busy roads. But the problem is that none is being implemented in its true sense because of the non availability of data about the traffic at runtime.

Many researchers have tried to predict the traffic patterns [31] for example Fatih Porikli and Xiaokun Li have proposed an unsupervised, low-latency traffic congestion estimation algorithm that operates on the MPEG video data. They extracted congestion features directly in the compressed domain, and employ Gaussian Mixture Hidden Markov Models (GM-HMM) to detect traffic condition. [32].

J.-S. Yang has used Neural Network with the weight estimation via the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to detect the traffic flow [33].



EXAMPLE OF A TYPICAL FLARED JUNCTION LAYOUT.

FIGURE 1

An alternative policy suggested by William is to consider changes in geometrical design, the objective being to achieve an increase in vehicular capacity, thus leading to a reduction in overall delay to vehicles. A flared approach to a set of traffic signals (or a roundabout) consists of a widened section of roadway in the proximity of the junction; the purpose is to provide extra lanes at the stop-line to enable greater numbers of vehicles to depart during the initial part of the green phase of the signals. The Williams's study is concerned with producing quantitative evidence of the advantages to be gained in providing wide flares at such junctions [39]. Example of wide flare is shown in figure 4.1.

In Saka's study [38], traffic arrival and service times at a given intersection are considered independent random variables, with known distribution. Typically, 95% confidence interval is considered adequate in selecting the appropriate distributions to fit the data. Because of the random nature of traffic arrival, they have used the Poisson distribution.

But now RFID offers a solution rather than a mere prediction that can be implemented in real time. The algorithm designed takes the data from the actual RFID readers at runtime, and controls the traffic signals so as to minimize unnecessary vehicle queuing at the crossings.

4.3 DEPLOYMENT OF READERS

RFID Reader's deployment is the key issue for intelligent traffic system design. The RFID readers should be deployed in such a way that they can deliver the accurate information about the traffic. Reader of a long range should be selected for the very wide roads, and for the roads having small width, even the readers of short range are sufficient. So range of the reader must be considered before the actual deployment of the readers on the roads [34].

Without loss of generality, let's consider the road network shown in figure 4.2. It consists of four main crossings.

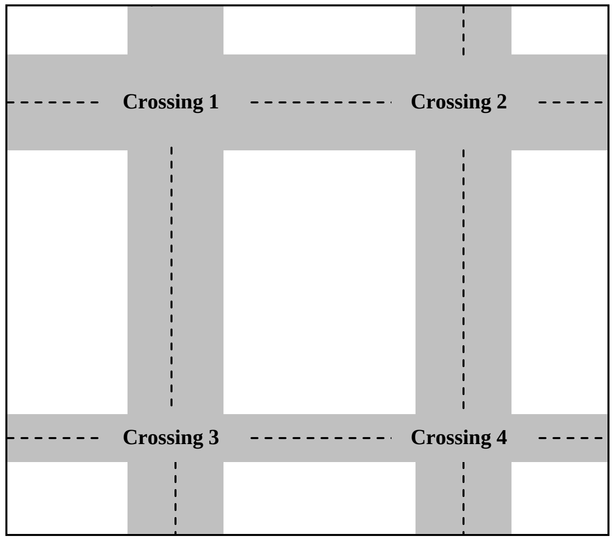


Figure 4.2: Crossings before Deployment

There are two types of readers deployed; Road readers and Crossing readers. The road readers give the static congestion information. The crossing readers give information about the dynamic traffic flow.

The readers are deployed in such a way that they cover only the area available for their particular road. The road network after the deployment of the readers is shown in figure 4.3.

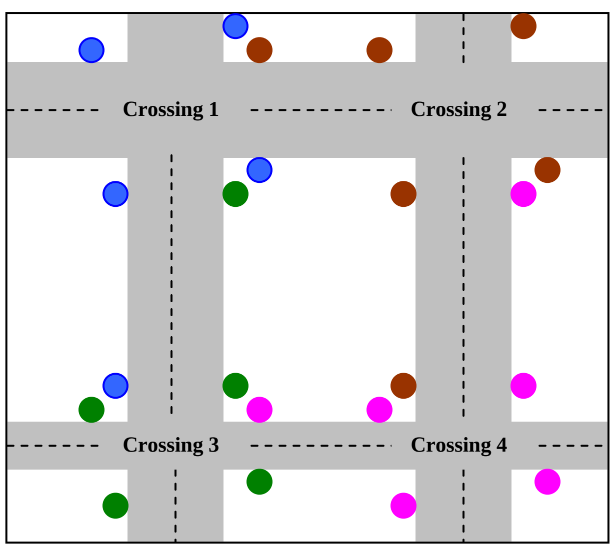


Figure 4.3: Crossings after deployment

The information available from the readers is used for decision making, i.e. controlling the traffic signals.

4.4 COLLECTION OF DATA

Once the readers are deployed then comes the process of collection of data from these readers. After regular intervals, the *'Road RFID readers'* execute the Inventory command. The inventory command returns the record of all the tags that are currently in the range of the reader. The number of tags read gives the information about the total cars waiting on a particular road.

For defining the signal interval, the 'Crossing RFID readers' are also inventoried, so as to take into consideration the traffic flow of all the connecting crossings so as to synchronize the traffic.

4.5 ALGORITHM

This information is delivered as an input to the algorithm that we have devised. The algorithm controls the traffic signals of a crossing by prioritizing the signals based on the congestion on these roads.

The algorithm is scalable and generic and can be applied to any road network. Each intersection or crossing is dependent on the data of its connected crossings so this as a whole forms a synchronous traffic control network. Moreover, the algorithm is recursive and continues processing for new traffic inflow.

The algorithm works as follows;

4.5.1 Inventory Function

First the inventory function of all the readers on the crossing is called. This will return the number of cars detected on the road at that time.

4.5.2 Setting Priority

Then after getting the results from the readers, the priorities are assigned to each road on the crossing. Priorities are numbered from 0 to 4. Where 0 is the lowest priority. Priority 0 is assigned to the road when no car is detected on that particular road. Priority 4 is assigned to the road having maximum number of cars. In case, more than one road has same number of cars, then any one of these roads is selected at random, and is assigned the priority 4. Similarly priority 3, 2, and 1 are assigned to the roads in descending order of the total number of cars read.

4.5.3 Selecting the Road

The road having the highest priority i.e. 4 is selected.

4.5.4 Calculating the Interval

The interval for which the signal remains green depends on the comparison of the congestion on all the roads of the crossing.

4.5.4.1 Inventory function for Connected Crossings

In order to define the signal interval, the cars headed towards the particular signal should be taken into account along with the waiting cars. For this the cars must be inventoried, which enter the particular road and will be a part of the queue on a specific crossing.

4.5.4.2 Time required

If the inventoried traffic flow can reach the crossing within the time the signal remains green, then the priorities are assigned to inflow of traffic towards the crossing.

Average speed = V

Distance b/w crossing s= S

then

Time Required = S / V (eq 4.1)

4.5.4.3 Calculating actual signal interval

Incase a signal is green and the incoming traffic has been assigned the top two priorities, only then the signal time is updated. The total congestion will now define the signal interval.

If the number of cars on the road having highest priority, is greater than or equal to twice the number of cars on the road having second highest priority, then the green signal remains on for maximum interval i.e. 72 seconds. The condition remains the same, even though the traffic heading towards the crossing can reach it within time, as the signal interval cannot be stretched beyond 72 sec.

If the number of the cars on the road having highest priority are less than or equal to 5 then the green signal remains on for minimum time i.e. 30 seconds. However, considering the incoming traffic towards the crossing, if the flow can reach the signal within time and it has one of the top two priorities, then the signal interval is updated to medium i.e. the signal remains on for 60 sec.

Otherwise the signal remains on for medium time i.e. 60 seconds. However, considering the incoming traffic towards the crossing, if the flow can reach the signal within time and it has one of the top two priorities, then the signal interval is updated to maximum i.e. the signal remains on for 72 sec.

If the number of cars on a particular road on the crossing is found to be zero, then priority0 is assigned to that road, and the green signal is skipped in such a case.

4.5.5 Turn on the Signal

After calculation of the interval of the signal, the green signal of the selected road is turned on, and traffic is allowed to flow.

After that same procedure is applied again for the remaining three roads. The inventory function of the remaining roads is called and priorities are assigned to the roads. Then the traffic signal of the road having highest priority is turned green for newly calculated interval. After regular interval the inventory function is called for remaining two roads, their priorities are set and the one with highest priority is allowed to move. At the end the traffic on the road, that is left, is allowed to move. After this, whole cycle is repeated over and over.

4.5.6 Algorithm

```
(If Crossing has four roads (a, b, c, d))
{
  While (true)
   {
     Get Inventory of roads a, b, c and d;
     Set Priorities of the roads a, b, c and d;
     Pick the road having highest priority;
     Calculate the interval for green signal;
             Determine inflow of traffic;
             Recalculate the interval;
     Turn on the green signal;
```

```
 // if 'a' is the road having highest priority then
// after the interval available for road 'a'
//expires and signal becomes red then

    Get Inventory of roads b, c and d;
    Set Priorities of the roads b, c and d;
    Pick the road having highest priority;
    Calculate the interval for green signal;
            Determine inflow of traffic;
           Recalculate the interval;
    Turn on the green signal;

 // if 'b' is the road having highest priority then
// after the interval available for road 'b'
//expires and signal becomes red then

    Get Inventory of roads c and d;
    Set Priorities of the roads c and d;
    Pick the road having highest priority;
    Calculate the interval for green signal;
           Determine inflow of traffic;
           Recalculate the interval;
    Turn on the green signal;

 // if 'c' is the road having highest priority then
// after the interval available for road 'c'
//expires and signal becomes red then

     Pick the road d;
     Calculate the interval for green signal;
           Determine inflow of traffic;
           Recalculate the interval;
     Turn on the green signal;

  //after the interval available for road 'd'
//expires and signal becomes red then
  }
}
(Else If Crossing has three roads (a, b, c))
```

```
{
 While (true)
  {
   Get Inventory of roads a, b, and c;
   Set Priorities of the roads a, b, and c;
   Pick the road having highest priority;
   Calculate the interval for green signal;
           Determine inflow of traffic;
           Recalculate the interval;
   Turn on the green signal;

  // if 'a' is the road having highest priority then
  // after the interval available for road 'a'
  //expires and signal becomes red then

    Get Inventory of roads b and  c;
    Set Priorities of the roads b and c;
    Pick the road having highest priority;
    Calculate the interval for green signal;
           Determine inflow of traffic;
           Recalculate the interval;
    Turn on the green signal;

  // if 'b' is the road having highest priority then
  // after the interval available for road 'b'
  //expires and signal becomes red then

      Pick the road c;
     Calculate the interval for green signal;
           Determine inflow of traffic;
           Recalculate the interval;
     Turn on the green signal;

   //after the interval available for road 'c'
  //expires and signal becomes red then
  }
}
```

4.6 COMPARISON

We have designed a simulator which simulates the traffic flow. The simulator is developed in Visual Basic.

The simulator has two parts. First part simulates the traffic controlled by the pre-programmed traffic signals; Second part simulates the traffic controlled by the traffic signals which are governed by the algorithm that we have devised.

Clearly, the waiting times for traffic in both cases differ in large extent. The part1 shows the number of congestion that occurs due to the traffic signals.

The part2 shows the case when algorithm is applied to the traffic flow. As RFID system is placed, so the cars on the road having more traffic are allowed to precede first, thus alleviating congestion.

The cars on roads having less traffic are not starved for their turns, but they are also given the proper share of time. Thus, in this way, the signals are controlled based on the information about the actual traffic flowing on the roads.

We have compared the average waiting time for the cars in both the cases, i.e. in part1 and in part2. The results are shown in figure 4.4. The results we have found shows that the waiting times of cars reduce enormously due to the use of RFID system.

This shows that once the RFID system delivers the correct information about the congestion on the roads and the when the signals are manipulated to reduce the congestion levels, then the average waiting time of the vehicles reduces.

The simulator screenshots are attached in appendix C.

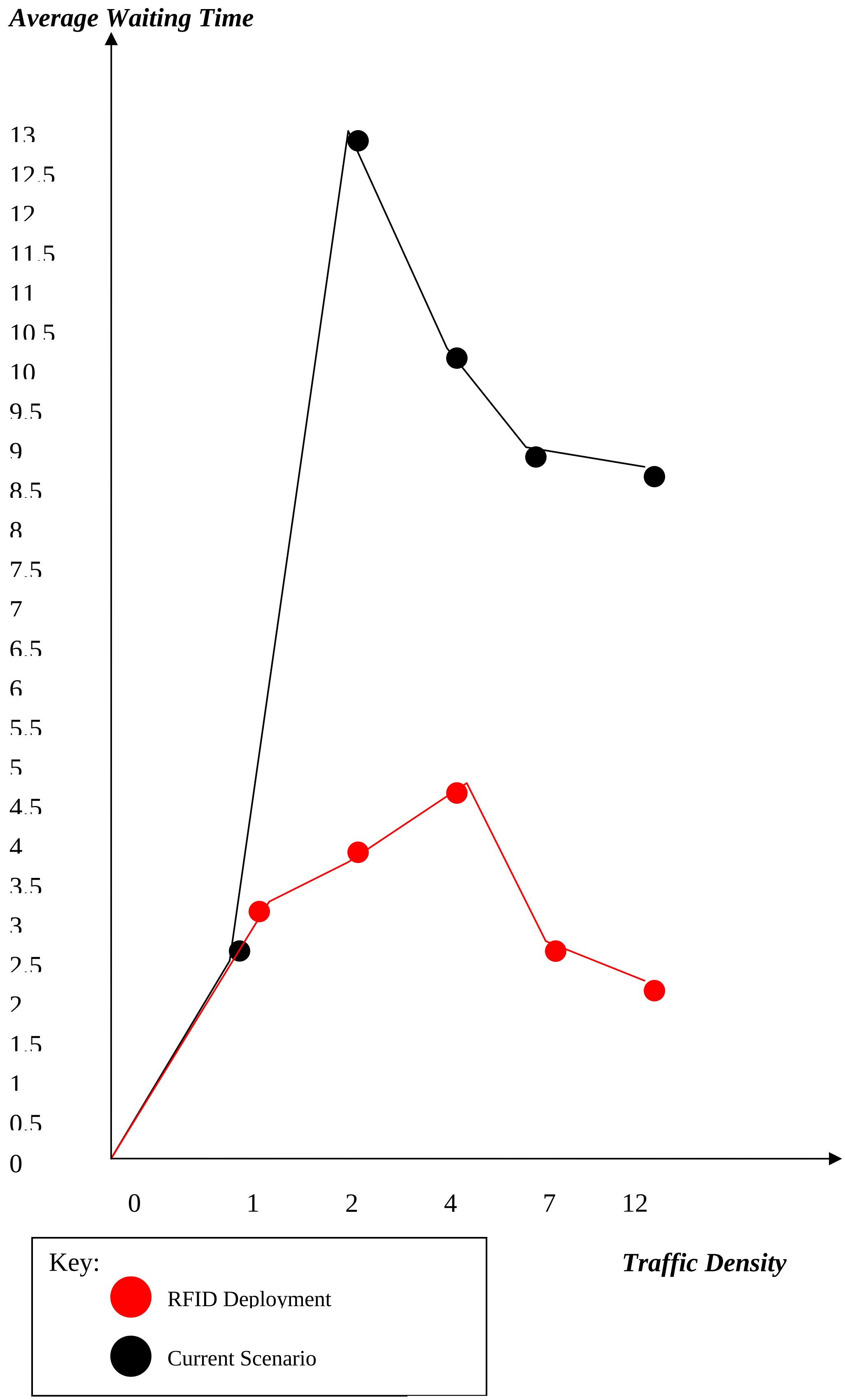


Figure 4.4: Comparative Study

4.7 CONCLUSION

Thus, the commonly occurring problem of congestion can be alleviated by the use of RFID technology. The algorithm that we have devised can form a landmark in the field of road traffic engineering as it opens the opportunities of having vehicle queuing free highways.

