

TI India

Analog Design Contest 2011



SMART LIBRARY MANAGEMENT SYSTEM

TI India Analog Design Contest 2011

1	Name of your College	Thapar University			
2	Name of the Project	Smart Library Management System			
3	Team Number	140			
4	CMT Paper ID	76			
5	Details of the participants		Name	Branch	Semester/Year
		1	Gursimran singh	Computer Science	6 th Semester
		2	Deepender Singla	Electrical Engineering	6 th Semester
		3	Amit Ranjan	Electrical Engineering	6 th Semester
6	Details of the mentor	Name Mr. M.D Singh Phone 9815605616 E-mail mdsingh@thapar.edu			
8	Has your report been read and approved by your mentor?	Yes			
9	YouTube Video Link	http://www.youtube.com/watch?v=sajW6u9C13E&feature=youtu.be			

SELF ASSESSMENT

We, authors of the report Smart Library Management System confirm that this report has not been submitted for any other forum such as another contest. We hereby state that we will not submit the same work for any other contest in the future. We understand that Texas Instruments has the right to use this report in its conferences/publications.

Originality of idea & inspiration

The project is based on our own idea which was a result of problems that we faced in our university library. We got in touch with the librarians of our university library to identify some crucial problems that administrators faced.

Persons who helped

Mr Mandeep Singh:

He guided us throughout the project. His helped us in algorithm development for image processing.

Dr.Rajesh Khanna:

He guided us about RFID Antenna Fabrication.

Mr.Prateek Bhatia:

He helped us with the Network Messenger development.

Technical Challenges:

Antenna Development: The biggest challenge in this project was to develop antennae. it was very costly to purchase ready-made antenna from FEIG. So we decided to fabricate antenna of our own. Initially it was very difficult but we studied literatures about its development from Texas Instruments. We also got expert advice from time to time through e2e community.

Synchronization: The project comprises of a significant part of software work, apart from the hardware. The project derives its innovative component from the tricky software that we have developed. Therefore it was a quite tedious task for us to synchronize all applications together.

Non-technical challenges:

The literature available online for class E amplifier development is very superficial. Hence it was very tough for us to understand its working and subsequent implementation.

The electronic components were not easily available in nearby market of Patiala and we had to get these from other cities.

Experience of the Analog Design Contest

We faced lot of problem in locating books at our university library. We wanted a viable solution for this. We organised our idea and submitted it for TI Analog Design Contest-2011. This is how our journey started. At the time of submission, our only goal was to get to the next round and nothing beyond that. But as we kept moving from one round to another, our goal kept shifting as well.

Once we started implementing the idea, we faced many ups and down. At times we even thought of giving up. But we kept motivating each other and kept on working on our project. Initially, we failed several times or rather in the words Thomas Edison We successfully found 10,000 ways that would not work. We kept switching from one software and hardware platform to another and to yet another. In the meantime we learned a lot.

Things that could have added further value to the project.

We would like to use this DSP platform for our face detection and recognition algorithm(Surveillance algorithm) in openCV, C6000 is a best choice for Porting openCV in embedded system like in our solution with its very high processing speed and this will also help us to minimize cost with its low cost compared to Intel Atom D-525, basically this is one of the reasons we developed our algorithm in openCV because it is open source and also C6000 processor supports it. Even though we will sacrifice TBB support here but still C6000 is a multi-core processor with less memory requirement as compared to others which makes it the best choice to us.

In our view C6000 platform is quite sufficient for us as the refinement which is going on in Face detection and recognition it is also able to meet more computational power required in future. Had we used some multi-box processor here we would not have used its computational power to the fullest and our system would be more expensive because of increased cost of processor, memory and also power requirement, so C6000 is a optimal choice for the solution.

In future we may replace TRF7960A with TRF7970A because newer chip has peer to peer communication which can be very useful for the refinement of our solution.

Check-list

Paper design of hardware <input type="checkbox"/>	Algorithm/software design <input type="checkbox"/>
Hardware implementation on breadboard <input type="checkbox"/>	System-level testing with examples <input type="checkbox"/>
Hardware implementation on PCB <input type="checkbox"/>	Benchmarking/Performance Analysis <input type="checkbox"/>
Hardware Testing <input type="checkbox"/>	Short Video on Project <input type="checkbox"/>

Gursimran Singh

Deepender Singla

Amit Ranjan

Table of Content

S.NO	Topic	Page No.
1	Introduction	7
2	Technical Background	8
3	Problem Statement	10
4	Objective of the Project	12
5	Proposed Solution	13
6	Hardware Implementation	17
7	Software Implementation	20
8	UML Diagrams	30
9	Cost of Hardware and Software used	34
10	Results	35
11	Conclusion	38
12	Picture of Finished Project	40
13	References	42
14	Appendix A	42
15	Appendix B	45

1.1 INTRODUCTION:

In the brain storming phase of the project, we got in touch with the librarians of our university library to identify some crucial problems that administrators faced. Since we ourselves are frequent library users, we were fully aware of the problems that we faced.

On analysis we found that one of the major problems that the current library management system faces is that most of the times books are not found at their assigned location. Tearing of pages and theft of books are also acute problems specific to Indian libraries. In this prototype, we have plugged-in the loop holes of the present system with our new smart library management system.

We have used RFID to establish a smart library system which knows the location of each book and updates the changes in location in real time. Thus the books are never misplaced. Though RFID is already in use in many libraries in India and across, there use has widely been limited to as a substitute of bar code. Our solution goes beyond conventional approach to present RFID systems to achieve a more robust, theft free, secure and fully automated library.

Moreover care has been taken so that the current prototype is compatible with the existing infrastructure so that minimal change is required to update the present semi-automated system to our fully automatic solution. Moreover installation cost at the start up is small and given the man power requirement and associated cost involved in the present setup, smart library system would be economically viable as well. Many 'out of box' ideas have been accommodated to make the system cost effective and indigenous.

We believe that our solution will be a stepping stone for the next generation libraries. This work will provide plethora of scope for many more research in this field. We can explore new horizons and integrate applications like GPS, image processing etc to achieve even more automation.

1.2 Technical Background

Existing Solution

The two most commonly used systems in libraries are

- Barcodes
- RFID

Both use a technique that embeds a unique tag with specific informations. One main requirement is that such tags must be passive and should not operate on battery.

BARCODE

Almost all the libraries across the globe use bar code that uniquely identifies a particular book. A barcode is an optical machine-readable representation of data, which shows data about the object to which it attaches. The barcode is read with the help of a barcode scanner and corresponding information or metadata can be located in the main database. Library at our university has bar codes attached with all books and even with all the member cards. All the allied information regarding each member and books are linked with the bar-code. To a large extent, issue and return of book becomes automated. User can search about all the information of the book and its assigned location through the computer connected to the library database.

But this arrangement has following loopholes.

- 1) Use of bar-code hardly fulfils our objective of automation. Each time a book has to be issued or returned, bar code on the book has to be scanned using a bar code scanner. It also calls for extra man power.
- 2) Incidents of theft can't be controlled using bar-code as it can't check unauthorised trespasses.
- 3) Most often than not, books are misplaced in the libraries from their designated location and hence it becomes impossible for users to locate books.
- 4) One of the serious flaws with barcode is that a bar code reader can scan only one bar code at a time. So it becomes cumbersome and time consuming to handle large volume of crowd.
- 5) Users have to wait in long queues to issue or return books.

RFID

RFID is being used in many libraries in India and across, but the use has widely been limited to as a substitute of bar code. No doubt, they have made check-in and check-out semi-automated with minimum human interference. The major loopholes of the system are:

- 1) There is no provision to locate misplaced books. Book that has been displaced has to be located manually and put back into proper place.
- 2) It doesn't absolutely curb the theft of books or tearing of pages

To locate a misplaced books, the method used till now in most 'automated libraries' are quite tedious. A person has to scan the RFID tag of all the books using a hand held device and if any misplaced book is found, hand held device responds either by glowing LED or turning on the buzzer. We have developed a fully automated Book location finder

using passive RFID tag. If a book is misplaced, the database of library will be likewise updated, so as to point to the location of the book.

Comparison:

Technology Capability	Bar Code	Present RFID	Our System
Line of Sight Requirement	Required	Not Required	Not Required
Book Theft	Possible	Possible	Theft Free
Tampering of book	Possible	Possible	Condition improved
Real time automatic Updating of book location	Not Possible	Not Possible	Possible
Unauthorized use of card	Possible	Possible	Condition improved
Book issue/return on-the-fly	Not Possible	Not Possible	Possible
Reviews of Books	Not Possible	Not Possible	Possible

In most advanced Present RFID system the book is issued at the library gateway , user with his card walks through the gate which has antenna attached and book are issued to him , there are certain limitation of this design like:

- 1.) Students could easily shield tags with foil or other metallic material and go through a portal.
- 2.) Student can tamper tag of book and go through a portal without detection.
- 3.) Student can throw the book outside library through window.

1.3 Problem Statement

Problem 1

Most often than not, books in libraries are displaced from their assigned location, and as such it is practically impossible for user to locate any particular book. It is cumbersome task on the part of administrator as well to put the books back in order.

Solution:

The main problem we are dealing with is identifying the location of the book. Embedded to the *unique tag*, we have a *location* field in our main database along with other details of that particular book. So when a user searches for a particular book, our automatic system can direct him where the book lies. Normally librarian's categorize the books on the basis of some location identifier. So the user can find out the book in the particular shelf tagged with that location identifier. So every time a malicious or careless user come and places the book in some other wrong location the user who is looking for that book can never find it where the automated system says.

In our present prototype, whenever a malicious or careless user places book in some wrong place, the book automatically updates its new location in the *location* field of the *metadata* attached to its unique ID. So this system will work in real time and update the new location of the book whenever it changes, so any user who will be looking for that book can find it easily.

Problem 2

Major concern for libraries has been to protect the theft of books. Increased human vigilance have had little impact on curbing this menace

Solution:

In our RFID based solution, User has to first present his card before the RFID reader before he picks up any book. In case the user tries to pick a book without presenting his card, administrator would be informed about this violation.

As soon as he picks up the book, book is issued in his name until he puts it back on any of the shelves (Not necessarily the shelf where the book was originally placed). The new location of the book would automatically be updated. In case the book is lost, the last user who used the book can easily be identified.

Problem 3:

What if a user tampers the book (tears pages/ writes on book) and puts it back

Solution:

The database would have complete history of the book with specific details of time duration for which a particular book was used by a user. The CCTV Camera footage of only those specific time spans has to be investigated to identify the culprit.

Problem 4

What if a malicious user gets hold of other user's library card and attempts to get the book issued through his account.

Solution

As soon as the systems detects a user, the installed Camera(CCTV) would automatically rotate so as to point out to the shelf where the user has been detected. To achieve this Camera has been attached to the shaft of the servo and is programmatically controlled. The snapshots from the Camera would be matched with the stored photograph of the registered user through Face recognition, In case they don't match an alarm would be sent to the administrator.

Problem 5

Users have to wait in long queues to borrow and return book.

Solution

As explained above, the books will automatically be issued and returned and hence the entire process would become user friendly and more efficient.

Problem 6

What are added advantages of 'Smart Library System'

Solution

'Smart Library System' has plethora of added advantages, most important of them being, the review of the book that a user picks can be displayed on the screen along with details of book issued in his account. Moreover it makes it easier on the part of librarians to manage books especially in large libraries. It improves the surveillance and reduces the manpower involved.

1.3 Objective of the project

We strive to make libraries more reader friendly, so that users can

- can locate any book in books
- get the reviews of the book
- get the book issued instantly

Administrators can

- Curb theft and tearing of pages
- Get specific information about users who used a particular book in a given span
- Detect users using other's library card

Thus leading to

- improved security, surveillance and ambiance.
- Hassle free experience for users
- Minimal man power requirement
- Cost effective and smart solution for administrators

2. Proposed Solution



Fig 2(a)- View of present Library Management System

We have used RFID antennas at each compartment of shelf which are connected to the master computer and hence when a book is picked up or placed in that particular compartment, the RFID antennae connected to that particular shelf will detect it and report to the main system. The system will in turn update the *location* field of the metadata immediately as per the requirement. This way we can track down the location of each book in real time and can guide a user looking for the book up to that location from the computer interface. We have also added a security check that when a user wants to pick up a particular book he can do so only if he shows his RFID based library card to the RFID antenna in that particular compartment. This way the system will know who picked up the book. If a malicious or careless user tries to pick the book without showing up his card, the system will alert the authorities and the user using an alarm and the CCTV camera will take a picture of the incident. Thus by using apt hardware and smart algorithm we have successfully automated the library management system

- Advantages
 - Library will have details of all users who pick a particular book within a given time frame. So user tearing the pages or tampering the book can be identified.
 - No security loophole
 - User will not be able to remove RFID tags and take the book along.
 - It is a more close approach to fully automatic 24x7 library.
 - Implementation of this system will involve minimal change.

