Palestine Polytechnic University

College of Engineering & Technology

Electrical and Computer Department

Graduation Project

Radio Frequency Identification technology on library

Project Team

Diyaa Isaeed

Mahmoud Zahda

Project Supervisor

Eng. Ayman Wazwaz

Hebron – Palestine

June,2010

Radio Frequency Identification technology

on libraries

Project Team

Diyaa Isaeed

Mahmoud Zahda

Project Supervisor

Eng. Ayman Wazwaz

This Under-Graduate Project Report submitted to Computer and Electrical engineering "Department in College of Engineering and Technology.

Palestine Polytechnic University

For accomplishment the requirements of the bachelor degree in Engineering field at Communication and Electronics Engineering.

Palestine Polytechnic University

Hebron – Palestine

December – 2009

جامعة بوليتكنك فلسطين

الخليل – فلسطين

كلية الهندسة والتكنولوجيا

دائرة الهندسة الكهربائية والحاسوب

اسم المشروع:

Radio Frequency Identification technology

on libraries

اسماء الطلاب:

محمود محمد زاهده ضياء وليد اسعيد

بناء على نظام كلية الهندسة والتكنولوجيا واشراف ومتابعة المشرف المباشر على المشروع ومتابعة اعضاء اللجنة الممتحنة تم تقديم هذا المشروع الى دائرة الهندسة الكهربائية والحاسوب وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة اتصالات والكترونيات.

توقيع المشرف:

....

توقع اللجنة الممتحنة:

....

توقيع رئيس الدائرة:

....

Dedication

To our parents who made it all possible

To our brothers and sisters for their courage

To Eng. Ayman Wazwaz

Who always encouraged and supported us

To all the people who like to know

and look for the knowledge

Diyaa isaeed
Mahoud Zahda

Acknowledgment

We acknowledge Palestine Polytechnic University for giving us the possibility to show some of what we have learned from it

And we acknowledge all the instructors in the electrical and computer department for their great impact in our education, especially the supervisor of this project eng. Ayman wazwaz, and every body who effected our education.

Finally we can’t forget to acknowledge our great parents who scarified themselves for educating us and facilate our life, and for all their coffee and tolerance.

Table of contents:

List of table:

ABSTRACT

The concept of Radio Frequency Identification (RFID) can be simplified to that of an electronic barcode and can be used to identify, track, sort or detect library holdings at the circulation desk and in the daily stock maintenance. This system, consist of smart RFID labels, hardware and software, provides libraries with more effective way of managing their collections while providing greater customer service to their patrons. The technology works through flexible, paper-thin smart labels, approximately "2cmX2cm" in size, which allows it to be placed on the inside cover of each book in a library's collection. The tag consists of an fixed antenna and a tiny chip which stores vital bibliographic data including a unique Accession number to identify each item. This contrasts with a barcode label, which does not store any information, but only points to a database. These smart labels are applied directly on library books and can be read with an RFID interrogator/scanner. Line of sight is not essential for reading the tags with the scanner, therefore, the books require much less human handling to be read and processed. A middleware or Savant software integrates the reader hardware with the existing Library Automation Software for seamless functioning of circulation.

CHAPTER ONE

INTRODUCTION

1.1 Problem overview.

1.2 Aim of the project.

1.3 Literature Review

1.4 Time Planning.

1.5 Estimated coast.

1.1 Problem overview:

RFID (Radio Frequency Identification) allows an item, for example a library book, to be tracked and communicated with by radio waves. This technology is similar in concept to a cell phone. RFID is a broad term for technologies that use radio waves to automatically identify people or objects. There are several methods of identification, but the most common is to store a serial number that identifies a person or object, and perhaps other information, on a microchip that is attached to an antenna (the chip and the antenna together are called an RFID transponder or an RFID tag). The antenna enables the chip to transmit the identification information to a reader. The reader converts the radio waves reflected back from the RFID tag into digital information that can then be passed on to computers that can make use of it.[1]

1.2 Aim of the project

RFID can be used library circulation operations and theft detection systems. RFID-based systems move beyond security to become tracking systems that combine security with more efficient tracking of materials throughout the library, including easier and faster charge and discharge, inventorying, and materials handling .

Fig 1.1 system overview

This technology helps librarians reduce valuable staff time spent scanning barcodes while charging and discharging items. RFID is a combination of radio -frequency-based technology and microchip technology. The information contained on microchips in the tags, affixed to library materials is read using radio frequency technology, regardless of item orientation or placement (i.e., the technology does not require line-of-sight or a fixed plane to read tags as do traditional theft detection systems).

1. To make it simple and easy way for customers in libraries, to self-checking for material loans and returns.
2. To make a fast scanning of the data stored on the RFID tag.
3. To reach better detection rates.
4. To reduce the number of false alarms at the exit sensors .
5. To make high speed inventory - reducing time to by staff for 'shelf-reading' and other inventory activities
6. Automated return of materials that speeds up sorting of materials and re-shelving for the next patron to access .
7. To make a longer life cycle than a barcode.

1.3 Literature Review:

RFID is becoming increasingly prevalent as the price of the technology decreases. An employee of Cisco says the company paid "well under ten cents" for each tag. The Japanese “HIBIKI “ initiative aims to reduce the price to 5 Yen (4 eurocents). And in January 2009 “Envego” announced a 5.9 cent tag.[2]

RFID Current Uses:

- Season Parking Ticket
- IT asset tracking
- Race timing
- Passports
- Mobile payment
- Product tracking
- Transportation and logistics
- Lap scoring
- Animal identification
- Inventory systems
- RFID mandates
- Promotion tracking
- Human implants

Our project uses RFID technology in libraries, we want to mention that there was some projects used the same application ,but our project include repeaters as a new feature to increase the coverage rang as increasing the speed of data transfer between reader and tags.

1.4 Time Planning

1.4.1 Time Schedule

The following table explains the expected timing plan.

Reading about the project	3 Weeks
Requirement Analysis	3 Weeks
System Design	4 Weeks
Calculations	4 Weeks
Documentation	12 Weeks

Table 1.1 Project Timing Plan Table.

1.4.2 Schedule Table

Weeks / Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Reading about the project															
Requirement Analysis															
System Design															
Calculations															
Documentation															

Table 1.2 Schedule Table.

1.5 Estimated coast:

1.5.1 hardware cost:

Component	Cost ($)
Transceivers	200
Antennas	100
Tags	0.05(per tag)
Reader components	50
Other HW components (IC's, wires ,chips, and…)	100
Total	450.05

1.5.2 Software cost:

Server	**1000**
Software	500
Database	500
Total	2000

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 The general block diagram.

2.2 Theory.

2.3 System data flow.

2.4 system data flow.

2.1 The general block diagram:

This is a simple block diagram for a RFID system which includes the tags on books ,repeaters placed on shelves ,a reader that communicate with tags through the repeater ,and finally the server which contains the data base that will be used by the librarian .

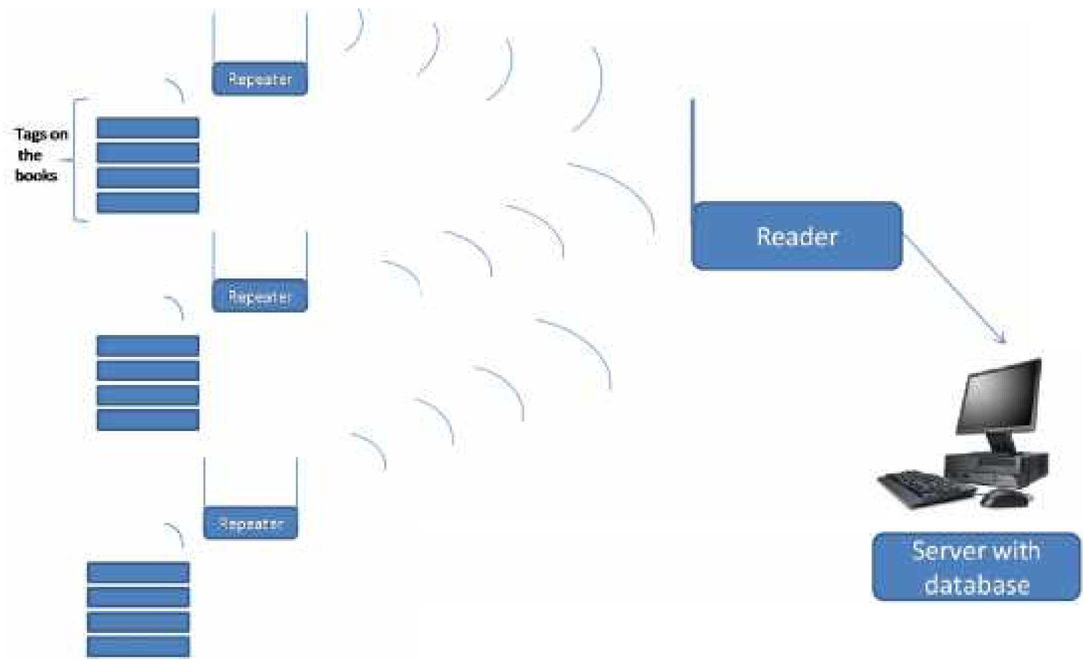


Fig 2.1 General block diagram.

This figure shows that Books are inserted by tags and repeaters amplify the tags signal ,communicate with the Reader .Reader can be connected to the server that contains the database.

2.2 Theory

2.2.1 Tag

An RFID tag is a device that can store, and transmit data to a reader ,in a contactless manner using radio waves.

RFID tags can be classified in two different ways. The following list shows the first classification, which is based on whether the tag contains an on-board power supply and/or provides support for specialized tasks: [2]

- Passive
- Active
- Semi-active (also known as semi-passive).

The following subsections discuss these in detail:

2.2.1.1 Passive Tags [2]

This type of RFID tag does not have an on-board power source (for example, a battery), and instead uses the power emitted from the reader, to energize itself and transmit its stored data to the reader. A passive tag is simple in its construction and has no moving parts. As a result, such a tag has a long life and is generally resistant to cruel environmental conditions. For example, some passive tags can resist corrosive chemicals such as acid, temperatures of 400°F (204°C approximately), and more.

In tag-to-reader communication for this type of tag, a reader always communicates first, followed by the tag. The presence of a reader is mandatory for such a tag to transmit its data.

A passive tag is typically smaller than an active or semi-active tag. It has a variety of read ranges starting with less than 1 inch to about 30 feet (9 meters approximately).

A passive tag is also generally cheaper compared to an active or semi-active tag.

A contactless smart card is a special type of passive RFID tag that is widely used today in various areas (for example, as ID badges in security and loyalty cards in retail). The data on this card is read when it is in close proximity to a reader. The card does not need to be physically in contact with the reader for reading.

A passive tag consists of the following main components:

- Microchip : as we will describe it in chapter three.
- Antenna : as we will also describe it in chapter three.

2.2.1.2 Active Tags [2]

Active RFID tags have an on-board power source (for example, a battery; other sources of power, such as solar, are also possible) and electronics for performing specialized tasks. An active tag uses its on-board power supply to transmit its data to a reader. It does not need the reader's emitted power for data transmission. The on-board electronics can contain microprocessors, sensors, and input/output ports powered by the on-board power source. Therefore, for example, these components can measure the surrounding temperature and generate the average temperature data. The components can then use this data to determine other parameters such as the expiry date of the attached item. The tag can then transmit this information to a reader (along with its unique identifier). You can think of an active tag as a wireless computer with additional properties (for example, like that of a sensor or a set of sensors).

In tag-to-reader communication for this type of tag, a tag always communicates first, followed by the reader. Because the presence of a reader is not necessary for data transmission, an active tag can broadcast its data to its surroundings even in the absence of a reader. This type of active tag, which continuously transmits data with or without the presence of a reader, is also called a transmitter. Another type of active tag enters a sleep or a low-power state in the absence of questioning by a reader. A reader wakes up such a tag from its sleep state by issuing an appropriate command. This state saves the battery power, and therefore, a tag of this type generally has a longer life compared to an active transmitter tag. In addition, because the tag transmits only when questioning, the amount of induced RF noise in its environment is reduced. This type of active tag is called a transmitter/receiver (or a transponder). As you can understand from this discussion, you cannot accurately call all tags transponders.

The reading distance of an active tag can be 100 feet (30.5 meters approximately) or more when the active transmitter of such a tag is used.

An active tag consists of the following main components:

- Microchip. The microprocessor size and capabilities are generally greater than the microchips found in passive tags.

- Antenna. This can be in the form of an RF module that can transmit the tag's signals and receive reader's signals in response. For a semi-active tag, this is composed of thin narrow piece(s) of metal such as copper, similar to that of a passive tag.
- On-board power supply.
- On-board electronics.

2.2.1.3 Semi-Active (Semi-Passive) Tags [2]

Semi-active tags have an on-board power source (for example, a battery) and electronics for performing specialized tasks. The on-board power supply provides energy to the tag for its operation. However, for transmitting its data, a semi-active tag uses the reader's emitted power. A semi-active tag is also called a battery-assisted tag. In tag-to-reader communication for this type of tag, a reader always communicates first, followed by the tag. Why use a semi-passive tag over a passive tag? Because a semi-active tag does not use the reader's signal, unlike a passive tag, to excite itself, it can be read from a longer distance as compared to a passive tag. Because no time is needed for energizing a semi-active tag, such a tag could be in the read zone of a reader for considerably less time for its proper reading (unlike a passive tag). Therefore, even if the tagged object is moving at a high speed, its tag data can still be read if a semi-active tag is used. Finally, a semi-active tag might offer better readability for tagging and RF-absorbent materials. The presence of these materials might prevent a passive tag from being properly excited, resulting in failure to transmit its data. However, this is not an issue with a semi-active tag.

The reading distance of a semi-active tag can be 100 feet (30.5 meters approximately) under ideal conditions using a modulated backscatter scheme (in UHF and microwave).

Now we want to make a new classification is based on the capability, to support data rewrites:

- Read-only (RO)
- Write once, read many (WORM)
- Read-write (RW)

Both active and passive tags can be RO, WORM, and RW. The following sections discuss these classifications in detail.

2.2.1.4 Read Only (RO) [3]

An RO tag can be programmed (that is, written) just once in its lifetime. The data can be burned into the tag at the factory during the manufacturing stage. To accomplish this, the individual fuses on the tag microchip are burned permanently using a fine-pointed laser beam. After this is done, the data cannot be rewritten for the entire lifetime of the tag. Such a tag is also called factory programmed.

The tag manufacturer supplies the data on the tag, and the tag users typically do not have any control over it. This type of tag is good for small applications only, but is impractical for large manufacturing or when tag data needs to be customized based on the application. This tag type is used today in small pilots and business applications.

2.2.1.5 Write Once, Read Many (WORM) [3]

A WORM tag can be programmed or written once, which is generally done not by the manufacturer but by the tag user right at the time when the tag needs to be created. In practice, however, because of buggy implementation, it is possible to overwrite particular types of WORM tag data several times (about 100 times is not uncommon)! If the data for such a tag is rewritten more than a certain number of times, the tag can be damaged permanently. A WORM tag is also called field programmable.

This type of tag offers a good price-to-performance ratio with reasonable data security, and is the most common type of tag used in business today.

2.2.1.6 Read Write (RW) [3]

An RW tag can be reprogrammed or rewritten a large number of times. Typically, this number varies between 10,000 and 100,000 times and above! This rewritability offers a great advantage because the data can be written either by the readers or by the tag itself (in case of active tags). An RW tag typically contains a Flash or a FRAM memory device to store its data. An RW tag is also called field programmable or reprogrammable. Data security is a challenge for RW tags. In addition, this type of tag is most expensive to produce. RW tags are not widely used in today's applications, a fact that might change in the future as the tag technology and applicability increases with a decrease in tag cost.

2.2.2 Repeater:

A repeater work as a transceiver in the system, that receive the data form the reader and transmit it to the tag, as an opposite it is receive the signal form the tag and transmit it to the reader.

We will put each repeater over a group of book or at each shelve, as it’s necessary. At the way that we use a reader to read data from the tags we have some problems such as :

1. waste of time
2. less coverage rang.

2.2.3 Reader

An RFID reader, also called an interrogator, is a device that can read from and write data to compatible RFID tags. Thus, a reader also doubles up as a writer. The act of writing the tag data by a reader is called creating a tag. The process of creating a tag and uniquely associating it with an object is called commissioning the tag. Similarly, decommissioning a tag means to disassociate the tag from a tagged object and optionally destroy it. The time during which a reader can emit RF energy to read tags is called the duty cycle of the reader. International legal limits apply to reader duty cycles.

The reader is the central nervous system of the entire RFID hardware system establishing communication with and control of this component is the most important task of any entity which seeks integration with this hardware entity.

A reader has the following main components:

- Transmitter
- Receiver

- Microprocessor
- Memory
- Input/output channels for external sensors, actuators (Although, strictly speaking, these are optional components, they are almost always provided with a commercial reader.)
- Controller (which may reside as an external component)
- Communication interface
- Power

2.2.3.1 Transmitter

The reader's transmitter is used to transmit AC power and the clock cycle via its antennas to the tags in its read zone. This is a part of the transceiver unit, the component responsible for sending the reader's signal to the surrounding environment, and receiving tag responses back via the reader antenna(s). The antenna ports of a reader are connected to its transceiver component. One reader antenna can be attached to each such antenna port. Currently, some readers can support up to four antenna ports.

2.2.3.2 Receiver

This component is also part of the transceiver module. It receives analog signals from the tag via the reader antenna. It then sends these signals to the reader microprocessor, where it is converted to its equivalent digital form (that is, the digital representation of the data that the tag has transmitted to the reader antenna).

2.2.3.3 Microprocessor

This component is responsible for implementing the reader protocol to communicate with compatible tags. It is performs decoding and error checking of the analog signal from the receiver. In addition, the microprocessor might contain custom logic for doing low-level filtering and processing of read tag data.

2.2.3.4 Memory

Memory is used for storing data such as the reader configuration parameters and a list of tag reads. Therefore, if the connection between the reader and the controller/software system goes down, not all read tag data will be lost. Depending on the memory size, however, a limit applies as to how many such tag reads can be stored at any one time. If the connection remains down for an extended period with the reader reading tags during this downtime, this limit might be exceeded and part of the stored data lost (that is, overwritten by the other tags that are read later).

2.2.3.5 Input/output Channels for External Sensors

Readers do not have to be turned on for reading tags at all times. After all, the tags might appear only at certain times in the read zone, and leaving readers perpetually on would just waste the reader's energy. In addition, as mentioned previously, regulatory limits apply to the reader duty cycle, too. This component provides a mechanism for turning a reader on and off depending on external events. A sensor of some sort, such as a motion or light sensor, detects the presence of tagged objects in the reader's read zone. This sensor can then set the reader on to read this tag. Similarly, this component also allows the reader to provide local output depending on some condition via an annunciator (for example, sounding an audible alarm) or an actuator (for example, opening or closing a security gate, moving a robot arm, and so forth). Sensors, actuators .

2.2.3.6 Controller

A controller is an entity that allows an external entity, either a human or a computer program, to communicate with and control a reader's functions and to control annunciators and actuators associated with this reader. Often, manufacturers integrate this component into the reader itself (as firmware, for example). However, it is also possible to package this as a separate hardware/software component that must be bought together with the reader. Controllers are discussed in detail later in this chapter.

2.2.3.7. Communication Interface

The communication interface component provides the communication instructions to a reader that allow it to interact with external entities, via a controller, to transfer its stored data and to accept commands and send back the corresponding responses. You can assume that this interface component is either part of the controller or is the medium that lies between a controller and the external entities. This entity has important characteristics that make it necessary to treat this as an independent component. A reader could have a serial as well as a network interface for communication. A serial interface is probably the most widespread type of reader interface available, but next-generation readers are being developed with network interfaces as a standard feature. Sophisticated readers offer features such as automatic discovery by an application, embedded Web servers that allow the reader to accept commands and display the results using a standard Web browser, and so forth.

2.2.3.8 Power

This component supplies power to the reader components. The power source is generally provided to this component through a power cord connected to an appropriate external electrical outlet.

2.2.4 Power supply

2.2.4.1 On-Board Power Supply

All active tags carry an on-board power supply (for example, a battery) to provide power to its on-board electronics and to transmit data. If a battery is used, an active tag generally lasts for about 2 to 7 years depending on the battery life. One of the determining factors of the battery life is the data-transmission rate interval of the tag the larger the interval, the longer the battery and hence the tag life. For example, suppose that an active tag is made to transmit once every few seconds. If you increase this so that the tag transmits once every few minutes or even once every few hours, you extend the battery life. The on-board sensors and processors consume power and can shorten the battery life, too.

When the battery of an active tag is completely discharged, the tag stops transmitting messages. A reader that was reading these messages does not know whether the tag's battery has died or whether the tagged product has disappeared from its read zone unless the tag transmits its battery status to this reader.

2.2.4.2 On-Board Electronics

The on-board electronics allow the tag to act as a transmitter, and optionally allow it to perform specialized tasks such as computing, displaying the values of certain dynamic parameters, acting as a sensor, and so on. This component can also provide an option for connecting external sensors. Therefore, depending on the sensor type attached, such a tag can perform a wide variety of sensing tasks. In other words, the range of functionality of this component is virtually limitless. Note that as the functionality and hence the physical size of this component grows, the tag might grow in size. This growth is acceptable because no hard limit applies to the size of an active tag as long as it can be deployed (that is, properly attached to the object that needs to be tagged). This means active tags can be applied to a wide range of applications, several of which might not even exist today.