5.1 FINALE

P&P RFID is designed for managing the immediate and future challenges of RFID event data management. Its architecture and performance capabilities lend to many advantages in an implementation looking to adapt to the world of RFID. Many applications used today to manage transactions in a particular field (supply-chain, healthcare, transportation, etc.) are facing the challenge of how to adapt to the changes RFID brings. This covers both the type of data and the types of transactions. P&P RFID mitigates the immediate need for change to these applications by:

> Acting as a buffer from the raw RFID event stream and passing along event data that has been transformed, through pipeline event processors, into a standard transaction message that the target application can handle with its existing interfaces.
>
> Acting as a repository for the wealth of new RFID event data. Today, the end user business application may not have processes or transactions that can take advantage of the additional depth that P&P RFID can serve up from its event repository. Over time as these new or improved processes emerge, P&P RFID can provide the data and visibility without burdening the business application with new data stores.
>
> Providing a single source for event data to all applications. Using different query pipelines for different application requests, RFID event data can be retrieved from P&P RFID in the form, and according to the specific needs of, the different applications in an implementation (inventory vs. logistics vs. billing).

On the software side, the challenges to RFID deployment include getting the right data onto the tags in the first place, and then leveraging the data the tags generate as they navigate across multiple enterprise systems. “Software is a huge issue,” AMR Research’s Romanow says. “Everyone’s underestimating it.”

Once the tags are loaded with data, the real software challenges begin, starting with managing the reader infrastructure. “There are no consistent mechanisms to manage them, check their health, do software upgrades, and turn them on and off,” says Javed Sikander, Microsoft’s program manager for industry solutions enablement. “There’s no standard interface across all the readers.”

Next comes the middleware challenge of managing how the readers filter data coming from the tags. “If I put a tag within the read field of an antenna, the reader reads it a couple hundred times a second,” VeriSign’s Brendsel explains. “You need a middleware layer which knows that the first time it sees it, it’s an event, and the next 10,000 times its just garbage.”

Once the data has been acquired and filtered, the work of feeding it to the appropriate applications begins. Most companies have multiple applications that need access to the RFID data, notes IBM’s Gilliam, including warehouse management systems, inbound supply chain systems, planning systems, order management systems, and data warehouse and analytics systems. “You’ve got an enormous amount of data that has to be moved around the network among applications,” Gilliam says.

Hutchinson SAYS “I think people are coming to the realization that they’re going to be looking at communication to their back-office systems and middleware challenges sooner than they thought.”

Mentioning the different views, it is a privilege to state that P&P RFID is one such middleware software that allows duplex communication between the reader and the application, moreover allowing the applications in the ERP to directly access the hardware through the P&P. P&P is scaleable, thus allowing the organizations to incorporate new developments conveniently.

TM&C was an effort to automate the traffic in the metropolitan cities of Pakistan so as to at least be one of the very few ones to initiate cooperation between the industry and the educational sector. ‘Traffic Monitoring and Control’ is software that has immense extensibility prospects. Thus it has been developed with the very idea that tomorrow the needs would be much more than what we anticipate today. Thus it has the potential of being extended as new requirements arise.

The world of today is wakening to the enormous bounties of *science* and *technology*. Thus nations would have to keep pace with the ever growing need to introduce technology to the masses so as to stay in the mainstream of developed nations.

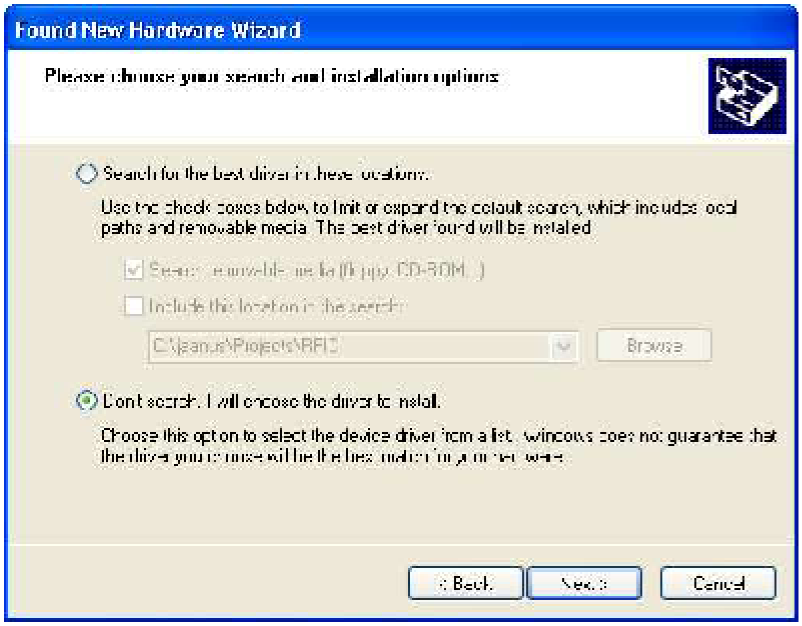
APPENDIX A

USER MANUAL OF P&P RFID

Reader Installations

Plug the reader device into a free USB port. The “Found New Hardware” dialog will appear. Below is described install procedure for Windows XP.

Choose “Install from a list or specific location (Advanced)” and click “Next ”.



Choose “Don’ t search. I will choose the driver to install” and click “Next”.

Now click “Have Disk … ” and select the folder that contains the driver file (RFIDReader.ini).Then click “Next”.

Click “Continue Anyway”. This will install the driver.

Now click “Finish”. Your computer should now have one more COM port – virtual COM port “RightTag Series 3000 RFID Reader (13.56 MHz)” .Communication between PC and reader can be done now exactly as it would be done through real COM port using the COM reader.

Java Installations

For running the software, you require a jdk1.5 installed in your computer. One extra package javax.comm is used for port communication, that separate package needs to be added manually. CD contains a folder name as package installations. Go to that folder and do the following steps.

1. Copy `win32com.dll` to your <JDK>\bin directory.

   ```
   C:\>copy c:\win32com.dll to c:\jdk1.5\bin
   ```
2. Copy `comm.jar` to your <JDK>\lib directory.

   ```
   C:\>copy c:\comm.jar c:\jdk1.5\lib
   ```
3. Copy `javax.comm.properties` to your <JDK>\lib directory.

   ```
   C:\>copy c:\javax.comm.properties c:\jdk1.5\lib
   ```

   The `javax.comm.properties` file must be installed. If it is not, no ports will be found by the system.

4. Add `comm.jar` to your classpath (do not do this step for a JRE installation).
   - If you don't have a classpath defined:

     ```
     C:\>set CLASSPATH=c:\jdk1.5\lib\comm.jar
     ```
   - If you already have a classpath defined:

     ```
     C:\>set CLASSPATH=c:\jdk1.5\lib\comm.jar;%classpath%
     ```

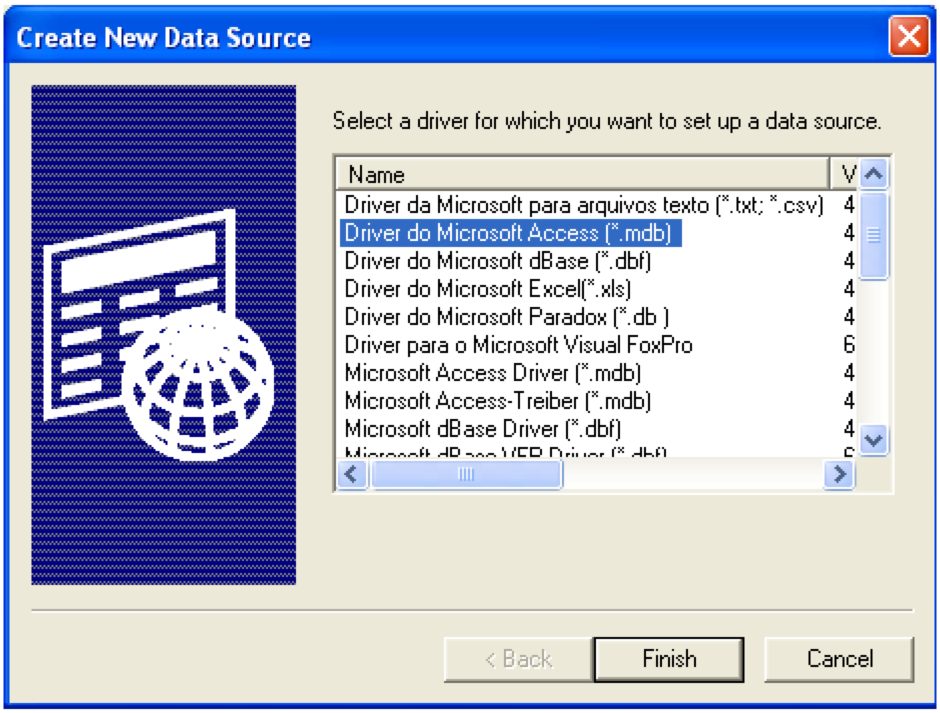
Following these steps will add javax.comm package to your jdk.

The program is tested using JCreator Pro version.

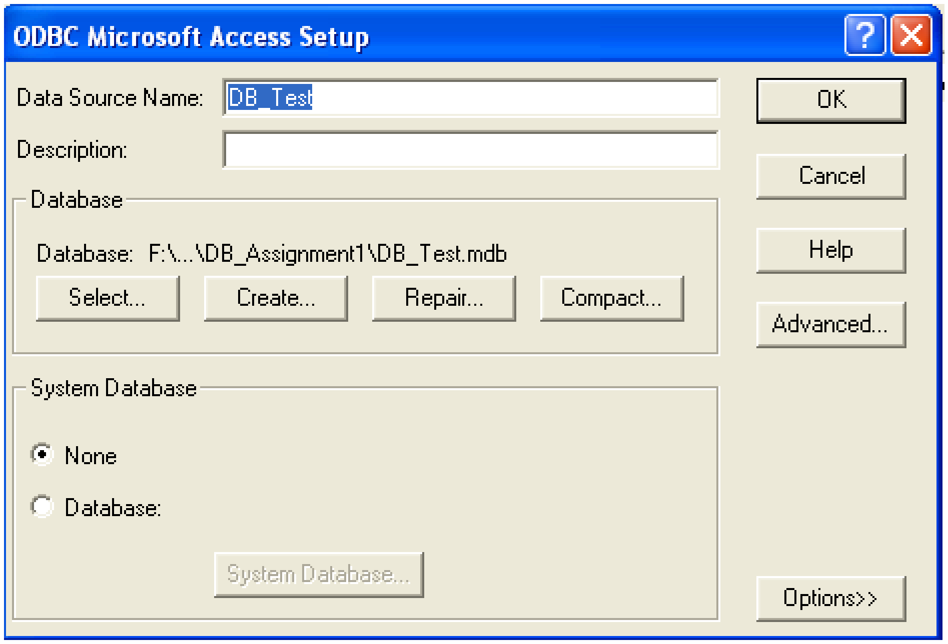
Database Driver Settings

Now add the database driver. Go to the control panel. Then select the administrative tools. Then go to Data Sources.

Now, press the add button. Select the option and press finish.



Give the name of the database and press the select button and select the location of the database. Press the OK button.



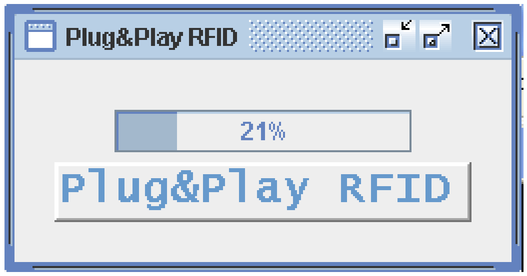
Press OK once again and this will load the Database driver into your system.

Running Software for Read and Write

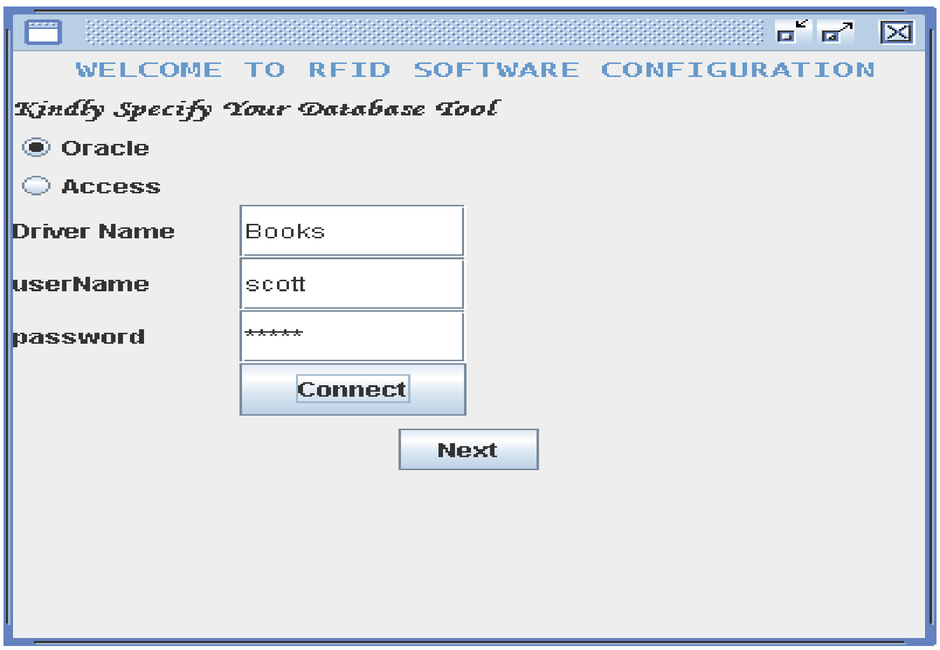
1. CD contains the jar file plugandplayrfid.jar.
2. Run the Command Prompt (DOS).
3. Include the above mentioned file in the classpath as follows(assuming the CD drive is the F drive in your system)