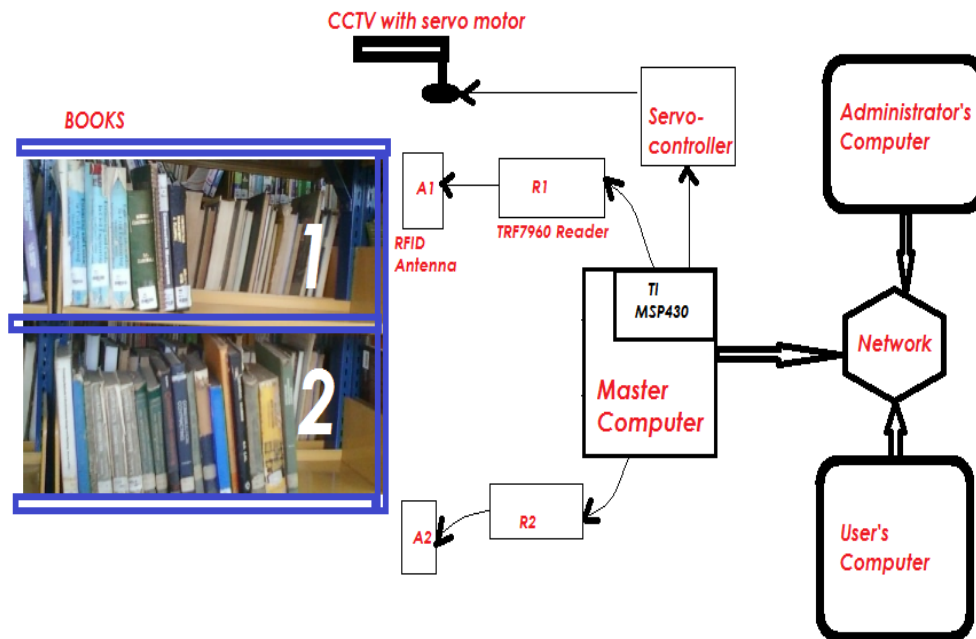


Fig 2(b)- Block diagram of set-up

We have a separate antenna for each compartment of the rack. The main system will know beforehand the location of various antenna. This information is conveyed to the main system by the microcontroller using a unique antenna number. This way we have a unique number, the *antenna number* tagged to each physical location or compartment of the shelf. So based on where the signal is coming from, the main system can recognize from where in the physical world the book is being picked up or kept. Along with this *antenna no* the unique RFID tag read by the RFID antennae is also sent to the main system. Thus the main system will be able to recognize which book is picked up or kept down from where in the physical world and hence update the location of each book in the main database.

- This is a brief scenario on how it works – Suppose if a user picks up the book the RFID antennae will detect that the book has been taken up so it will update the location field to PICKED_UP and when another user searching for that particular book will come to main system interface and search for that book using name, author name, etc he will be immediately informed that the book is with some other user. When the user who has the book keeps it back anywhere on the shelf, the RFID antenna of those particular shelves will detect its entry and update the location of the book immediately in the main database.

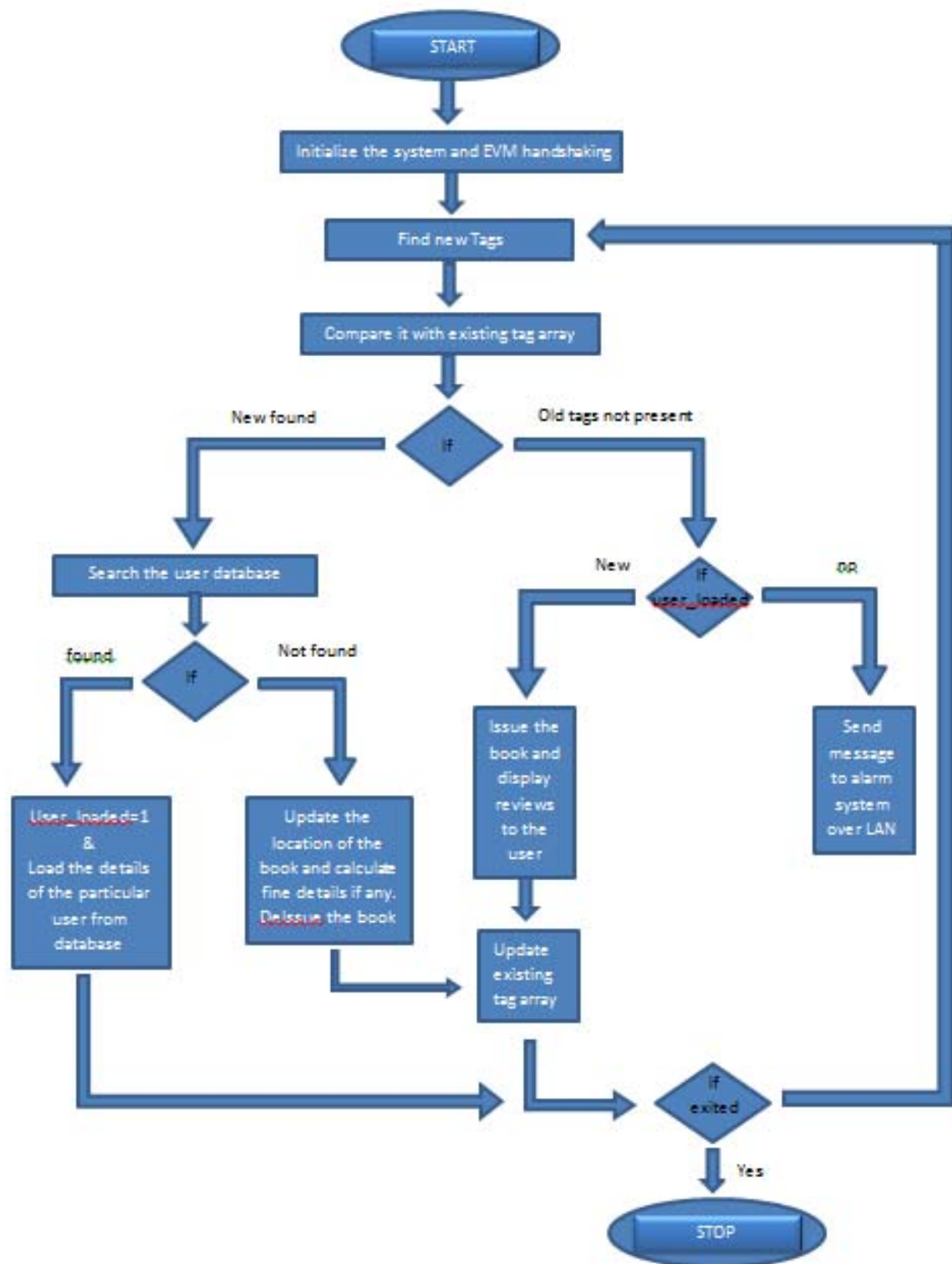


Fig 2(c) Flow Diagram of Smart Library Management System

- When person picks a book he has to show his library card in front of concerned antenna. Now there are two possibilities:
 1. User shows his card before lifting the book hence in our database we would know the exact details of the user who has picked up a particular book. If any user tries to tear away the tag for stealing the book, he can be caught very easily.
 - i) If a user tears the book page and keeps the book again in library and authorities come to know about it after 2 days. In this case, we know the details of the persons who read this book in the given time span. We get to know the exact duration for which a particular book was with a user. All the libraries have CCTV camera installed. Hence we can see the recording of all the concerned users and will be able to detect which person among them is the culprit, therefore there will be no need of a person watching CCTV camera every time.
 2. User doesn't show his card and lifts the book. Antenna will be able to detect that book has been lifted but microcontroller won't receive initial signal of person identification tag hence an interrupt will shoot out from our microcontroller to run siren or alarm. Microcontroller is communicating with master computer which in turn is in communication with library CCTV camera and it knows the location of any particular microcontroller and hence CCTV will take picture of that person and will be stored in its database. This message can in future be sent to admin's mobile via GSM module.

3. Implementation

3.1 Hardware Implementation

➤ RFID Sensor:

Initially we analyzed different aspects of RFID with TRF7960A evm ,and found that there were two main changes we needed:

- 1.)Use power amplifier with variable range (which can be set manually while installation) as different library compartment will have different length of stacks of books
- 2.)The size of Antenna is to be optimized as per the power Output from Reader TRF7960 i.e (200 MW).

Chips used:

➤ TRF7960A

The TRF7960A is an integrated analog front end and data-framing system for a 13.56-MHz RFID reader system. Built-in programming options make it suitable for a wide range of applications for proximity and vicinity RFID system.

We selected this chip as library management systems work on 13.56-Mhz Frequency This chip is an ideal reader for that purpose with its low power consumption , multiple reading and writing mode and also with lots of community support for using it for our design. But in future we may replace TRF7960A with TRF7970A because newer chip has peer to peer communication which can be very useful for the refinement of our solution.

➤ MSP430F2370

We used this ultra-low-power microcontroller which is connected with RFID reader and Communicates with master PC with UART –USB(silicon Lab) chip in between.

➤ Rectangular Inlay(45mm*76mm, RI-I02-112A)

Texas Instruments Tag-it™ HF-I plus transponder inlays consist of 13.56-MHz high-frequency (HF) transponders that are compliant with the ISO/IEC 15693 and ISO/IEC 18000-3 global open standards..

It is a passive tag we have used it for its cost-effectiveness .It has bigger size which results in improved detection range. In our solution we use this inlay on our Books.

➤ Square Inlay(45mm*45mm, RI-I11-112B-03)

Texas Instruments Tag-it™ HF-I plus transponder inlays consist of 13.56-MHz high-frequency (HF) transponders that are compliant with the ISO/IEC 15693 and ISO/IEC 18000-3 global open standards.

This passive tag is used within the user card as less detection range is needed with user card

Refer Fig.5.(d) for the detection range for the above tags.

➤ Serial servo controller

The servo controller is used to rotate camera according to the preferred location as discussed in our solution .The servo controller is made using ATMEGA-168 , basically we chose ATMEGA-168 because we had worked on it earlier as well so we were familiar with its working. In future we may replace this with Texas Instruments' MSP430 family as they have multiple power options which is a necessity in today's scenario or may be some multi-core Microcontroller like Cocerto from TI which can control servo and can integrate also with RF reader.

We replicated this PCB schematic for our solution .
Refer fig 9.(b)

➤ LM317DCY:

This adjustable Voltage Regulator is used for developing class-E power amplifier for the TRF7960.

➤ LM1085A:

This adjustable Voltage Regulator is used for developing class-E power amplifier for the TRF7960.

➤ Power Amplifier:

For development of power Amplifier we followed Texas literature and PCB schematic Refer fig 9.(c) .

We have planned our system according to the design constraint like different library compartment will have different length of stacks of books so there is a need that output Power can be varied Between (1 watt to 4 watt) manually while installing the components

In our design we have taken V_{dsat} as 2 volt as shown In Refer fig 5.(e).

➤ Antenna Fabrication:

The changes we made for the antenna fabrication is shown below, we took TRF7960 EVM, made some modifications in the PCB Schematic as shown for connecting external antenna. It is required to make 50 ohm impedance for maximum power transfer. Antenna inductance should be aprox 1 micro Henry.

Refer fig 9.(a) , 5.(a) , 5(b)

3.2 Software Implementation:

The project comprises of a significant part of software work, apart from the hardware. The project derives its innovative component from the tricky software that we have developed. In a nutshell we have used the same hardware of the existing RFID technology, modified it a bit using analogue components (for signal amplification) and developed software algorithms that can control it as per our idea.

The software that we have developed was quite crucial and big, so adequate planning and analysis was required in terms of formal frameworks of software engineering. So we started the planning phase right from the beginning, parallel to antennae range research. We tried to make the complete software as per plan but the scope and size of the application was too large and would require much more time and labour than provided in this competition. However we finally realised that for such large projects spiral model of software development is used, which is designed in incremental versions. The software that we have developed so far is just a prototype of our planned final release. This method was adopted to demonstrate and depict our idea in the minimum possible time. In the following sections we have included some elements of the planning, analysis, requirement gathering and development phase of the software.

Overall Description of software

➤ Product Perspective

The major concern of 'Smart Library Management System' is the 'automation of Library'. This must be of major concern to any university or organization. Because already a semi-automatic system exist, care has been taken the proposed system is compatible with the existing infrastructure so minimal change is required to update the present system to our fully automatic solution.

➤ Product Features

1. For Users

- a. Search a book using various options like title/ authors/ publishers/ etc
- b. Issue a book on-the-fly
- c. Return the book on-the-fly
- d. Display the reviews of the book issued.

2. For Administrator

- a. Add book to database
- b. Change/ edit the details of an existing book
- c. Manage other library managers
- d. Manage sensors
- e. Manage Users
- f. Manage security systems
- g. Manage navigation systems.

➤ Project Scope

Smart library management system” is designed to fit in the existing infrastructure of the existing library. The overall solution and software has been designed to minimise changes to be made to existing infrastructure. In libraries where existing RFID system is implemented, the changes required is even less. The solution proposes to install new RFID sensors in the existing racks without any problem. It will include participation ranging from faculty, library staff and students. These ‘actors’ will have different access rights according to their roles, so that coordination can be achieved and knowledge maintained without any hassle.

➤ Operating Environment

The current product can run in any system having a basic platform to run Microsoft Windows and VB/ VC++ runtime environment. Same computer has been used in the prototype as the database server however a separate system can be deployed as a database server, depending on load of the server. The separate machine for database will also facilitate a distributed design which will ease the process of taking backups/ ensuring enough redundancy and security.

However the product can be modified to run in a webserver, supporting PHP scripting and a database management system like MySQL. The servers should provide a robust and fast environment for the product. It should cater the needs of its many users both faculty and users and support parallel access without any lag and avoid downtime.

These possible features are required in the webserver where the product is supposed to be hosted

- Servers with Top Line Hardware specs.
- Dedicated MySQL server to handle SQL Queries load.
- Remote weekly backups of server’s data for safety of data.

System Features

➤ Search a book

➤ Description and priority

This is used to search a book using its various details. The details can include the various parameters as follows. Various security and data integrity features has been embedded to maintain the correctness of data being entered.

- Name of Book
- Authors
- Publisher
- Code of Subject

➤ Stimulus/Response Sequences

The server will receive data from the form based interface and will execute an SELECT query for searching through the database. The results are displayed in the list box.