A stationary reader can generally operate in the following modes:

- Autonomous
- Interactive

2.2.4.3 Autonomous Mode

In autonomous mode, a reader continuously read tags in its read zone. Every time a tag is read, it is saved to a list, usually called a *tag list*. An item on the tag list is associated with what is generally called a persist time. If the associated tag cannot be read for a period of time exceeding its persist time, it is dropped from the tag list. An application running on a host machine can register itself to receive the tag list periodically. A tag list includes information such as the following:

- Unique tag identifiers

- Reading time
- How many times a particular tag has been read since it has been discovered (that is, first read by the reader)
- The antenna ID that read a particular tag
- Reader name

2.2.4.4 Interactive Mode

In interactive mode, a read receives and executes commands from an application running on a host machine or from a user using a vendor-supplied client to communicate with the reader. After the reader fully executes the current command, it waits for the next. A reader can execute a range of commands, from sending the current tag list to the command invoker to changing the reader's configuration parameters.

2.2.5 Radio Wave:

2.2.5.1 Low Frequency (LF)

Frequencies between 30 KHz and 300 KHz are considered low, and RFID systems commonly use the 125 KHz to 134 KHz frequency range. A typical LF RFID system operates at 125 KHz or 134.2 KHz. RFID systems operating at LF generally use passive ,have low data-transfer rates from the tag to the reader, and are especially good if the operating environment contains metals, liquids, dirt, snow, or mud (a very important characteristic of LF systems). Active LF tags are also available from vendors. Because of the maturity of this type of tag, LF tag systems probably have the largest installed base. The LF range is accepted worldwide.

2.2.5.2 High Frequency (HF)

HF ranges from 3 MHz to 30 MHz, with 13.56 MHz being the typical frequency used for HF RFID systems. A typical HF RFID system uses passive tags, has a slow data-transfer rate from the tag to the reader, and offers fair performance in the presence of metals and liquids. HF systems are also widely used, especially in hospitals (where it does not interfere with the existing equipment). The HF frequency range is accepted worldwide.

The next frequency range is called very high frequency (VHF) and lies between 30 and 300 MHz Unfortunately, none of the current RFID systems operate in this range. Therefore, this frequency type is not discussed any further.

2.2.5.3 Ultra High Frequency (UHF)

UHF ranges from 300 MHz to 1 GHz. A typical passive UHF RFID system operates at 915 MHz in the United States and at 868 MHz in Europe. A typical active UHF RFID system operates at 315 MHz and 433 MHz A UHF system can therefore use both active and passive tags and has a fast data-transfer rate between the tag and the reader, but performs poorly in the presence of metals and liquids (not true, however, in the cases of low UHF frequencies such as 315 MHz and 433 MHz). UHF RFID systems have started being deployed widely because of the recent RFID mandates of several large private and public enterprises, such as several international and national retailers, the U.S. Department of Defense, and so on. The UHF range is not accepted worldwide.

2.2.5.4 Microwave Frequency

Microwave frequency ranges upward from 1 GHz. A typical microwave RFID system operates either at 2.45 GHz or 5.8 GHz, although the former is more common, can use both semi-active and passive tags, has the fastest data-transfer rate between the tag and the reader, and performs very poorly in the presence of metals and liquids. Because antenna length is inversely proportional to the frequency, the antenna of a passive tag operating in the microwave range has the smallest length (which results in a small tag size because the tag microchip can also be made very small). The 2.4 GHz frequency range is called Industry, Scientific, and Medical (ISM) band and is accepted worldwide.

2.3 Communication Between a Reader and a Tag

Depending on the tag type, the communication between a reader and a tag can be one of the following:

- Modulated backscatter

- Transmitter type
- Transponder type

Before delving into the details of these communication types, it is important for you to understand the concepts of near field and far field.

The area between a reader antenna and one full wavelength of the RF wave emitted by the antenna is called “near field”. The area beyond one full wavelength of the RF wave emitted from a reader antenna is called” far field”. Passive RFID systems operating in LF and HF use near field communication, whereas those in UHF and microwave frequencies use far field communication. The signal strength in near field communication attenuates as the cube of the distance from the reader antenna. In far field, it attenuates as square of the distance from the reader antenna. As a result, far field communication is associated with a longer read range compared with near field communication.

A comparison between tag read and tag write :

Tag write takes a longer time than tag read under the same conditions because a write operation consists of multiple additional steps, including an initial verification, erasing any existing tag data, writing the new tag data, and a final verification phase. In addition, the data is written on the tag in blocks in multiple steps. As a result, a single tag write can take hundreds of milliseconds to complete and increases with the increase in data size. In contrast, several tags can be read in this time interval by the same reader. Also, tag write is a sensitive process that needs the target tag to be closer (compared to its corresponding read distance) to the reader antenna for the entire write operation. This closer proximity ensures the tag antenna can derive sufficient energy from the reader antenna signal to power its microchip so that it can execute the write instructions. The power requirement for write operation is generally significantly higher than that required for reading. The write operation might fail otherwise. However, a tag does not have to stay close to the reader during a read operation. Also, during tag write operation, any tag other than the target should not be in write range of the reader. Otherwise, in some cases, this other tag might accidentally get written rather than the target tag. This write range issue is clearly not relevant during a read operation, when multiple tags can exist in the read range of the reader at the same time.

2.4 System data flow:

This figure (fig2.5) explains how the system works in six steps :

1. a requesting of data command , coming from the server database has to initialize the process .

2. the receiver and transmit this request to the repeater .

3. the repeater arrives this data to the Tag ,then the tag is turned on . which means a request of Tag's data has been done.

4. The desired data is sent from the tag to the repeater .

5. the repeater repeat the data and the reader receive it .

6. finally the desired data will arrive to the data base and we can reach our desired book .

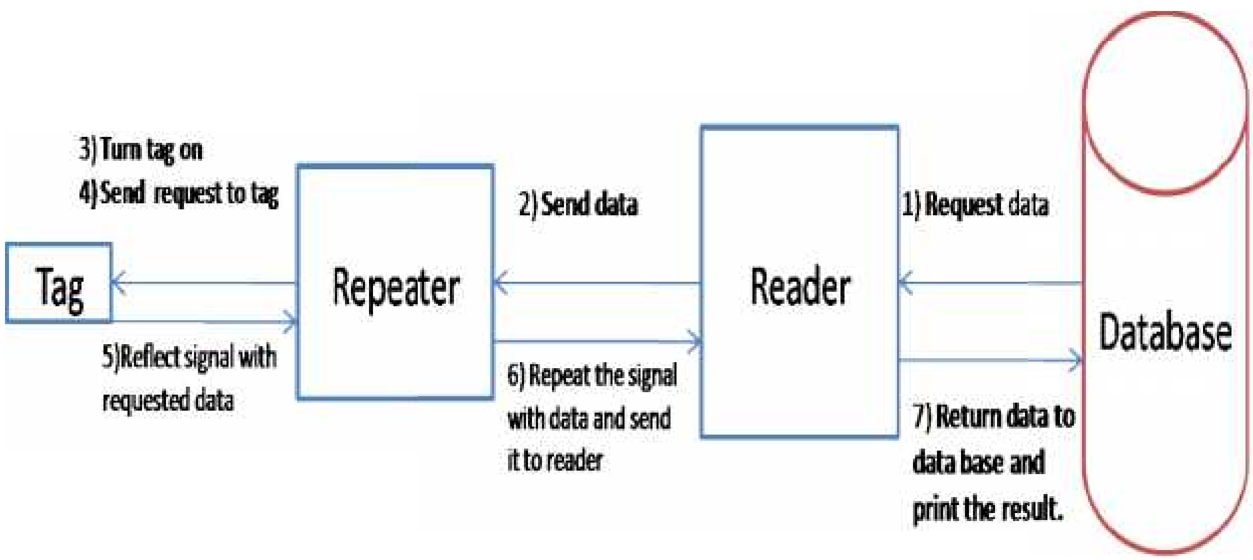


Fig 2.4 data flow of RFID system.

CHAPTER THREE

SYSTEM DESIGN

3.1 System Parts

3.2 System modeling

3.3 How the system works

3.4 Conclusion

3.1 System Parts

An RFID system is an integrated collection of components (in singular form) from an end-to-end view:

- Tag: This is a mandatory component of any RFID system.
- Reader: This is a mandatory component, too.
- Reader antenna: This is another mandatory component. Some current readers available today have built-in antennas.
- Controller: This is a mandatory component. However, most of the new-generation readers have this component built in to them.
- Sensor, actuator, and annunciator: These optional components are needed for external input and output of the system.
- Host and software system: Theoretically, an RFID system can function independently without this component. Practically, an RFID system is close to worthless without this component.
- Communication infrastructure: This mandatory component is a collection of both wired and wireless network and serial connection infrastructure needed to connect the previously listed components together to effectively communicate with each other.

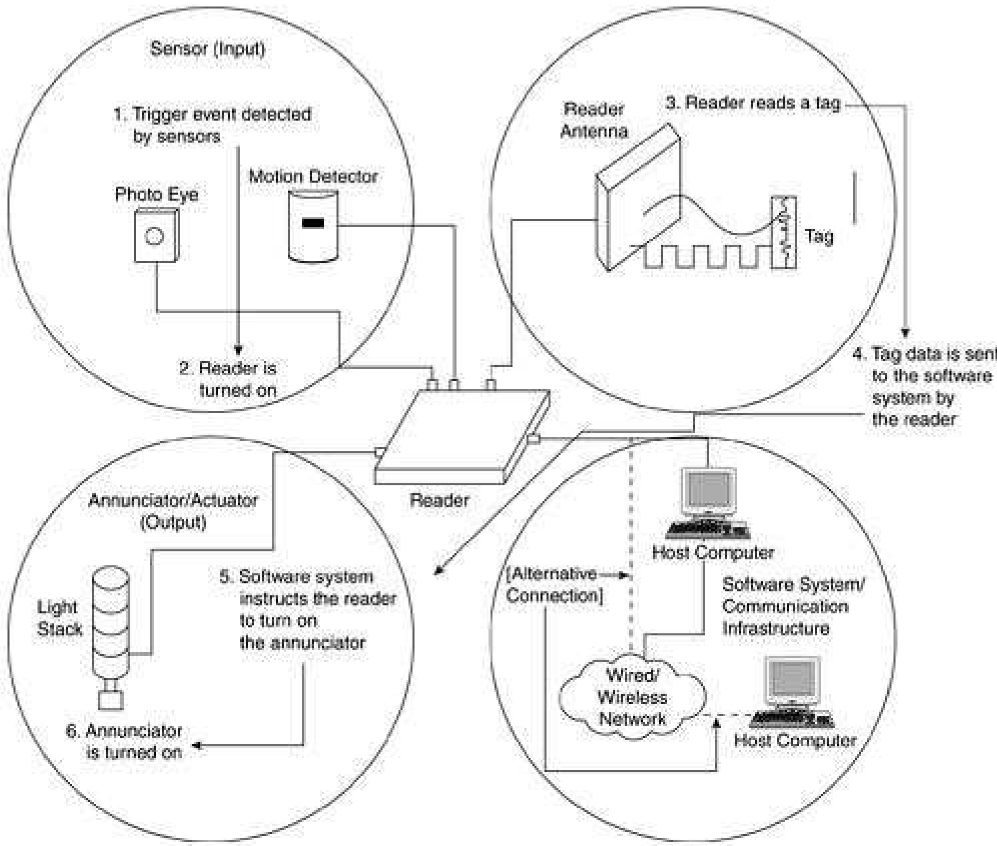


Figure 3.2-1. An RFID system with example components.

This diagram explains steps of how our system works and the connection of the reader with these devices:

1. Input (sensor=photo eye + motion detector) : trigger event detected by sensors .
2. Reader is turned on
3. Reader reads a tag by their antennas.
4. Tag's data is sent to the software system through the reader.
5. Software system instructs the reader to turn on the annunciator.
6. Annunciator is turned on.

3.2 System modeling

The following subsections discuss these previously identified RFID system components in detail

3.2.1 Tag

3.2.1.1 Passive Tags:

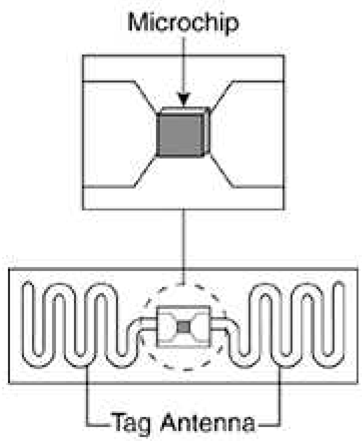


Figure 3.2-2. Components of a passive tag.

This figure shows a passive tag that consists of a two main components: Microchip and Antenna.

3.2.1.1.1 Microchip

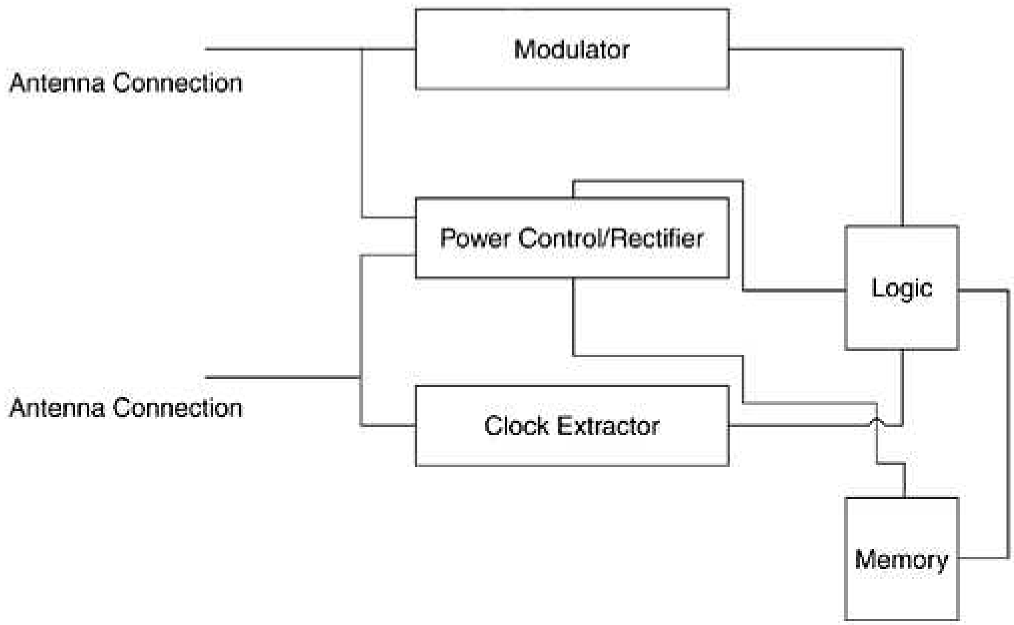


Figure 3.2-3. Basic components of a microchip.

The power control/rectifier converts AC power from the reader antenna signal to DC power. It supplies power to the other components of the microchip. The clock extractor extracts the clock signal from reader antenna signal. The modulator modulates the received reader signal. The tag's response is embedded in the modulated signal, which is then transmitted back to the reader. The logic unit is responsible for implementing the communication protocol between the tag and the reader. The microchip memory is used for storing data. This memory is generally segmented (that is, consists of several blocks or fields). Addressability means the ability to address (that is, read or write) the individual memory of a tag's microchip. A tag memory block can hold different data types, such as a portion of the tagged object identifier data, checksum (for example, cyclic redundancy check [CRC]) bits for checking the accuracy of the transmitted data, and so on. Recent advances in technology have shrunk the size of the microchip to less than the size of a grain of sand. However, a tag's physical dimensions are not determined by the size of its microchip but by the length of its antenna.

3.2.1.1.2 Antenna

A tag's antenna is used for drawing energy from the reader's signal to energize the tag and for sending and receiving data from the reader. This antenna is physically attached to the microchip. The antenna geometry is central to the tag's operations. Infinite variations of antenna designs are possible, especially for UHF, and designing an effective antenna for a tag is as much as an art as a science. The antenna length is directly proportional to the tag's operating wavelength. A dipole antenna consists of a straight electric conductor (for example, copper) that is interrupted at the center. The total length of a dipole antenna is half the wavelength of the used frequency to optimize the energy transfer from the reader antenna signal to the tag. A dual dipole antenna consists of two dipoles, which can greatly reduce the tag's alignment sensitivity. As a result, a reader can read this tag at different tag orientations. A folded dipole consists of two or more straight electric conductors connected in parallel and each half the wavelength (of the used frequency) long. When two conductors are involved, the resulting folded dipole is called *2*-wire *folded dipole*. A *3-wire folded dipole* consists of three conductors connected in parallel. 3.2-6 shows these antenna types.

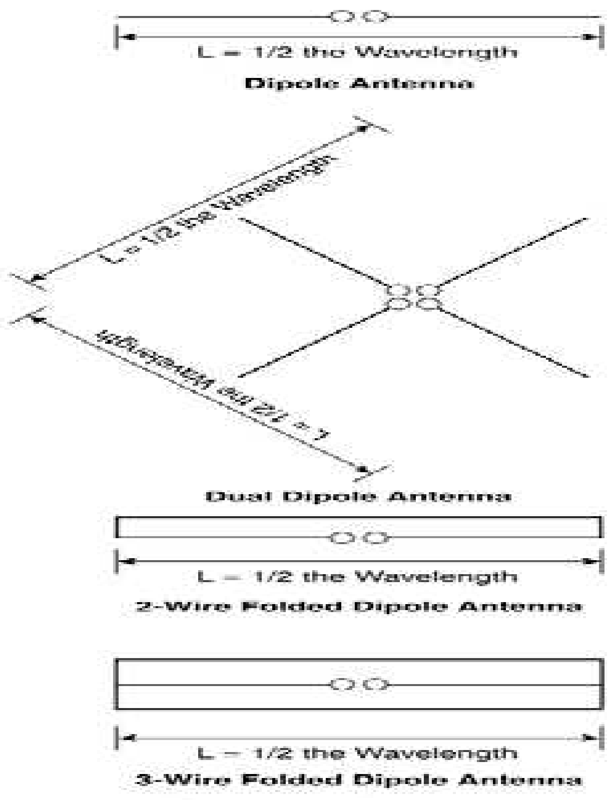


Figure 3.2-4. Dipole antenna types.

A tag's antenna length is generally much larger than the tag's microchip, and therefore ultimately determines a tag's physical dimensions. An antenna can be designed based on several factors, such as the following:

- Reading distance of the tag from the reader
- Known orientation of the tag to the reader
- Arbitrary orientation of the tag to the reader
- Particular product type(s)
- Speed of the tagged object
- Specific operating condition(s)
- Reader antenna polarization

The connection points between a tag's microchip and the antenna are the weakest links of the tag. If any of these connection points are damaged, the tag might become nonfunctional or might have its performance significantly degraded. An antenna designed for a specific task (such as tagging a case) might perform poorly for a different task (such as tagging an individual item in the case). Changing antenna geometry randomly (just "hacking around;" for example, cutting or folding it) is not a good idea because this can detune the tag, resulting in suboptimal performance. However, someone who knows what he is doing can deliberately modify a tag's antenna to detune it (drilling a hole into it, for example) and actually increase the readability of the tag!

Currently, a tag antenna is constructed with a thin strip of a metal (for example, copper, silver, or aluminum). In the future, however, it will be possible to print antennas directly on the tag label, case, and product packaging using a conductive ink that contains copper, carbon, or nickel. Effort is also currently underway to determine whether the microchip might be printed with such an ink, too. These future enhancements may enable you to print an RFID tag just as you do a bar code on the case and item packaging. As a result, the cost of an RFID tag might drop substantially below the anticipated $.05 per tag. Even without the ability to print a microchip, a printed antenna can be attached to a microchip to create a complete RFID tag much faster than attaching a metal antenna. examples of passive tags which are produced by some companies is shown in the appendix.

3.2.1.2 Active Tags

An active tag consists of the following main components:

- Microchip. The microprocessor size and capabilities are generally greater than the microchips found in passive tags.
- Antenna. This can be in the form of an RF module that can transmit the tag's signals and receive reader's signals in response. For a semi-active tag, this is composed of thin strip(s) of metal such as copper, similar to that of a passive tag.
- On-board power supply.

- On-board electronics.

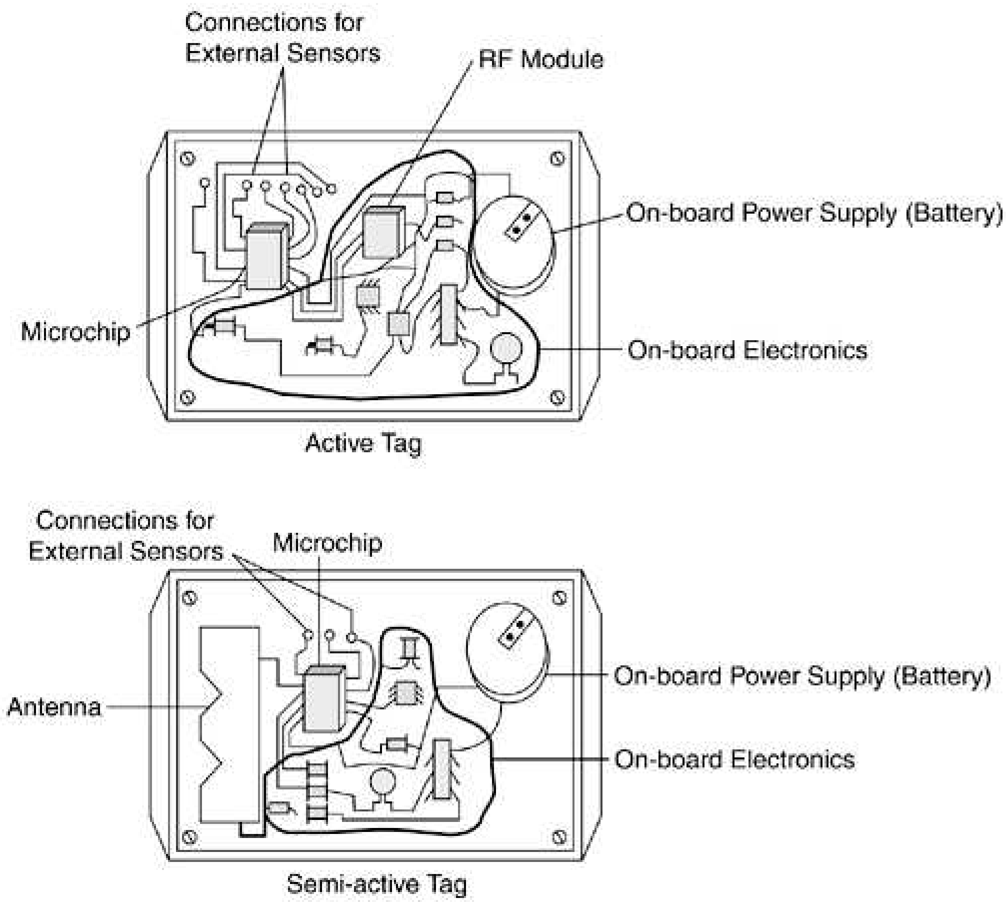


Figure 3.2-5. Example active and semi-active tags.