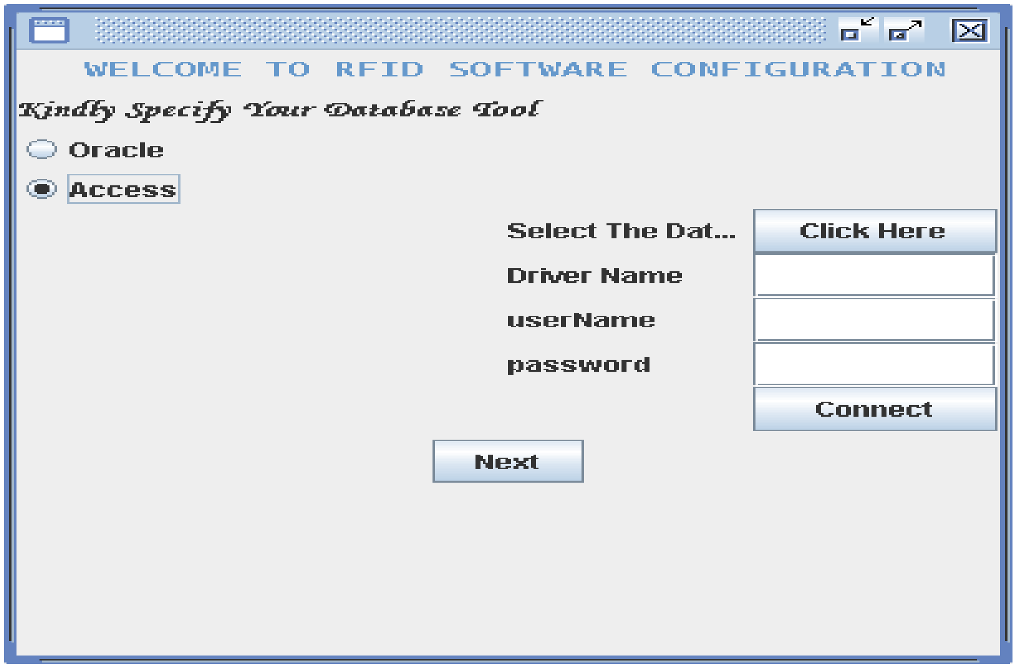
   Set classpath = %classpath%;f:\plugandplayrfid.jar;
4. To run the application, enter the following command in the Command Prompt

   Java GUI_Frame
5. This will display the following window:

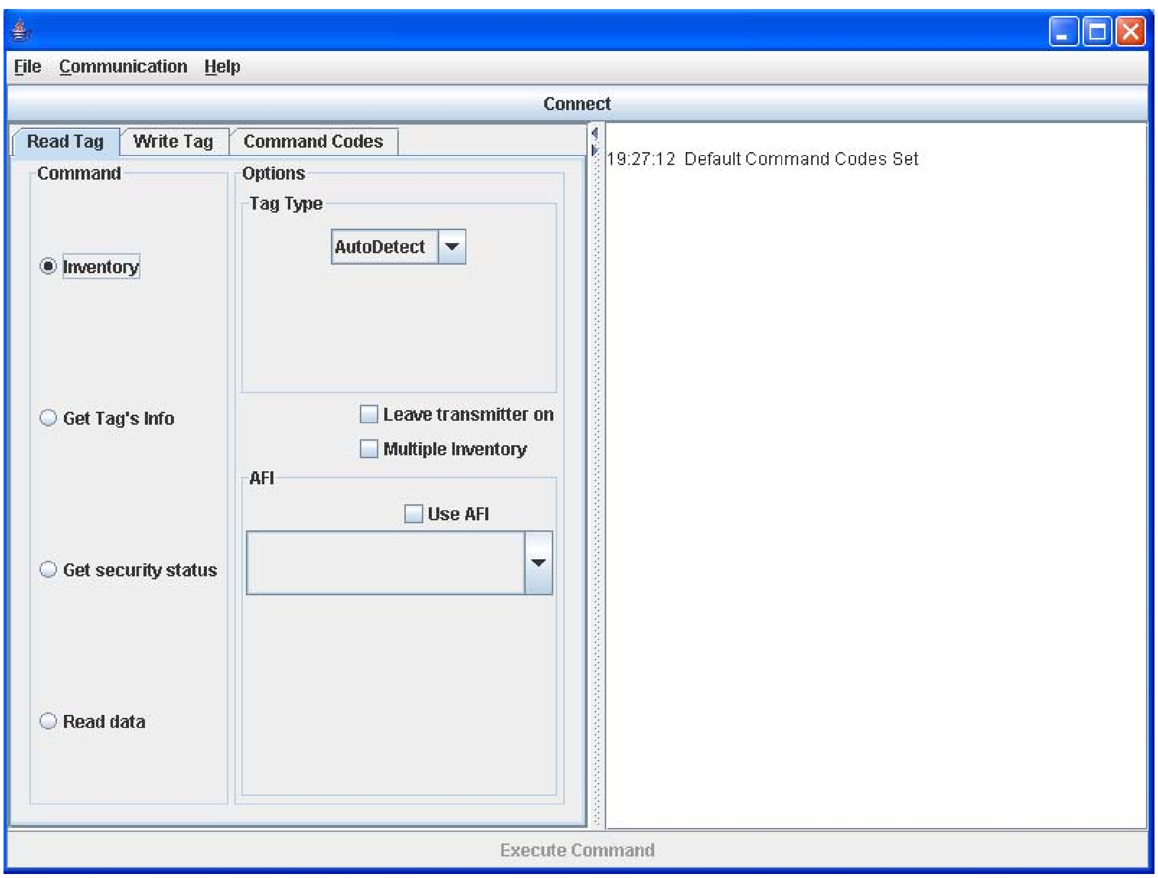


6. Now, do the database configuration. The following window will be visible to you. You are given with two options. The database can be
   - Oracle Database
   - Access Database

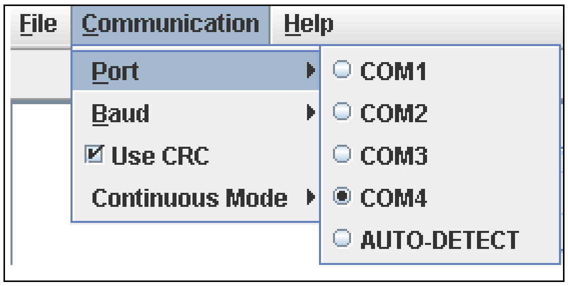


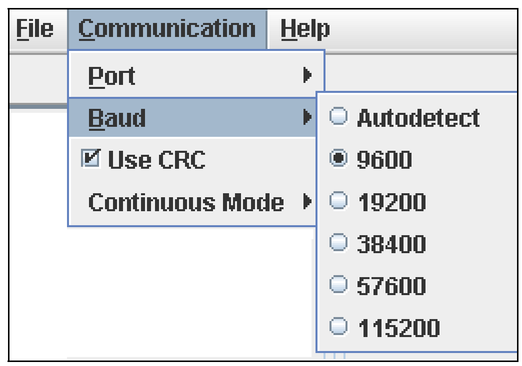


7. Once connection is established, the following window will be displayed.

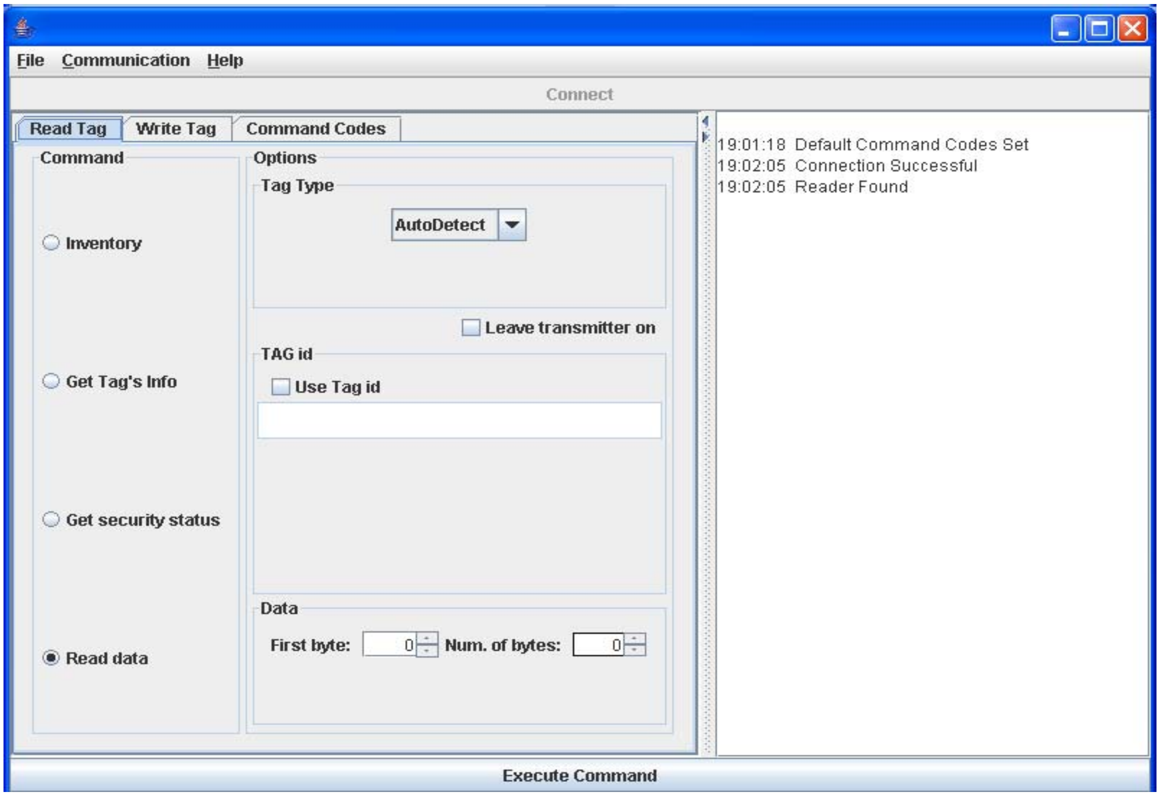


8. Firstly set the connection with the RFID system. You can just click the ‘connect’ button and software will automatically detect the RFID reader connected to your system, and will establish the connection with the reader. Or you can manually select the port and baud rate.



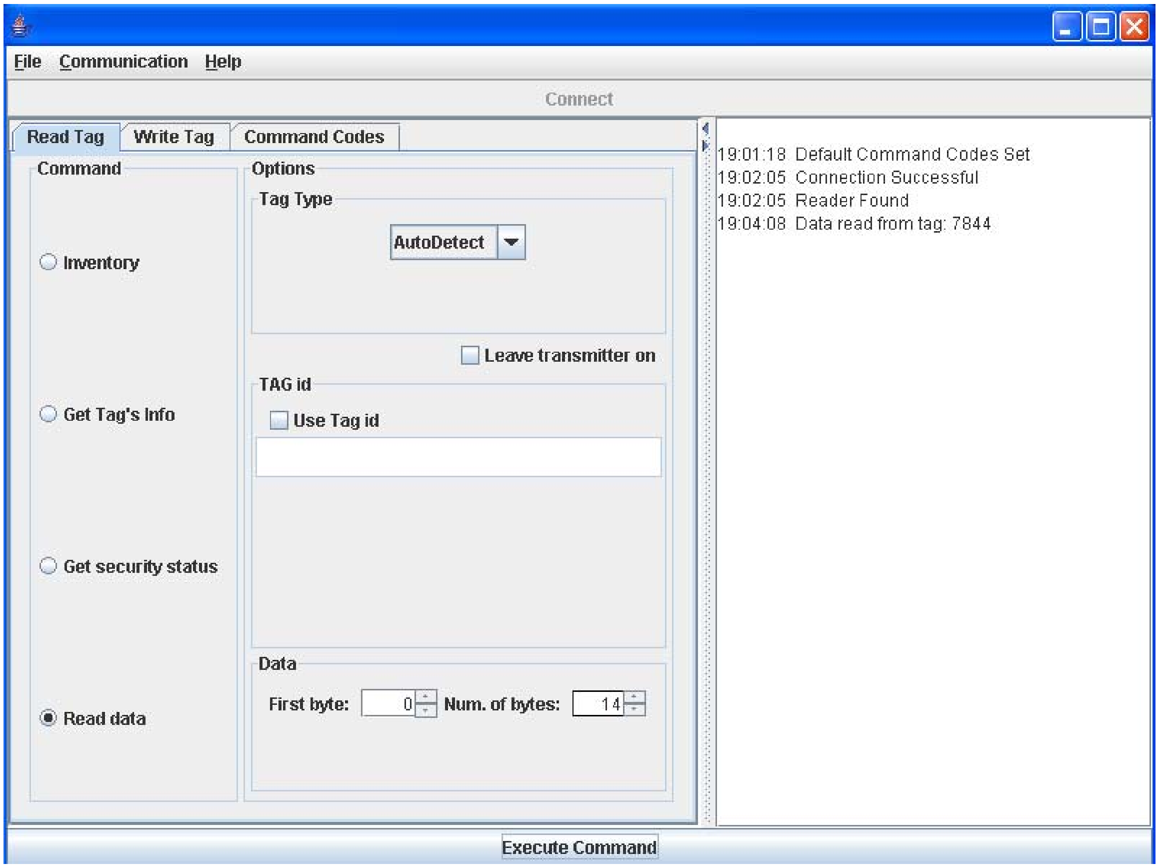


Once the port and baud settings are in place, press the ‘Connect’ button. The connection status will be displayed in the output area.



9. Once the connection is established, now you are ready to perform operations on the Tag and on the data that it carries.

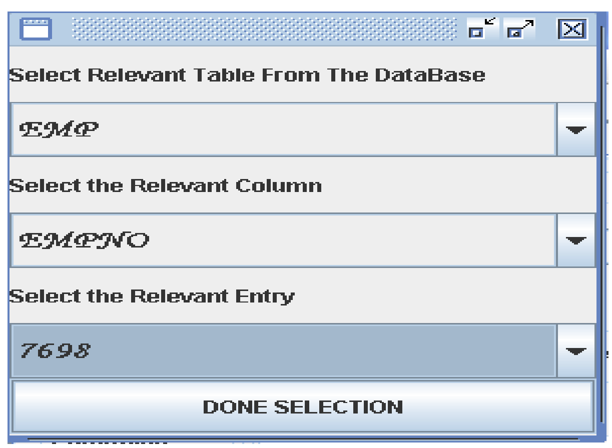
10. To execute the read command, set the option parameters and press the execution button. The command will be executed and the data read in will be displayed.



11. For writing enable the write Tab, select the write command, and set the appropriate 'Option' settings. Two schemes are provided for data writing:

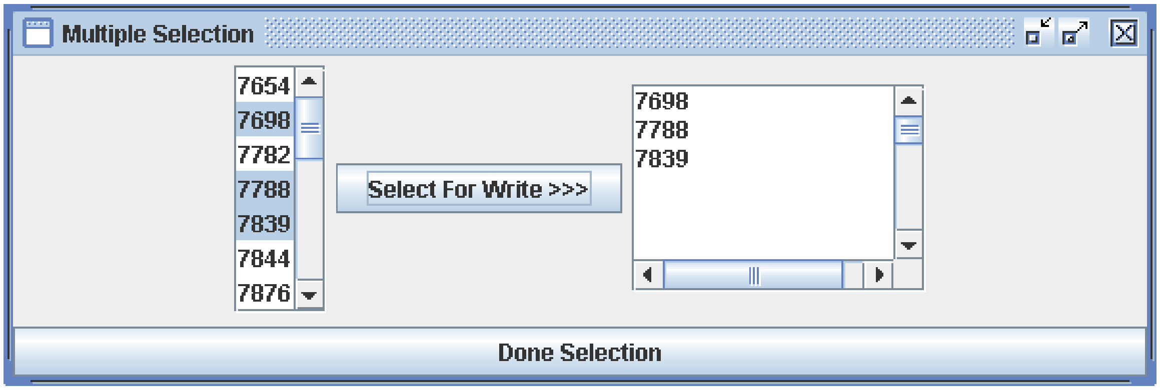
➢ **Single tag**

In this, you select a single entry which is then written on the tag.