Screenshot

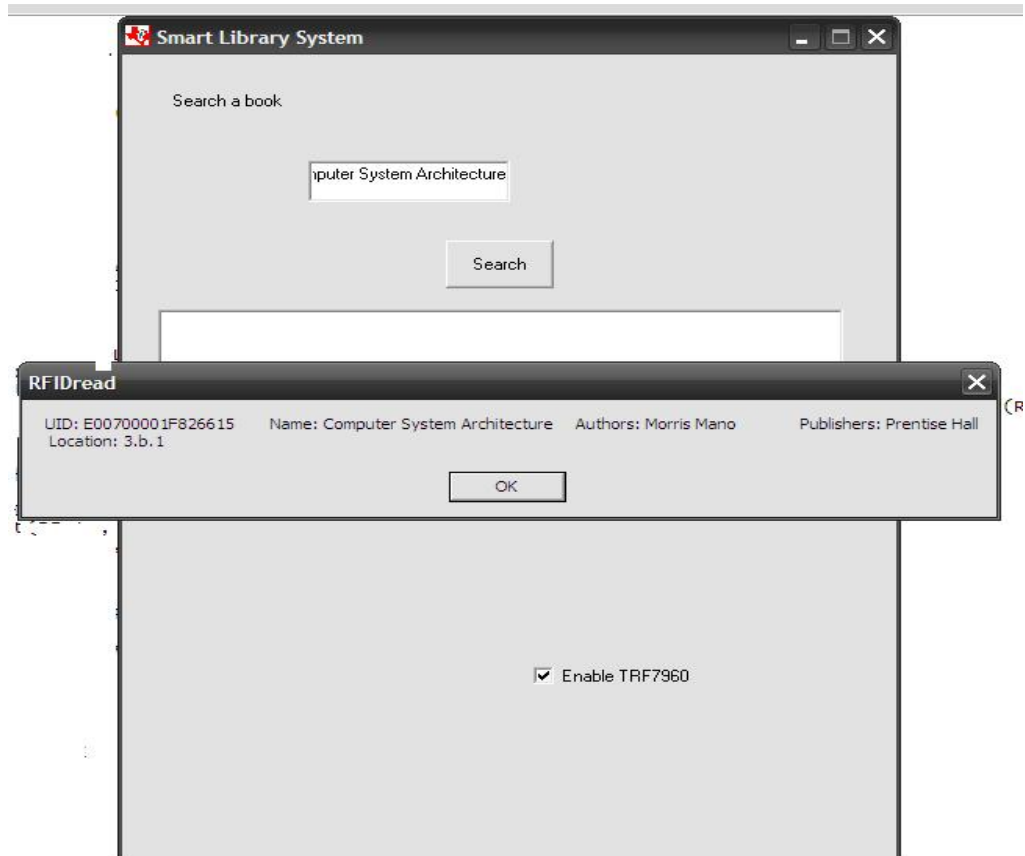


Fig 3(c) GUI to search books from database

- Issue Book on-the-fly
- Description and priority

As per our proposed innovative solution, the user has the provision to issue the book right on the shelf as soon as he picks up the book.

- Stimulus/Response Sequences

As per the protocol, the system will require the user to identify himself/ herself by showing his card in front of the RFID sensor. Once the user gets identified he can see the following image on the screen depicting that system has identified the user and has granted him access.

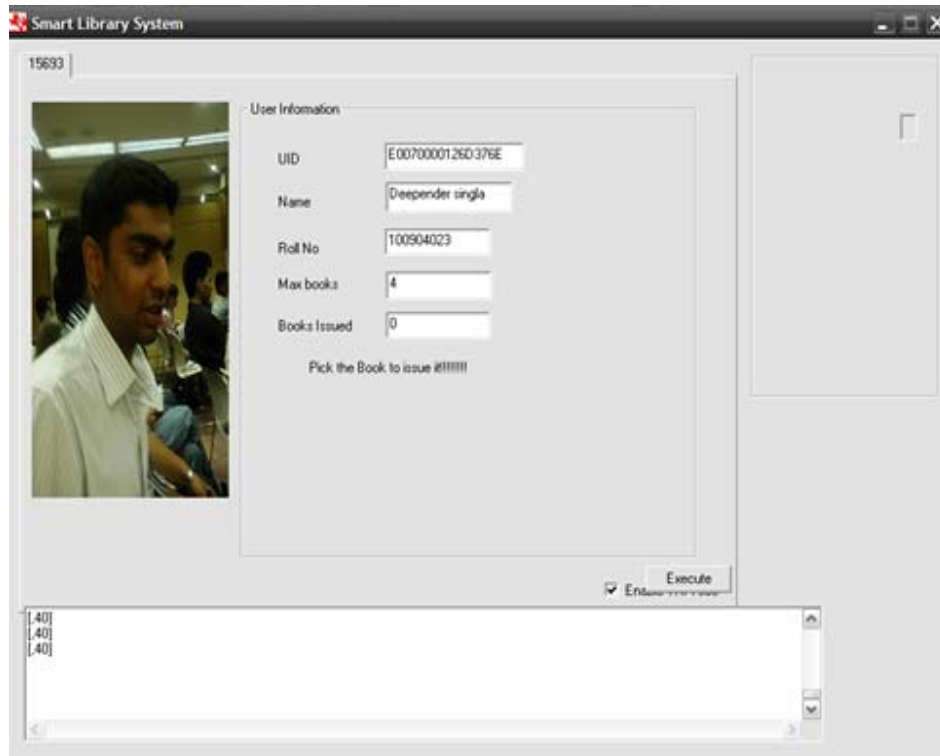


Fig 3(d) Welcome screen when User shows his card

Having granted the access, the user can pick any book from the rack and the system issues it instantly to the specified user. This action can be confirmed with the following image on the screen.

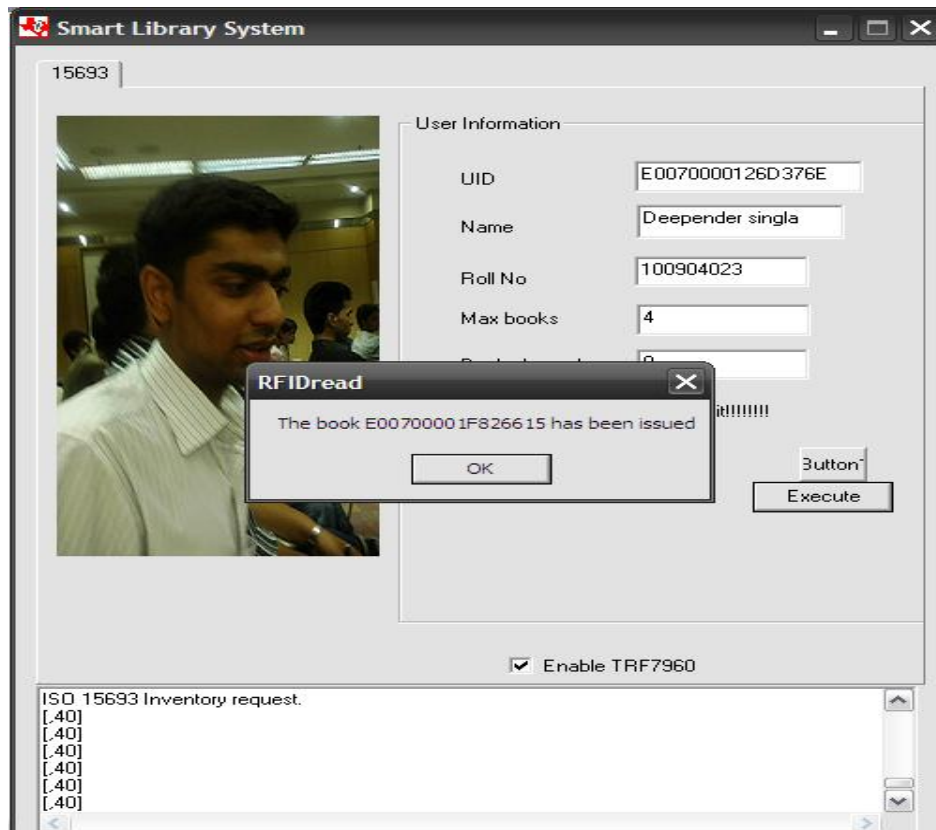


Fig 3(e) conformation of book issued

As soon as the book gets issued, the system automatically INSERT INTO the *issue_history* database the entry of the transaction for security and validation. This entry can later be viewed by administrator for various details.

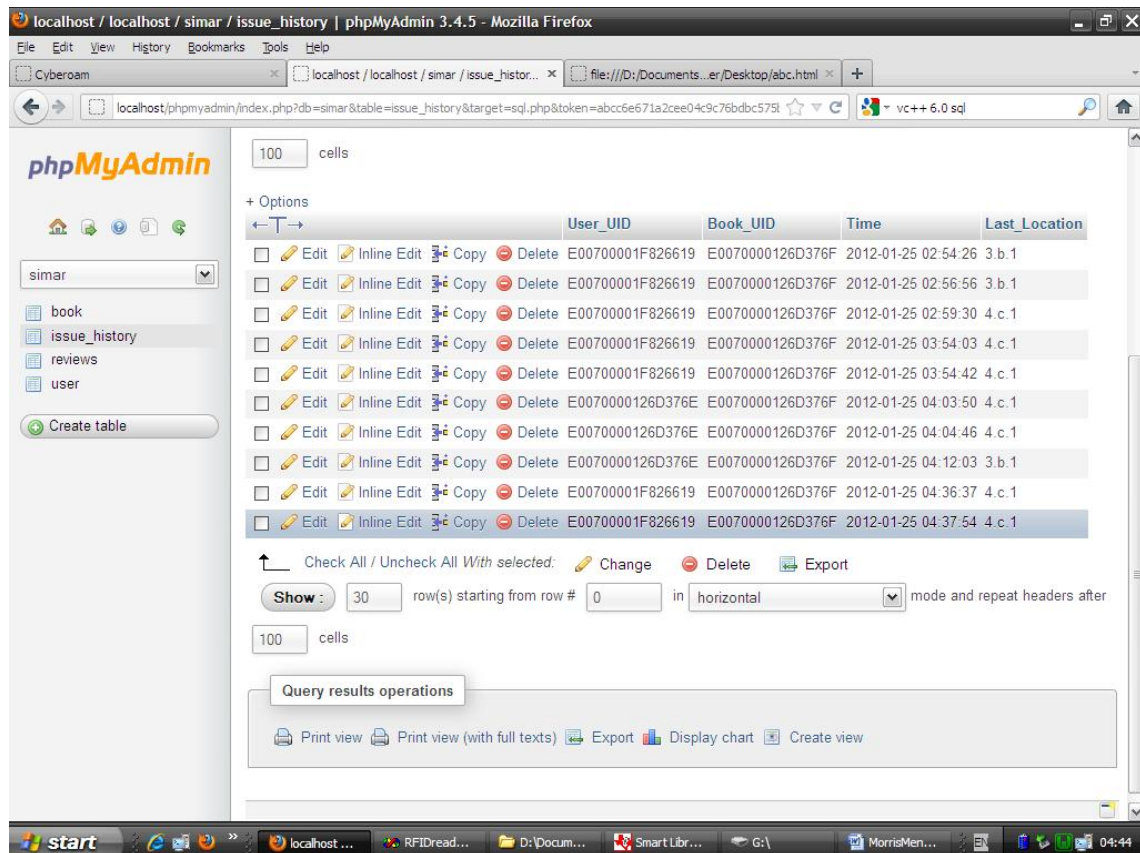


Fig 3(f) Change reflected in Database

Following which the user database is updated to change bi (number of books issued).

➤ Return the book on-the-fly

Description and priority

Just as in the case of issuing the book, returning the book is even simpler. The user will just come and place the book on the rack. The system (RFID sensors) will detect the book and will de-issue them, the system will calculate fine and will update the database accordingly.

Stimulus/Response Sequences

Now the interesting point here is that the system, even if the user does not place the book at its right place, the system will update the new location of the book and keep a track on the book. So any new user who comes for searching the book in the library will get its new updated location and will be able to locate the book very easily.

Screenshot

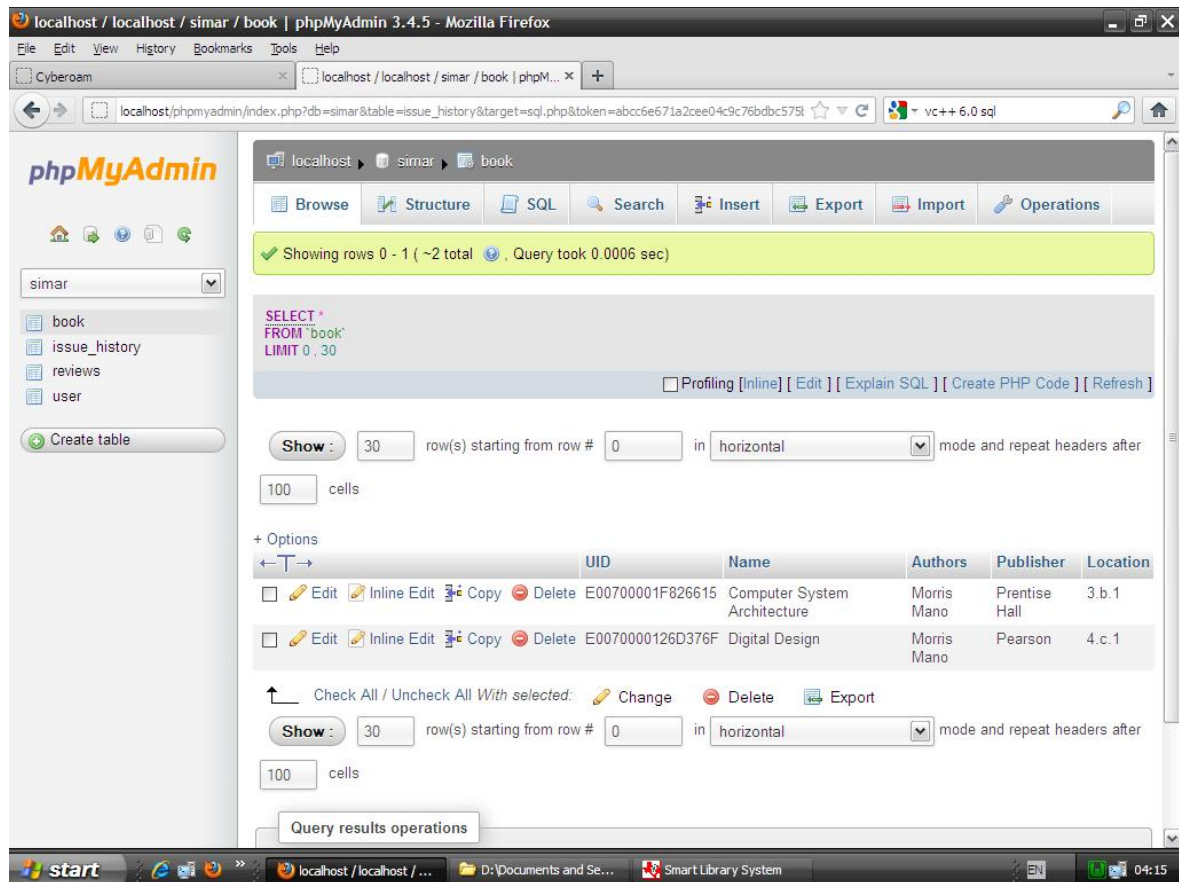


Fig 3(g) Change reflected in Book Database

The new location of the book Digital Design is updated in database, as soon as the user places the book in the 4.c.1 rack.