The first two components have already been described in the previous section. The last two components are discussed now. examples of active tags and semi-active tags which are produced by some companies is shown in the appendix.

3.2.1.2.3 Semi-Active (Semi-Passive) Tags

The next classification, as shown here, is based on the capability to support data rewrites:

- Read-only (RO)
- Write once, read many (WORM)
- Read-write (RW)

Both active and passive tags can be RO, WORM, and RW.

3.2.2 Reader

A reader has the following main components:

- Transmitter
- Receiver
- Microprocessor
- Memory
- Input/output channels for external sensors, actuators, and annunciators (Although, strictly speaking, these are optional components, they are almost always provided with a commercial reader.)
- Controller (which may reside as an external component)
- Communication interface
- Power

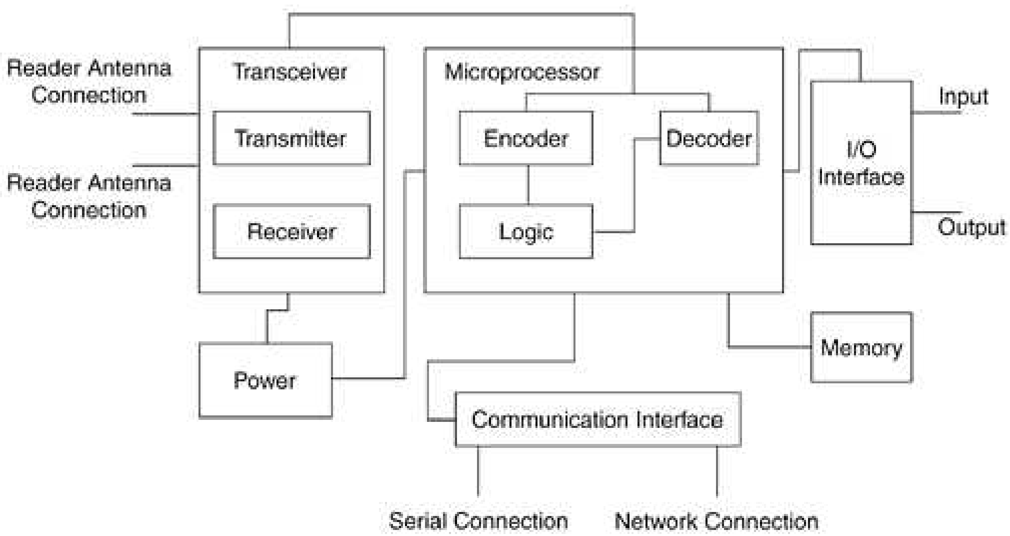


Figure 3.2-6. The components of an example reader.

examples of some readers , RFID smart label tags, RFID printer which are produced by some companies is shown in the appendix.

3.2.2.1. Handheld Reader

A handheld reader is a mobile reader that a user can operate as a handheld unit. A handheld reader generally has built-in antenna(s). Although these readers are typically the most expensive (and few are commercially available), recent advances in reader technology are resulting in sophisticated handheld readers at lower prices. Figures 3.2-20 shows a handheld reader . An example of a handheld reader and which are produced by some companies is shown in the appendix.

3.2.2.2. Modulated Backscatter

Modulated backscatter communication applies to passive as well as to semi-active tags. In this type of communication, the reader sends out a continuous wave (CW) RF signal containing AC power and clock signal to the tag at the carrier frequency (the frequency at which the reader operates). Through physical coupling (that is, a mechanism by which the transfer of energy takes place from the reader to the tag), the tag antenna supplies power to the microchip. The word excite is frequently used to indicate a passive tag microchip drawing power from a reader's signal to properly energize itself. About 1.2 volts are generally necessary to energize the tag microchip for reading purposes. For writing, the microchip usually needs to draw about 2.2 volts from the reader signal. The microchip now modulates or breaks up the input signal into a sequence of on and off patterns that represents its data and transmits it back. When the reader receives this modulated signal, it decodes the pattern and obtains the tag data.

Thus, in modulated backscatter communication, the reader always "talks" first, followed by the tag. A tag using this scheme cannot communicate at all in the absence of a reader because it depends totally on the reader's power to transmit its data. Figure 3.2-21 shows backscatter communication.

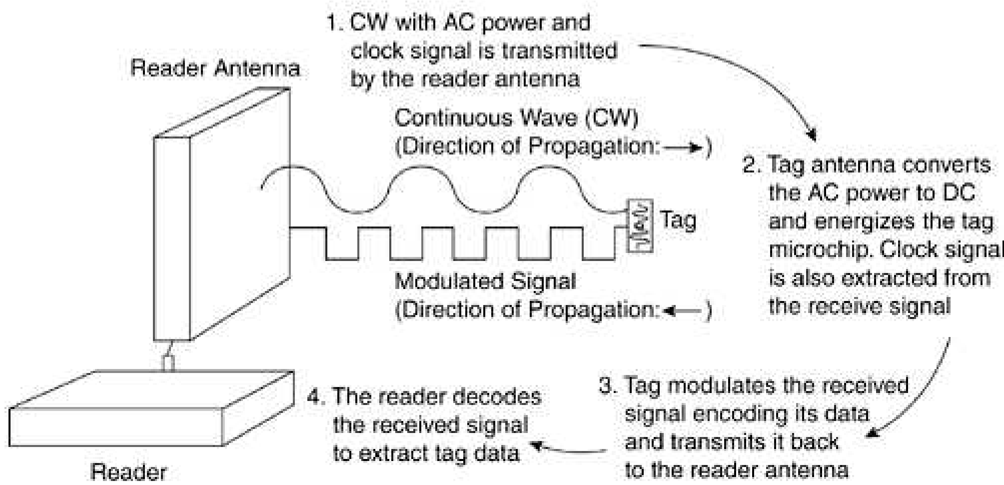


Figure 3.2-7. Backscatter communication.

A related term, beam power, is also used in this context, and means that a tag is using the reader's power to modulate the reader signal back. Note that a passive tag exclusively uses beam power to transmit its data. A semi-active tag uses beam power to clock its oscillator and generate the transmit signal back. Thus, in essence, a semi-active tag also uses beam power to transmit its data.

3.2.3. Reader Antenna

A reader communicates to a tag through the reader's antennas, a separate device that is physically attached to a reader, at one of its antenna ports, by means of a cable. This cable length is generally limited to between 6 and 25 feet. (However, this length limit may vary.) As mentioned previously, a single reader can support up to four antennas (that is, have four physical antenna ports). A reader antenna is also called the reader's coupling element because it creates an electromagnetic field to couple with the tag. An antenna broadcasts the reader transmitter's RF signal into its surroundings and receives tag responses on the reader's behalf. Therefore, proper positioning of the antennas, not the readers, is essential for good read accuracy (although a reader has to be located somewhat close to an antenna because of the limitation of the antenna cable length). In addition, some stationary readers might have in-built antennas. As a result, in this case, positioning the antennas for a reader is equivalent to positioning the reader itself. In general, RFID reader antennas are shaped like rectangular or square boxes. Figures 3.2-24 and 3.2-25 show some reader antennas.

3.2.3.1. Antenna Footprint

The footprints of the reader's antennas determine the read zone (also called the read window) of a reader. In general, an antenna footprint, also called an antenna pattern, is a three-dimensional region shaped somewhat like an ellipsoid or a balloon projecting out of the front of the antenna. In this region, the antenna's energy is most effective; therefore, a reader can read a tag placed inside this region with the least difficulty. Figure 3.2-26 shows such a simple antenna pattern.

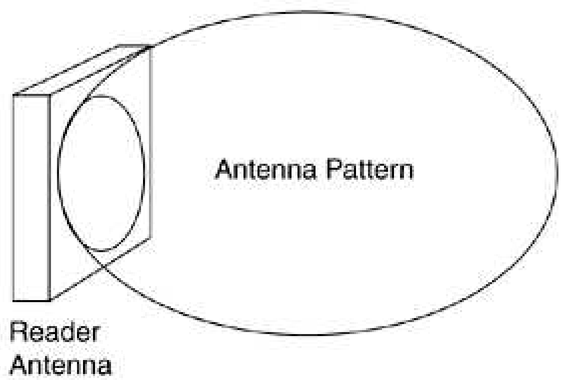


Figure 3.2-8. Simple antenna pattern.

In reality, because of antenna characteristics, the footprint of an antenna is never uniformly shaped like an ellipsoid but almost always contains deformities or protrusions. Each protrusion is surrounded by dead zones. Such dead zones are also called nulls. Figure 3.2-27 shows an example of such an antenna pattern.

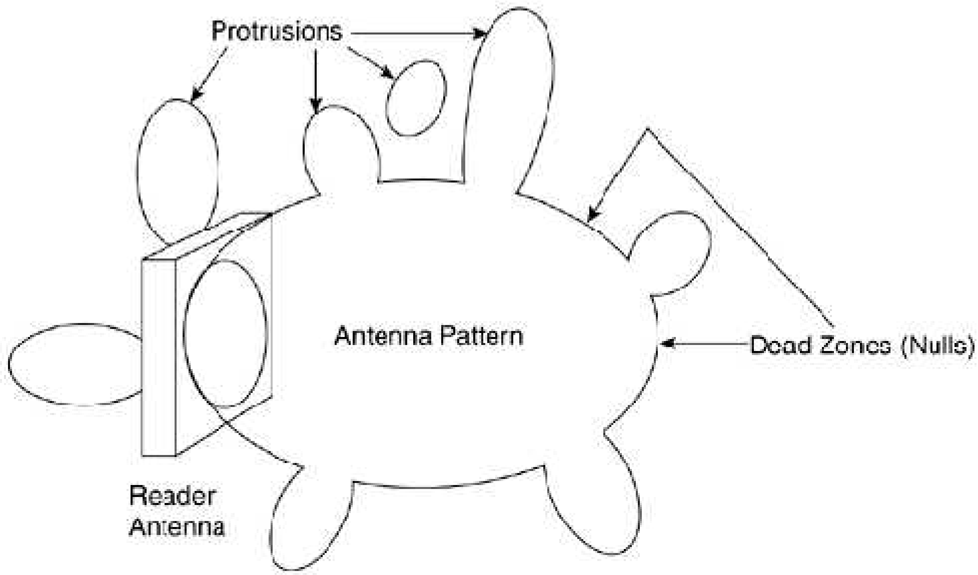


Figure 3.2-9. An example antenna pattern containing protrusions.

The reflection of reader antenna signals on RF-opaque objects causes what is known as *multipath*. In this case, the reflected RF waves are scattered and can arrive at the reader antenna at different times using different paths. Some of the arriving waves could be in phase (that is, exactly match with the original antenna signal's wave pattern). In this case, the original antenna signal is enhanced when these waves impose with the original waves giving rise to protrusions. This phenomenon is also known as constructive interference. Some of the waves could also arrive out of phase (that is, the exact opposite of the original antenna wave pattern). In this case, the original antenna signal is cancelled when these two wave types impose on each other. This is also called *destructive interference*. Nulls are created as a result. Figure 3.2-28 shows an example of multipath.

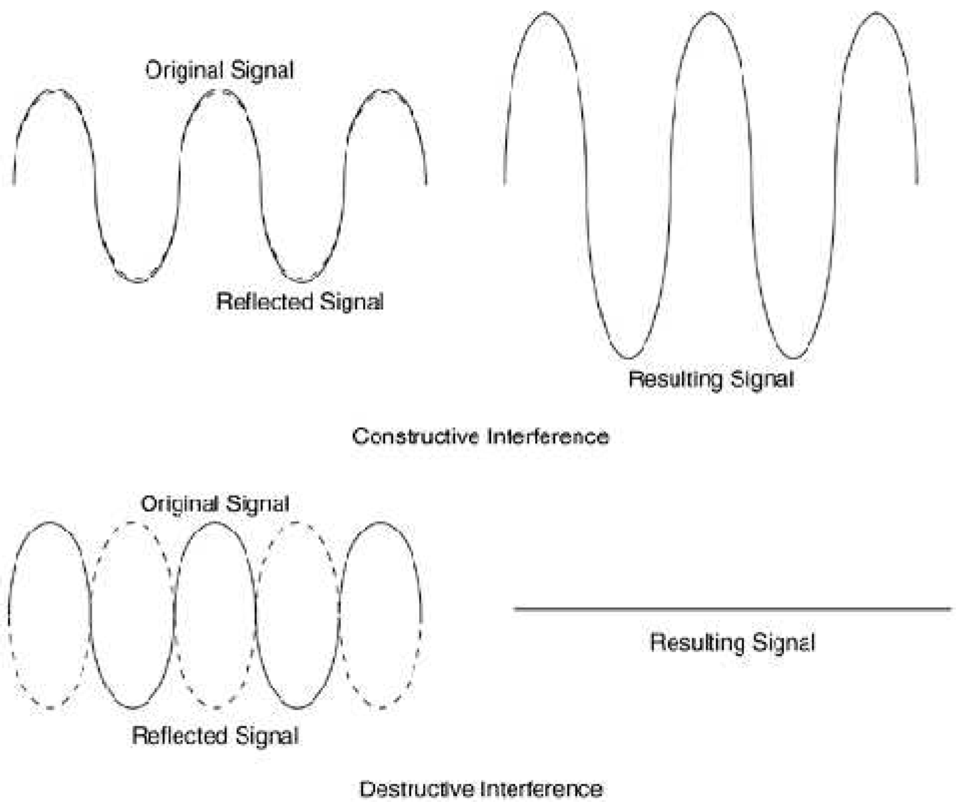


Figure 3.2-10. A multipath schematic.

A tag placed in one of the protruded regions will read, but if this tag moves slightly so that it is inside the surrounding dead region, the tag cannot be read (which might lead to non intuitive tag-reading behavior). For example, when placed a certain distance away from a reader, a tag does not read, but when moved slightly in one direction, it can be read by the reader; if this tag is then moved slightly in another direction, however, it cannot be read! The read behavior of a tag near a protruded region is thus unreliable. Therefore, when you place an antenna to cover a read area, it is important that you not depend on these protruded regions to maximize the read distance. The best strategy is to stay inside the main ellipsoid-shaped region even if it means sacrificing the read range by a few feet better safe than sorry.

It is extremely important to determine the antenna footprint; the antenna footprint determines where a tag can or cannot be read. The manufacturer might provide the antenna footprint as part of the antenna's specifications. However, you should use such information as a guideline only, because the actual footprint will most likely vary depending on the operating environment. You can use well-defined techniques such as signal analysis to map an antenna footprint. In signal analysis, the signal from the tag is measured, using equipment such as a spectrum analyzer and/or a network analyzer, under various conditions (for example, in free space, different tag orientations, and on conductive materials or absorptive materials). By analyzing these signal strengths, you can precisely determine the antenna footprint.

Antenna polarization, another important concept of reader antenna design, is discussed in the following section.

3.2.3.2. Antenna Polarization

As discussed previously, an antenna emits electromagnetic waves into its surroundings. The direction of oscillation of these electromagnetic waves is called the polarization of the antenna. What does this mean to tag readability? A great deal! The readability of a tag, together with its reading distance and reading robustness, greatly depends on the antenna polarization and the angle at which the tag is presented to the reader.

The main antenna types in UHF, based on polarization, are

- Linear polarized
- Circular polarized

The following subsections discuss these two types of antennas.

3.2.3.2.1. Linear Polarized Antenna

In this antenna type, the RF waves emanate in a linear patter from the antenna. These waves have only one energy field. Figure 3.2-29 shows the resulting wave pattern emanating from a linear polarized antenna.

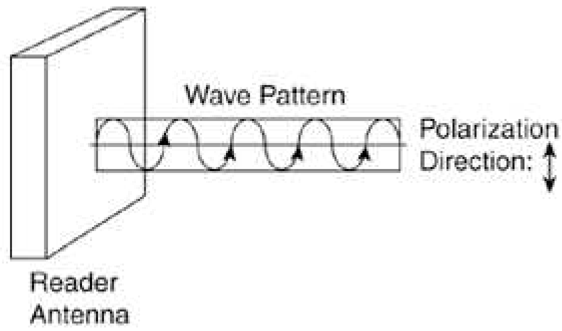


Figure 3.2-11. Wave pattern from a linear polarized antenna

A linear polarized antenna has a narrower radiation beam with a longer read range compared to a circular polarized antenna. In addition, a narrower radiation beam helps a linear polarized antenna to read tags within a longer, narrow but well-defined read region (compared to a circular polarized antenna), instead of reading tags randomly from its surroundings. However, a linear polarized antenna is sensitive to tag orientation with respect to its polarization direction. These types of antenna are therefore useful in applications where the tag orientation is fixed and predictable. Figure 1-31 shows how a tag should be oriented with respect to a linear antenna for its proper reading in case of backscatter communication.

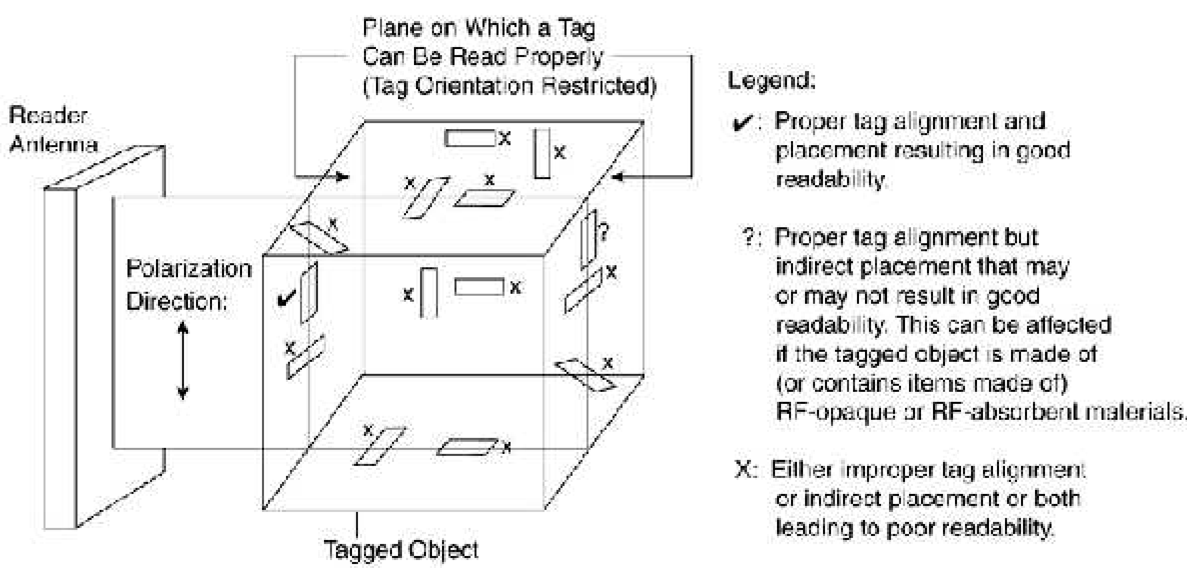


Figure 3.2-12. Proper tag orientation for a linear polarized antenna.

3.2.3.2.2. Circular Polarized Antenna

RF waves radiate from a circular polarized antenna in a circular pattern. These waves have two constituting energy fields that are equal in amplitude and magnitude, but have a phase difference of 90°. Therefore, when a wave of an energy field is at its highest value, the wave of the other field is at its lowest. Figure 3.2-13 shows the resulting wave pattern emanating from a circular polarized antenna.

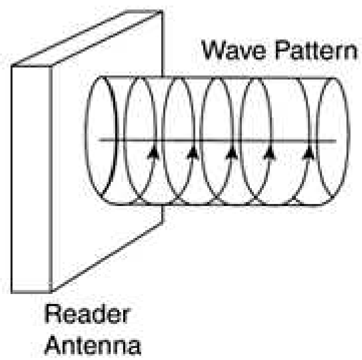


Figure 3.2-13. Wave pattern from a circular polarized antenna.

Because of the nature of polarization, a circular polarized antenna is largely unaffected by tag orientation. Therefore, this type of antenna proves ideal for applications where the tag orientation is unpredictable. A circular polarized antenna has a wider radiation beam and hence reads tags in a wider area compared to a linear polarized antenna. This antenna is preferred for an RFID system that uses high UHF or microwave frequencies in an operating environment where there is a high degree of RF reflectance (due to presence of metals and so forth). Figure 3.2-14 shows how a tag should be oriented with respect to a circular antenna for its proper reading in case of backscatter communication.