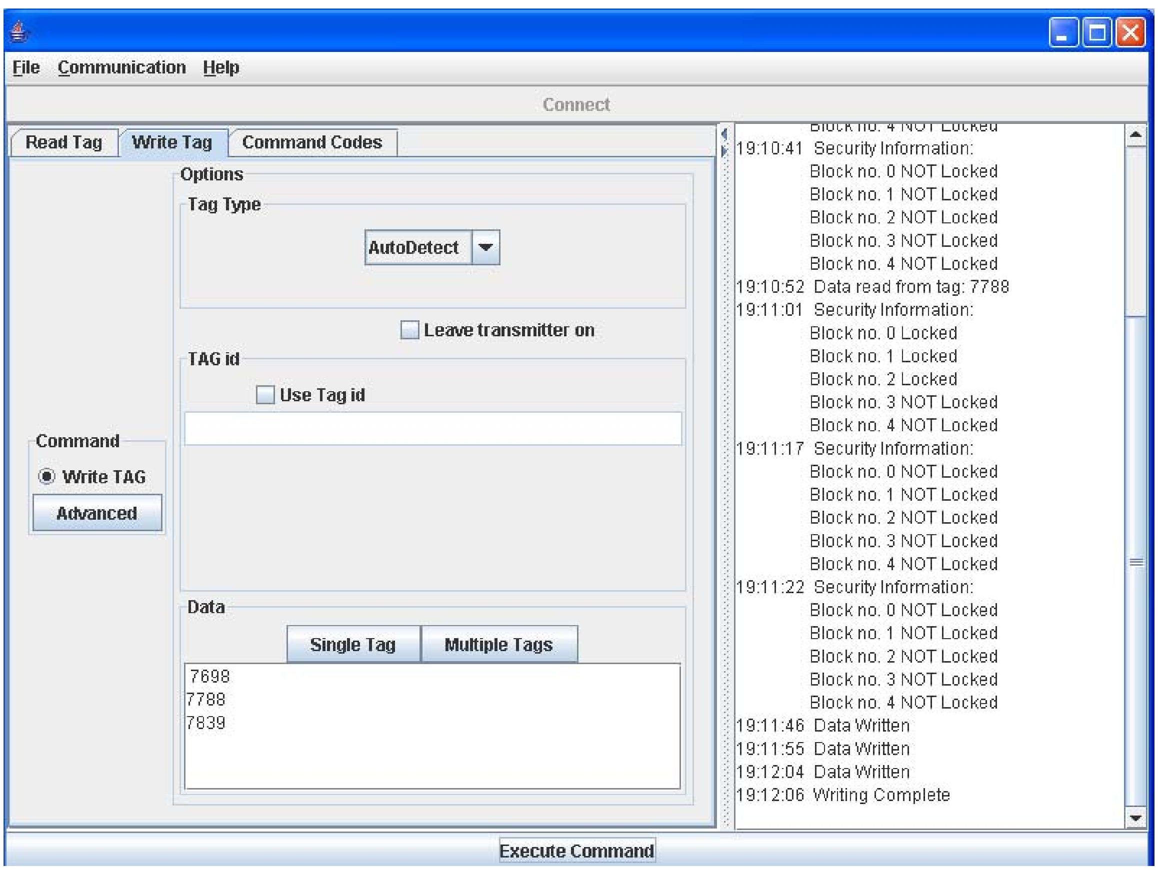


- **Multiple Tag**

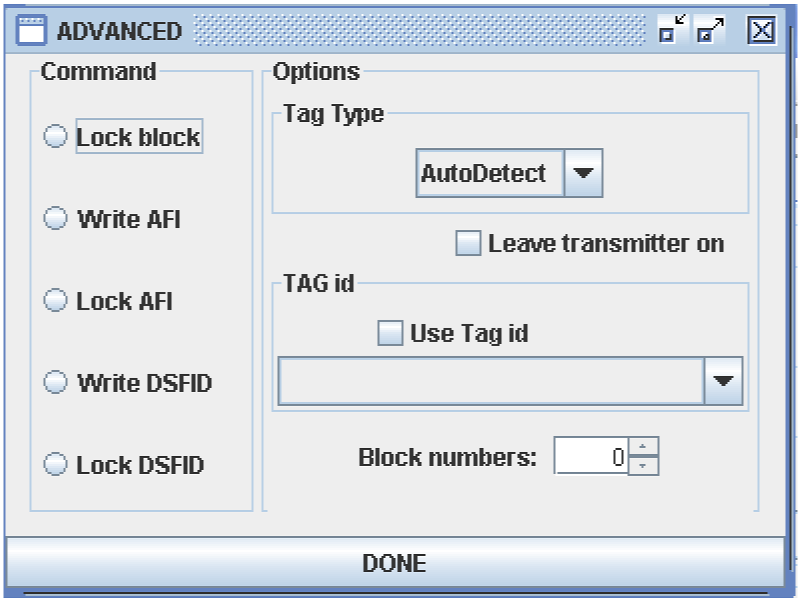
Multiple Values are selected simultaneously, each value is then written on individual tag sequentially with the click of the Execute button.



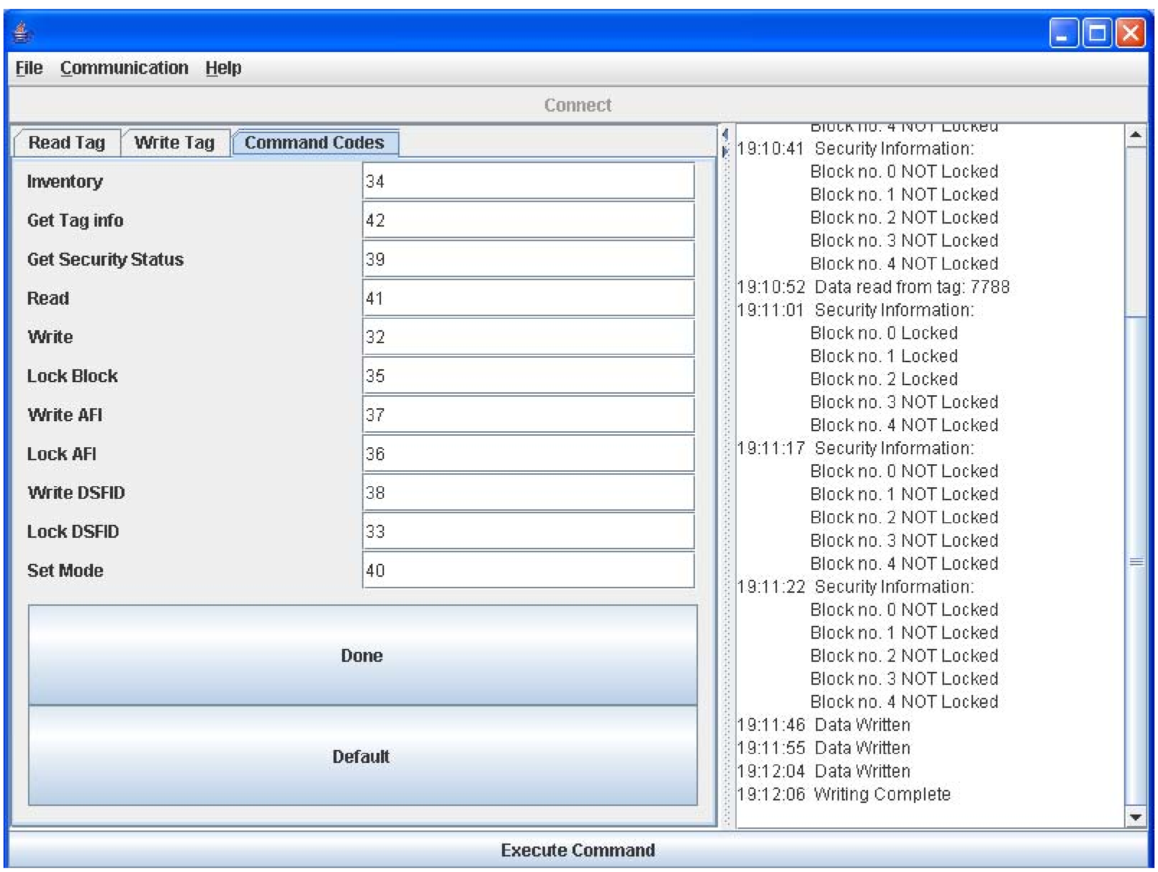
Make sure the tag is present within the reader field or else the reader will respond with error. Once the tag is written appropriate message is displayed in the GUI.



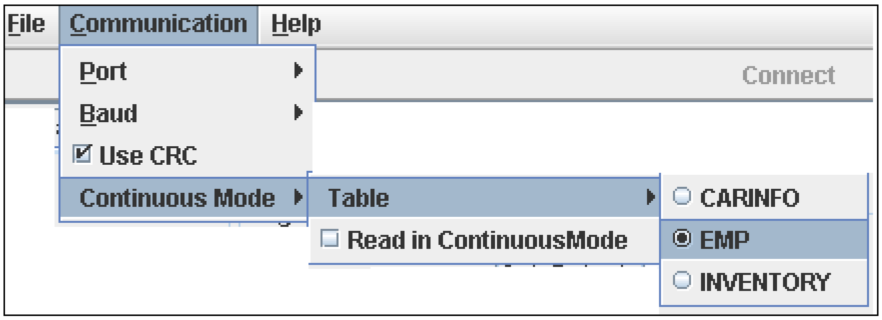
12. Advanced command options are provided to perform advanced operations on the tag information.

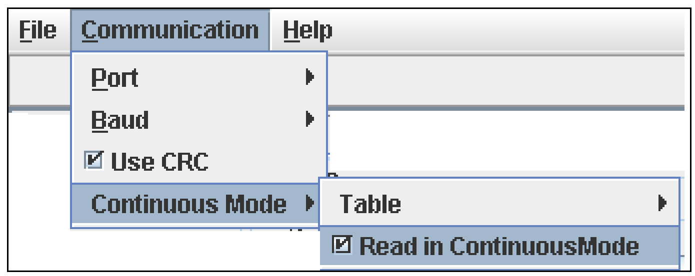


13. In order, to connect to a different reader, change the command codes by moving to the command codes tab.

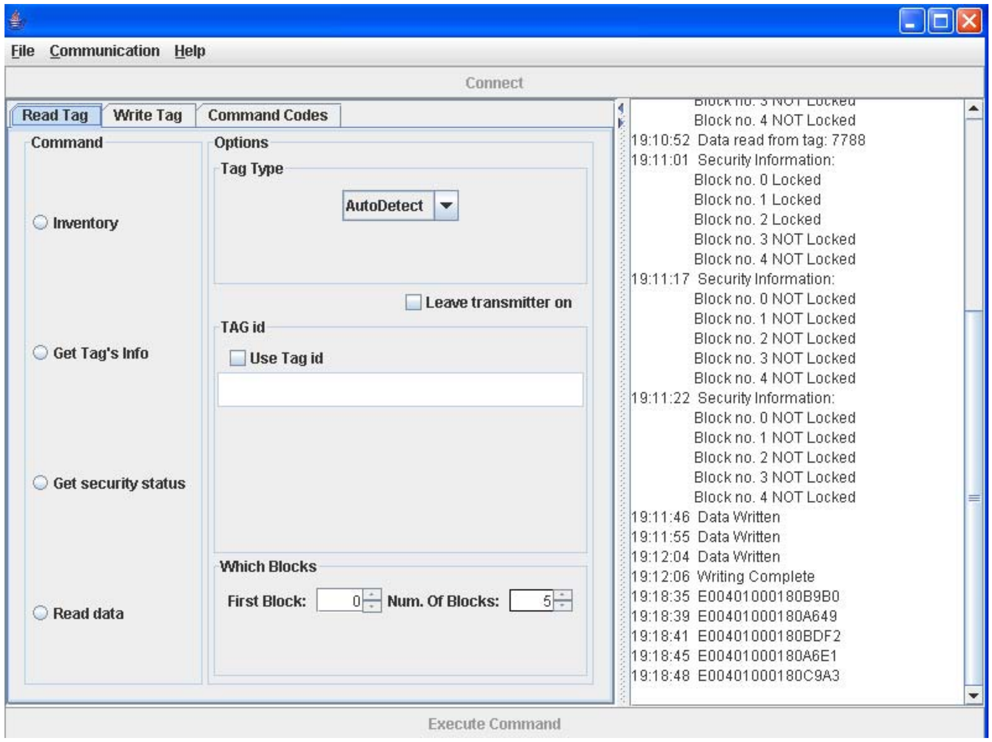


14. For changing the read mode of the reader from on click mode to the continuous mode, go to the communication menu, select continuous mode, select particular table for which the tags are meant for, then turn on the continuous mode by clicking on the checkbox.

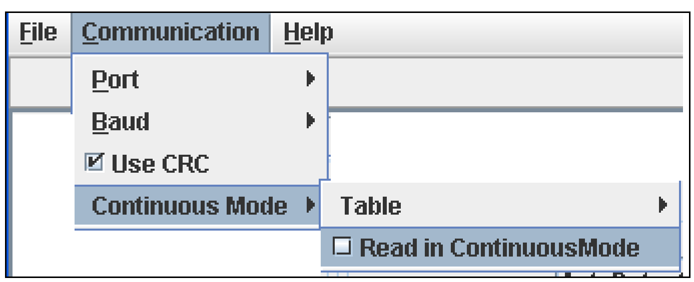




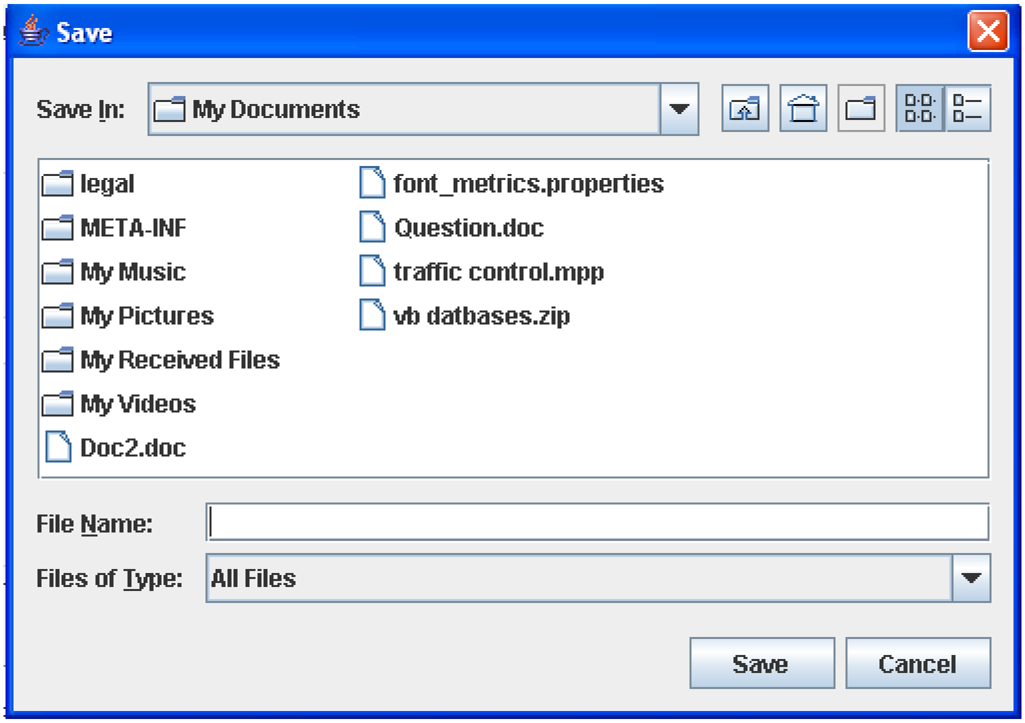
When the reader is set to the continuous mode, it will read the tags passing through its field automatically and update the database at the back end. The ID's of some of the tags that are read is shown below.



15. When the reader is in continuous mode operation, then other commands can not be executed. To turn off the continuous mode, go to communication menu and in continuous mode menu, uncheck the ‘continuous mode on’ check box.