Hardware Interfaces

The server running software and maintaining database is required to run the product properly. The server should have sufficient hardware resources like primary memory, secondary memory to satisfy the needs of the server software and the database. Because the project is based on distributed model, the server should be equipped with a network interface from which it should be able to communicate using TCP/IP within the LAN/ WAN. Just as in case the system keeping track of RFID sensors communicate with the security admin's computer in case, if any user picks up a book without authenticating himself with the system.

Security considerations

Because library is essentially a public service system, security and integrity is a primary requirement for the proper working and implementation of any automation system inside the library. The system in the proposed solution immediately report of any such incident of book theft, if a user picks up a book before getting identified by the system and the alarm and various security cameras will automatically capture the incident.

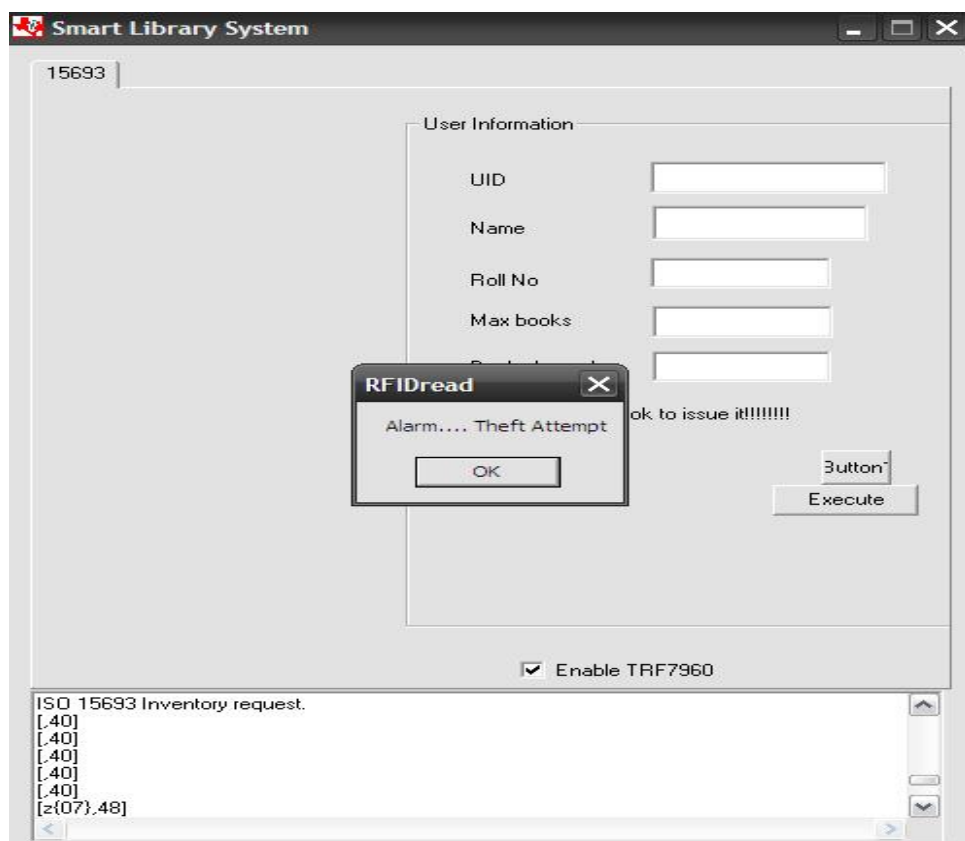


Fig 3(h) Theft Alarm

When the user gets himself identified, the system will issue the book the very instant, when he picks books. Now the book belongs to the user, so it is his responsibility. Thus

the new system is not only much easier, cheaper but has features of added security as well.

Software Quality Attributes

- Interoperability
- Maintainability
- Portability
- Reliability
- Reusability

Administrator Computer:

Admin computer always get a LAN message containing *description regarding whether a user card is shown or not and location where a user is detected*. LAN message is processed and two algorithm run simultaneously. *If a card is shown, face detection and recognition is performed to verify the identity of the user. In case user doesn't show card, admin is alerted by a audio message along with the specific location. Simultaneously camera rotates to the desired location*

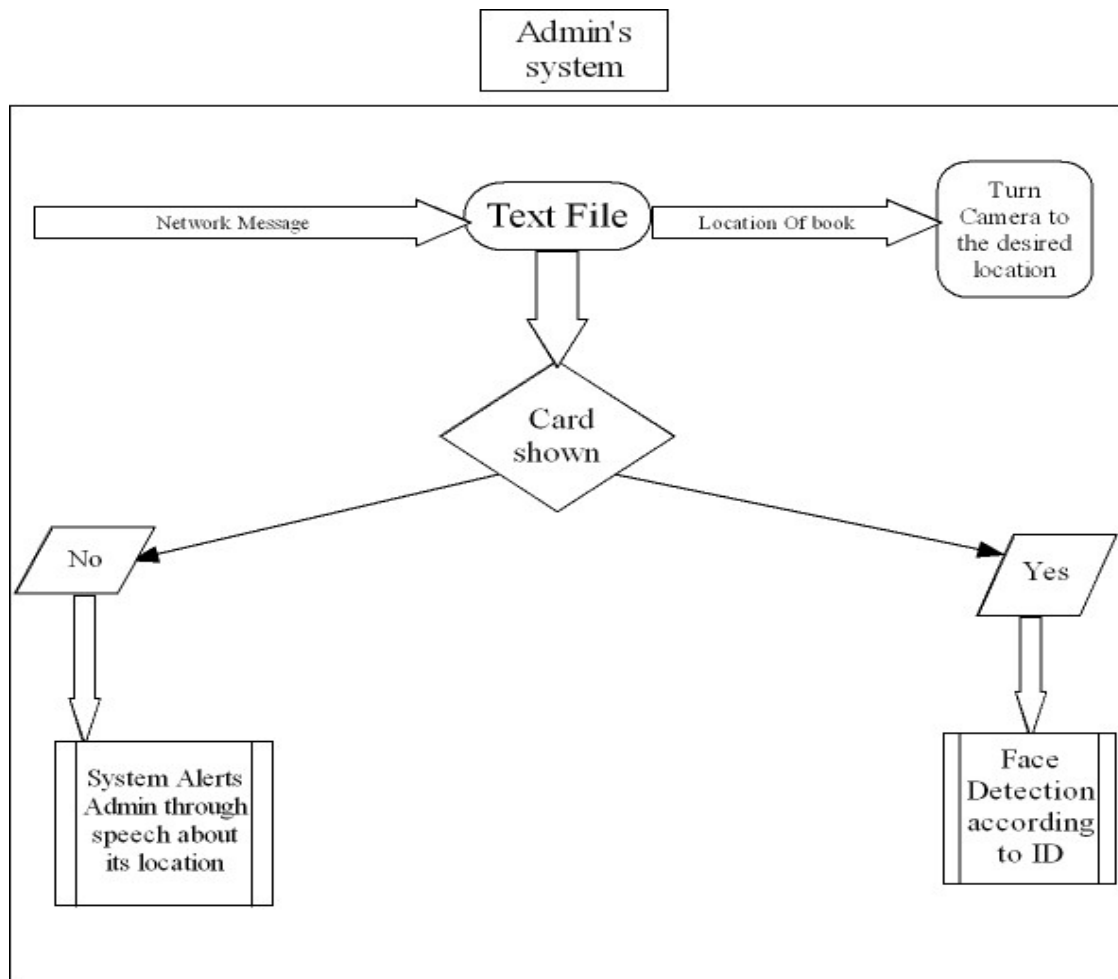


Fig 3(i) Flow Diagram for Admin's system

Face Detection is performed using openCV library in python, this algorithm is still being refined and needs thorough overhaul for real life set-up.