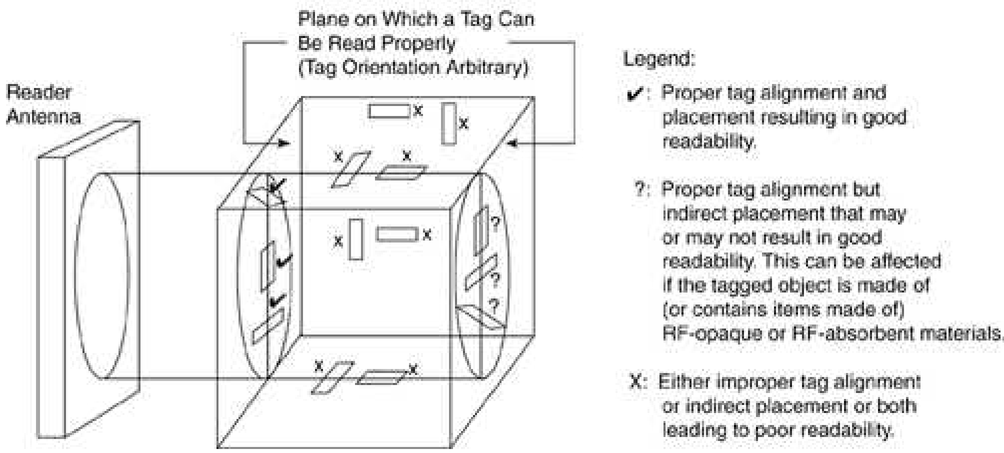


Figure 3.2-14. Proper tag orientation for a circular polarized antenna.

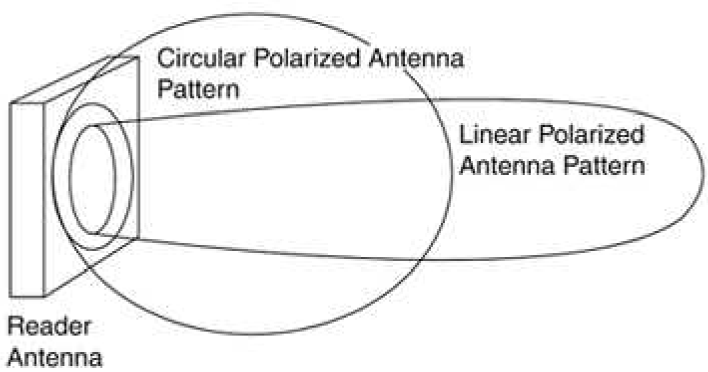


Figure 3.2-15. Circular and linear polarized antenna patterns.

Often, a patch antenna is used for making UHF antennas, as described in the following subsection.

3.2.3.2.3. Patch Antenna

A patch antenna, also called a micro strip or planar antenna, in its basic form consists of a rectangular metal foil or a plate mounted on a substrate such as Teflon. The other side of the substrate is coated with a metallic substance. A micro strip connected to the rectangular metal foil supplies power to the antenna (see Figure 3.2-16). The power supply type can be varied to make a patch antenna circular or linear polarized.

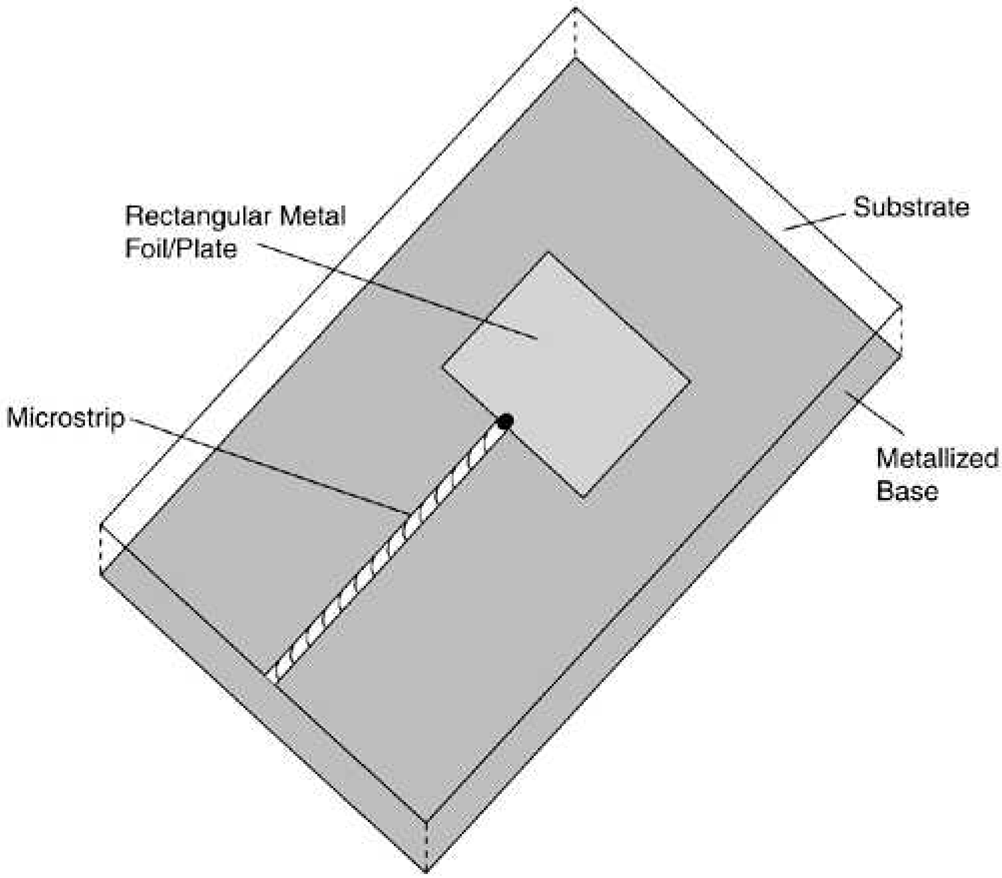


Figure 3.2-16. An basic patch antenna.

3.3 server with data base:

We will use PPU's library data base instead of installing new software and reconstruct new data base. The data base must contain the book title and code with serial number, ISBN, author, publish year, and the publisher. We can connect the data base with internet to access from any where.

3.4 How the system works :

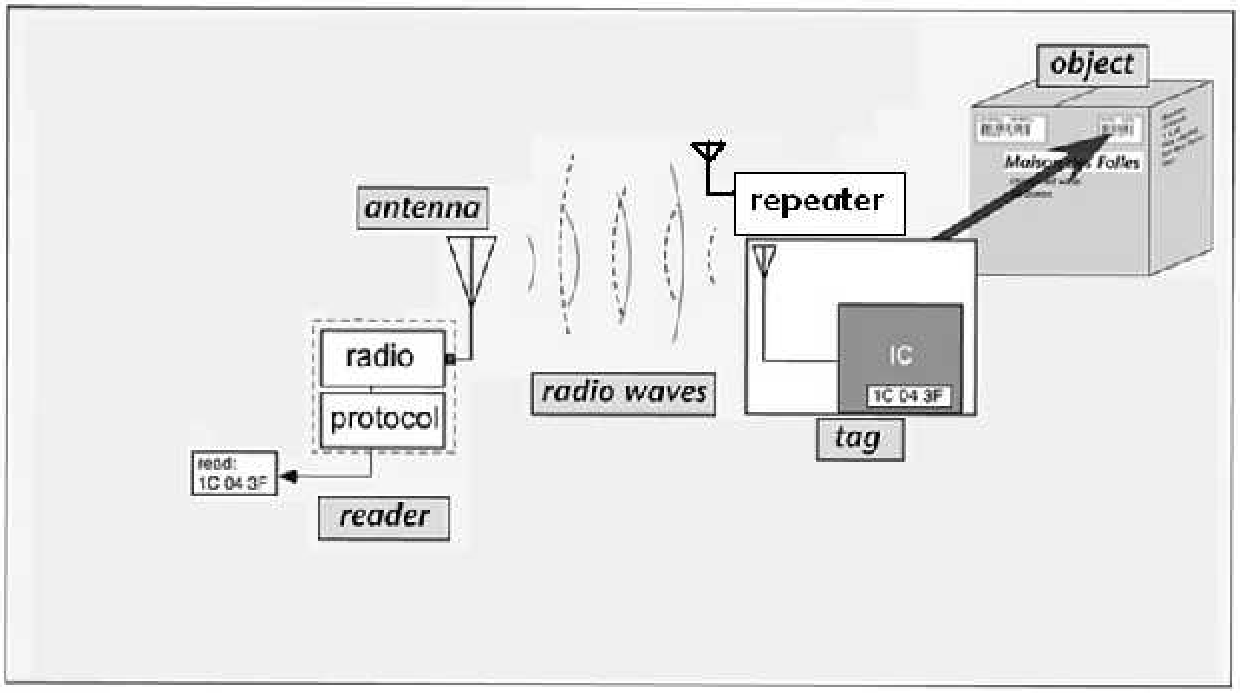


Figure 3.4-1 : a simple diagram to show a steps for how the system works .

In chapter one – introduction – we have described how the system should work at a high level of abstraction , but now we are going to describe how the system works in more details :

1. Borrowers present their identification to the unit. (This might be another RFID enabled card, a barcode or some other technology) , and then here we are start .
2. Depending on library policy they may be required to enter a PIN code or password and this entered on the computer (server) at which the data base exist .
3. Items are placed on the reading table ,through a special cable connecting each other .
4. Items are read and passed for checking (using SIP) , this happens through sending the signal to the repeater after it had been loaded on the reader , the signal will be repeated and then sent to the all tags .

5. The term return to the Self Service Unit (or SSU) (again via SIP) , through an opposite way (tags send the signal to the repeaters and next it transmit it to the reader) .
6. Any items that may not be loaned will be advised on screen
7. Security data is written to the tags to allow or deny them to pass the security gates.
8. A receipt may optionally be printed

3.5 Conclusion

RFID in the library speeds up book borrowing, monitoring, books searching processes and thus frees staff to do more user-service tasks. But the performance varies with respect to the vendors of RFID readers and tags. The efficient utilization of the technology also depends upon the information to be written in tag. Experimental results with respect to effectiveness of RFID reader position, tag position are presented in the paper. Developments in RFID technology continue to yield larger memory capacities, wider reading ranges, and faster processing.

CHAPTER FOUR

Hardware Implementation

Hardware :

The RFID reader consists of transmitting and receiving sections. It transmits a carrier signal (13.56 MHz),receives the backscattered signal from the tag, and performs data processing. The reader also communicates with an external host computer.

4.1 Transmitting Section:

The transmitting section contains a 13.56 MHz signal oscillator (74HC04) which is shown in fig4.2 by raw no.1, power amplifier (Q2) which is shown in fig4.2 by raw no.2, and RF tuning circuits which is shown in fig4.2 by raw no.3. The block diagram shown in the fig 4.1, and circuit in details shown in fig 4.2, The tuning circuit matches impedance between the antenna coil circuit and the power driver at 13.56 MHz. The radiating signal strength from the antenna must comply with government regulations. For best performance, the antenna coil circuit must be tuned to the same frequency of the tag.

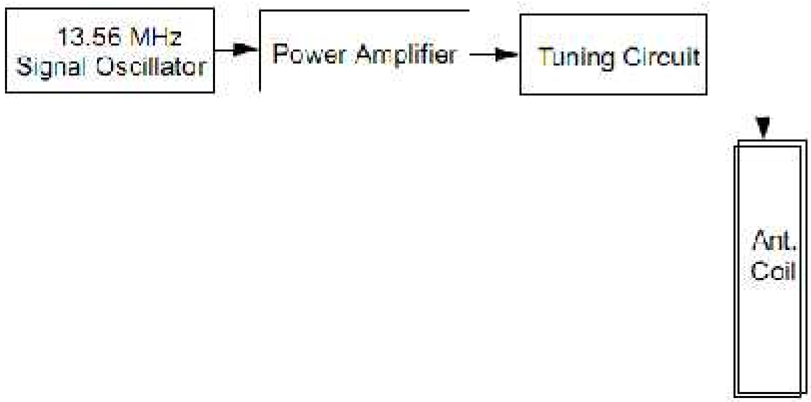


fig 4.1 Transmission Section

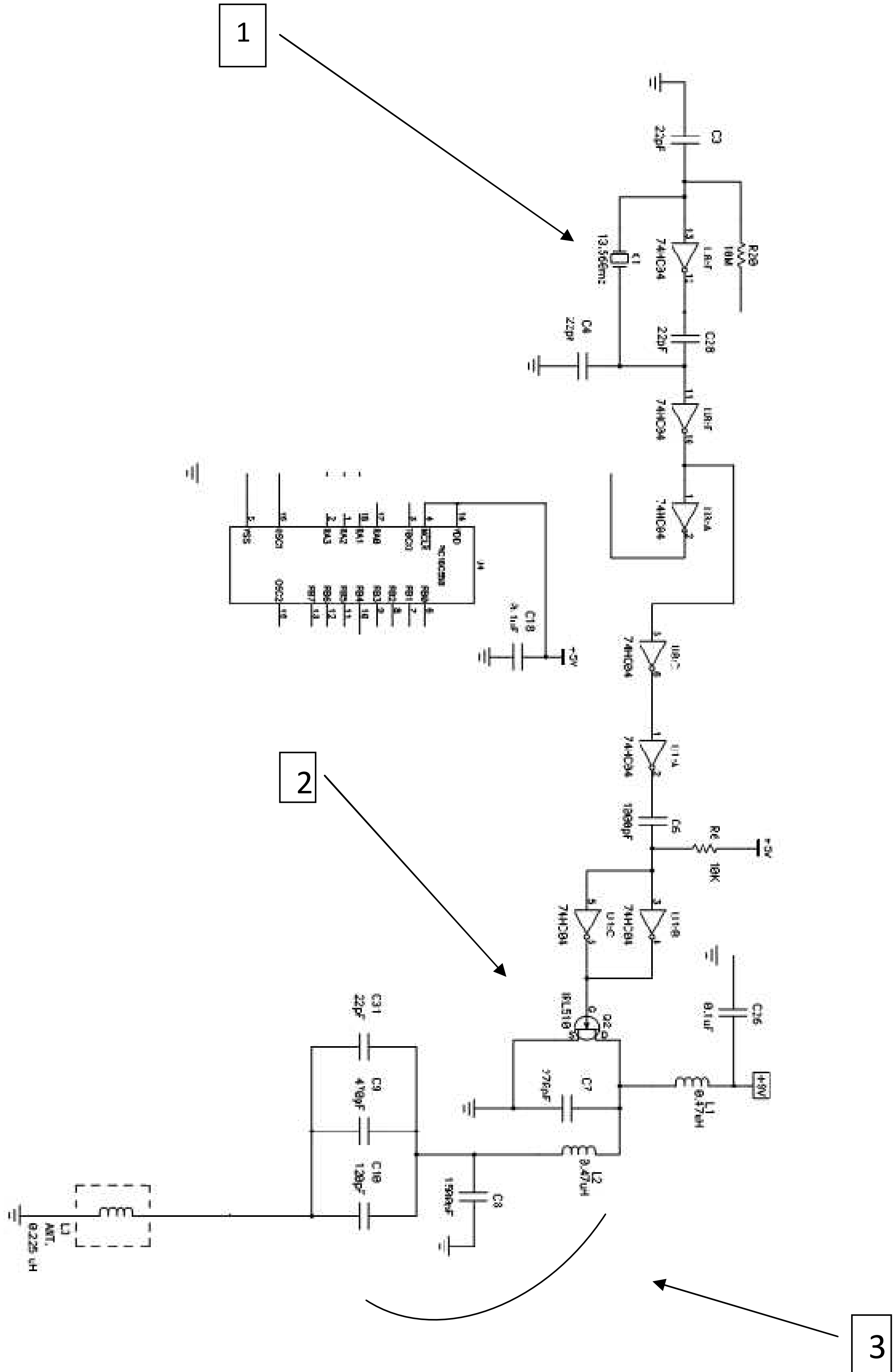


fig 4.2 circuit of transmission section

4.2 Receiving Section:

The receiving section contains an envelope detector (D6)) which is shown in fig4.2 by raw no.1, hi-pass filters, and amplifiers (U2 and U3)) which is shown in fig4.2 by raw no.2. after that signal is processed by the PIC that is illustrated by raw no.3. befor entering it to the last stage. When the tag is energized, as shown in fig 2.3 ,it transmits 154 bits of data that is encoded in Biphase-L (Manchester). In the Manchester encoding, data '1' is represented by a logic high-to-low level change at mid clock, and data '0' is represented by a low-to-high level change at midclock. There is always a level change at middle of every bit clock.

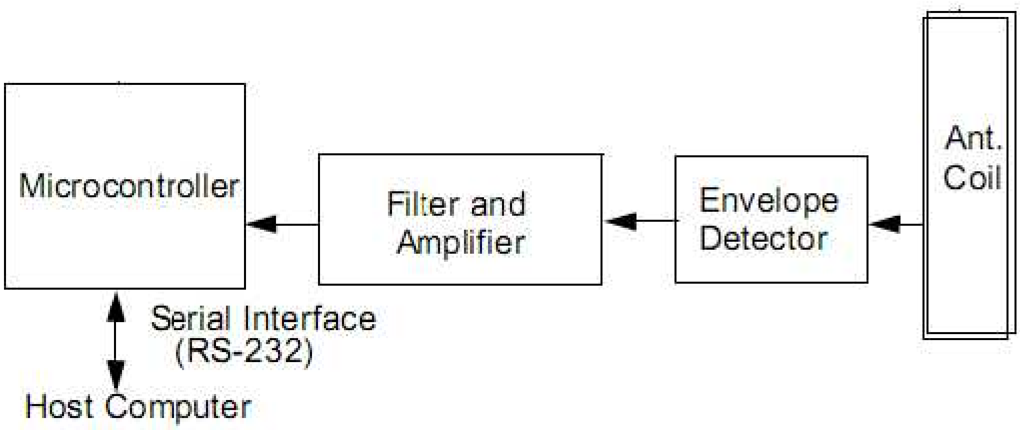


Fig 4.3 Receiving section

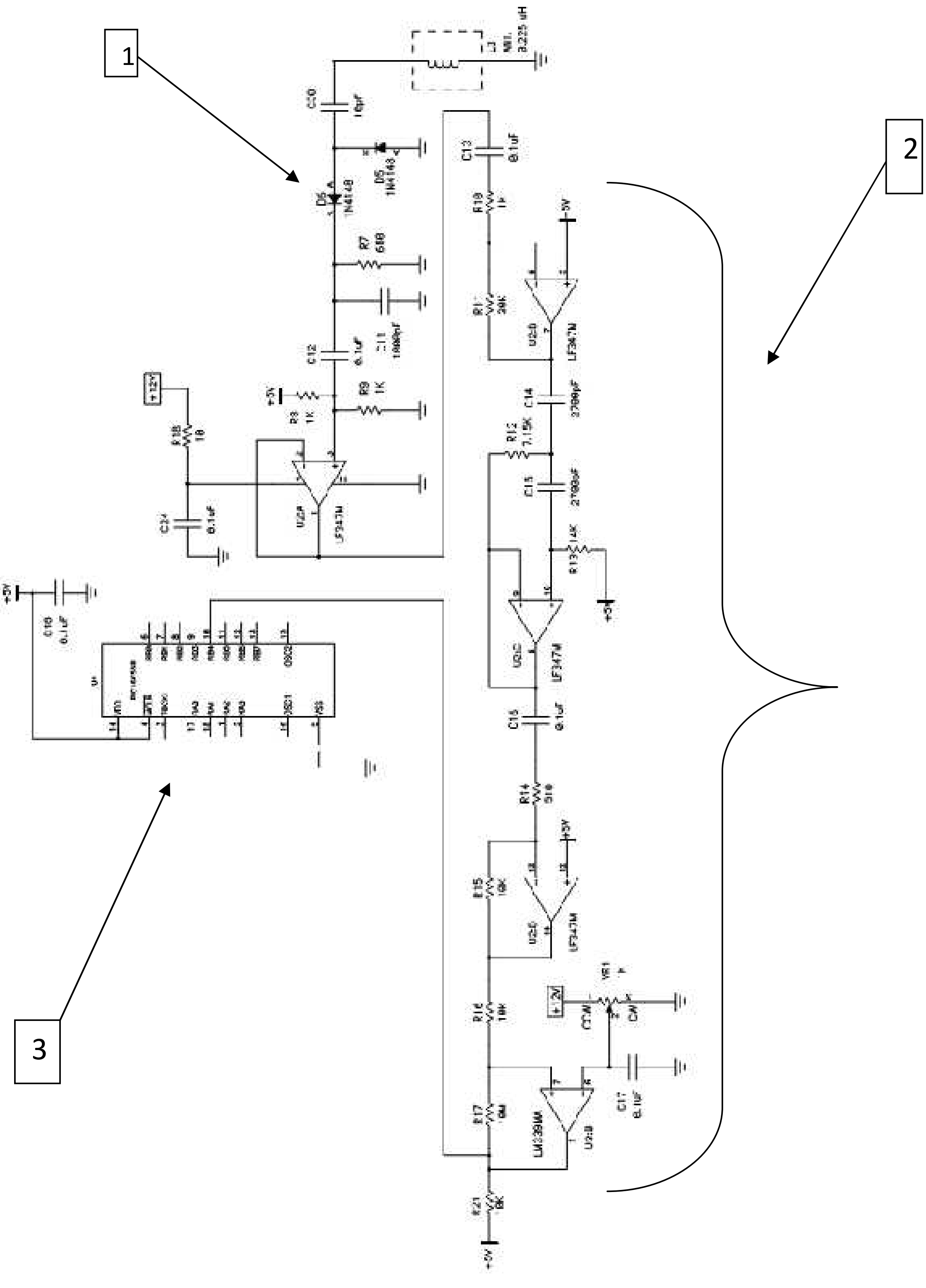


fig 4.4 circuit of Receiving section

4.3 Tag Section:

When the tag is energized by the reader’s carrier signal, it transmits back with an amplitude modulated signal. This results in a perturbation in the voltage amplitude across the reader antenna coil. The envelope detector detects the changes in the voltage amplitude and passes it into an RC filter (R7, C11), The charged signal in the capacitor passes through active filters and amplifiers. The signal that is passing through this receiving section is the data signal. This filtered shaped data signal is fed into Pin 10 of the microcontroller for data processing.