16. For saving the all the records of processing, go to ‘File’ menu and click ‘SAVE’ option, a dialog box will appear, select the location and save the record.



17. For exiting, go to 'File' menu and click on 'EXIT' option.

TROUBLE SHOOTING

1. **Problem:** Reader seems to do nothing (no sound, no light).

Check if the USB or COM cable is connected properly. In case you have the COM version, ensure that there is power supply +5V a pin 1 of COM connector.

2. **Problem:** Reader always makes the "fail" beep and blinks the red light.

Your tag may be of wrong type or damaged. Try to check if the reader was meant for those tags. Also try some more tags.

3. **Problem:** Only garbage appears into the GUI window during read operation.

Probably he COM port baud rate was wrong. The possible values the reader can be configured: 9600, 19200, 38400, 57600, 115200. Try to change the baud rate.

4. **Problem:** ‘NO Reader Found’ error occurs each time the connect button is pressed.

   Try connecting to a different COM port in the communication menu.

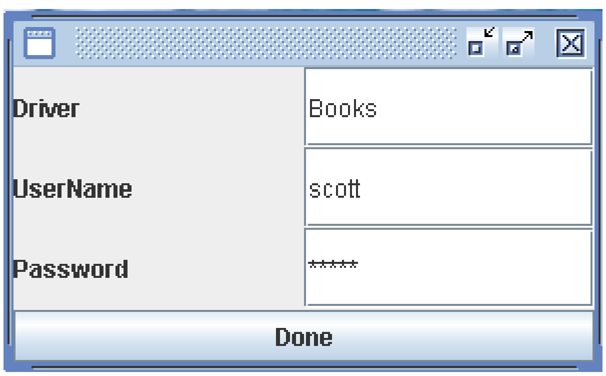
APPENDIX B

User Manual Of TM&C

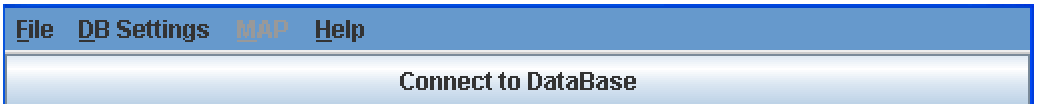
Firstly the server has to be bound, DB settings be in place, map generated and then the client software is run. Kindly follow the sequence below for correct operation of TM&C.

Running Server Software

1. Run the Command Prompt (DOS)
2. Set the path for where the Application is placed
3. start the registry by typig the following command in the command prompt
   …….>start rmiregistry
4. A new Command prompt window appears. Minimize the new window
5. Now type the following command in the DOS prompt
   …….>java –Djava.security.policy=java.policy ComputeEngine
6. The server is bound and the server software runs.
7. Click on *DB Settings* and then on *Settings*
8. Enter the driver name, username and password and press the *Done* button

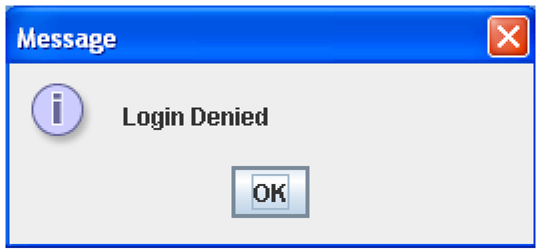


9. The window disappears. Now click the *Connect to Database* button below the menu bar



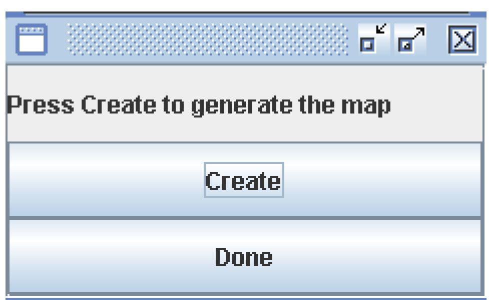
10. If the connection is successful, the MAP menu is enabled

11. Otherwise if connection is unsuccessful, connection error is displayed

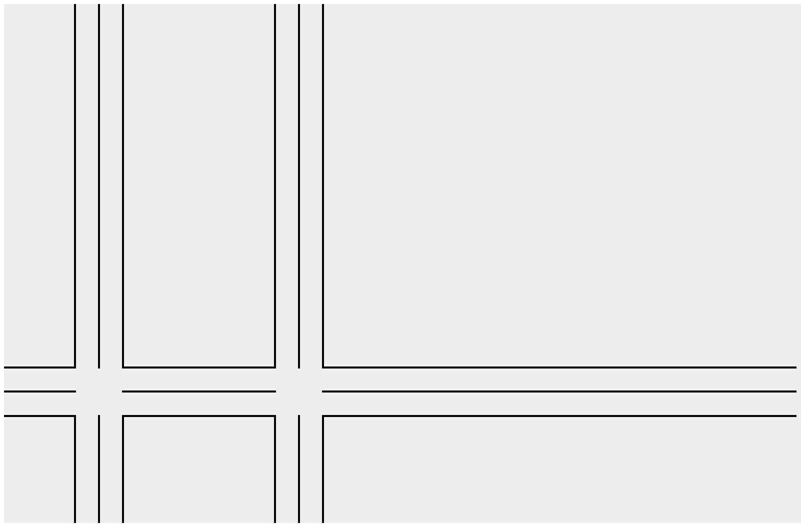


12. Now the Client Software can be run
13. Click *MAP* and then *Generate Map*

14. Press the *Create* button and then *Done*

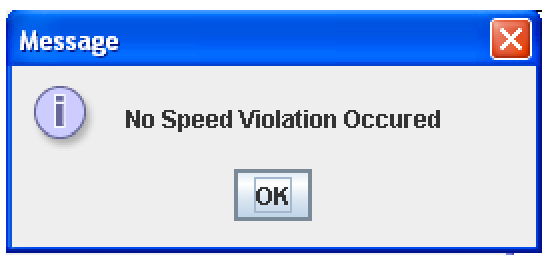


15. A map is generated in the main window, depending on the network at the back end



16. Now the *Tickets* and *Track* buttons will be enabled. Thus the operations can now be performed.

17. If we click on *Tickets*, the to-date violations will be noted and corresponding tickets are automatically generated which can then be saved. If however, no speed violation is noted, it is reported to the client.



18. Any car id can be tracked and its path graphically displayed. Along with the map, a tabular display is also provided. Option is also given to track the car on a particular date. Reports can be generated as and when desired. Multiple report generation is also supported.

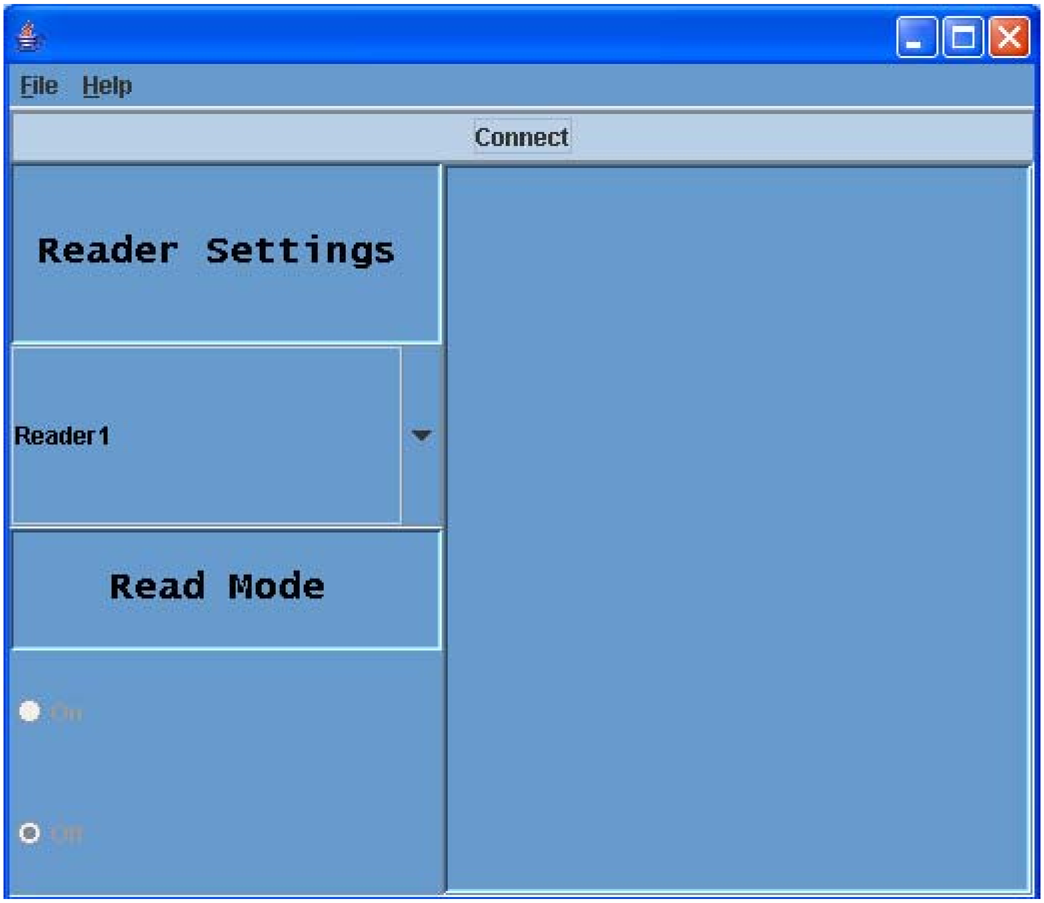
```
                         CAR NO.   Tha 01

****************************************************************

ROAD NO.          1
READER NO.        R1
DATE              25-FEB-2006
TIME              12:28:39  AM

****************************************************************

ROAD NO.          1
READER NO.        R1
DATE              16-MAR-2006
TIME              10:19:59  AM

****************************************************************

ROAD NO.          4
READER NO.        R16
DATE              22-MAR-2006
TIME              12:28:43  AM

****************************************************************
```

Running Client Software

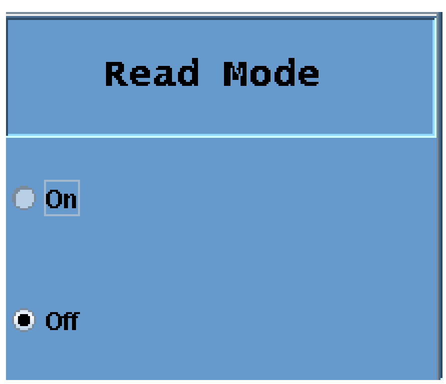
1. Run the Command Prompt (DOS)
2. Set the path for where the Application is placed
3. start the registry by typing the following command in the command prompt

   …….>start rmiregistry
4. A new Command prompt window appears. Minimize the new window
5. Before proceeding to the next step, make sure the server is bound(step12 has been reached in server execution)
6. Now type the following command in the DOS prompt

   …….>java –Djava.security.policy=java.policy ComputeEngine
7. Set the reader connected to the system



8. Once the reader is set, turn the mode *on*



9. The client is now fully operational.

TROUBLE SHOOTING

Problem: Ever time a car passes by a error message is displayed, and no value is read at the server side.
Make sure the DB settings are in place at the server side.

Problem: Each time client attempts to establish the connection with the server, 'Access Denied Error' is displayed.
Make sure when writing to a DOS prompt to extra spaces are included in the command. It should be typed exactly as below:

java –Djava.securiyt.policy=java.policy ComputePi

Problem: Each time server attempts to register itself, 'Access Denied Error' is displayed.
Make sure when writing to a DOS prompt to extra spaces are included in the command. It should be typed exactly as below:

java –Djava.securiyt.policy=java.policy ComputeEngine

APPENDIX C

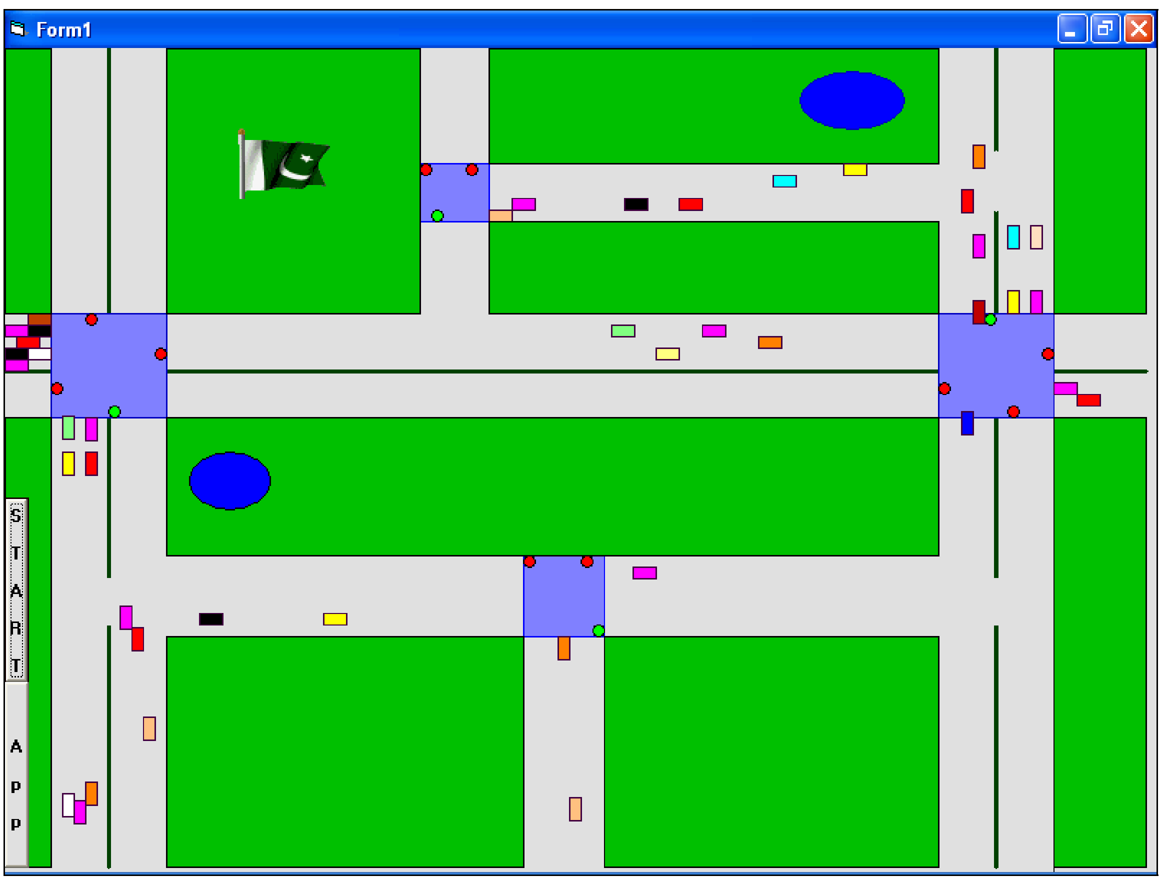
Screen Shots of

Intelligent Traffic Control System for Congestion Alleviation Simulator

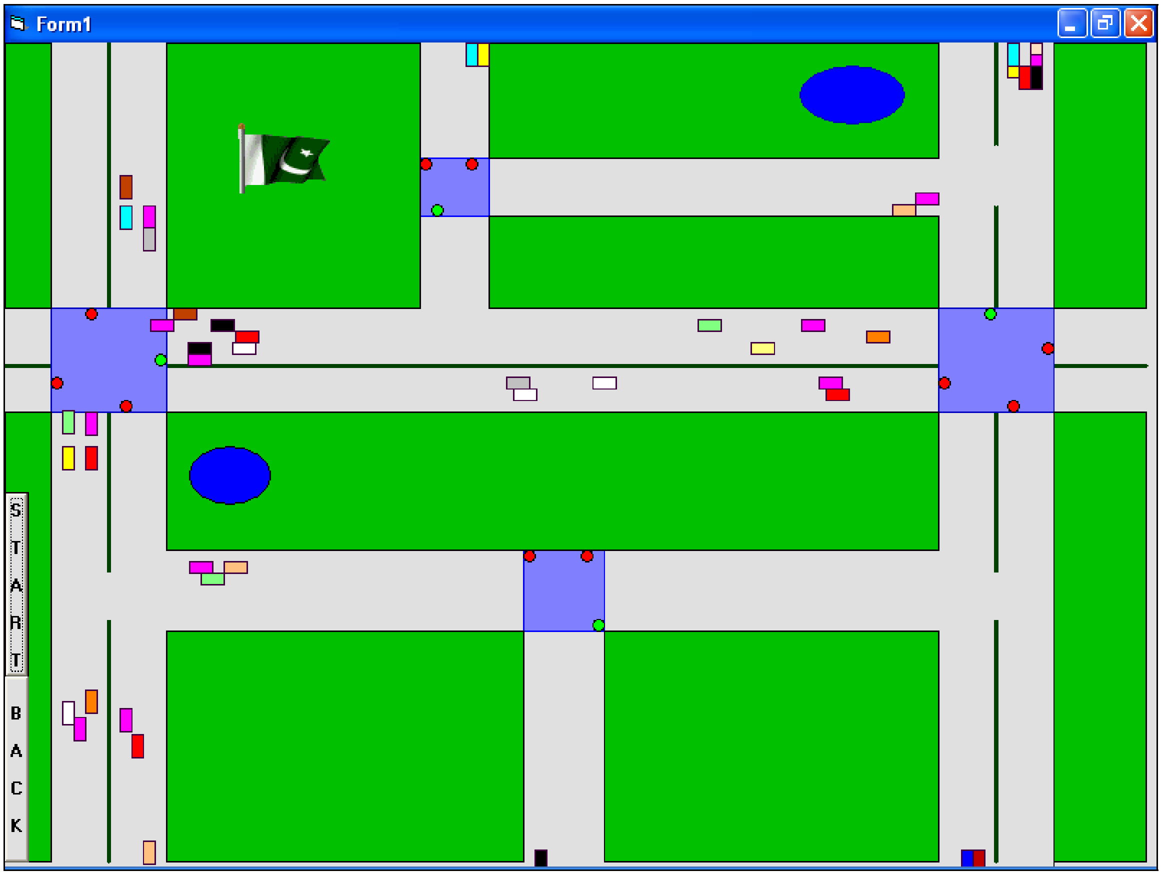
The simulator is developed for verifying the algorithm that we have developed to alleviate the congestion on the major highways of the metropolitan cities.

The simulator has two major sections:

- Section one simulates the traffic before the implementation of the algorithm. Congestions and vehicle queuing occurring on the roads is shown in this section.



- Section 2 deals with the scenario where the algorithm has been applied, and thus reduced vehicle queuing is shown in this section.



APPENDIX D

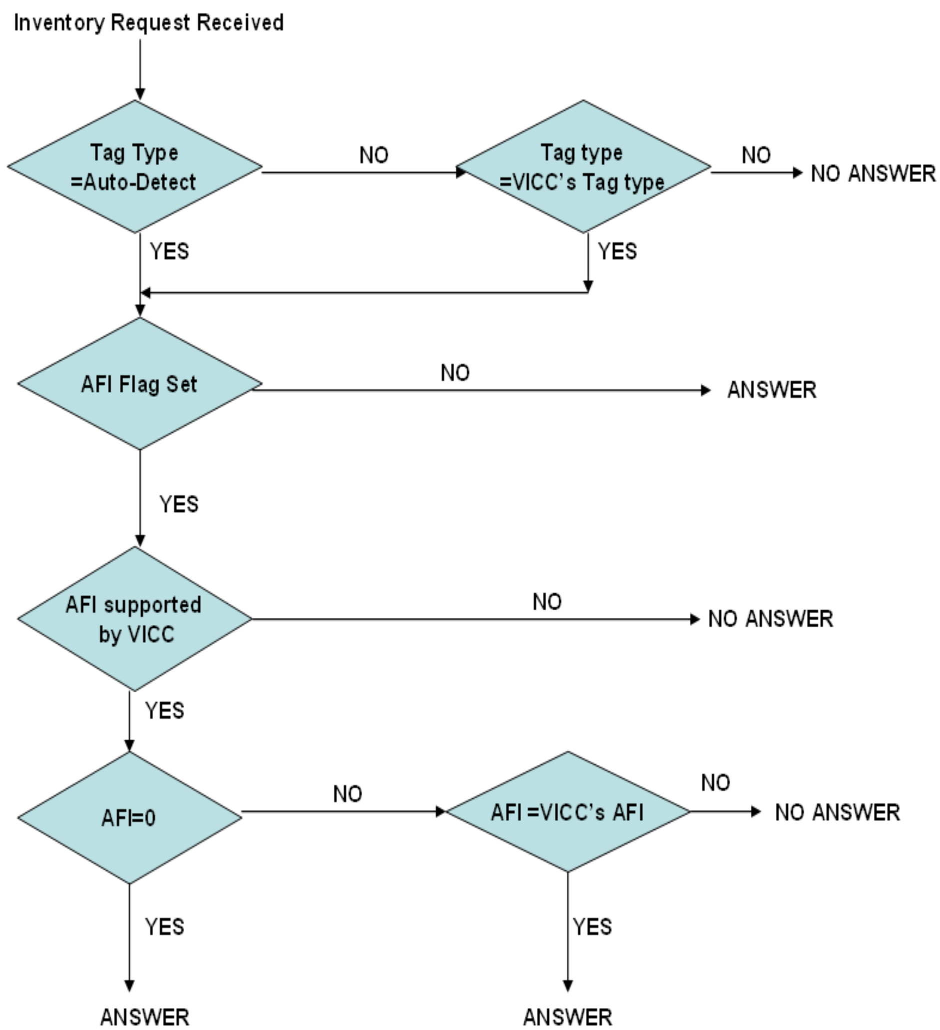
Memory Organization of RFID Tag

	Byte 0	Byte 1	Byte 2	Byte 3	
Block –4	UID0	UID1	UID2	UID3	Unique Identifier (lower bytes)
Block –3	UID4	UID5	UID6	UID7	Unique Identifier (higher bytes)
Block –2	Internally used	EAS	AFI	DSFID	EAS, AFI, DSFID
Block –1	00	00	00	00	Write Access Conditions
Block 0	x	x	x	x	User Data
Block 1	x	x	x	x	
Block 2	x	x	x	x	
Block 3	x	x	x	x	
Block 4	x	x	x	x	
Block 5	x	x	x	x	
Block 6	x	x	x	x	
Block 7	x	x	x	x	
Block 8	x	x	x	x	
Block 9	x	x	x	x	
Block 10	x	x	x	x	
Block 11	x	x	x	x	
Block 12	x	x	x	x	
Block 13	x	x	x	x	
Block 14	x	x	x	x	
Block 15	x	x	x	x	
Block 16	x	x	x	x	
Block 17	x	x	x	x	
Block 18	x	x	x	x	
Block 19	x	x	x	x	
Block 20	x	x	x	x	
Block 21	x	x	x	x	
Block 22	x	x	x	x	
Block 23	x	x	x	x	
Block 24	x	x	x	x	
Block 25	x	x	x	x	
Block 26	x	x	x	x	
Block 27	x	x	x	x	User Data