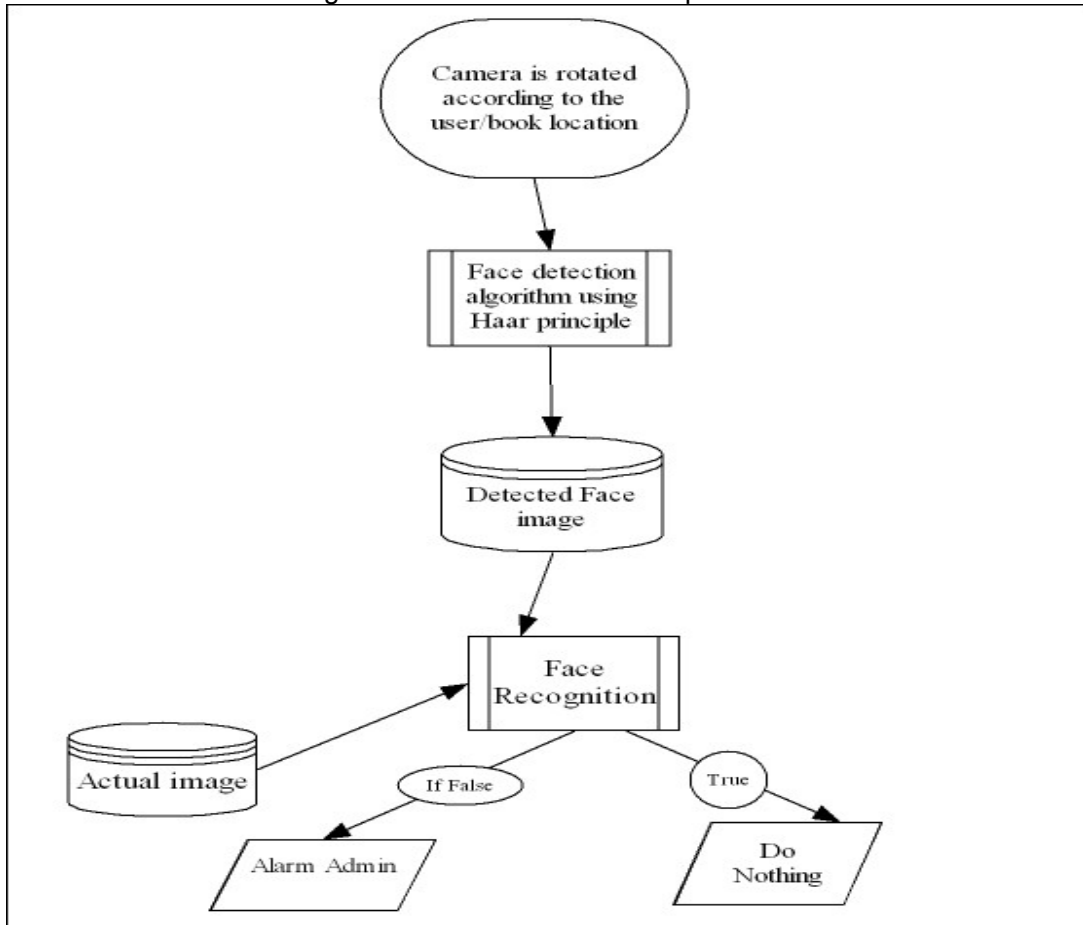


Fig 3(j) Face Detection and Recognition Flow Diagram

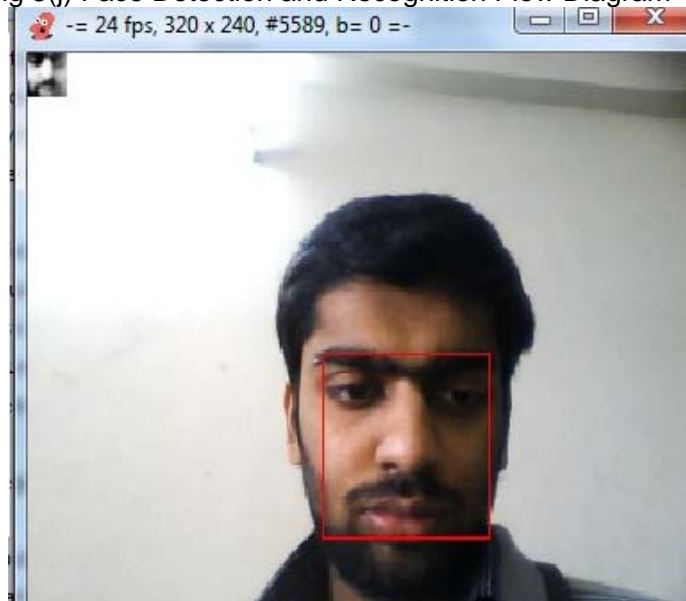


Fig 3(k) Face Detection in admin's computer

3.3 UML DIAGRAMS

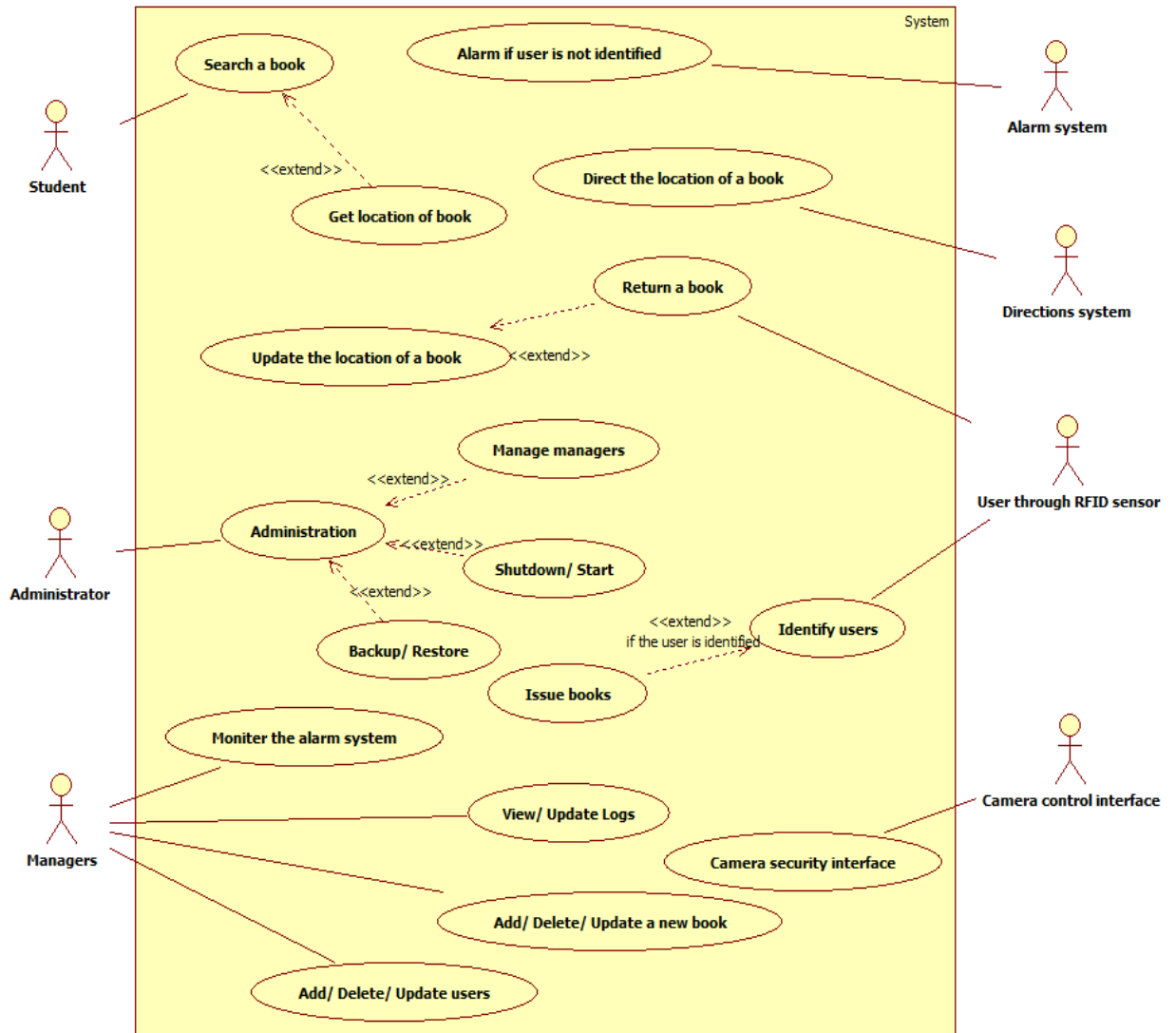


Fig 3(l) Use case diagram

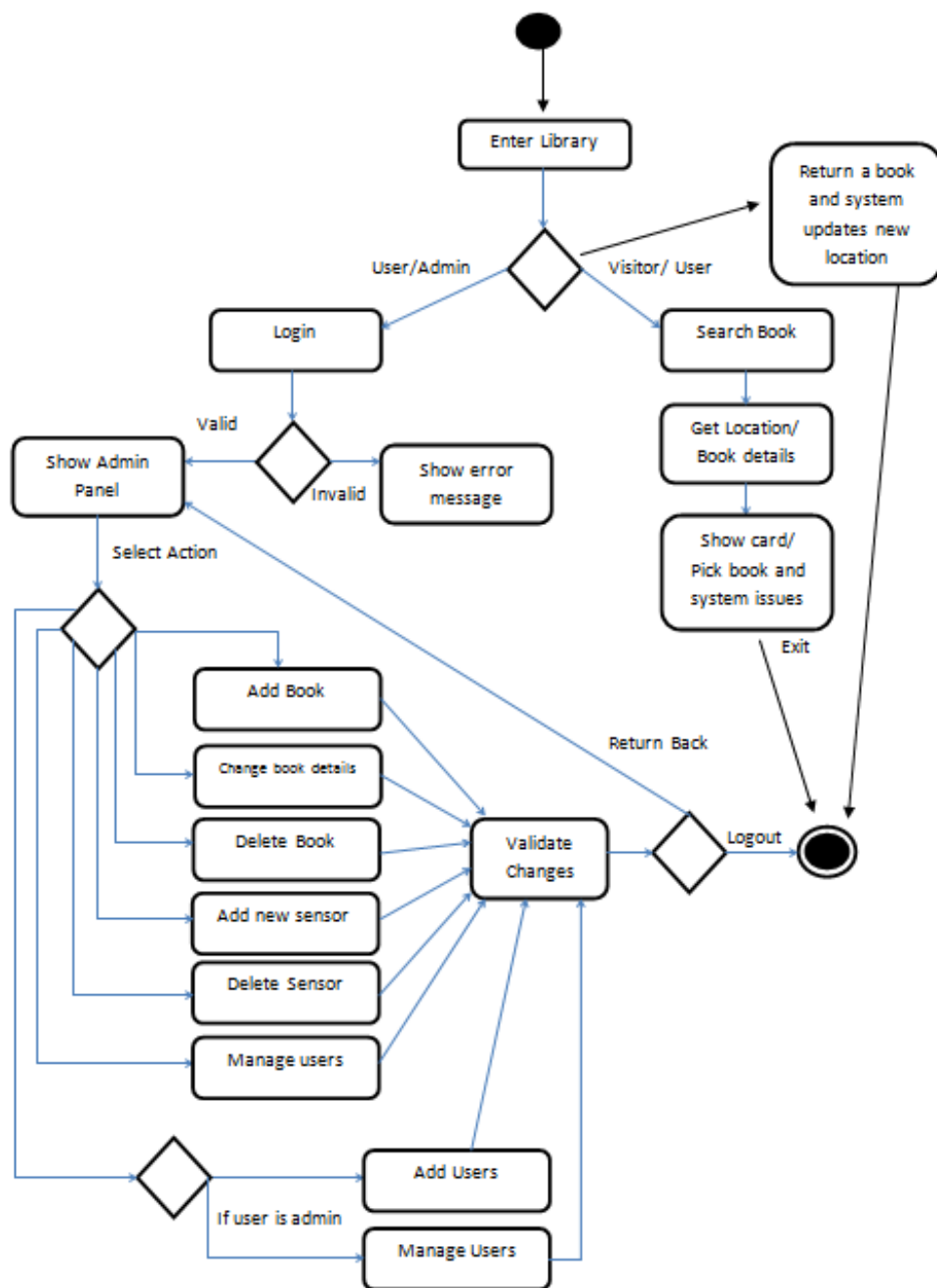


Fig 3.(m)Activity Diagram

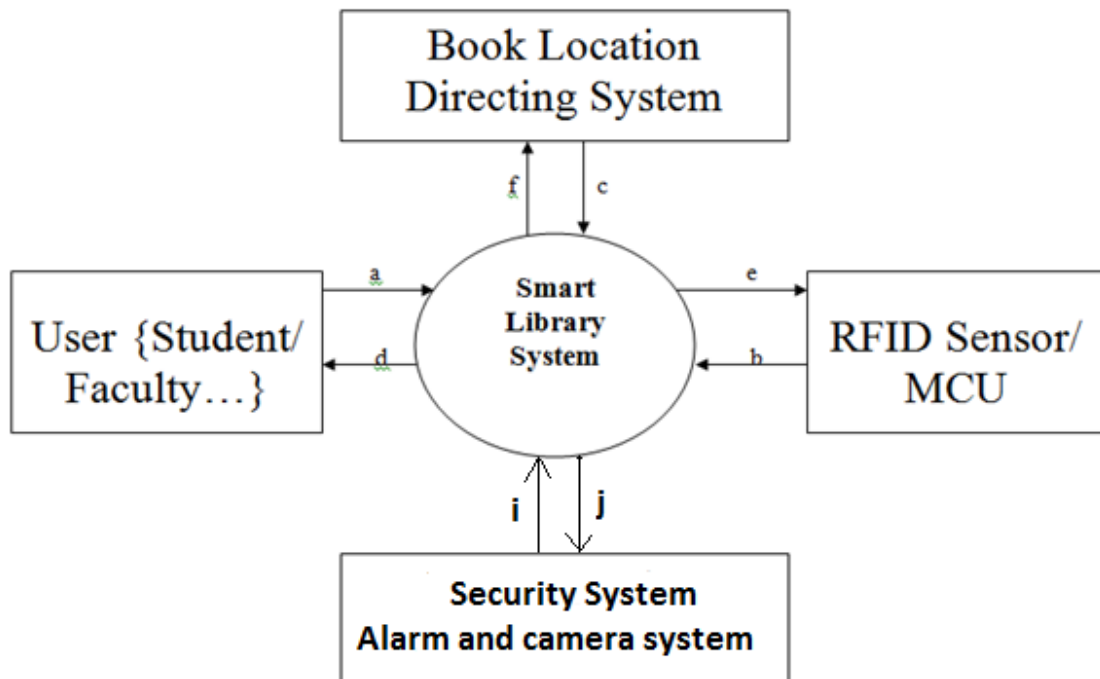


Fig 3.(n) Context Free diagram

State Machine Diagram

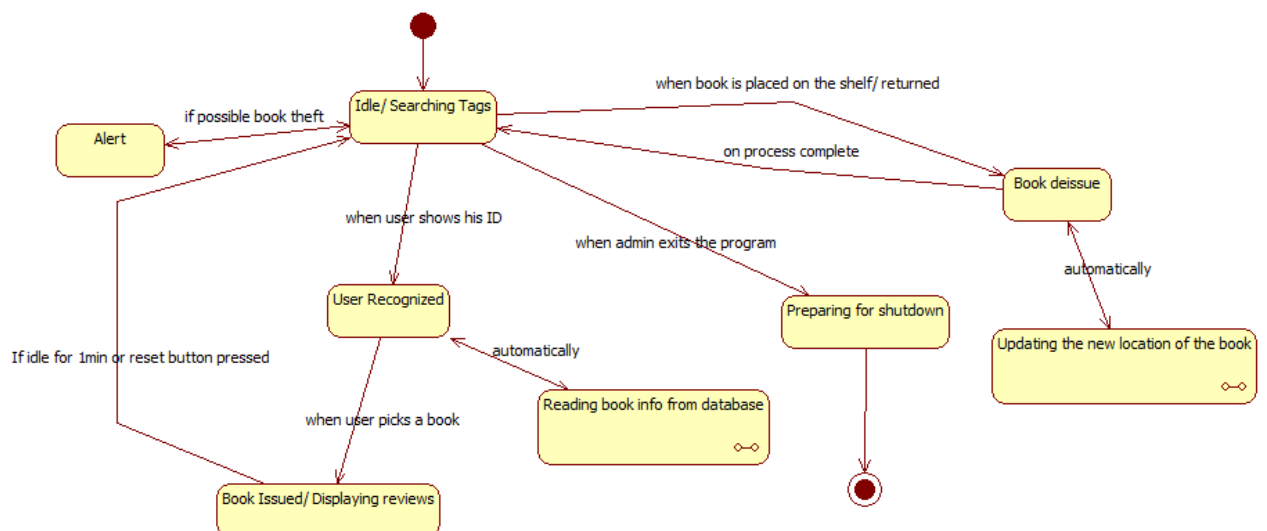


Fig 3.(o) RFID Object

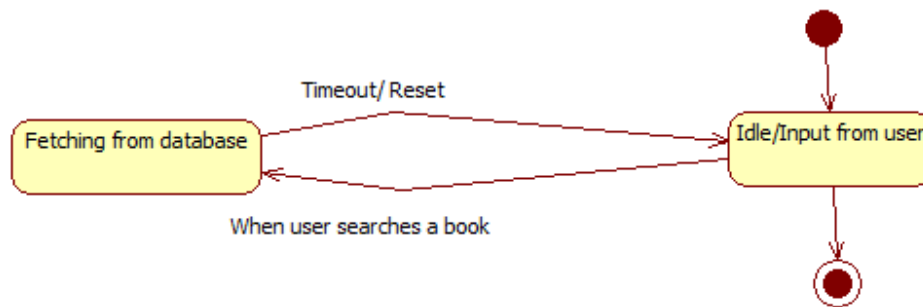


Fig 3.(p) Search Object

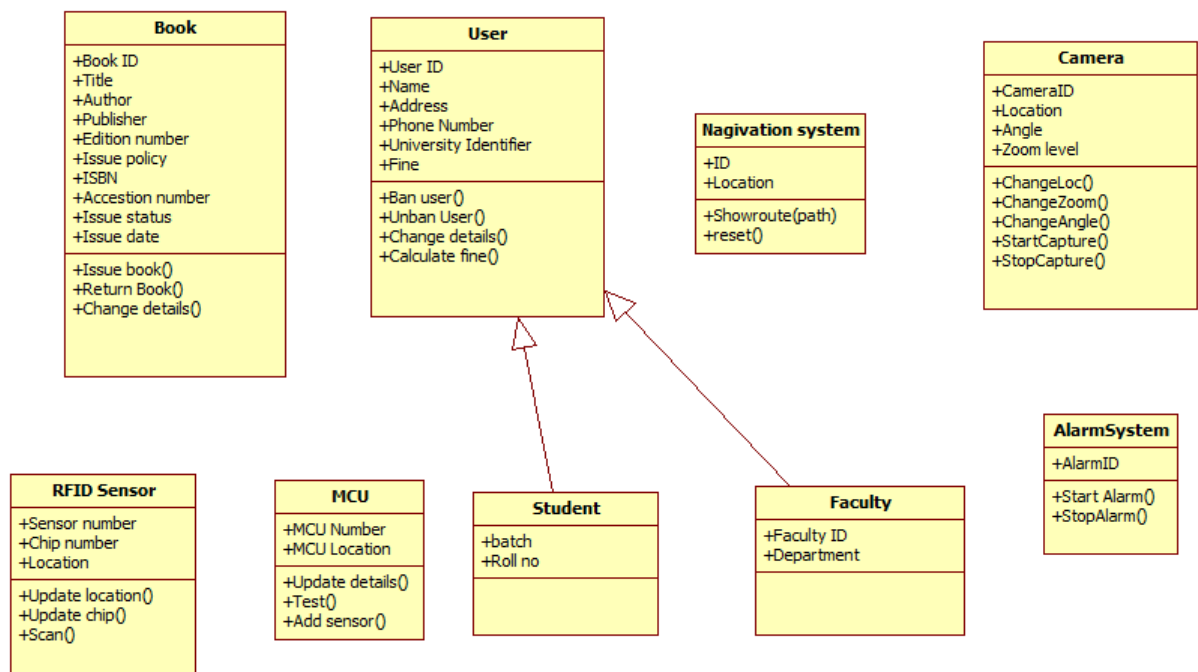


Fig 3.(q) Class diagram

4. Cost of Hardware and software used in the project:

	Component	Manufacturer	Cost per component	Quantity	Total cost of component	TI Supplied/ Purchased
1	TRF7960A	TI	\$3.25	2	\$6.5	Ti Supplied
2	RI-I02-112A	TI	\$.34	10	\$3.40	Ti Supplied
3	MSP430F2370	TI	\$1.6	2	\$3.2	Ti Supplied
4	RI-I11-112B-03	TI	\$.30	3	\$1	Ti Supplied
5	LM317DCY	TI	\$.43	4	\$2	Ti Supplied
6	Atmega-168	Atmel Corporation	250	1	250	Purchased
7	Servo Motor	Robosoft System	500	1	500	Purchased
8	Web Camera	Kou	450	1	450	Purchased
9	Antennas	Self-Designed	250	2	500	Designing Expenditure

Software has entirely been developed by us and tools used for simulation and development purposes are mostly open sources.