4.4 Interface Section:

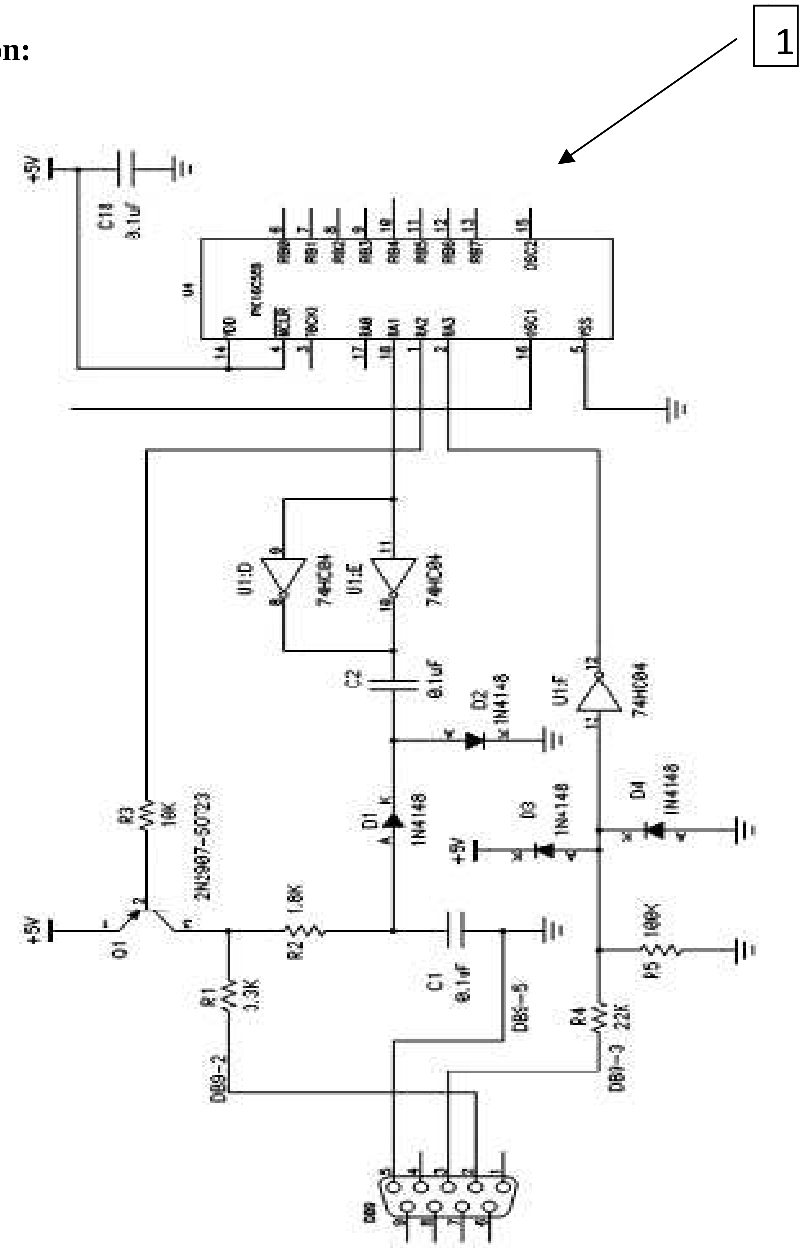


fig 4.5 interface circuit.

This section as shown in fig 4.5 is responsible of processing the signal by using the PIC shown by raw no.1, before entering to the Computer and implement the application by the librarian.

CHAPTER FIVE

Software Design Implementation

5.1 Visual Basic Design

5.2 Orcad Simulation

Software:

5.1 Visual basic design :

We made the database through connecting two software, first we build database using Microsoft access 2007 then connecting it with microsoft Visual basic to build a main program to control the operation of the RFID system.

These are captures form the software:

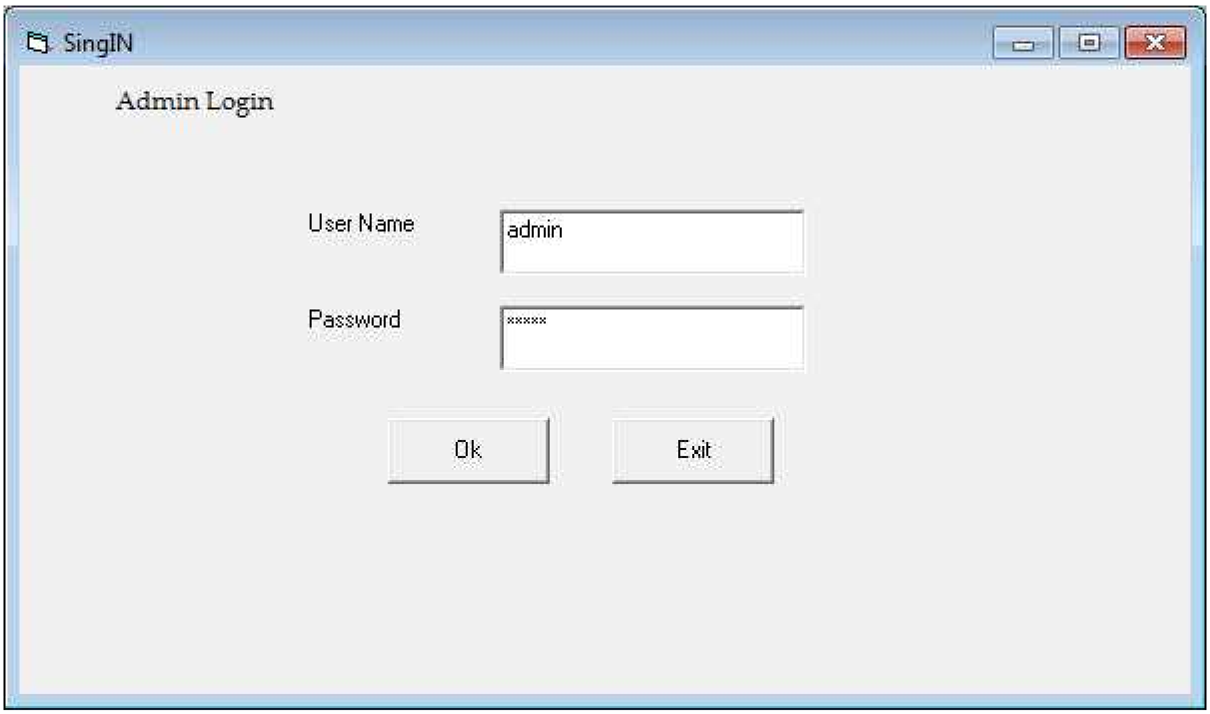


fig 5.1 sing in window

This window controlling librarian to accesses to data base and software for authentication .

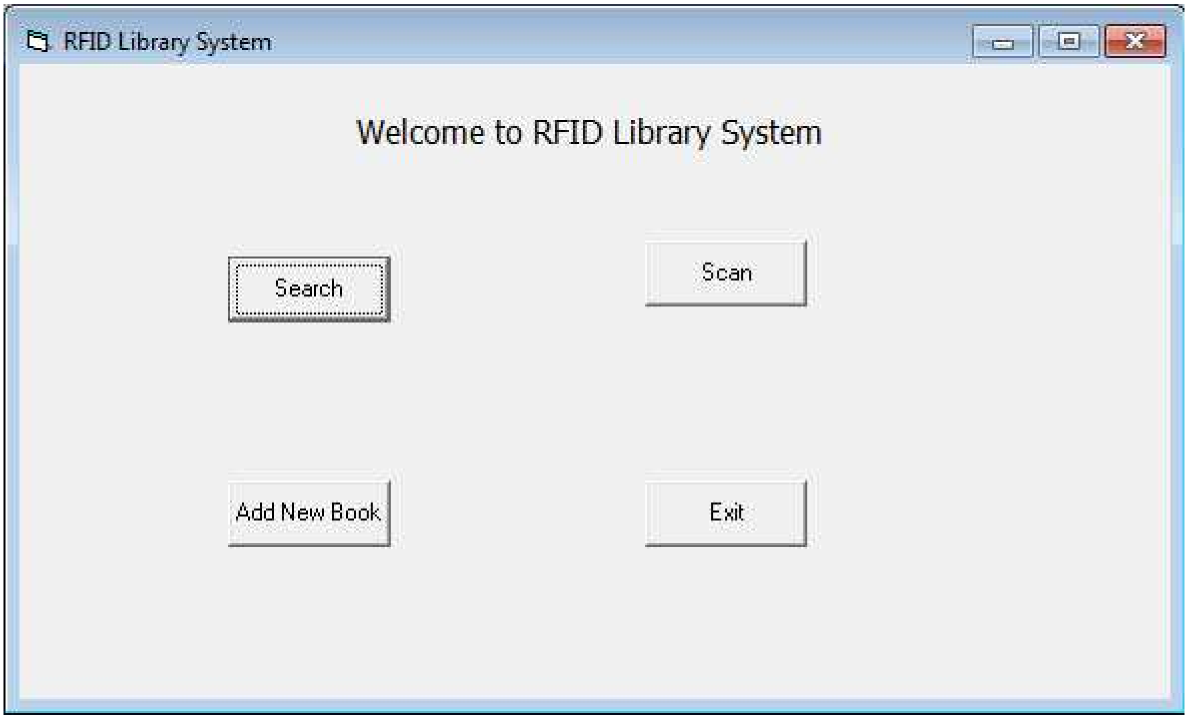


Fig 5.2 Main page

This window allows for user to select one of the choices which is Scan for book, search inside database and add new book.

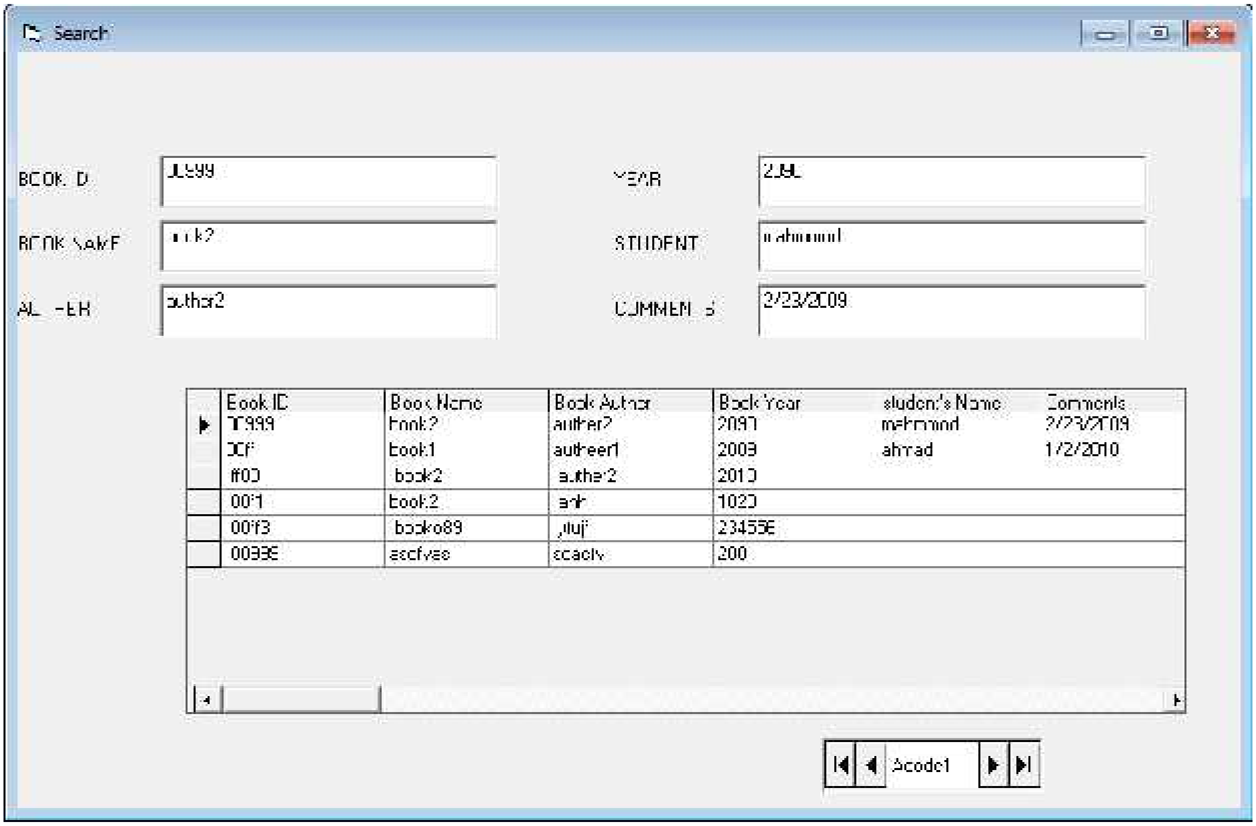


fig 5.3 Search Window

The librarian can search for tag or book from this window shown in fig 5.2

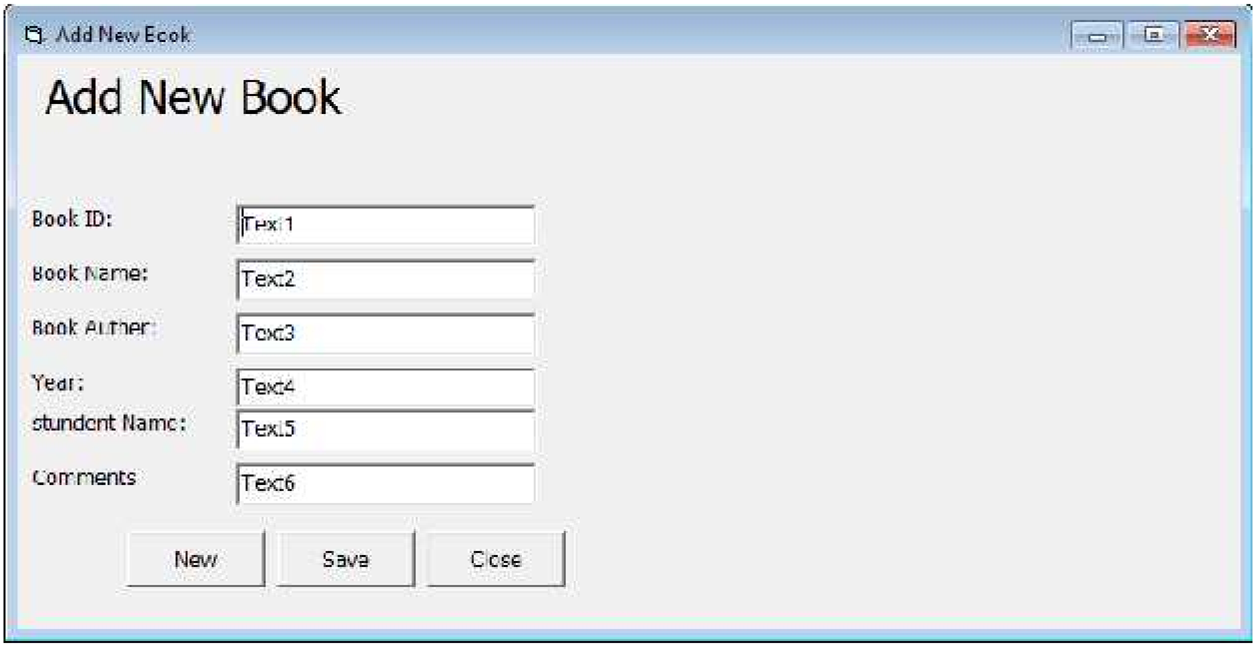


fig 5.4 Add Book Window

This window allow for user or librarian to add new book to database with new tag and code.

5.2 Orcad simulation:

we use orcad capture to simulate the circuit on computer which is shown in fig5.5 , then to convert the simulated circuit to the layout shape as shown in fig.5.6 to print it on a board which is responsible to accomplish all the connection wires.

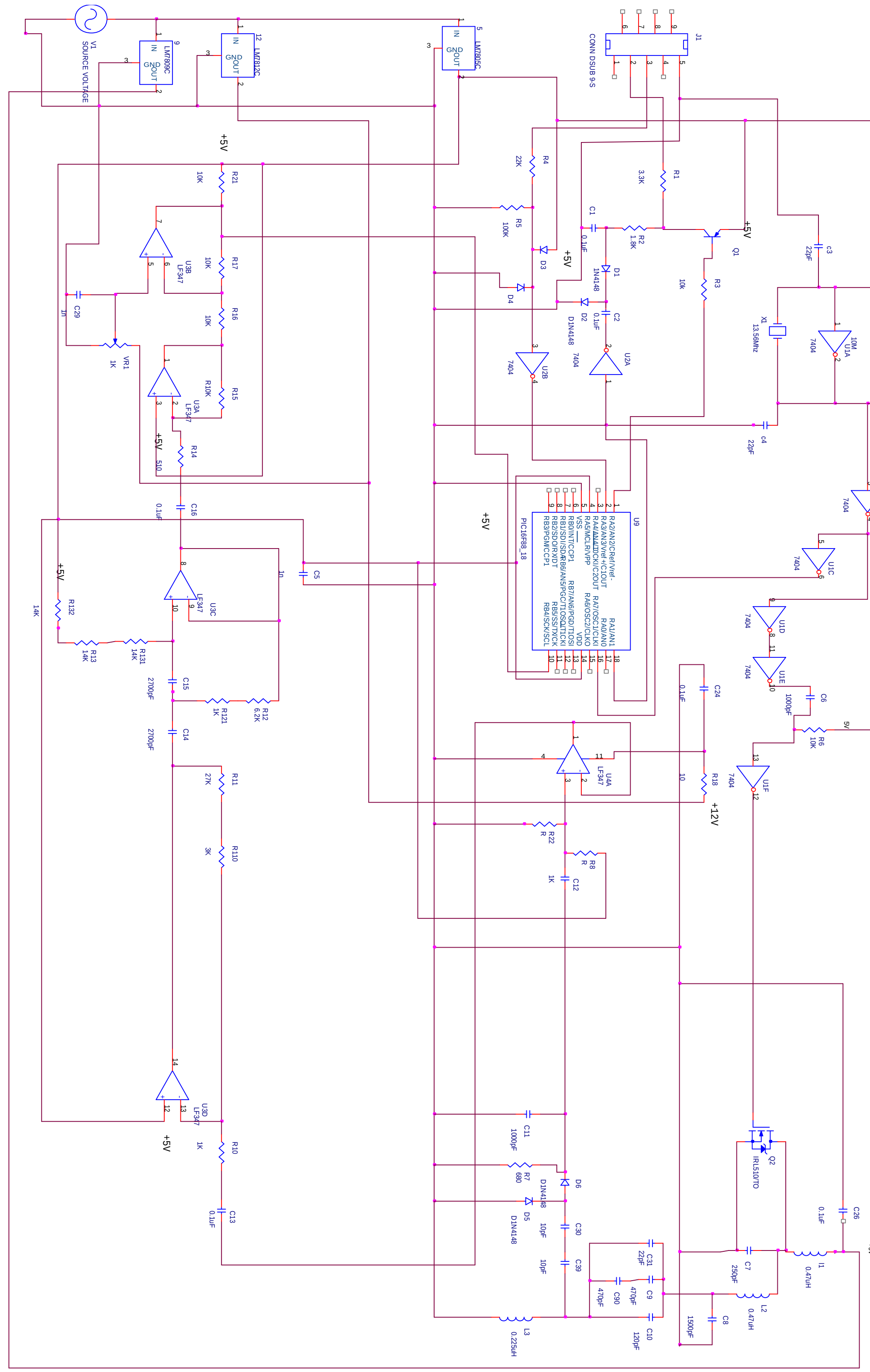


fig 5.5 capture on orcad.

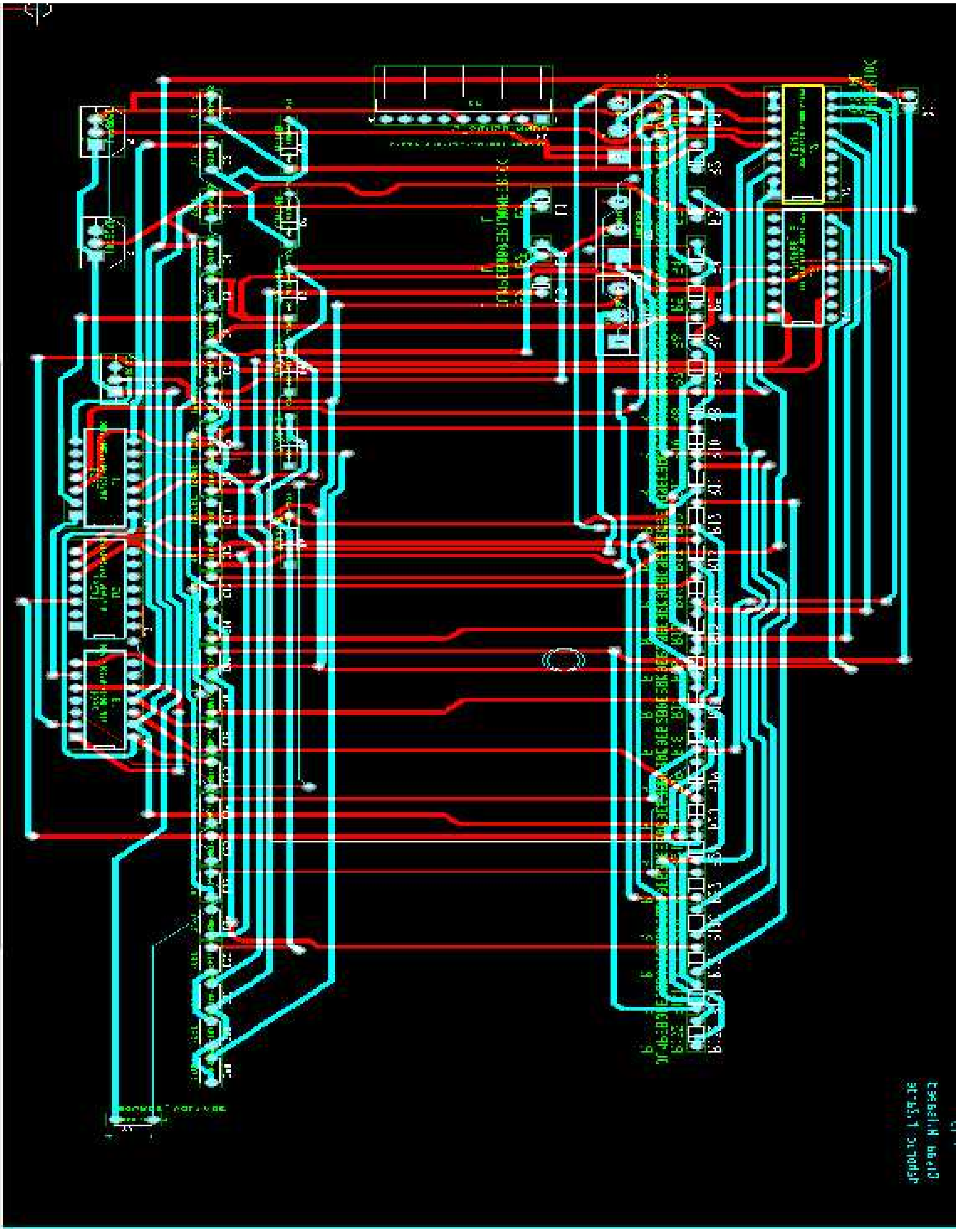


Fig 5.6 layout circuit

CHAPTER SIX

System Testing

6.1. Introduction

The testing steps are very important to the system. The system after all sections have been designed , was placed under spot to see if it is working as expected, and to find out the mistakes, problems.