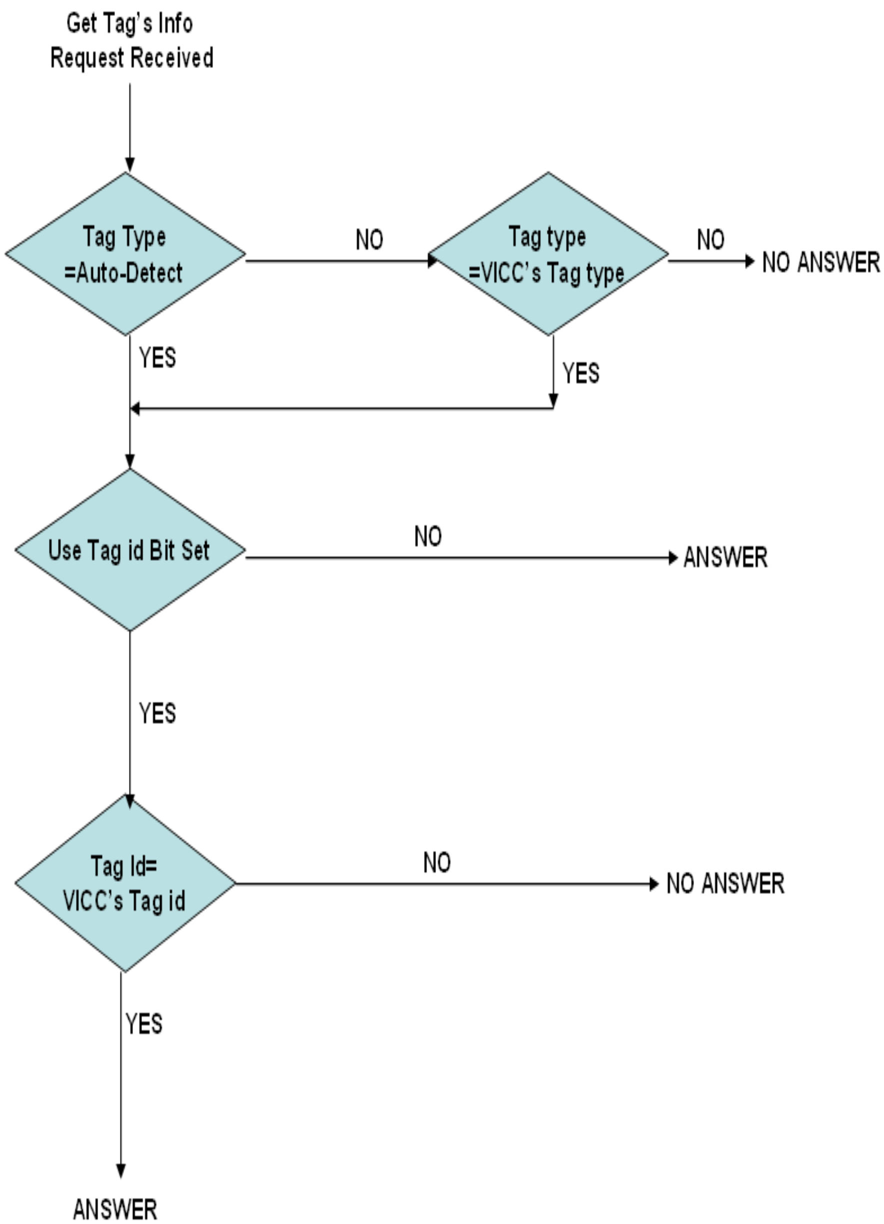
APPENDIX E

FLOW CHARTS

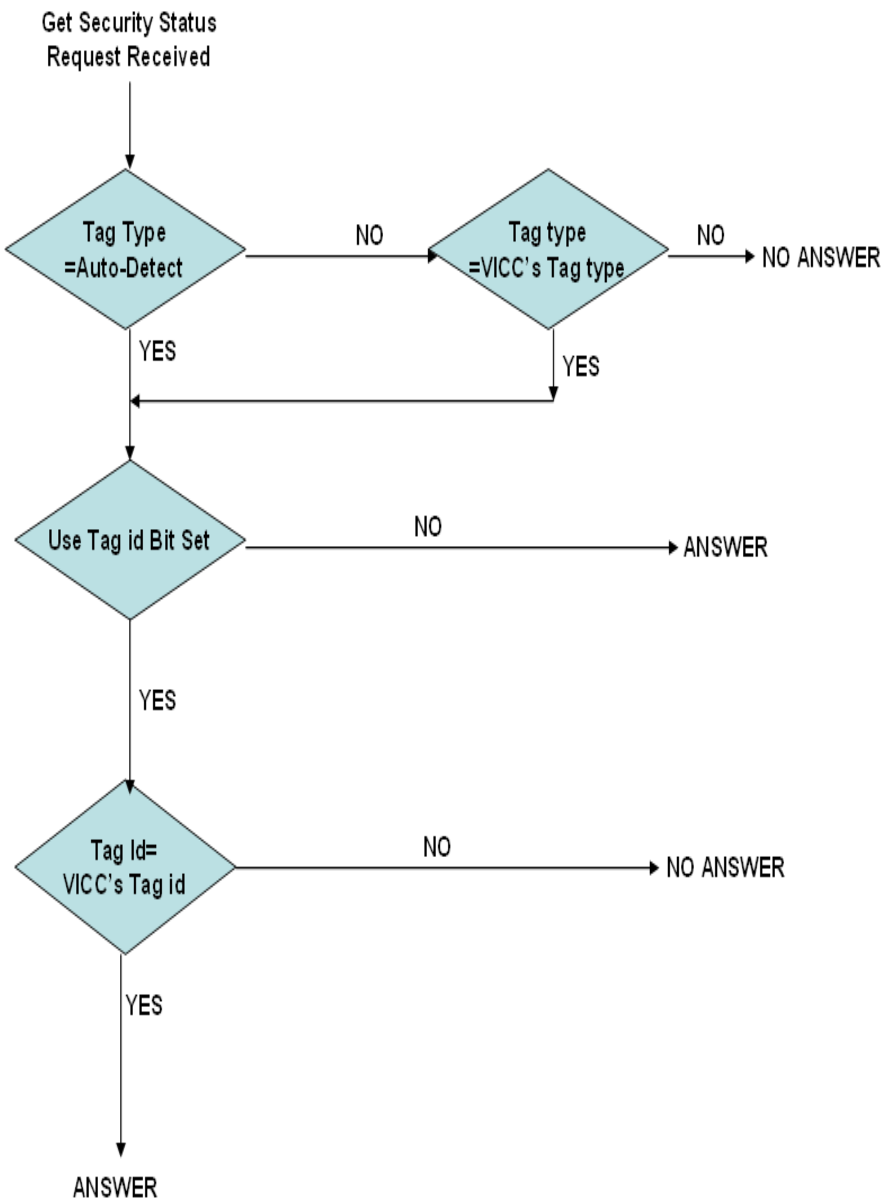
<u>Inventory Command:</u>



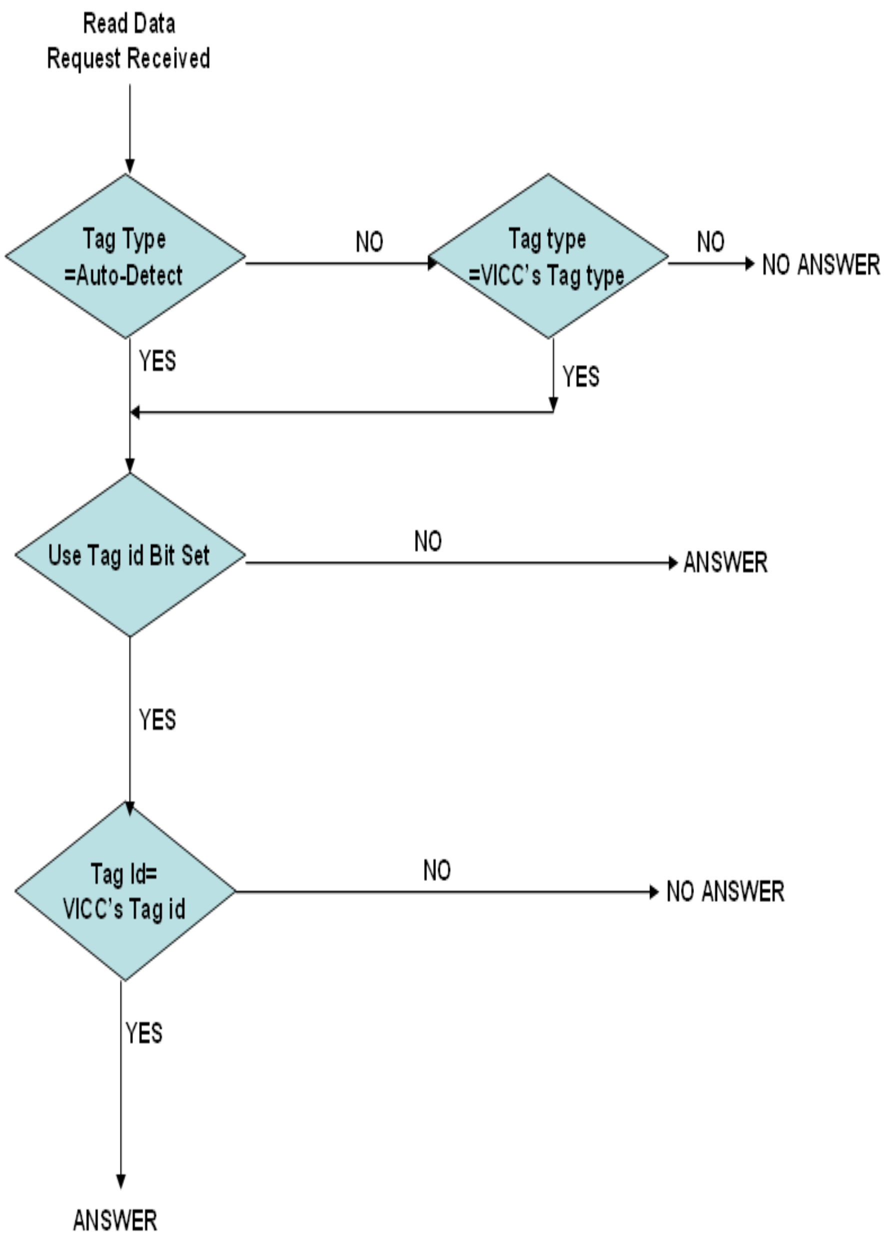
Get Tag's Info Command:



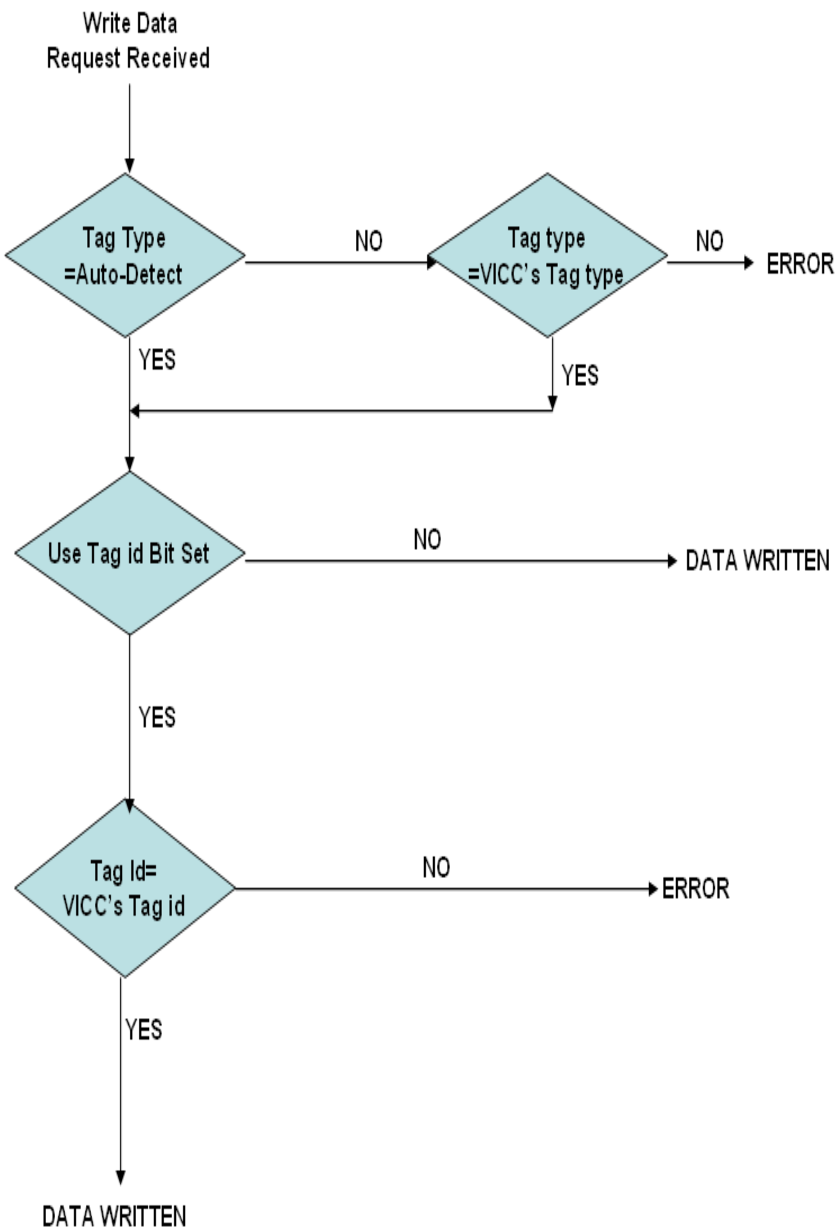
Get Security Status Command:



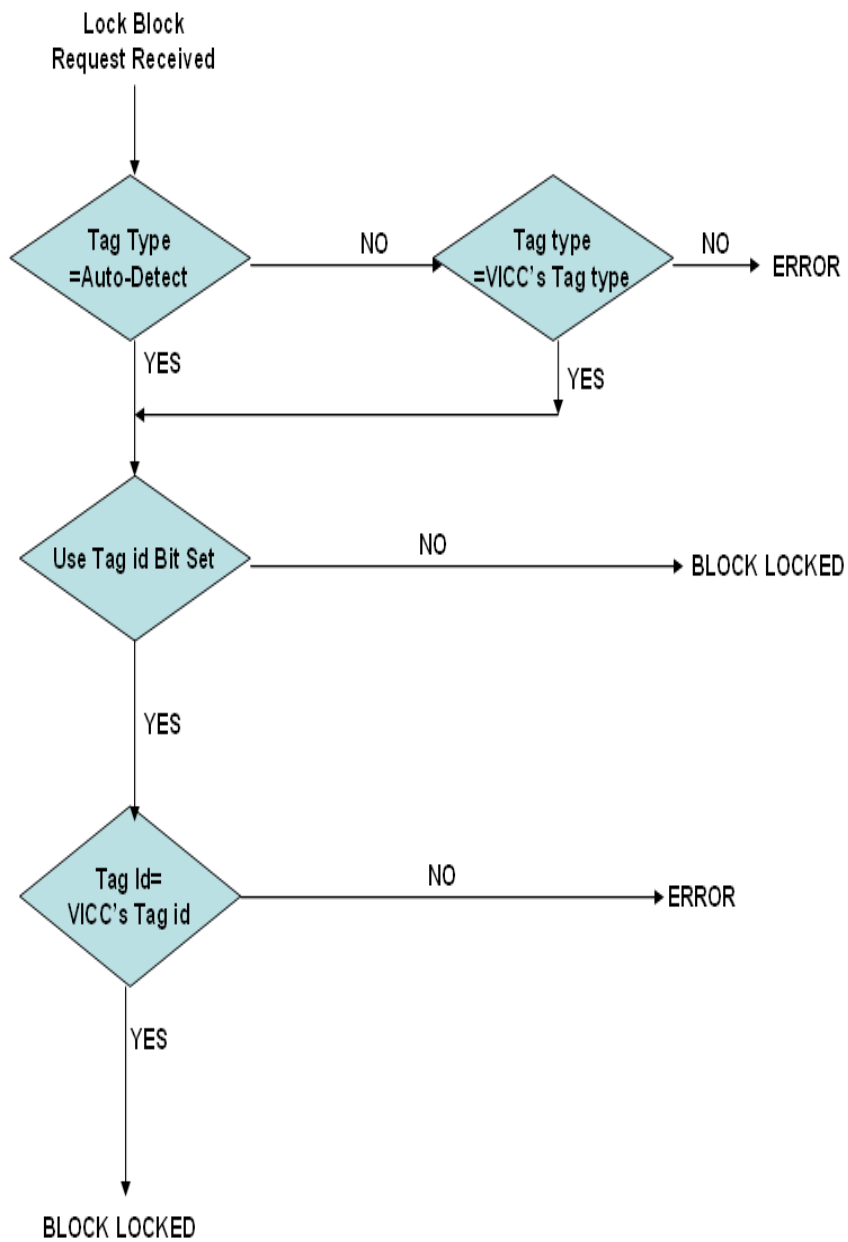
Read Data Command:



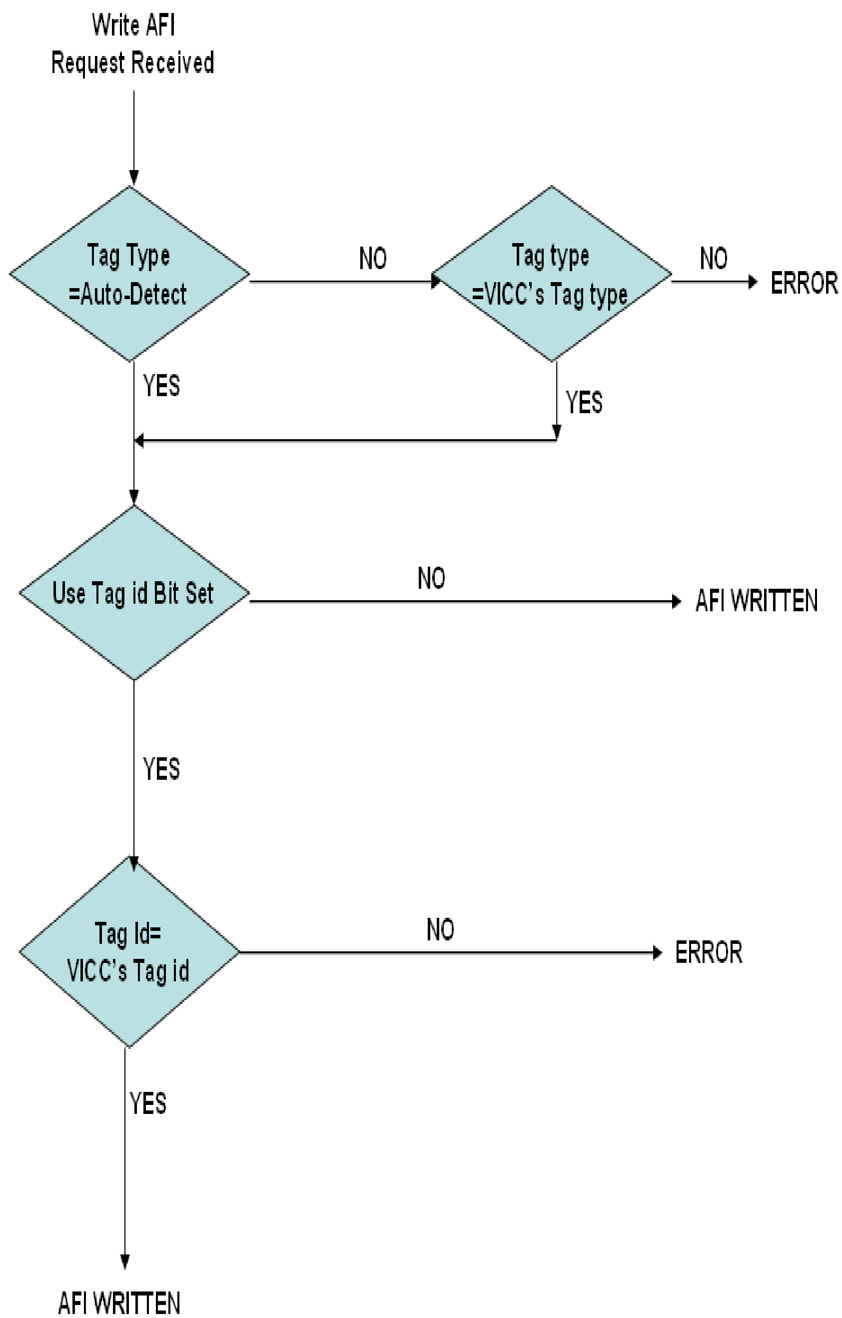
Write Data Command:



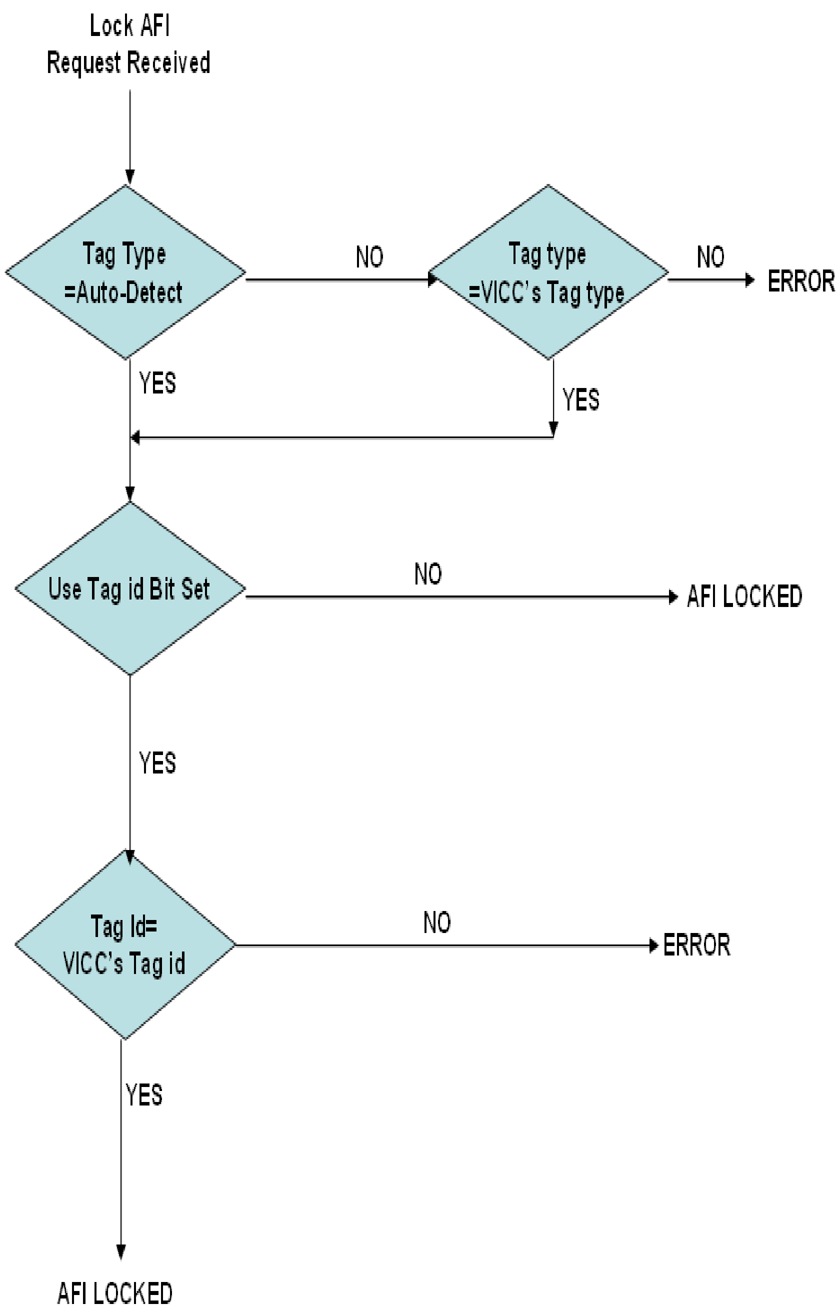
Lock Block Command:



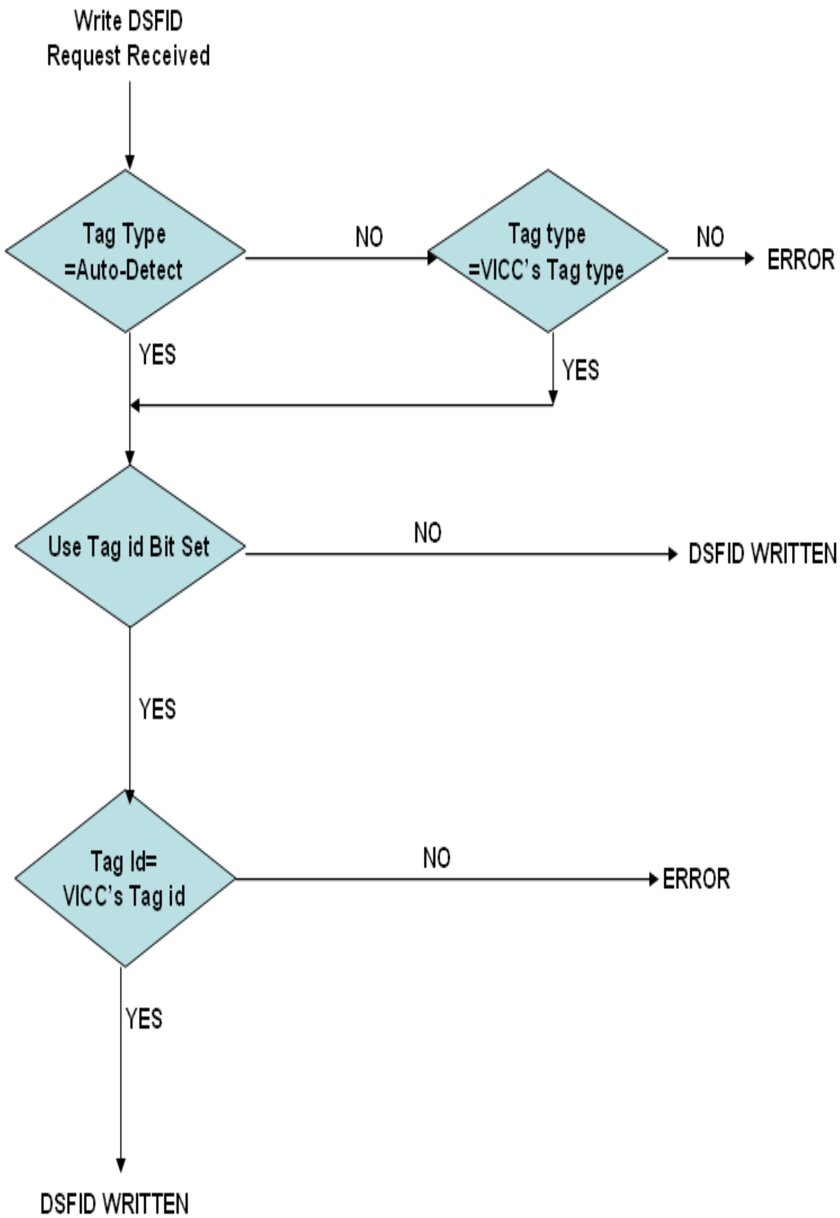
Write AFI Command:



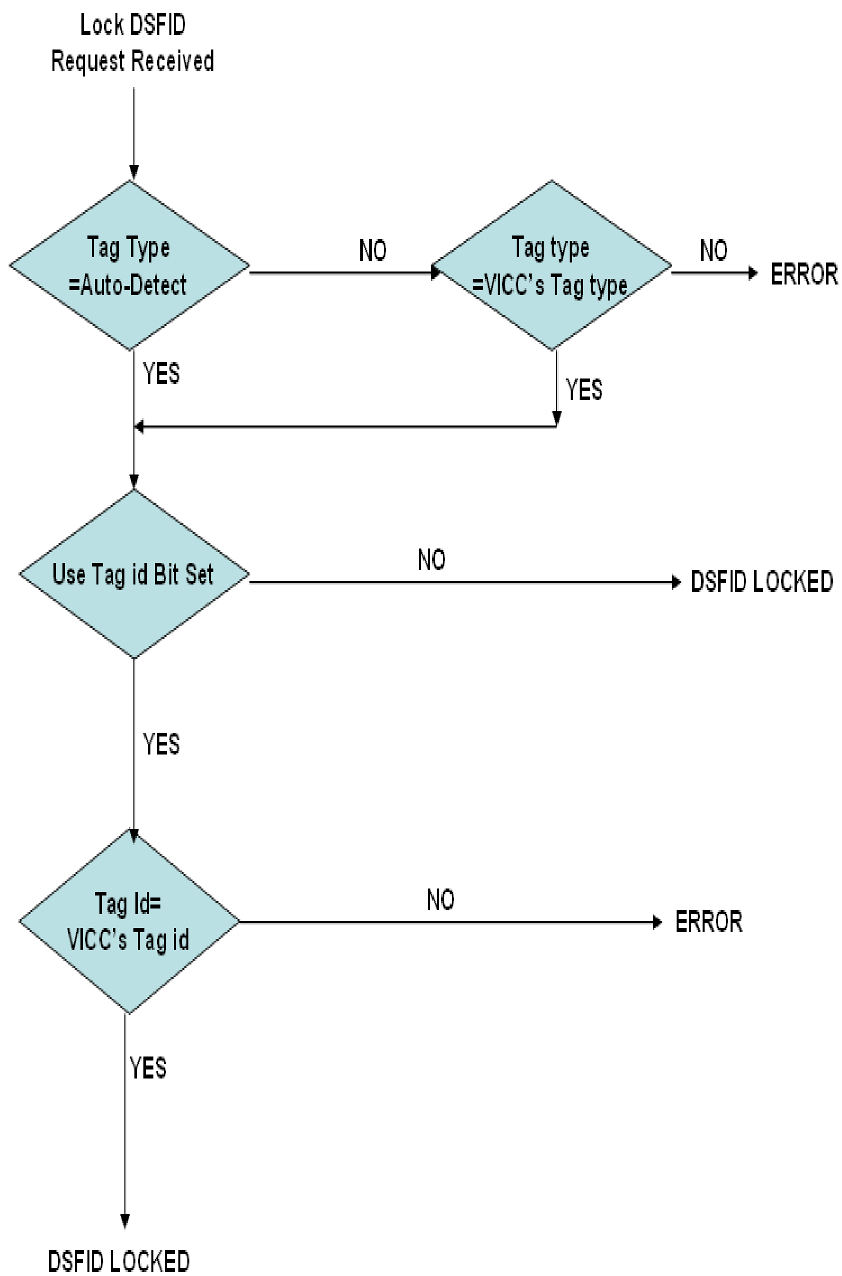
Lock AFI Command:



Write DSFID Command:



Lock DSFID Command:



BIBLIOGRAPHY

[1] “*Wikipedia, the free encyclopedia”*.

[2] “*RFID Unites the Supply Chain*”, March 23, 2005.

[3] Steve Meloan, *“Toward a Global Internet of Things",* November 11, 2003.

[4] Bret Kinsella, *“RFID- it’s more than tags and standards”,* Published by UsingRFID.com in March 2004.

[5] Kim Watson, *“What it going to cost me?”*, August 31, 2004.

[6] Susyd’Hont, “*The Cutting Edge of RFID Technology and Applications for Manufacturing and* Distribution”, Texas Instrument TIRIS.

[7] Karen M. Furlani and William C. Stone, *“Architecture for Discrete Construction Component Tracking”*.

[8] Dora Karali, “Integration of RFID and Cellular Technologies”.

[9] Savi , “Learn About RFID”.

[10] Intel Technology Journal, *“RFID: The Real and Integrated Story”*, August 3, 2005.

[11] Gearoid Morley, *“Radio Frequency ID Tags What They Are*, *How They Can Be Used and How Production Costs Can Be Reduced”*, Automation Research Centre, University of Limerick, August 2003.

[12] Jonathan Collins*, “RFID System Keeps Track of Evidence”*.

[13] Kelvin J Delaney, “*Beyond Barcodes*”.

[14] editor@rfidjournal.com , “*Glossary of RFID terms*”.

[15] Harry Stockman, *"Communication by Means of Reflected Power"*, Proceedings of the IRE, pp1196-1204, October 1948.

[16] Laura J. Smart, “Considering RFID”, California State Polytechnic University, Pomona.

[17] Association for Automatic Identification and Mobility. Radio Frequency Identification (RFID) home page, “*<http://www.aimglobal.org/technologies/rfid/>*”, 10 February 2004.

[18] Richard Boss, “*RFID Technology for Libraries*”, Library Technology Reports (Nov/Dec 2003), pp8-10.

[19] Homburger, Kell and Perkins, “*Fundamentals of Traffic Engineering, 13th Edition*”, Institute of Transportation Studies, University of California (Berkeley), 1992.

[20] Das, Shantanu and Levinson, D. (2004), “*A Queuing and Statistical Analysis of Freeway Bottleneck Formation*”, ASCE Journal of Transportation Engineering Vol. 130, No. 6, November/December 2004, pp787-795.

[21] B. Kerner, “*The Physics of traffic*”, Springer Verlag (2004)

[22] N. Bellomo, V. Coscia, M. Delitala, “*On the Mathematical Theory of Vehicular Traffic Flow I. Fluid Dinamic and Kinetic Modelling*”.

[23] Mark Roberti, “*RFID aided Marines in Iraq*”.

[24] Mary Catherine O'Connor, *"RFID Takes Attendance- and heat"*.

[25] Laura J. Smart, *"Considering RFID"*, Electronic Resources Cataloger

[26] Southwest Research Institute "*Automated Vehicle Identification; Chapter 1"*.

[27] "*Intelligent Traffic System Alleviates Congestion for Londoners*", Aug 2004.

[28] Southwest Research Institute, "*Automated Vehicle Identification; Chapter 2"*.

[29] Susy d'Hont, *"The Cutting Edge of RFID Technology and Applications for Manufacturing and Distribution",* Texas Instrument TIRIS.

[30] *Steve Meloan, "Toward a Global "Internet of Things"*, November 11, 2003.

[31] Annie Chen, Navendu Jain, Angelo Perniola, Tadeusz Pietraszek,Sean Rooney, Paolo Scotton, "*Scaling Real- Time Telematics Applications using Programmable Middleboxes: A Case Study in Traffic Prediction",* IEEE 2004 Consumer Communications and Networking Conference, 7 January 2004.

[32] Fatih Porikli and Xiaokun Li, *"Traffic Congestion Estimation Using HMM Models Without Vehicle Tracking"*, TR2004-019, December 2004.

[33] J.-S. Yang (USA), "*Traffic Signal Timing Control for a Small-scale Road Network",*2004

[34] Gearoid Morley, "*Radio Frequency ID Tags What They Are, How They Can Be Used and How Production Costs Can Be Reduced"*, Automation Research Centre, University of Limerick, August 2003.

[35] Southwest Research Institute, "*Automated Vehicle Identification Model Deployment Initiative System Design Document, Version 1.0*", January 23, 1998.

[36] Southwest Research Institute, "*Automated Vehicle Identification Model Deployment Initiative Software User's Manual, Version 1.0*", June 18, 1998.

[37] Science Applications International Corporation, "*Automated Vehicle Identification Tags in San Antoni*".

[38] Anthony A. Saka, G. Anandalingam and Nicholas J. Garber, *"Traffic Signal Timing at Isolated Intersections using Simulation Optimization"*, Proceedings of the 1986 Winter Simulation Conference.

[39] Janet E. Williams, Jeff Griffiths, "*The Geometrical Design of Signalized Road Traffic Junctions"*, Proceedings of the 1987 Winter Simulation Conference

OTHER REFERNCES

Jonathan Collins, *"Rotary Screen used to print UHF Antennas"*

Zoe Davidson, Botson University, *"RFID right to know act of 2003"*

Frank Byrt ,"*Small Firm Tunes in winning business"*

Bill Abelson , *"Symbol Technologies Expands Enterprise Mobility Portfolio with New RFID Products and Solutions"*, September 29, 2004

"Symbol Technologies Names Industry Leader Philip Lazo Vice President and General Manager of RFID Division", Jan. 20, 2003

Purdue Pharma, *"Symbol RFID Systems Will Support Law Enforcement to Reduce Drug Counterfeiting"*, February 14, 2005

"RFID Pioneers Symbol Technologies and Philips Announce RFID Generation 2 Early Adopter Initiative, April 11, 2005

Jonathan Collins , *"RFID System Keeps Track of Evidence"*, 2006

"Gillette Conducts RFID Risk-Assessment Survey", May 27, 2005, www.bpm-today.com/story.xhtml?story_id=13100002SFC5

"RFID Solutions – Enterprise scalable, end to end mobility products", www.symbol.com/products/rfid/rfid.html

"Panel Urges Suppliers To Start Developing Their RFID Strategies", May 16, 2005, www.bpm-today.com/story.xhtml?story_id=34974

"Wal-Mart Starts Game of RFID Tag", May 16, 2005, www.bpm-today.com/story.xhtml?story_id=34975

Mick Brady , *"Genesta Offers Multi-Protocol RFID Handheld Reader"*, May 25, 2005, www.bpm-today.com/story.xhtml?story_id=35629

"IBM Has Some Tall RFID Plans", May 20, 2005, www.bpm-today.com/story.xhtml?story_id=35345

"Reader/writer for use with non-contact IC card and reader/writer system", October 28, 1997, www.patentstorm.us/patents/5682143.html

"RFID Printer/Encoders", www.zebra.com/id/zebra/na/en/index/products/printers/rfid.html

"RFID Smart Labels",

www.zebra.com/id/zebra/na/en/index/products/supplies/rfid_smart_labels.html

Ronald Reagan,*"Symbol Technologies Sponsors RFID and the Enterprise Mobility Advantage"*, 26 April, 2003

Press Release, *"The History of the Barcode",* 26 May, 2005

"RFID Webinars", www.zebra.com/id/zebra/na/en/index/rfid/webinars.html

www.symbol.com/products/rfid/rfid.html

www.rfidtribe.com

www.rfidjournal.com

www.rfidjournalalive.com

www.rfidgazette.com

www.rfidbuzz.com

www.rfidlog.com

www.rfidnews.com

www.rfidupdate.com

www.rfidtalk.com

www.manager-tool.com/default.asp

www.righttag.com/RFIDInfo.html , www.manager-tool.com/TheISO9000Toolkit.html