Miscellaneous cost like fabrication of PCB , capacitors , resistors sum upto Rs1,500 and we have also used two TRF7960evm of Rs 5,000 each:

Total Cost=4005+10,000(Evm Cost)=Rs14,005

Taking exchange rate Rs50

4.1 BUDGET:

This section deals with the estimated budget to convert a simple RFID library to a smart library proposed by us. The whole detail is for library containing nearly 10,000 books with 26 shelf containing 250 compartment and 10 CCTV camera.

COMPONENT	Estimated COST
Sensor Cost with antenna	50,000
Servo Controller with motor	5,000
Software	Nil (As Developed by us)
DSP Processor	5,000
Total	60,000

We have assumed our University library as the model for the project. The above cost is for converting thapar university library to a fully automated library if books are tagged with passive RFID tags.

5. Results:

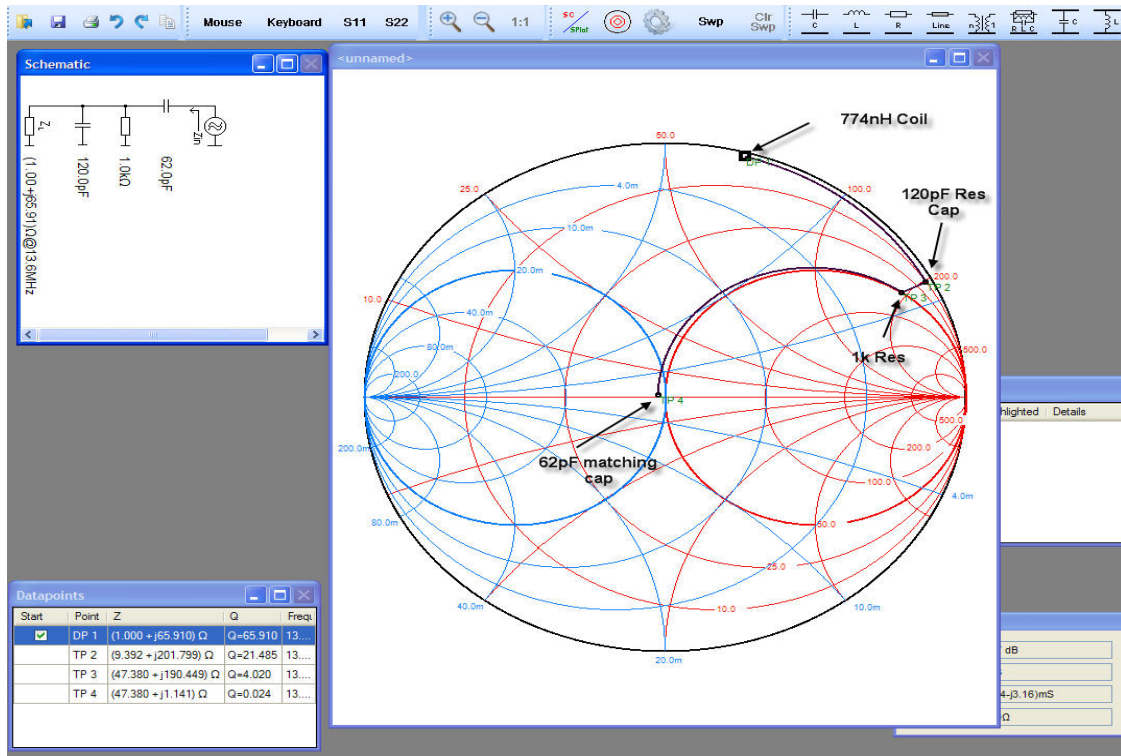


Fig 5.(a) Use of smith chart software for tuning values of external antenna to be attached.

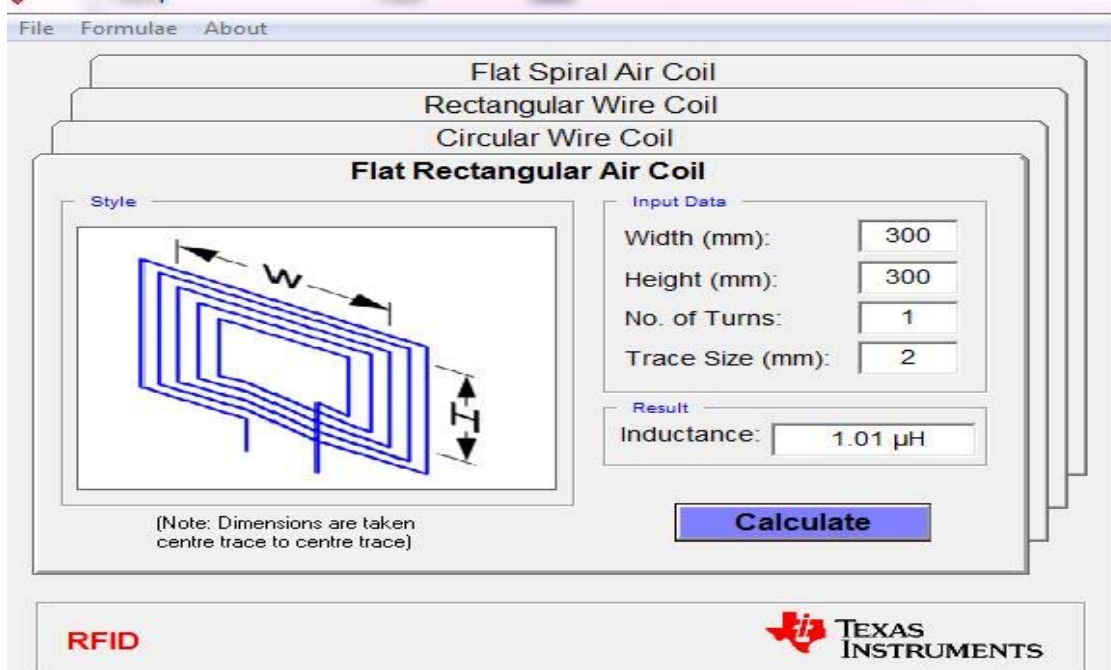
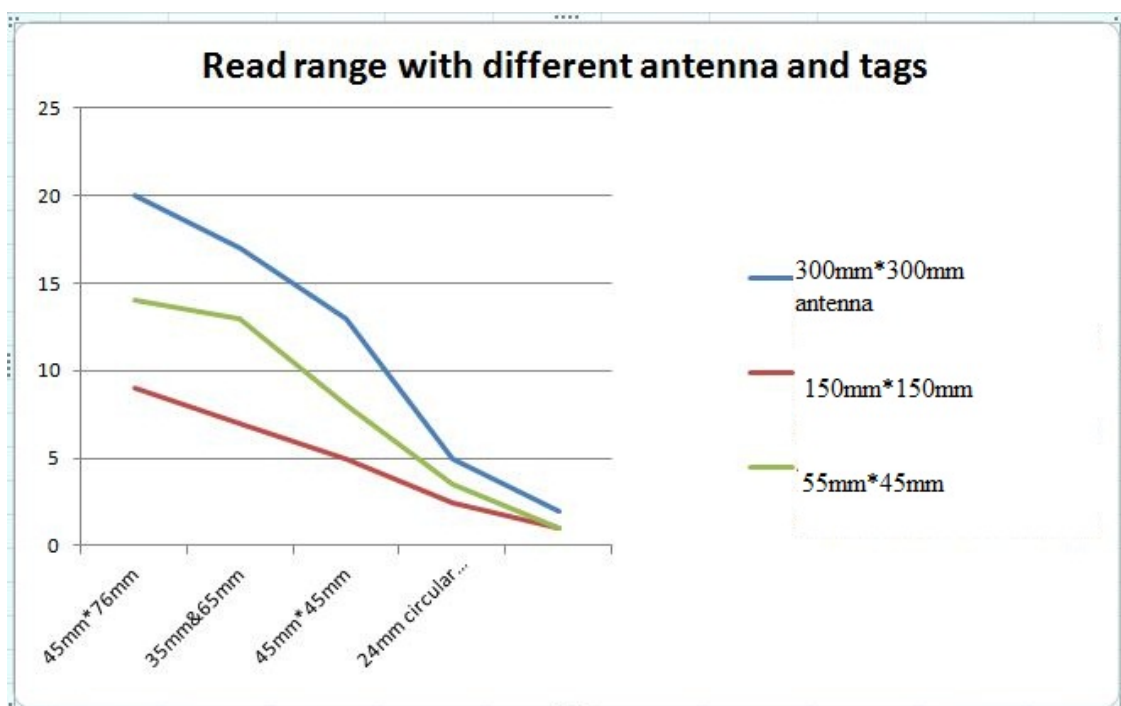
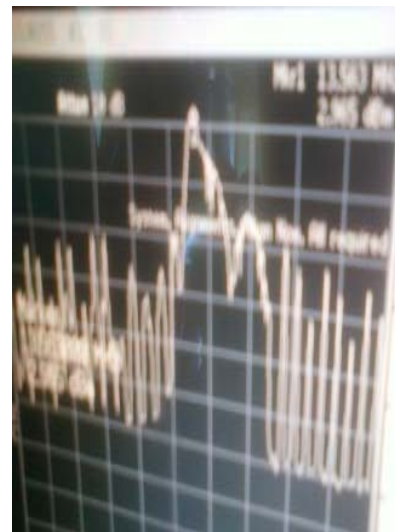
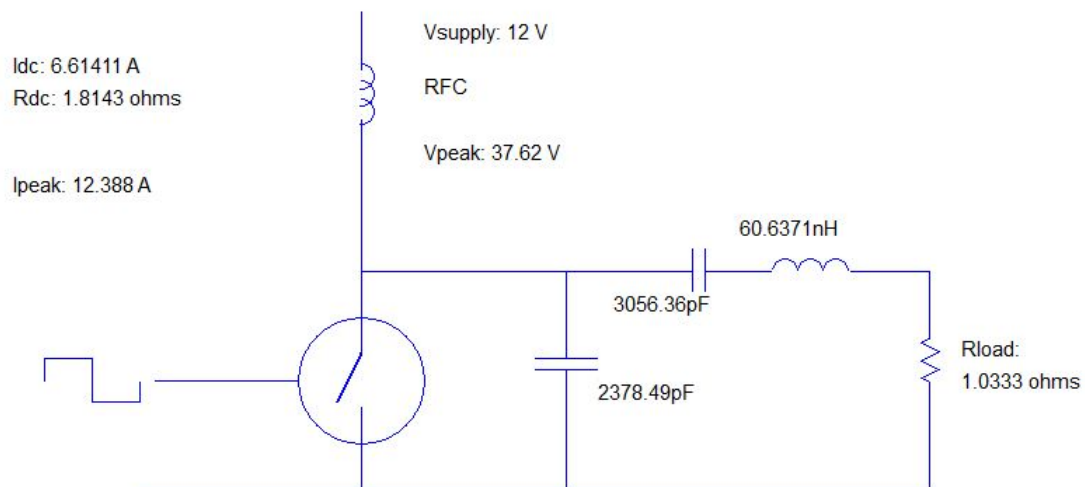


Fig 5(b) Calculation of Inductance for Antenna by using different combination





-1%	Frequency	+1%	13.56MHz
-1%	Supply voltage	+1%	12 V
-1%	Saturation voltage	+1%	2 V
-1%	Power	+1%	50 W
-1%	Network Q	+1%	5
-1%	Falltime	+1%	14ns

Fig 5.(e) Designing of Class E amplifier with Tonne software

Observations:

- 1.)The antenna is tuned slightly off frequency in order to prevent transmitter power from saturating receiver inputs.
- 2.)It is preferable to make system standalone using linux rather than using windows .
- 3.)It is better to go with one development platform for development of software part rather than using different platforms because ultimately it will create problem of synchronization.
- 4.)Better to draw reference design of ckt first in Simulation Software like (Pspice , LTspice) suite exactly as the reference design circuit , and play around with the circuit until you get a real feel for how it works.

6. Conclusions

Our main objective was to develop a library management system where the location of the books can be automatically updated even if the books have been misplaced. We have successfully accomplished this objective. In this new RFID based set-up, system automatically updates the location of the book.