6.2 Testing Steps

The steps below illustrate the testing operation for our project:

1) Transmitting Testing
2) Antenna Testing
3) Tags Testing
4) Receiving Testing
5) Software testing

6.3 Testing Procedures

6.3.1 Transmitting Testing

in this section we took each stage in the transmitting section alone , we started by crystal (13.56 MHz) by oscilloscope and the result shown in fig 6.1.

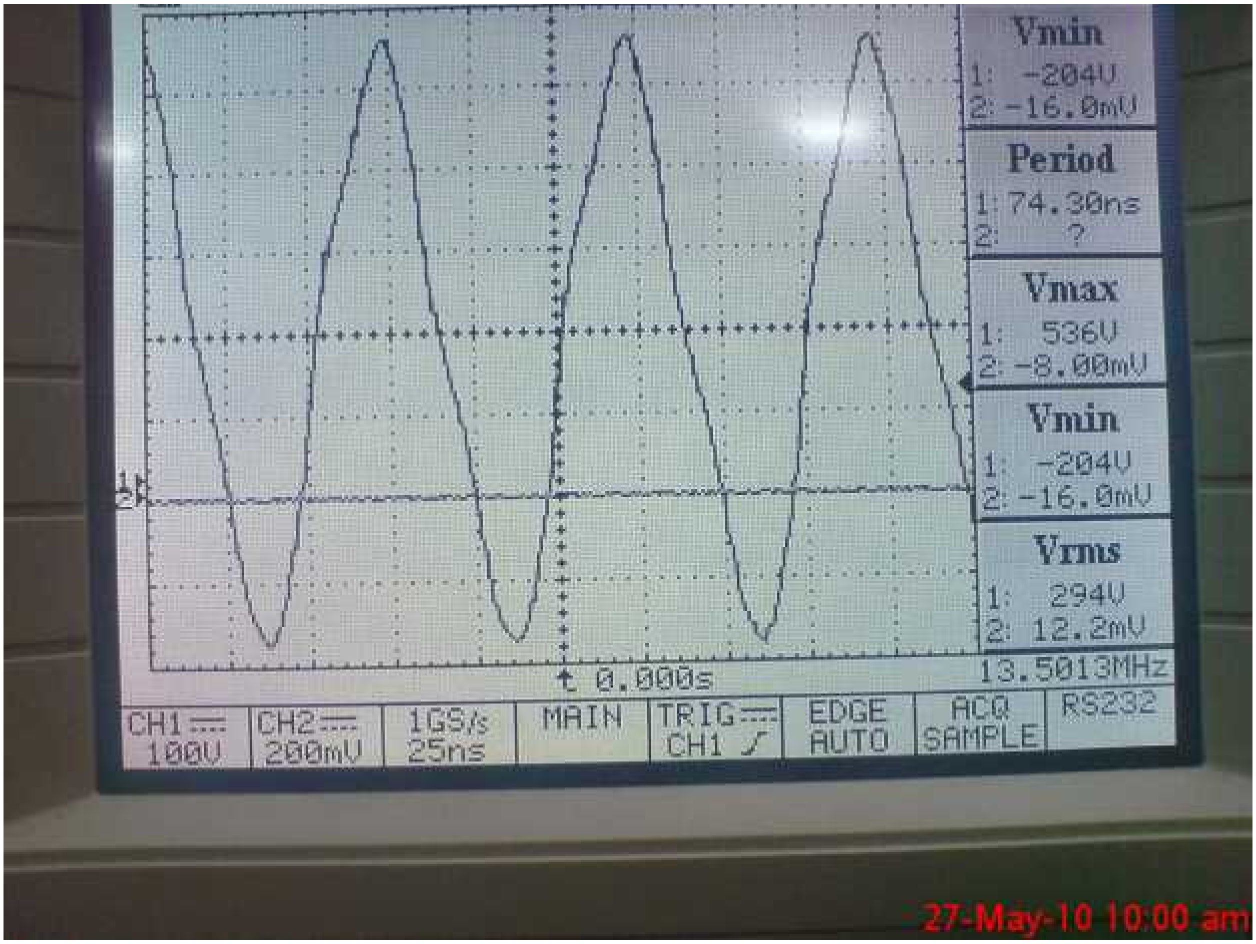


fig 6.1 crystal output

The next stage in transmitting section is the power amplifier Q2 as shown in fig 4.2 in chapter 4 through raw no.2 . which provide the best combination of fast switching and amplifying the signal , as shown in figure 6.2 ,

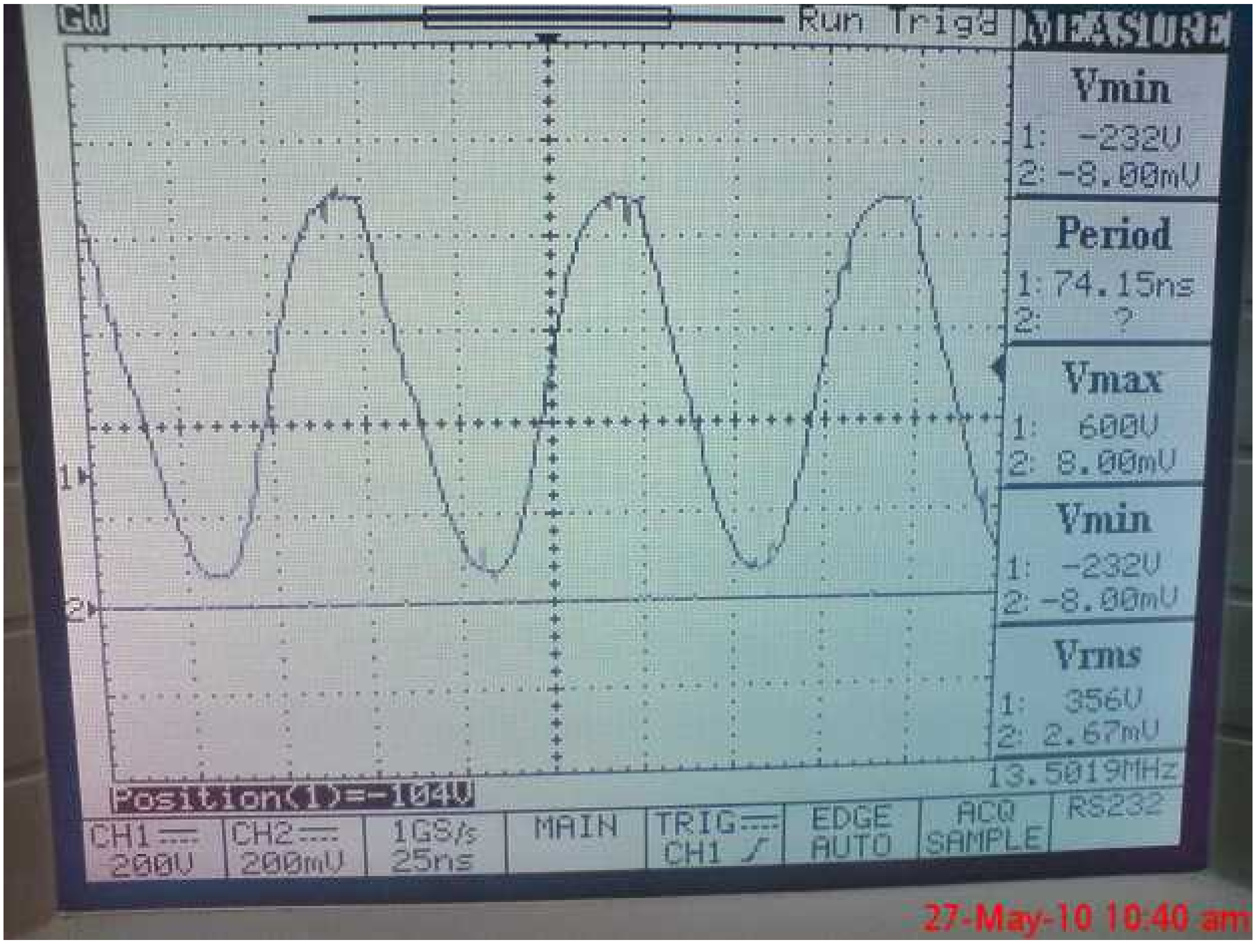


fig 6.2 signal after Q2

6.3.2 antenna testing :

Antenna used to transmit the carrier signal to tags and receive the backscattered signal (modulated signal) . as shown in the figure 6.3.

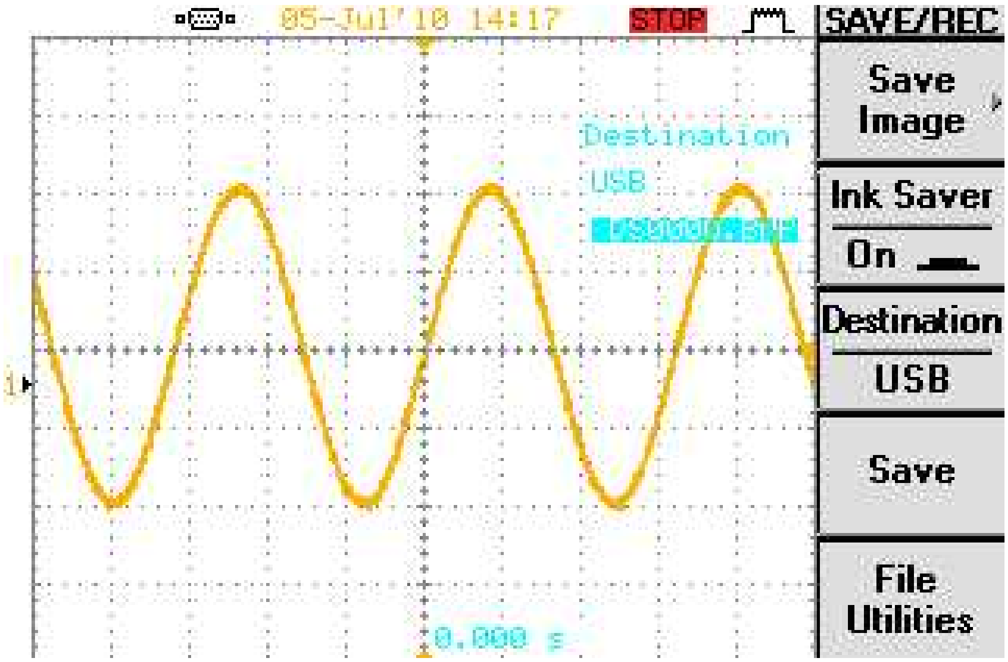


Fig 6.3 transmitted signal of the antenna

6.3.3 Receiving section:

The receiving section contains an envelope detector (D6), hi-pass filters, and amplifiers (U2 and U3). When the tag is energized ,it transmits 154 bits of data that is encoded in Biphase-L (Manchester). In the Manchester encoding, data ‘1’ is represented by a logic high-to-low level change at mid clock, and data ‘0’ is represented by a low-to-high level change at midclock. There is always a level change at middle of every bit clock.

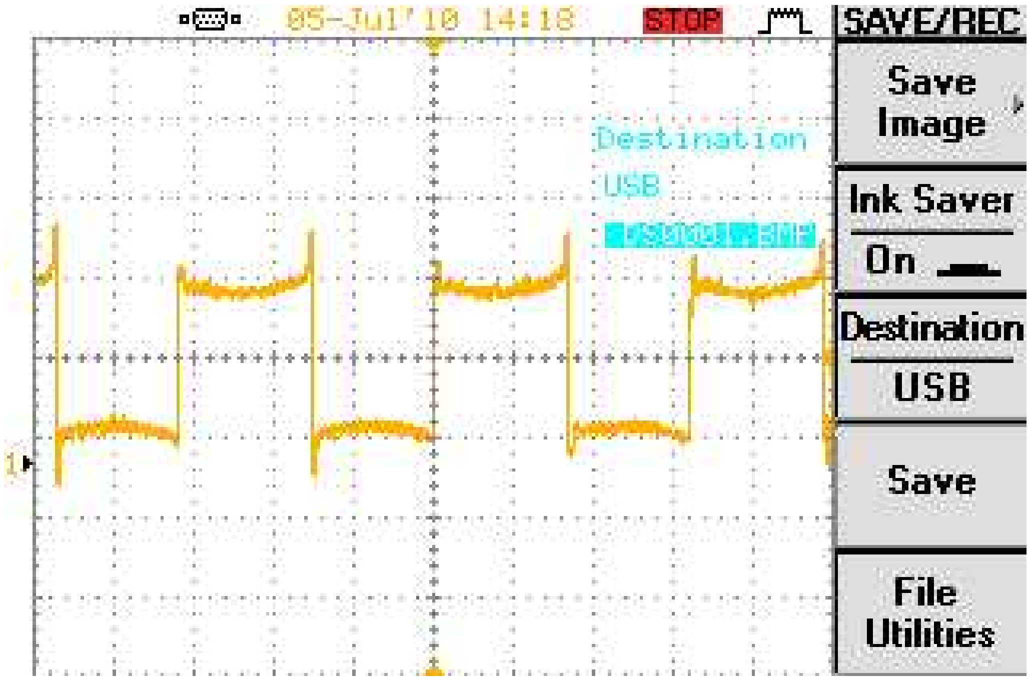


fig 6.4 received signal

this figure shows the receiving signal .

6.3.4 Software testing:

1-tseting librarian log in to access to the database, the user can add ,edit, save ,and remove data form database.

2- testing add new book window which allow user to add new book to database and save it .

3- testing search window which allow user to search for some book on library and find name or code and location of the book in the library .

4- testing for scanning .

6.4 Black Box Testing

Operation	Expected results	Actual results	Comments
Transmission	Transmitted Wave should be sinusoidal	The carrier signal is been sent as sinusoidal signal	---
Antenna	The antenna should be compatible with a specific properties indeed the signal should be sent on the required frequency .	Sinusoidal signal is actually sent at the required frequency .	We designed different shapes of antennas ,not all gave us the required signal ,but we selected the correct one of them.
Tag	Must receive the carrier signal and modulate it with the data that is stored in the memory inside tag after that it transmit the modulated signal through the antenna coil to the reader .	The tags we used received the signal and transmitted the modulated signal .	---

Receiving	Modulated signal is passing through three stages to be convert to digital data (Manchester code).	The backscattered signal is converted to digital signal (Manchester code)	-----

Table 6.5 Black Box Testing.

Chapter Seven

Conclusion and Future Work

7.1 Future Work

7.2 conclusion

7.1 Future Work

In this project, there are some ideas that could be done or added to improve its performance, or add some capabilities, some techniques that are efficient and meet worldwide needs. Some of these ideas are mentioned below:

APPLICATIONS:

- RFID could be used for other applications like super markets, passports in the airports, livestock .etc…
- Access to system through Internet from any place .
- Locate the location of the books through connect RFID with GPRS.
- To build system for anti thief books from library.

DESIGN:

- Changing frequency , therefore to improve the range .
- Repeaters could be placed within the project to increase the coverage range.
- Decrease the size of tags and readers.
- Use more techniques for security

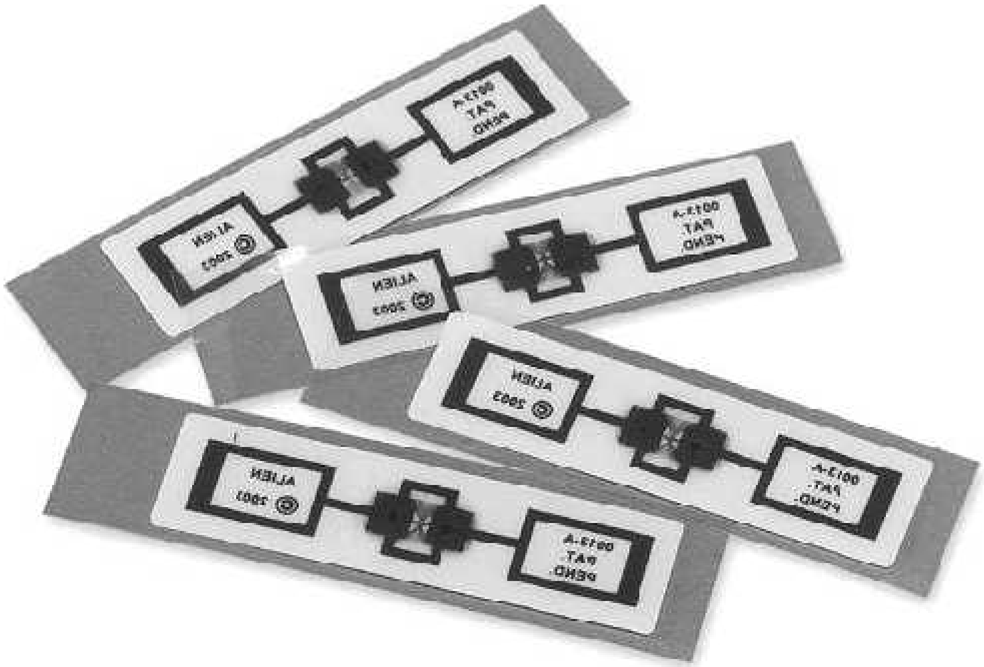
7.2 conclusion:

RFID in the library speeds up book borrowing, monitoring, books searching processes and thus frees staff to do more user-service tasks. But the performance varies with respect to the vendors of RFID readers and tags. The efficient utilization of the technology also depends upon the information to be written in tag. Experimental results with respect to effectiveness of RFID reader position, tag position are presented in the paper. Developments in RFID technology continue to yield larger memory capacities, wider reading ranges, and faster processing.

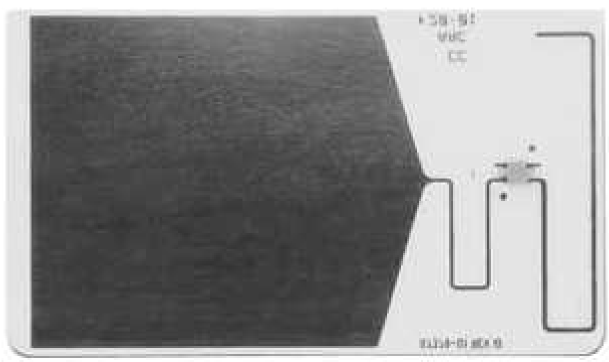
Appendix A:

1. Examples of passive tags which are produced by some companies.

Family of LF tags from Texas Instruments.



2.45 GHz tags from Alien Technology.



915 MHz tag from Intermec Corporation.

2. Examples of active and semi-active tags from various vendors.

Mantis low UHF (303.8 MHz) active tag with built-in motion detector from RF Code , Inc.

2.45 GHz semi-active tags from Alien Technology.

Figure 3.2-12. 915 MHz/2.45 GHz semi-active tags from Trans Core .

Low UHF (303.8 MHz) fixed wired/wireless (802.11b) network reader from RF Code , Inc.

RFID smart label from Zebra Technologies.

RFID printer from Zebra Technologies.

UHF handheld reader from Intermec Corporation.

UHF Circular polarized reader antenna from Alien Technology.