Books are automatically issued as soon as a user lifts the book and as soon as he places the book back to any shelf, book is returned. This not only saves user from long queues but also minimizes the chances of theft and book tempering.

Face Detection algorithm developed is crude at present and needs further refinement to tackle real life situations.

The system should have been developed in linux to make it stand-alone.

The software has not been developed on a common platform.

Future scope of the project

- We believe that our solution will be a stepping stone for the next generation libraries. This work will provide plethora of scope for many more research in this field. We can explore new horizons and integrate applications like GPS, image processing etc to achieve complete automation.
- We would like to use this DSP platform for our face detection and recognition algorithm(Surveillance algorithm) in openCV , C6000 is a best choice for Porting openCV in embedded system like in our solution with its very high processing speed and this will also help us to minimize cost with its low cost compared to Intel Atom D-525 , basically this is one of the reasons we developed our algorithm in openCV because it is open source and also C6000 processor supports it. Even though we will sacrifice TBB support here but still C6000 is a multi- core processor with less memory requirement as compared to others which makes it the best choice to us.

- we strive to optimize antenna used in the project. In future we will use grid arrangement of antenna to locate books. We can find location of book through x, y coordinates.

For eg-

If antenna 1 from vertical direction and antenna 2 from horizontal direction detect the book simultaneously. It would imply that book location is (1,2). Hence lesser number of antenna will be required.



Fig6.(a) CAD view showing proposed location of antenna

7. Pictures of the Finished Product:



Fig 7.(a) Antenna 300mm *300mm with 2 mm trace width and 1 turn



Fig 8.(b) Servo controller

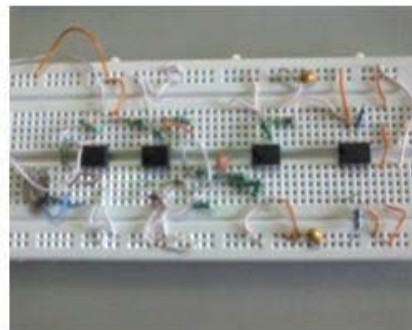


fig 8.(c) Class-E amplifier design on bread board



Fig 8.(d) Camera with Servo arrangement



Fig 8.(d) TRF7960A EVM

8.References

- [1]. N.O. Sokal and A.D. Sokal, "Class E- A New Class of High-Efficiency Tuned Single-Ended Switching Power Amplifiers," *IEEEJ. Solid-State Circuits*, Vol. SC-10, June 1975, pp. 168-176.
- [2.] N. O. Sokal and F. H. Raab, "Harmonic output of Class-E RF power amplifiers and load coupling network design," *IEEE J. Solid-State Circuits*, vol. SC-12, no.1, pp. 86-88, Feb. 1977.
- [3.] M.K. Kazimierczuk, "Class E Tuned Power Amplifier with Shunt Inductor," *IEEE J. Solid-State Circuit*, Vol. SC-16, February 1981, pp. 2-7.
- [4] Accenture's Mike Gorshe, Mary Rollman and Russ Beverly, "Item-Level RFID: A Competitive Differentiator" RFID journal-white paper
- [5.] Learning OpenCV, By Gary Bradski and Adrian Kaehler "O Reilly Publisher"

9. Appendix A

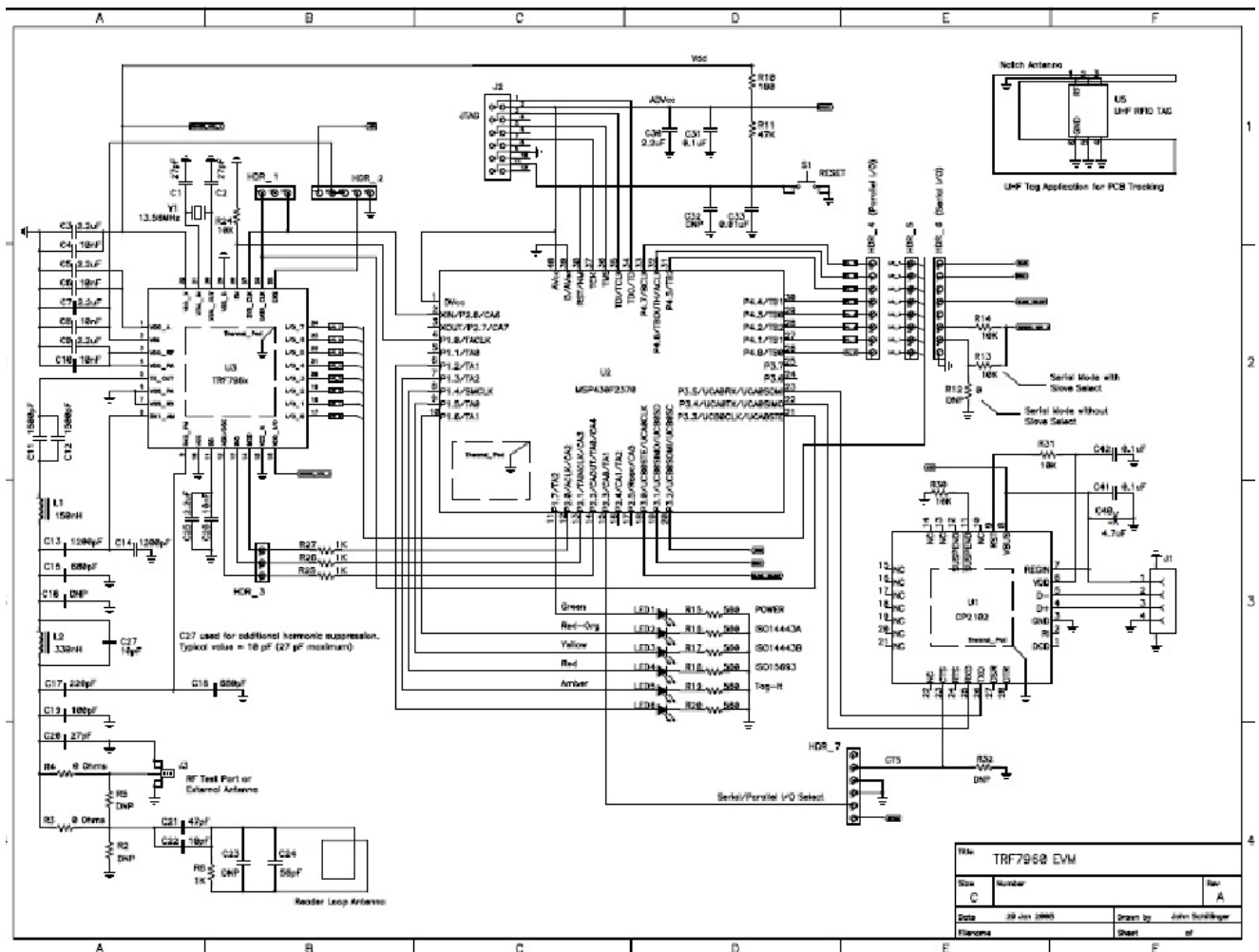
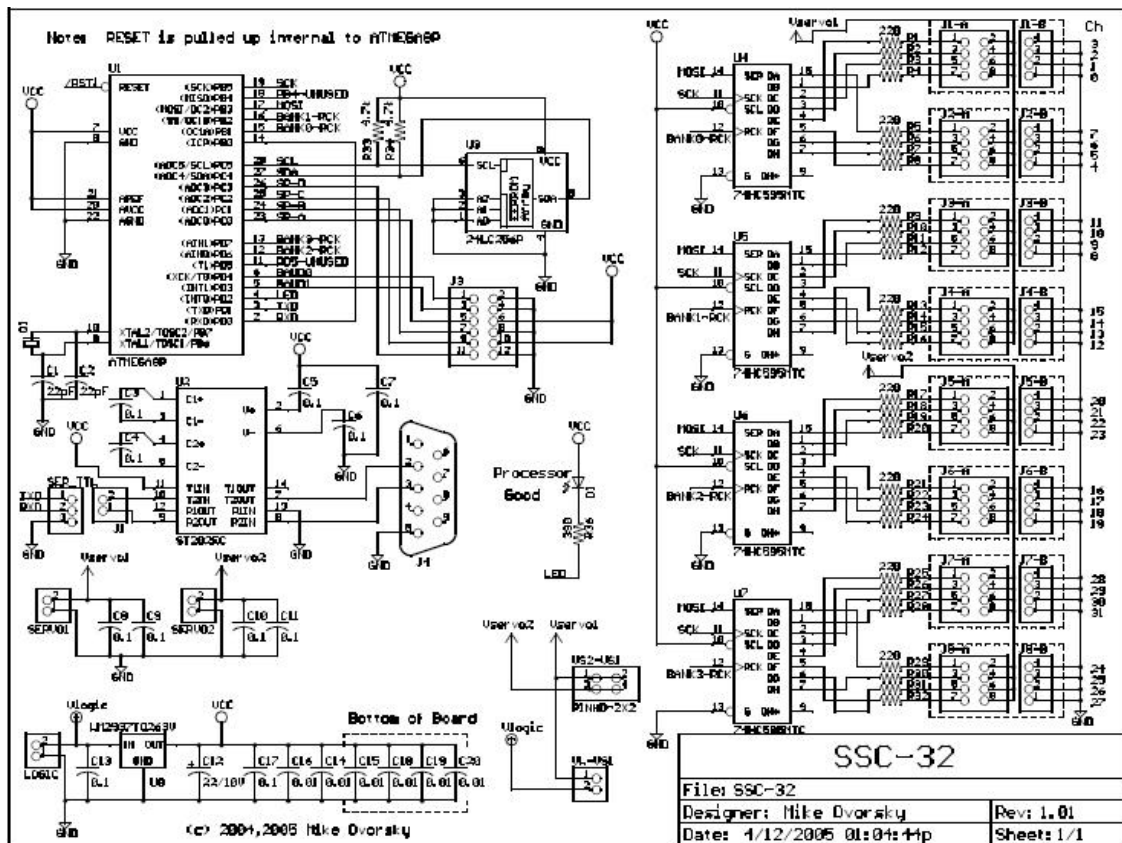


Fig 9.(A) PCB Schematic –TRF7960 EVM Source(Texas Instruments)



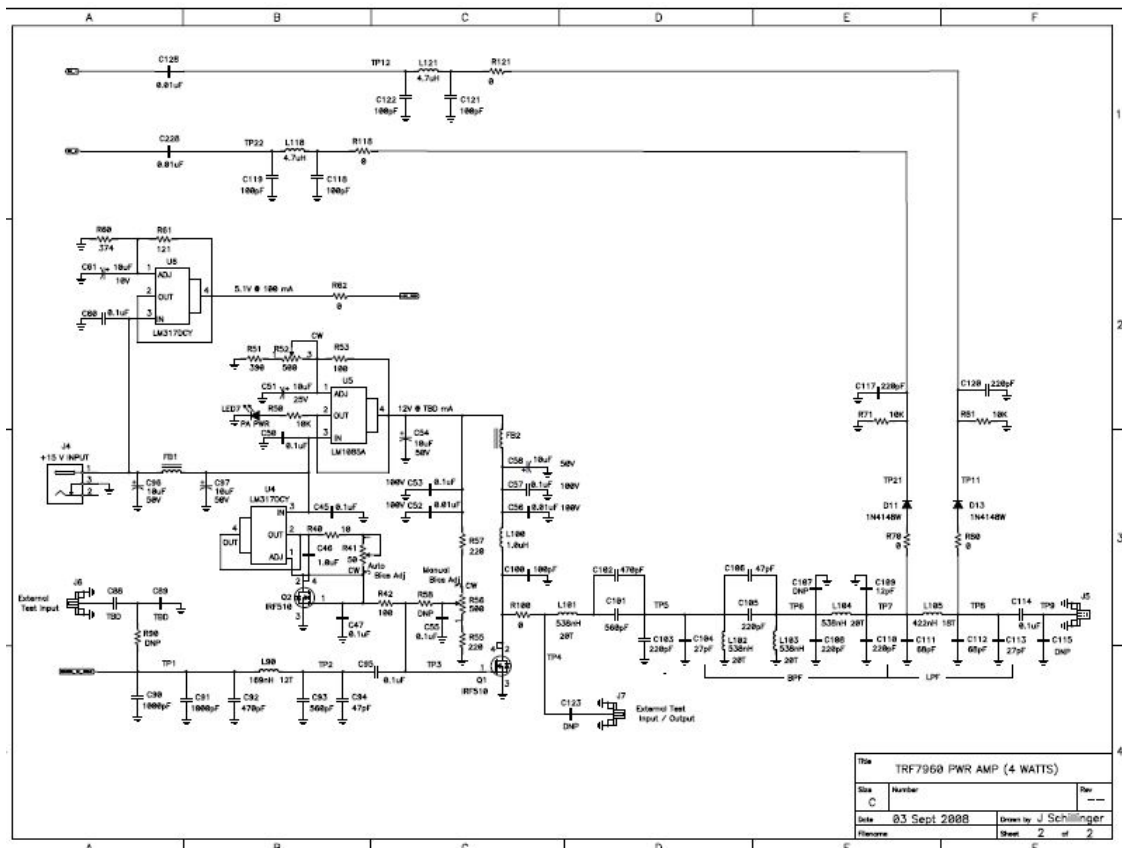


Fig.9.(C): 4 watt High Frequency Power Amplifier Souce (Texas Instruements)

10.Appendix B:

This is a part of the code that we have developed:

```
////////////////////////////////RFIDreadDlg.cpp////////////////////////////////

*/
void CRFIDreadDlg::OnButton1() //On clicking on 'Select Port' button
{
    /*~~~~~*/
    CString str;
    /*~~~~~*/

    //m_ComPortMan.GetWindowText(str); //Get text from EditBox
    str = "COM5";

    if(!strstr(str, "com") && !strstr(str, "COM"))
        MessageBox("Invalid Entry!", "Error", MB_OK);
    else
    {
        comPort.Format("\\\\.\\%s", str);
        portFindSinglePort(); //Open the port specified
    }

}

void CRFIDreadDlg::OnClear()
{
    // TODO: Add your control notification handler code here
    m_sLogger = " ";
    logBuf = "";
    UpdateData(FALSE);
}

////////////////////////////////Tab1593.cpp////////////////////////////////
#include "stdafx.h"
#include "RFIDread.h"
#include <string>
#include <cstdlib>
#include "Tab15693.h"
#include <iostream>
#include <time.h>

#include <sqlext.h>
```

```

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

#define DATA_LEN 256
#define MAX_UID 32
#define UID_LEN 68
#define DSFID_LEN 4

// 15693 Command codes
#define CC_INVENTORY "01"
#define CC_QUIET "02"
#define CC_READ_SB "20"
#define CC_WRITE_SB "21"
#define CC_LOCK "22"
#define CC_READ_MP "23"
#define CC_WRITE_MP "24"
#define CC_SELECT "25"
#define CC_READY "26"
#define CC_WRITE_API "27"
#define CC_LOCK_API "28"
#define CC_WRITE_DSFID "29"
#define CC_LOCK_DSFID "2A"
#define CC_GET_INFO "2B"
#define CC_GET_SEC_STAT "2C"
#define CC_WRITE_TWO_BLOCKS "A2"
#define CC_LOCK_TWO_BLOCKS "A3"

#define INVENTORY 0
#define READ_SB 1
#define WRITE_SB 2
#define LOCK_B 3
#define READ_MB 4
#define WRITE_MB 5
#define QUIET 6
#define SELECT 7
#define READY 8
#define WRITE_AFI 9
#define LOCK_AFI 10
#define WRITE_DSFID 11
#define LOCK_DSFID 12
#define SYSTEM_INFO 13
#define MBS_STATUS 14
#define PROTOCOL 15
#define FLAG 16
#define SET 17

#define WRITE_TWO_BLOCKS 15

```

```

#define LOCK_TWO_BLOCKS          16

#define MFG_CODE "07"

static char *commands[] = { "inventory",
                             "read_single_block",
                             "write_single_block",
                             "lock_block",
                             "read_multiple_blocks",
                             "write_multiple_blocks",
                             "stay_quiet",
                             "select",
                             "reset_to_ready",
                             "write_AFI",
                             "lock_AFI",
                             "write_DSFD",
                             "lock_DSFD",
                             "get_system_info",
                             "get_multi_blk_sel_status",
                             "set_protocol",
                             "flag",
                             "set",
                             NULL};

static char *dataCodingMode[] = { "4", "256",
                                   NULL};

static char *dataPowerMode[] = { "full", "half",
                                  NULL};

#define F_SUB_CARRIER 0
#define F_DATA_RATE 1
#define F_SELECT 2
#define F_ADDRESS 3
#define F_OPTION 4
#define F_AFI 5
#define F_ONE_SLOT 6

static char *flags[] = { "double_sub_carrier",
                         "high_data_rate",
                         "select",
                         "address",
                         "option",
                         "AFI_is_present",
                         "one_slot",
                         NULL};

#define D_UID 0
#define D_FIRST_B 1
#define D_NUM_BL 2
#define D_DATA 3

```

```

#define      D_DSFDID      4
#define      D_AFI      5
#define      D_CODING      6
#define      D_POWER      7

static char *dataKeys[] = {
    "UID",
    "first_block",
    "number_of_blocks",
    "data",
    "DSFDID",
    "AFI",
    "data_coding_mode",
    "power",
    NULL};

static char DSFDIDs[MAX_UID][DSFDID_LEN], UIDs[MAX_UID][UID_LEN];
static int selectedProtocol = -1,
          t15bSubCarrier, t15bDataRate, t15bInventory, t15bSelect,
          t15bAddress, t15bOption, t15bNextSlotC, t15iFullPower;
static CString t15UID, t15FirstBN, t15NumBI, t15Data, t15DSFDID, t15AFI,
          t15Protocol;

static bool set;

void changeByteOrder_MB(char *, char *, int);
////////////////////////////////////
// CTab15693 dialog

// modified my simar

SQLHANDLE hEnv;           // ODBC Environment handle
SQLHANDLE hDbc;           // ODBC Connection handle
SQLHANDLE hStmt;          // ODBC Statement handle

SQLHANDLE hOdbcEnv; //ODBC Environment handle
SQLHANDLE hDbConn; //ODBC Connection handle

void sleep(unsigned int mseconds)
{
    clock_t goal = mseconds + clock();
    while (goal > clock());
}

```



```

CTab15693::CTab15693(CWnd* pParent /*=NULL*/)
    : CDialog(CTab15693::IDD, pParent)
{
   //{{AFX_DATA_INIT(CTab15693)
    m_i15CommandRB = -1;
    m_s15NumBI = _T("");
    m_s15AFI = _T("");
    m_s15Data = _T("");
    m_s15DSFID = _T("");
    m_b15DataRate = FALSE;
    m_s15Protocol = _T("1 out of 4");
    m_s15FirstBN = _T("");
    m_b15Select = FALSE;
    m_s15UID = _T("");
    m_b15Option = FALSE;
    m_b15Address = FALSE;
    m_b15SubCarrier = FALSE;
    m_15UIDNo = _T("");
    m_s15TagBS = _T("");
    m_s15TagNB = _T("");
    m_i15FullPower = 0;
   //}}AFX_DATA_INIT
    selectedProtocol = -1;
}

void CTab15693::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CTab15693)
    DDX_Control(pDX, IDC_EDIT4, m_mbi);
    DDX_Control(pDX, IDC_EDIT5, m_bi);
    DDX_Control(pDX, IDC_EDIT3, m_rollno);
    DDX_Control(pDX, IDC_EDIT2, m_name);
    DDX_Control(pDX, IDC_EDIT1, m_uid);
    DDX_Control(pDX, IDC_PICTURE, m_Picture);
    DDX_Control(pDX, IDC_15_RSSI, m_15RSSI);
    DDX_Control(pDX, IDC_15_F_ADDRESS, m_15Address);
    DDX_Control(pDX, IDC_15_F_SELECT, m_15Select);
    DDX_Control(pDX, IDC_15_PROTOCOL, m_15Protocol);
    DDX_Control(pDX, IDC_15_UID, m_15UID);
    DDX_Radio(pDX, IDC_15_INVENTORY_RB, m_i15CommandRB);
    DDX_Text(pDX, IDC_15_NUM_BL, m_s15NumBI);
    DDV_MaxChars(pDX, m_s15NumBI, 2);
    DDX_Text(pDX, IDC_15_AFI, m_s15AFI);
    DDV_MaxChars(pDX, m_s15AFI, 2);
    DDX_Text(pDX, IDC_15_DATA, m_s15Data);
    DDX_Text(pDX, IDC_15_DSFID, m_s15DSFID);
    DDV_MaxChars(pDX, m_s15DSFID, 2);
    DDX_Check(pDX, IDC_15_F_DATA_RATE, m_b15DataRate);
    DDX_CBString(pDX, IDC_15_PROTOCOL, m_s15Protocol);
    }
}

```

```

        DDX_Text(pDX, IDC_15_FIRST_BN, m_s15FirstBN);
        DDV_MaxChars(pDX, m_s15FirstBN, 2);
        DDX_Check(pDX, IDC_15_F_SELECT, m_b15Select);
        DDX_CBString(pDX, IDC_15_UID, m_s15UID);
        DDV_MaxChars(pDX, m_s15UID, 16);
        DDX_Check(pDX, IDC_15_F_OPTIONS, m_b15Option);
        DDX_Check(pDX, IDC_15_F_ADDRESS, m_b15Address);
        DDX_Check(pDX, IDC_15_F_SUB_CARRIER, m_b15SubCarrier);
        DDX_Text(pDX, IDC_15_UID_NO, m_15UIDNo);
        DDX_Text(pDX, IDC_15_TAG_BS, m_s15TagBS);
        DDX_Text(pDX, IDC_15_TAG_NB, m_s15TagNB);
        DDX_Radio(pDX, IDC_15_FULL_POWER, m_i15FullPower);
    //}}AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CTab15693, CDialog)
   //{{AFX_MSG_MAP(CTab15693)
    ON_BN_CLICKED(IDC_15_SEND, On15Send)
    ON_BN_CLICKED(IDC_15_INVENTORY_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_EXECUTE, On15Execute)
    ON_BN_CLICKED(IDC_15_F_SELECT, On15FSelect)
    ON_CBN_SELCHANGE(IDC_15_UID, On15UID)
    ON_BN_CLICKED(IDC_TICC, OnTicc)
    ON_BN_CLICKED(IDC_15_READ_SB_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_WRITE_SB_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_LOCK_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_READ_MB_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_WRITE_MB_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_QUIET_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_SELECT_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_READY_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_WRITE_AFI_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_LOCK_AFI_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_WRITE_DSIF_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_LOCK_DSIF_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_SYS_INFO_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_15_SECURITY_RB, On15CommandRB)
    ON_BN_CLICKED(IDC_WRITE_TWO, On15CommandRB)
    ON_BN_CLICKED(IDC_LOCK_TWO, On15CommandRB)
    ON_BN_CLICKED(IDC_BUTTON1, OnButton1)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

```

////////////////////////////////////
// CTab15693 message handlers

```

```

void CTab15693::On15FSelect()
{
    UpdateData(TRUE); // Get curret values from the screen
}

```

```

        if (! m_i15CommandRB) {
            if (m_b15Select)
                GetDlgItem(IDC_15_AFI)->EnableWindow(TRUE);
            else
                GetDlgItem(IDC_15_AFI)->EnableWindow(FALSE);
        }
    }

void CTab15693::On15CommandRB()
{
    UpdateData(TRUE); // Get current values from the screen
    // Change 2 flag captions
    if (! m_i15CommandRB) {
        m_15Select.SetWindowText("AFI is present");
        m_15Address.SetWindowText("One slot");
    }
    else {
        m_15Select.SetWindowText("Select");
        m_15Address.SetWindowText("Addressed");
    }

    // Disable all command dependant controls
    GetDlgItem(IDC_15_UID)->EnableWindow(FALSE);
    GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(FALSE);
    GetDlgItem(IDC_15_NUM_BL)->EnableWindow(FALSE);
    GetDlgItem(IDC_15_DATA)->EnableWindow(FALSE);
    GetDlgItem(IDC_15_DSFIID)->EnableWindow(FALSE);
    GetDlgItem(IDC_15_AFI)->EnableWindow(FALSE);
    // Selectively enable controls
    switch (m_i15CommandRB) {
        case 0:
            if (m_b15Select)
                GetDlgItem(IDC_15_AFI)->EnableWindow(TRUE);
            break;
        case 1:
            GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
            GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
            break;
        case 2:
            GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
            GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
            GetDlgItem(IDC_15_DATA)->EnableWindow(TRUE);
            break;
        case 3:
            GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
            GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
            break;
        case 4:
            GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
            GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
            GetDlgItem(IDC_15_NUM_BL)->EnableWindow(TRUE);
    }
}

```

```

        break;
    case 5:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_NUM_BL)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_DATA)->EnableWindow(TRUE);
        break;
    case 6:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        break;
    case 7:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        break;
    case 8:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        break;
    case 9:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_AFI)->EnableWindow(TRUE);
        break;
    case 10:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        break;
    case 11:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_DSfid)->EnableWindow(TRUE);
        break;
    case 12:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        break;
    case 13:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        break;
    case 14:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_NUM_BL)->EnableWindow(TRUE);
        break;
    case 15:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_DATA)->EnableWindow(TRUE);
        break;

    case 16:
        GetDlgItem(IDC_15_UID)->EnableWindow(TRUE);
        GetDlgItem(IDC_15_FIRST_BN)->EnableWindow(TRUE);
        break;
}
}
////////// Code of face detection in python//////////

```

```

import sys
import cv

class FaceDetect():
    def __init__(self):
        cv.NamedWindow ("CamShiftDemo", 1)
        device = 0
        self.capture = cv.CaptureFromCAM(device)
        capture_size = (320,200)
        cv.SetCaptureProperty(self.capture, cv.CV_CAP_PROP_FRAME_WIDTH,
capture_size[0])
        cv.SetCaptureProperty(self.capture, cv.CV_CAP_PROP_FRAME_HEIGHT,
capture_size[1])

    def detect(self):
        cv.CvtColor(self.frame, self.grayscale, cv.CV_RGB2GRAY)

        #equalize histogram
        cv.EqualizeHist(self.grayscale, self.grayscale)

        # detect objects
        faces = cv.HaarDetectObjects(image=self.grayscale, cascade=self.cascade,
storage=self.storage, scale_factor=1.2,\
min_neighbors=2, flags=cv.CV_HAAR_DO_CANNY_PRUNING)

        if faces:
            #print 'face detected!'
            for i in faces:
                if i[1] > 10:
                    cv.Circle(self.frame, ((2*i[0][0]+i[0][2])/2,(2*i[0][1]+i[0][3])/2), (i[0][2]+i[0][3])/4,
(128, 255, 128), 2, 8, 0)

    def run(self):
        # check if capture device is OK
        if not self.capture:
            print "Error opening capture device"
            sys.exit(1)

        self.frame = cv.QueryFrame(self.capture)
        self.image_size = cv.GetSize(self.frame)

        # create grayscale version
        self.grayscale = cv.CreateImage(self.image_size, 8, 1)

        # create storage
        self.storage = cv.CreateMemStorage(128)
        self.cascade = cv.Load('haarcascade_frontalface_default.xml')

        while 1:
            # do forever
            # capture the current frame

```

```

self.frame = cv.QueryFrame(self.capture)
if self.frame is None:
    break

# mirror
cv.Flip(self.frame, None, 1)

# face detection
self.detect()

# display webcam image
cv.ShowImage('CamShiftDemo', self.frame)
# handle events
k = cv.WaitKey(10)

if k == 0x1b: # ESC
    print 'ESC pressed. Exiting ...'
    break
    sys.exit(1)

if __name__ == "__main__":
    print "Press ESC to exit ..."
    face_detect = FaceDetect()
    face_detect.run()

```