Appendix B

PIC16F88

Microcontroller

PIC 16F88 Datasheet Snapshots:

PIC16F87/88

MICROCHIP

18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
  - Primary Run: RC oscillator, 76 μA, 1 MHz, 2V
  - RC_RUN: 7 μA, 31.25 kHz, 2V
  - SEC_RUN: 9 μA, 32 kHz, 2V
  - Sleep: 0.1 μA, 2V
- Timer1 Oscillator: 1.8 μA, 32 kHz, 2V
- Watchdog Timer: 2.2 μA, 2V
- Two-Speed Oscillator Start-up

Oscillators:

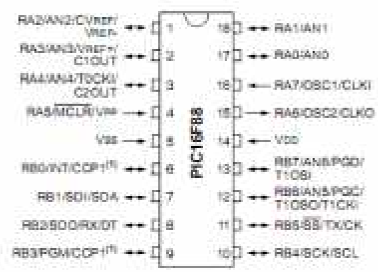
- Three Crystal modes:
  - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
  - ECIO: up to 20 MHz
- Internal oscillator block:
  - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

Peripheral Features:

- Capture, Compare, PWM (CCP) module:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I^{2}C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART/SCI) with 9-bit address detection:
  - RS-232 operation using internal oscillator (no external crystal required)
- Dual Analog Comparator module:
  - Programmable on-chip voltage reference
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Pin Diagram

18-Pin PDIP, SOIC



Note 1: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

Special Microcontroller Features:

- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
  - Programmable period from 1 ms to 268s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	SSP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	16	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	16	7	1	Y	2	Y	2/1

MICROCHIP

PIC16F87/88

18/20/28-Pin Enhanced Flash MCUs with nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
  - Primary Run: RC oscillator, 76 μA, 1 MHz, 2V
  - RC_RUN: 7 μA, 31.25 kHz, 2V
  - SEC_RUN: 9 μA, 32 kHz, 2V
  - Sleep: 0.1 μA, 2V
- Timer1 Oscillator: 1.8 μA, 32 kHz, 2V
- Watchdog Timer: 2.2 μA, 2V
- Two-Speed Oscillator Start-up

Oscillators:

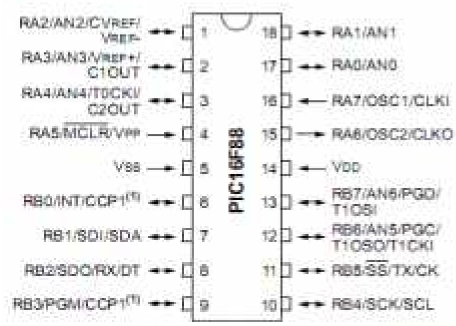
- Three Crystal modes:
  - LP, XT, HS: up to 20 MHz
- Two External RC modes
- One External Clock mode:
  - ECIO: up to 20 MHz
- Internal oscillator block:
  - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz

Peripheral Features:

- Capture, Compare, PWM (CCP) module:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit, 7-channel Analog-to-Digital Converter
- Synchronous Serial Port (SSP) with SPI™ (Master/Slave) and I²C™ (Slave)
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART/SCI) with 9-bit address detection:
  - RS-232 operation using internal oscillator (no external crystal required)
- Dual Analog Comparator module:
  - Programmable on-chip voltage reference
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Pin Diagram

18-Pin PDIP, SOIC



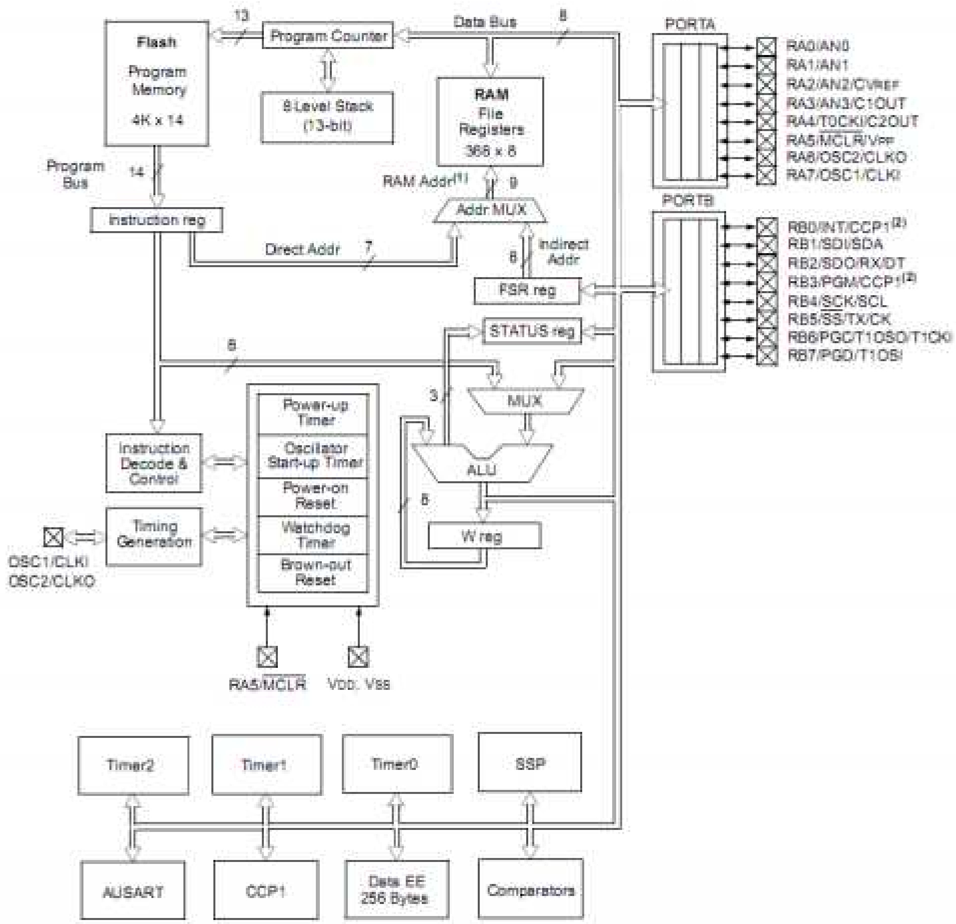
Note 1: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

Special Microcontroller Features:

- 100,000 erase/write cycles Enhanced Flash program memory typical
- 1,000,000 typical erase/write cycles EEPROM data memory typical
- EEPROM Data Retention: > 40 years
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Processor read/write access to program memory
- Low-Voltage Programming
- In-Circuit Debugging via two pins
- Extended Watchdog Timer (WDT):
  - Programmable period from 1 ms to 268s
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O Pins	10-bit A/D (ch)	CCP (PWM)	AUSART	Comparators	SSP	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)							
PIC16F87	7168	4096	368	256	16	N/A	1	Y	2	Y	2/1
PIC16F88	7168	4096	368	256	16	1	1	Y	2	Y	2/1

FIGURE 1-1: PIC16F87 DEVICE BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.

2: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

TABLE 1-2: PIC16F87/88 PINOUT DESCRIPTION

Pin Name	PDIP/ SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
						PORTA is a bidirectional I/O port.
RA0/AN0	17	19	23			
RA0				I/O	TTL	Bidirectional I/O pin.
AN0				I	Analog	Analog input channel 0.
RA1/AN1	18	20	24			
RA1				I/O	TTL	Bidirectional I/O pin.
AN1				I	Analog	Analog input channel 1.
RA2/AN2/CVREF/VREF-	1	1	26			
RA2				I/O	TTL	Bidirectional I/O pin.
AN2				I	Analog	Analog input channel 2.
CVREF				O		Comparator VREF output.
VREF-(4)				I	Analog	A/D reference voltage (Low) input.
RA3/AN3/VREF+/C1OUT	2	2	27			
RA3				I/O	TTL	Bidirectional I/O pin.
AN3				I	Analog	Analog input channel 3.
VREF+(4)				I	Analog	A/D reference voltage (High) input.
C1OUT				O		Comparator 1 output.
RA4/AN4/T0CKI/C2OUT	3	3	28			
RA4				I/O	ST	Bidirectional I/O pin.
AN4(4)				I	Analog	Analog input channel 4.
T0CKI				I	ST	Clock input to the TMR0 timer/counter.
C2OUT				O		Comparator 2 output.
RA5/$\overline{MCLR}$/VPP	4	4	1			
RA5				I	ST	Input pin.
$\overline{MCLR}$				I	ST	Master Clear (Reset). Input/programming voltage input. This pin is an active-low Reset to the device.
VPP				P	–	Programming voltage input.
RA6/OSC2/CLKO	15	17	20			
RA6				I/O	ST	Bidirectional I/O pin.
OSC2				O	–	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	–	In RC mode, this pin outputs CLKO signal which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA7/OSC1/CLKI	16	18	21			
RA7				I/O	ST	Bidirectional I/O pin.
OSC1				I	ST/CMOS(3)	Oscillator crystal input.
CLKI				I	–	External clock source input.

Legend: I = Input O = Output I/O = Input/Output P = Power
– = Not used TTL = TTL Input ST = Schmitt Trigger Input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
4: PIC16F88 devices only.
5: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

TABLE 1-2: PIC16F87/88 PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP/SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT/CCP1[5]	6	7	7			
RB0				I/O	TTL	Bidirectional I/O pin.
INT				I	ST[1]	External interrupt pin.
CCP1				I/O	ST	Capture input, Compare output, PWM output.
RB1/SDI/SDA	7	8	8			
RB1				I/O	TTL	Bidirectional I/O pin.
SDI				I	ST	SPI™ data in.
SDA				I/O	ST	I^{2}C™ data.
RB2/SDO/RX/DT	8	9	9			
RB2				I/O	TTL	Bidirectional I/O pin.
SDO				O	ST	SPI data out.
RX				I		AUSART asynchronous receive.
DT				I/O		AUSART synchronous detect.
RB3/PGM/CCP1[5]	9	10	10			
RB3				I/O	TTL	Bidirectional I/O pin.
PGM				I/O	ST	Low-Voltage ICSP™ Programming enable pin.
CCP1				I	ST	Capture input, Compare output, PWM output.
RB4/SCK/SCL	10	11	12			
RB4				I/O	TTL	Bidirectional I/O pin. Interrupt-on-change pin.
SCK				I/O	ST	Synchronous serial clock input/output for SPI.
SCL				I	ST	Synchronous serial clock input for I^{2}C.
RB5/$\overline{SS}$/TX/CK	11	12	13			
RB5				I/O	TTL	Bidirectional I/O pin. Interrupt-on-change pin.
$\overline{SS}$				I	TTL	Slave select for SPI in Slave mode.
TX				O		AUSART asynchronous transmit.
CK				I/O		AUSART synchronous clock.
RB6/AN5/PGC/T1OSO/T1CKI	12	13	15			
RB6				I/O	TTL	Bidirectional I/O pin. Interrupt-on-change pin.
AN5[4]				I		Analog input channel 5.
PGC				I/O	ST[2]	In-Circuit Debugger and programming clock pin.
T1OSO				O	ST	Timer1 oscillator output.
T1CKI				I	ST	Timer1 external clock input.
RB7/AN6/PGD/T1OSI	13	14	16			
RB7				I/O	TTL	Bidirectional I/O pin. Interrupt-on-change pin.
AN6[4]				I		Analog input channel 6.
PGD				I	ST[2]	In-Circuit Debugger and ICSP programming data pin.
T1OSI				I	ST	Timer1 oscillator input.
VSS	5	5, 6	3, 5	P	–	Ground reference for logic and I/O pins.
VDD	14	15, 16	17, 19	P	–	Positive supply for logic and I/O pins.

Legend: I = Input; O = Output; I/O = Input/Output; P = Power; – = Not used; TTL = TTL Input; ST = Schmitt Trigger input

Note
1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
4: PIC16F88 devices only.
5: The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

PIC source code for reader:

```
;receiver.asm


;Processor: PIC16C558 operating at 13.56 MHz
;           Ti= 295 nsec


        processor 16c558
        #include "P16c558.inc"
        __config h'3ff2'    ;protection off,PWRT enabled,watchdog disabled,HS oscillator


#define _CARRY           STATUS,0
#define _ZERO            STATUS,2
#define _125KHZ          PORTA,1
#define _RS232TX         PORTA,2
#define _RS232RX         PORTA,3
#define _RS232           PORTA
#define SIGNAL           PORTB,4
invmask                  = h'2'
;Define variables and constants here
delay       =h'20'
```

```
wait         =h'21'

acctime       =h'22' ;accumulated sync interval sum--also used as halfbit interval threshold

#define       halfthr      acctime    ;halfbit interval threshold

halfthr       =acctime ;halfbit interval threshold

recv_csumhi =h'23' ;2 bytes for storing received checksum

recv_csumlo =h'24'

bitcnt        =h'25' ;RS232 bit counter

cycle_cnt     =h'26'

halfthr       =h'27' ;threshold value between halfbit and fullbit intervals

ptr1          =h'28' ;temporary FSR storage

ptr2          =h'29' ;temporary FSR storage

TXchar        =h'2a' ;character to transmit over RS232

temp          =h'2b' ;temporary storage

shiftcnt      =h'2c' ;used to strip the framing '0' bits from the rec'd data array

letters     =h'2d' ;storage area for next character to send

charcnt       =h'2e'

lastbit       =h'2f' ;the LSb stores the last rec'd bit--flip it by complementing f


;bit storage area--16 bytes of storage, indirectly addressed

;;;Note that s/w tests for MSb to detect end of area--be careful if move to different

;;;processor or relocate this storage area
```

recvbits =h’40’ ;32 bytes set aside for storing the received bits--actual number of bytes

;in transmission is 18

;;Note that main loop uses bit tests to determine bit receive or runaway condition (to limit processing time). Keep this in mind if recvbits storage area changed in the future.

;;40h-60h is reserved for received bits--actual bit receiving area 40h-51h, rest is overrun area

;;52h-73h set aside for ASCII conversion of received bytes before RS232 transmission. Note that

;;52h-60h contains no useful information from the use during receive of demodulated bits. Also,

;; bits are not being received while the ASCII conversion and serial transmission are

;; taking place.

;; ‘G’ 1st character: “go”

;; Character 2-37: ASCII representation of received 18 bytes (until checksum used)

;; Character 38: ‘\n’ newline

sendascii =h’52’ ;begin of storage area for ASCII conversion of received bytes

xfercnt =d’14’ ;defines number of received bytes to convert to ASCII & transmit
;..

;Overall function- To recover Manchester encoded RFID message after AM demodulation and

; comparator decision. The comparator input trips the interrupt on PORTB change.

;The steps are:

;

; 1- Initialize registers to seek synch field.

; 2- Determine bit width from synch field by averaging the periods between transitions

; over the synch field. TMR0 is cleared at each edge. If the timer overflows before

; the next edge, synch seek starts over. The synch field is composed of 9 bits.

; 3- Use the measured bit width to establish a threshold period between repeat bits and

; complement of previous bit. This is due to the Manchester encoding method. Since there

; is always a transition in the middle of each bit interval transmitted, a repeated bit

; will appear as a pair of edges that occur with a halfbit interval period. A bit that

; is the complement of the last received bit will appear as an interval between edges

; of a full bit interval period.

; 4- Shift in bits as they are received into the storage array. When the timer overflows,

; consider the data field over. The received data format is MSb to LSb, where the MSb

; is the first bit received.

; 5- There are 16 bytes in the message, followed by a 16 bit checksum of the message

; contents. The remaining bit is unused.

; 6- Compute the checksum of the received 16 byte message and compare to the received

; checksum.

```
;       7- If checksums match, convert the message and the checksum into ASCII form and
transmit
;           over the RS232 serial link. The message format is:
;                 "GG" :the go characters (start of message)
;                  36 bytes which are the ASCII representation of the 18 bytes received
;                 "\n" : closing newline character
;           The serial data rate is 9600 bps, 8 data bits, 1 stop, no parity


        org h'000'              ;RESET vector location
        goto    init
        org h'004'              ;interrupt vector location


;;isr(): interrupt service routine
;         interrupts enabled for transition on PORTB
;
;         1- BEWARE! To minimize interrupt response time, the w & status register are NOT
;            archived.
;       2- The isr execution path is determined by w register and uses calculated goto's.
;            The w for next isr is set at end of current isr execution and is dependent on
;            signal context (i.e. sync start, w/in sync, w/in data, etc.)
;          Be very cautious here--must stay w/in 255 instructions for this to work!
```

```
;       3- Sync field processed as follows:
;              -Ignore the first 4 transitions, they may be in response to tag power on reset
;              -Accumulate the sum of next 8 intervals
;              -Establish half bit width from full bit width threshold value based on  average
interval measured above. Due to Manchester encoding, repeat of previous
;               bit will be a series of 2 halfbit width intervals, complement of previous bit
;               will be a fullbit width interval. halfbit defined as 1.5x(average sync).
;                 -wait for interval over the fullbit threshold. This is end of sync. In accordance
;               w/ Manchester encoding, the sync field will be: 1 1 1 1 1 1 1 1 0
isr
      addwf   PCL,f         ;4 calculated goto
;first sync edge is calculated goto here
      clrf    TMR0          ;5
      movf    PORTB,f       ;6 must read PORTB before clearing RBIF
      bcf     INTCON,RBIF ;7 just in case timer interrupt happened just at 1st edge
      bcf     INTCON,T0IF ;8
      movlw   (first_cycle - isr-d'1') ;9 next isr calculated goto offset
      clrf    lastbit       ;10   lastbit @ end of sync = 0
      retfie                ;12
;end of first cycle here. Note that first 4 transitions are ignored, because sync start is
;corrupted by tag power on reset.
```

```
first_cycle
      clrf    TMR0          ;5
      movf    PORTB,f       ;6 must read PORTB before clearing RBIF
      bcf     INTCON,RBIF ;7
      movlw   (second_cycle - isr-d'1') ;8 next isr calculated goto offset
      retfie                ;10
;end of 2nd cycle here. Note that first 4 transitions are ignored, because sync start is
;corrupted by tag power on reset.
second_cycle
      clrf    TMR0          ;5
      movf    PORTB,f       ;6 must read PORTB before clearing RBIF
      bcf     INTCON,RBIF ;7
      movlw   recvbits      ;8
      movwf   FSR           ;9 set up to store data bits
      movlw   (third_cycle - isr-d'1') ;10 next isr calculated goto offset
      retfie                ;12
;end of 3rd cycle here. Note that first 4 transitions are ignored, because sync start is
;corrupted by tag power on reset. The 3rd cycle is the 4th transition, so from here we
measure
;the longest interval in sync field.
third_cycle
```

```
    clrf   TMR0        ;5
    movf   PORTB,f     ;6 must read PORTB before clearing RBIF
    bcf    INTCON,RBIF ;7
    clrf   acctime     ;8  reset accumulated sync interval for average
    movlw  (fourth_cycle - isr-d'1') ;9 next isr calculated goto offset
    retfie             ;11
;end of 4th cycle here. Start looking for longest sync interval here.
fourth_cycle
    movf   TMR0,w      ;5
    clrf   TMR0        ;6
    movf   PORTB,f     ;7
    bcf    INTCON,RBIF ;8
    addwf  acctime,f   ;9  first measured sync cycle, must be the largest
    movlw  (fifth_cycle - isr-d'1') ;10
    retfie             ;12
;end of 5th cycle here.
fifth_cycle
    movf   TMR0,w      ;5
    clrf   TMR0        ;6
    movf   PORTB,f     ;7
    bcf    INTCON,RBIF ;8
```

```
    addwf   acctime,f     ;9    acctime = acctime + TMR0
    movlw   (sixth_cycle - isr-d'1') ;10
    retfie                ;12
;end of 6th cycle here.
sixth_cycle
    movf    TMR0,w        ;5
 clrf    TMR0          ;6
    movf    PORTB,f       ;7
    bcf     INTCON,RBIF ;8
    addwf   acctime,f     ;9    acctime = acctime + TMR0
    movlw   (seventh_cycle - isr-d'1') ;10
    retfie                ;12
;end of 7th cycle here.
seventh_cycle
    movf    TMR0,w        ;5
    clrf    TMR0          ;6
    movf    PORTB,f       ;7
    bcf     INTCON,RBIF ;8
    addwf   acctime,f     ;9    acctime = acctime + TMR0
    movlw   (eighth_cycle - isr-d'1') ;10
    retfie                ;12
```

```
;end of 8th cycle here.
eighth_cycle
        movf    TMR0,w        ;5
        clrf    TMR0          ;6
        movf    PORTB,f       ;7
        bcf     INTCON,RBIF ;8
        addwf   acctime,f     ;9    acctime = acctime + TMR0
        movlw   (nineth_cycle - isr-d'1') ;10
        retfie                ;12
;end of 9th cycle here.
nineth_cycle
        movf    TMR0,w        ;5
        clrf    TMR0          ;6
        movf    PORTB,f       ;7
        bcf     INTCON,RBIF ;8
        addwf   acctime,f     ;9    acctime = acctime + TMR0
        movlw   (tenth_cycle - isr-d'1') ;10
        retfie                ;12
;end of 10th cycle here.
tenth_cycle
        movf    TMR0,w        ;5
```

```
    clrf    TMR0          ;6
    movf    PORTB,f       ;7
    bcf     INTCON,RBIF ;8
    addwf   acctime,f     ;9    acctime = acctime + TMR0
    movlw   (eleventh_cycle - isr-d'1') ;10
    retfie                ;12
;end of 11th cycle here. --this is last of sync cycles to be accumulated. Average the result
;and determine halfbit threshold in remaining sync cycles.
eleventh_cycle
    movf    TMR0,w        ;5
    clrf    TMR0          ;6
    movf    PORTB,f       ;7
    bcf     INTCON,RBIF ;8
    addwf   acctime,f     ;9    acctime = acctime + TMR0
    movlw   (twelfth_cycle - isr-d'1') ;10
    retfie                ;12
;end of 12th cycle here. Start averaging the sync interval accumulated time
twelfth_cycle
    movf    PORTB,f       ;5
    bcf     INTCON,RBIF ;6
    rrf     acctime,f     ;7 acctime/2
```

```
    rrf     acctime,f     ;8 acctime/4
    rrf     acctime,f     ;9 avg interval = acctime/8
    movlw   h'1f'         ;10    clear 3 MSbs that may have been set by carry
    andwf   acctime,f     ;11
    movlw   (cycle13 - isr-d'1') ;12
    retfie                ;14
;end of 13th cycle here. Calculate the halfbit threshold = 1.5(sync interval avg) Note that
;that the threshold value will be kept in acctime (=halfthr)
cycle13
    clrf    TMR0          ;5
    movf    PORTB,f       ;6
    bcf     INTCON,RBIF ;7
    rrf     acctime,w     ;8 half the sync interval avg
    addwf   acctime,f     ;9 halfthr = 1+1.5x(sync interval avg)
    incf    acctime,f     ;10
    movlw   (sync_end - h'100'-h'1'-isr) ;11
    bsf     PCLATH,0      ;12   adjust for origin @ 100h
    retfie                ;14

    org     h'100'
;sync end wait. End of sync is distinguished by a fullbit interval. ( T > halfthr )
```

```
sync_end
      movf   TMR0,w        ;5
      clrf   TMR0          ;6
      movf   PORTB,f       ;7
      bcf    INTCON,RBIF   ;8
      subwf halfthr,w      ;9     Test interval to detect end of sync field (halfthr - w)
      movlw (sync_end - h’100’-isr-d’1’) ;10
      btfss STATUS,C       ;12    Carry set for halfthr >= w
      movlw (bit1 - h’100’-isr-h’1’);12        If T > halfbit, end of sync detected. Proceed to
data
processing
      retfie               ;14
;rec’d bit processing here --bit1 is 1st bit of 8 bit block
bit1
      movf   TMR0,w        ;5
      clrf   TMR0          ;6
      movf   PORTB,f       ;7
      bcf    INTCON,RBIF   ;8
      subwf halfthr,w      ;9    Test interval to determine bit. C = 1 for repeated bit
      btfsc STATUS,C       ;11
      goto   halfabit1     ;12
```

```
;fullbit processing here
      comf   lastbit,f      ;12   Complement lastbit for fullbit measurement
      rrf    lastbit,w      ;13
      rlf    INDF,f         ;14   shift in the new bit
      movlw (bit2 - h'100'-isr-h'1') ;15
      retfie                ;17
halfabit1
;repeated bit (1 of 8)
      rrf    lastbit,w      ;13
      rlf    INDF,f         ;14
      movlw (half21-h'100'-isr-h'1') ;15
      retfie                ;17
;2nd half of bit interval processing
half21        ;2nd half, bit1
      clrf   TMR0           ;5
      movf   PORTB,f        ;6
      bcf    INTCON,RBIF    ;7
      movlw (bit2-h'100'-isr-h'1');8
      retfie                ;10

;rec'd bit processing here --bit2 is 2nd bit of 8 bit block
```

```
bit2
    movf   TMR0,w         ;5
    clrf   TMR0           ;6
    movf   PORTB,f        ;7
    bcf    INTCON,RBIF    ;8
    subwf halfthr,w       ;9    Test interval to determine bit. C = 1 for repeated bit
    btfsc STATUS,C        ;11
    goto   halfabit2      ;12
;fullbit processing here
    comf   lastbit,f      ;12   Complement lastbit for fullbit measurement
    rrf    lastbit,w      ;13
  rlf    INDF,f         ;14   shift in the new bit
    movlw (bit3 - h’100’-isr-h’1’) ;15
    retfie                ;17
halfabit2
;repeated bit (2 of 8)
    rrf    lastbit,w      ;13
    rlf    INDF,f         ;14
    movlw (half22-h’100’-isr-h’1’) ;15
    retfie                ;17
;2nd half of bit interval processing
```

```
half22        ;2nd half, bit2
      clrf   TMR0          ;5
      movf   PORTB,f       ;6
      bcf    INTCON,RBIF    ;7
      movlw (bit3-h'100'-isr-h'1');8
      retfie               ;10
;rec'd bit processing here --bit3 is 3rd bit of 8 bit block
bit3
      movf   TMR0,w        ;5
      clrf   TMR0          ;6
      movf   PORTB,f       ;7
      bcf    INTCON,RBIF    ;8
      subwf halfthr,w      ;9   Test interval to determine bit. C = 1 for repeated bit
      btfsc STATUS,C       ;11
      goto   halfabit3     ;12
;fullbit processing here
      comf   lastbit,f     ;12   Complement lastbit for fullbit measurement
      rrf    lastbit,w     ;13
      rlf    INDF,f        ;14   shift in the new bit
      movlw (bit4 - h'100'-isr-h'1') ;15
      retfie               ;17
```

```
halfabit3
;repeated bit (3 of 8)
    rrf   lastbit,w     ;13
    rlf   INDF,f        ;14
    movlw (half23-h'100'-isr-h'1') ;15
    retfie              ;17
;2nd half of bit interval processing
half23      ;2nd half, bit3
    clrf  TMR0          ;5
    movf  PORTB,f       ;6
    bcf   INTCON,RBIF   ;7
    movlw (bit4-h'100'-isr-h'1');8
    retfie              ;10
;rec'd bit processing here --bit4 is 4th bit of 8 bit block
bit4
    movf  TMR0,w        ;5
    clrf  TMR0          ;6
    movf  PORTB,f       ;7
    bcf   INTCON,RBIF   ;8
    subwf halfthr,w     ;9   Test interval to determine bit. C = 1 for repeated bit
    btfsc STATUS,C      ;11
```

```
      goto   halfabit4      ;12

;fullbit processing here

      comf   lastbit,f      ;12   Complement lastbit for fullbit measurement

      rrf    lastbit,w      ;13

      rlf    INDF,f         ;14   shift in the new bit

      movlw (bit5 - h’100’-isr-h’1’) ;15

      retfie                ;17

halfabit4

;repeated bit (4 of 8)

      rrf    lastbit,w      ;13

      rlf    INDF,f         ;14

      movlw (half24-h’100’-isr-h’1’) ;15

      retfie                ;17

;2nd half of bit interval processing

half24        ;2nd half, bit4

      clrf   TMR0           ;5

      movf   PORTB,f        ;6

      bcf    INTCON,RBIF    ;7

      movlw (bit5-h’100’-isr-h’1’);8

      retfie                ;10

;rec’d bit processing here --bit5 is 5th bit of 8 bit block
```

```
bit5
      movf   TMR0,w         ;5
      clrf   TMR0           ;6
      movf   PORTB,f        ;7
      bcf    INTCON,RBIF    ;8
      subwf halfthr,w       ;9    Test interval to determine bit. C = 1 for repeated bit
      btfsc STATUS,C        ;11
      goto   halfabit5      ;12
;fullbit processing here
      comf   lastbit,f      ;12   Complement lastbit for fullbit measurement
      rrf    lastbit,w      ;13
      rlf    INDF,f         ;14   shift in the new bit
      movlw (bit6 - h'100'-isr-h'1') ;15
      retfie                ;17
halfabit5
;repeated bit (5 of 8)
      rrf    lastbit,w      ;13
      rlf    INDF,f         ;14
      movlw (half25-h'100'-isr-h'1') ;15
      retfie                ;17
;2nd half of bit interval processing
```

```
half25       ;2nd half, bit5
    clrf   TMR0         ;5
    movf   PORTB,f      ;6
    bcf    INTCON,RBIF   ;7
    movlw (bit6-h'100'-isr-h'1')      ;8
    retfie              ;10

;rec'd bit processing here --bit6 is 6th bit of 8 bit block
bit6
    movf   TMR0,w       ;5
    clrf   TMR0         ;6
    movf   PORTB,f      ;7
    bcf    INTCON,RBIF   ;8
    subwf halfthr,w      ;9    Test interval to determine bit. C = 1 for repeated bit
    btfsc STATUS,C       ;11
    goto   halfabit6     ;12
;fullbit processing here
    comf   lastbit,f     ;12   Complement lastbit for fullbit measurement
    rrf    lastbit,w     ;13
    rlf    INDF,f        ;14   shift in the new bit
    movlw (bit7 - h'100'-isr-h'1') ;15
```

```
    retfie              ;17
halfabit6
;repeated bit (6 of 8)
    rrf   lastbit,w     ;13
    rlf   INDF,f        ;14
    movlw (half26-h'100'-isr-h'1') ;15
    retfie              ;17
;2nd half of bit interval processing
half26      ;2nd half, bit6
    clrf  TMR0          ;5
    movf  PORTB,f       ;6
    bcf   INTCON,RBIF   ;7
    movlw (bit7-h'100'-isr-h'1') ;8
    retfie              ;10
;rec'd bit processing here --bit7 is 7th bit of 8 bit block
bit7
    movf  TMR0,w        ;5
    clrf  TMR0          ;6
    movf  PORTB,f       ;7
    bcf   INTCON,RBIF   ;8
    subwf halfthr,w     ;9   Test interval to determine bit. C = 1 for repeated bit
```

```
      btfsc STATUS,C       ;11
      goto   halfabit7     ;12
;fullbit processing here
      comf   lastbit,f     ;12   Complement lastbit for fullbit measurement
      rrf    lastbit,w     ;13
      rlf    INDF,f        ;14   shift in the new bit
      movlw (bit8 - h'100'-isr-h'1') ;15
      retfie               ;17
halfabit7
;repeated bit (7 of 8)
      rrf    lastbit,w     ;13
      rlf    INDF,f        ;14
      movlw (half27-h'100'-isr-h'1') ;15
      retfie               ;17
;2nd half of bit interval processing
half27        ;2nd half, bit7
      clrf   TMR0          ;5
      movf   PORTB,f       ;6
      bcf    INTCON,RBIF   ;7
      movlw (bit8-h'100'-isr-h'1') ;8
      retfie               ;10
```

```
;rec'd bit processing here --bit8 is 8th bit of 8 bit block
bit8
    movf   TMR0,w        ;5
    clrf   TMR0          ;6
    movf   PORTB,f       ;7
    bcf    INTCON,RBIF   ;8
    subwf halfthr,w      ;9   Test interval to determine bit. C = 1 for repeated bit
    btfsc STATUS,C       ;11
    goto   halfabit8     ;12
;fullbit processing here
    comf   lastbit,f     ;12  Complement lastbit for fullbit measurement
    rrf    lastbit,w     ;13
    rlf    INDF,f        ;14  shift in the new bit
    movlw (bit1 - h'100'-isr-h'1') ;15
    incf   FSR,f         ;16
    retfie               ;18
halfabit8
;repeated bit (8 of 8)
    rrf    lastbit,w     ;13
    rlf    INDF,f        ;14
    movlw (half28-h'100'-isr-h'1') ;15
```

```
        retfie              ;17

;2nd half of bit interval processing

half28        ;2nd half, bit8

        clrf   TMR0          ;5

        movf   PORTB,f       ;6

        bcf    INTCON,RBIF   ;7

        movlw (bit1-h'100'-isr-h'1') ;8

        incf   FSR,f         ;9    advance to next byte in recvbits storage array

        retfie              ;11


;The negative RS232 supply is generated by an inverter clocked at ~125 KHz by port pin RA1.

          ;first pump up the -5V, i.e. generate 125 KHz clock (T=8 usec, ~27 Ti)

          ;run for a total of 128 cycles before sending data

         ;put line at stop bit level

alphabet

          clrwdt

        bcf    INTCON,GIE    ;make sure interrupts are off

        movlw   sendascii

        movwf   FSR
```

```
    movlw   xfercnt     ;# of ASCII represented received bytes to xfer
    addlw   xfercnt     ;x2
    addlw   h'3'        ;plus 2 start character "G" and newline character at end
    movwf   charcnt
;;set up registers in bank 1
     bsf     STATUS,RP0    ;point to bank 1
     movlw    h'8'
     movwf    TRISA        ;RA3 input, RA2-0 output
     movlw    h'10'
     movwf    TRISB        ;RB7-5,3-0 output, RB4 input
     movlw    b'00001100' ;set up timer option for internal clock, prescale-->watchdog/16
     movwf    OPTION_REG    ;port B pullups enabled
     bcf     STATUS,RP0    ;point back to bank 0
;;done setting up registers in bank 1, back to bank 0
     bsf     _RS232TX   ;default is mark mode
     call    gen125khz


;start the test transmission
sendA
     movf    INDF,w
     movwf    TXchar
```

```
        movlw   d'8'
        movwf   bitcnt
       ;stop bit last
        bsf     _RS232TX
        call    TX_RS232     ;stop bit = 3Ti


        call    ti17         ;burn 17Ti (includes the 2Ti for the call)
        ;start bit first
        bcf     _RS232TX
        call    TX_RS232
        call    ti17        ;burn 17Ti (includes the 2Ti for the call, adjusts the bit timing)
sendchar
        btfsc   TXchar,0     ;1Ti
        goto    setbit       ;3Ti
        bcf     _RS232TX
        goto    nextbit
setbit
        bsf     _RS232TX     ;4Ti
nextbit
        call    TX_RS232     ;6Ti
        rrf     TXchar,f     ;7Ti
```

```
        call     ti10          ;17Ti
        decfsz bitcnt,f        ;18Ti
        goto    sendchar       ;20Ti
       ;stop bit last
        bsf      _RS232TX
        call     TX_RS232      ;stop bit = 3Ti


      incf    FSR,f        ;1
      decfsz charcnt,f     ;2
      goto    inalpha      ;4
      movlw   d’255’
      movwf   charcnt
      movlw   d’10’
      movwf   bitcnt
waiting
call    ti17
      decfsz charcnt,f
      goto    waiting
      decfsz bitcnt,f
      goto    waiting
      goto    seekinit
```

```
inalpha
    call    ti10
    goto    sendA
TX_RS232
      movlw    d'17'     ;time out 104 usec, Ti=295 nsec
      movwf    wait
to104
      movlw    invmask    ;flip voltage inverter bit
      xorwf    _RS232,f
      movlw    d'4'
      movwf    delay
wait4usec
      decfsz delay,f        ;4 usec is half inverter clock period
      goto     wait4usec
      decfsz wait,f
      goto     to104
      movlw    invmask
      xorwf    _RS232,f
      nop
      nop
      nop
```

```
        return
gen125khz
        movlw    d'128'
        movwf    cycle_cnt
next125
        bsf      _125KHZ
        movlw    d'4'
        movwf    delay
highside
        decfsz delay,f
        goto     highside
        bcf      _125KHZ
        movlw    d'4'
        movwf    delay
lowside
        decfsz delay,f
        goto     lowside
        decfsz cycle_cnt,f
        goto     next125
        return
```

```
ti17
        movlw    d'3'      ;1
        movwf    delay     ;2
burn9
        decfsz delay,f
        goto     burn9     ;11
        clrwdt             ;12
        nop                ;13
        return             ;15+2 for call ti17=17Ti

ti15
        movlw    d'3'      ;1
        movwf    delay     ;2
burn9Ti
        decfsz delay,f
        goto     burn9Ti   ;11
        return             ;13+2 for call ti15=15Ti

ti12
      nop
      clrwdt
```

```
ti10
        goto    dly1      ;2Ti
dly1
        goto    dly2      ;4Ti
dly2
        goto    leaveti10 ;6Ti
leaveti10
        return            ;8Ti+2Ti=10Ti
init
        ;1st set up the I/O configuration--note that setting PORTB 7,6,5,4,0 as outputs
disables
        ;them as external interrupt sources. In this application PORTB-4 is utilized as an
        ;external interrupt source upon change of state. All other external interrupt sources
are
        ;set as outputs to disable them as interrupts.

;;set up registers in bank 1
        bsf     STATUS,RP0   ;point to bank 1
        movlw   h'8'
        movwf   TRISA        ;RA3 input, RA2-0 output
        movlw   h'10'
        movwf   TRISB        ;RB7-5,3-0 output, RB4 input
```

```
movlw    b'00001000' ;set up timer option for internal clock, no prescaler
        movwf    OPTION_REG    ;port B pullups enabled
        bcf      STATUS,RP0    ;point back to bank 0
;;done setting up registers in bank 1, back to bank 0
      movlw   HIGH isr
      movwf   PCLATH        ;setup for calculated goto's dependent on context when entering
                            ;isr
seekinit
        clrwdt
      movlw   d'19'
      movwf   bitcnt      ;clear the bit storage field
      movlw   recvbits
      movwf   FSR
clrbits
      clrf    INDF
      incf    FSR,f
      decfsz bitcnt,f
      goto    clrbits

        movlw    recvbits
        movwf    FSR         ;start of the received bits field
```

```
      movf    PORTB,w ;read PORTB before clearing INTCON to be sure RBIF=0
      clrf    INTCON
      clrf    TMR0
movlw   d'0'
      clrf    PCLATH
      bsf     INTCON,RBIE    ;enable portB change interrupt enable
      bsf     INTCON,GIE     ;global interrupts are now enabled.
seeksync
      bcf      INTCON,RBIE
      movlw    d'0'     ;calculated goto offset for 1st sync edge processing
      clrf     PCLATH
      clrf     FSR       ;FSR = 0 to indicate not gathering bits
      bsf      INTCON,RBIE
      bcf      INTCON,T0IF
main
        clrwdt
      btfsc    FSR,6
      goto     datamain     ;receiving data, monitor progress
      btfsc    INTCON,T0IF
      goto     seeksync     ;if TMR0 overflows w/o receiving bits, seeksync
        goto     main
```

```
datamain
      clrwdt
      btfsc   INTCON,T0IF
      goto    calc_checksum ;if timer overflows, calculate checksum of received data
      btfsc   FSR,5         ;if bit 5 set, FSR > 5fh and has overrun its proper area.
      goto    seeksync       ;search for sync.
      goto    datamain
;Data received at this point. Two processing tasks remain:
;1- the framing ‘0’ bits must be removed from the received 14 data bytes and 16 bit
checksum
;2- the checksum of the 14 data bytes must be calculated and compared to the received
;    16 bit checksum
;If checksums match, transmit data over RS232 link.


calc_checksum
      clrf    INTCON
clrgie
      bcf     INTCON,GIE
       btfsc    INTCON,GIE    ;make sure it’s clear before proceeding
      goto    clrgie
      movf    PORTB,f
```

```
        clrf     INTCON          ;disable all interrupts while processing received data
;remove the framing ‘0’ bits by bit shifting the data array left until all framing 0s are
;shifted out

        movlw    d’17’
        movwf    bitcnt
        movwf    shiftcnt
shiftout
        movlw    recvbits+d’17’
        movwf    FSR
roll_left
        rlf      INDF,f
        decf     FSR,f
        decfsz   shiftcnt,f
        goto     roll_left       ;rotate left shiftcnt # of bytes
        decfsz   bitcnt,f
        goto     next_RL
        goto     framestripped
;bit shift left through the array (successively 1 byte less each time)
next_RL
        movf     bitcnt,w
```

```
        movwf    shiftcnt
        goto     shiftout
framestripped

;1st check for all 0s in data--This is an illegal combination
        movlw    recvbits
        movwf    FSR
        movlw    d’14’
        movwf    bitcnt
zerotest
        movf     INDF,w
        btfss    STATUS,Z
        goto     nonzero
        decfsz   bitcnt,f
        goto     zerotest
        goto     seekinit       ;all zeros received. Ignore the message
nonzero
;do 16 bit checksum of first 14 bytes received. It should match the last 2 bytes received.
        movlw    recvbits
        movwf    FSR
        movlw    d’14’
```

```
    movwf    bitcnt
    clrf     recv_csumlo
    clrf     recv_csumhi
sumbytes
    movf     INDF,w
    addwf    recv_csumlo,f
    btfsc    STATUS,C
    incf     recv_csumhi,f     ;carry into high byte as necessary
    incf     FSR,f             ;point to next data byte
    decfsz   bitcnt,f
    goto     sumbytes
;now compare the received checksum w/ the calculated checksum. Transmit data if they match.
    movf     recv_csumhi,w
    subwf    INDF,f
    btfss    STATUS,Z
    goto     seekinit
    incf     FSR,f            ;point to received checksum LSB
    movf     recv_csumlo,w
    subwf    INDF,f
    btfss    STATUS,Z
```

```
    goto    seekinit
;message passes checksum. Convert to ASCII and transmit.
;now convert to ASCII form
    movlw   recvbits
    movwf   ptr1        ;keep track of where in conversion
    movlw   sendascii
    movwf   ptr2
    movwf   FSR
    movlw   “G”
    movwf   INDF
    incf    ptr2,f
    incf    FSR,f
    movwf   INDF        ;double “G” to indicate start
    incf    ptr2,f      ;next ascii character
    movlw   xfercnt     ;how many bytes to convert to ASCII
    movwf   bitcnt
    movlw   h’4’
    movwf   PCLATH      ;set up PCLATH for lookup table
asciiconv
    movf    ptr1,w
    movwf   FSR
```

```
swapf    INDF,w
andlw    h'f'        ;isolate the MSN
call     hex2ascii
movwf    temp        ;hold the ASCII character
movf     ptr2,w
movwf    FSR
movf     temp,w      ;store ASCII representation of received byte MSN
movwf    INDF
incf     ptr2,f      ;advance ASCII ptr
movf     ptr1,w      ;back to received bytes
movwf    FSR
movf     INDF,w
andlw    h'f'        ;isolate the LSN
call     hex2ascii
movwf    temp
movf     ptr2,w
movwf    FSR
movf     temp,w      ;store ASCII representation of received byte LSN
movwf    INDF
incf     ptr2,f      ;advance ASCII ptr
incf     ptr1,f      ;advance received byte ptr
```

```
        decfsz    bitcnt,f
        goto      asciiconv
;done data conversion, now indicate newline before sending
        movlw     "\n"        ;newline character
        incf      FSR,f
        movwf     INDF
;cleared for RS232 transmission
        goto      alphabet


;hexadecimal to ASCII conversion table
        org       h'3ff'
hex2ascii
        addwf     PCL,f
        retlw     "0"   ;ascii 0
        retlw     "1"   ;ascii 1
        retlw     "2"   ;ascii 2
        retlw     "3"   ;ascii 3
        retlw     "4"   ;ascii 4
        retlw     "5"   ;ascii 5
        retlw     "6"   ;ascii 6
        retlw     "7"   ;ascii 7
```

```
retlw    "8"   ;ascii 8

retlw    "9"   ;ascii 9

retlw    "A"   ;ascii A

retlw    "B"   ;ascii B

retlw    "C"   ;ascii C

retlw    "D"   ;ascii D

retlw    "E"   ;ascii E

retlw    "F"   ;ascii F


 end
```

References:

1. RFID Design Principles, Harvey Lehpamer, 2008 ARTECH HOUSE.

2. Wikipedia Encyclopedia

   http://en.wikipedia.org/wiki/Radio-frequency_identification.

3. RFID source book, Sandip Lahiri.

4. RFID Handbook ,Fundamentals and Applications in Contactless Smart Cards andidentification ,Second Edition, Klaus Finkenzeller,Giesecke & Devrient GmbH, Munich, Germany, Translated by Rachel Waddington.

5. Radio-Frequency Electronics ,Circuits and Applications ,Second Edition ,Jon B. Hagen.

6. RFID-A GUIDE TO RADIO FREQUENCY IDENTIFICATION ,V. DANIEL HUNT,ALBERT PUGLIA ,MIKE PUGLIA ,WILEY-INTERSCIENCE,A John Wiley & Sons, Inc., Publication