Microcontroller Based RFID Car Security System

GROUP NAME

Zohaib Iftikhar

(01-113092-029)

Asfand Yar Khaliq

(01-113081-002)

Supervised by:

Mr. Arslan Qamar Malik

A Report is submitted to the Department of Computer Science,

Bahria University, Islamabad.

In partial fulfillment of requirement for the degree of BS (ETM).

بسم الله الرحمن الرحيم

CERTIFICATE

We accept the work contained in the degree project report titled“RFID Car Security System”as a confirmation to the required standard for the partial fulfilment the degree of BS-ETM.

Project Coordinator

Supervisor **Internal Examiner** **External Examiner**

Head of Department

Acknowledgements

We want to acknowledge all the people who made their efforts in supporting our work in one way or other. We predominantly want to be thankful to our families who meant to be real support for us in this whole time. We are obliged to many of faculty members and fellow students of this university who supported us and helped us to effectivelyenterprise this project. In particular, we want to mention our project supervisor **Mr.ArslanQamar**who has steered us bit by bitthrough every module of the project. His unswervingsupervision gives us approach to accomplish the chores of this project.

Dedication

Success of a project is always dependent upon gratitude, which is built by personal motivations and support. In our case parents were our total motivation through all the hard time we were been through and they were the ones who showed their total support to us while we were facing hard times. From deepness of our heart we respect all the efforts of our parents and we dedicate our whole work to our parents.

Abstract

Idea behind the project is to design a special security systemfor cars that is more users friendly and harder to be intervened. A system that turns a simple car into a key free car while switching Unlock/Lock control to RFID tag detection and reading system. A system that controls car's power supply and itself controls engine's ignition and which is monitored and protected by password protection. So the whole system adds up into a smart security system that can itself manage car's security and engine control.

Table of Contents

Table of Figures

1 Introduction

Cars and personal vehicles has become need of every common man running down the streets. In modern age most of the people own their own personal cars. But as they say with a great power comes great responsibility, will be interpreted it as “with great power comes greater chances of being harmed”.

As number of cars increasing, crime rate of car theft is increasing gradually. Number of cars you see in society brings along danger of more cars being theft. When something is in danger somebody in society always take step to eradicate that specific danger. As far as car security is concerned there are different companies which introduced different methods to ensure car safety. Some came up with idea of locking devices, some came up with car trackers, and everybody has a different mindset, so they purposed solutions what they could approach. A brief look on most commonly used security setups for car safety is as follows.

1.1 Remote Control Alarm System

Remote control alarm system is one of the most widely world-wide used security systems for vehicle protection. It is a basic type of circuitry but which is considered efficient for vehicle protection. Remote control alarm system is implemented as an alarm is installed in car that is raised when someone tries to access that car. Car is controlled by a remote that locks and unlock the car. When somebody tries to open the car with a key or anything (even the car’s original key) alarm is raised that will tend to alert owner about the circumstances and hence car is secured. Car can only be opened via its specially designed remote and there’s no other way of opening car. Car’s remote looks like:

Figure 1(Chapter1) Car Remote[1]

1.2 Car tracking system

Another device is introduced in the market, few years’ back which was considered one of the reasonable security systems, known as car trackers. Car tracking system is also one of the most widely adopted security systems for ensuring one’s car security. Car trackers are devices, you can consider them as navigation devices, which were originated from the idea of navigation systems. Idea was introduced as when you can follow the navigations while traveling or making a trip, then why shouldn’t be these navigations tracked to ensure car’s security.

Car tracking helps a user tracking whole trip of that car by continuously monitoring its navigation. So user can access a stolen car by monitoring those navigations and can get his car back. Also by setting car’s navigations limit, owner is always sent an alert when his car leaves that boundary limit. So by keeping the track record of one's car, car is protected. Whole tracking system can be elaborated by following diagram:

Figure 2(chapter1) Car Tracking System[2]

There are a lot of other security systems which are introduced in market in different periods which are promising car's security for a long time. These systems include keys with built in RF ID tags, password protected alarm locks, etc.

1.3 Problem statement

Problem that runs in the veins of society is car security problem that makes people feared of for owning their own cars. Though there are security systems introduced in market but there still exists problems, problems which need serious attention.

First problem lies in the systems which have introduced RFID systems as a door opener. RFID tag system is used for unlocking cars. They have embedded RFID tags in key itself which ensures security but itself raise a problem, problem of taking care of a bit larger key, that needs an extra care, and if that key is lost there is no other backup to access that car at once, i.e. if you're in emergency and you lost your key somewhere you can't access your car. And if someone learns how to intervene your RFID tag, he can even access your car, so in bigger perspective security system itself drags you into pile of problems you don't want to face.

Keys of cars are specially design to limit access to your car but what keys are designed; they have specific designs but common purpose, to ignite a car which most of people can do even without any key easily. Key system is not good to keep car secured.

Remote controls have provided an ease about car safety for a long time but they also do have some problems, one of the battery problems in remote device. Also remote control car alarm system has greater possibilities of being intervened. Remote control system is not enough for maintaining car's security.

There are other modules which provide password protected security and many other security protocols but they are way too expensive to be afforded by a common man. They are never an option for ones who have low budgets.

1.4 Proposed Solution

The solution proposed is to make a security system for car which does not have a key and RFID remote control embedded in key. A security system is built which will only be operated by RFID tag and keypad for password and control.Mechanism we are following is, RFIDtag is been detected by RF reader which will match RF tag and car will be unlock/locked. A keypad (or a touch panel in better scenario) will be installed on car dashboard, which will control car's power supply and will ignite car's engine to start it. We don't have to use any other security system for car's security because car is already safe by using RFID tag detection for door's control. Password will be protecting the car and will let the car start when someone types the right password. So by maintaining a secure password and RFID tag we can have a better security system for car which has very less chances to be intervened.

1.5 Objectives and Goals

- To develop a car security system
- Develop a cost efficient security system
- To build a system that is hard to be intervened
- Eradicate any key factors or remote problems
- Build a ignite control that don't let anyone start car by another means.

2 Literature Review

2.1 Microcontrollers:

Every system involves a MCU (Master Control Unit), which has the first priority when building a scenario for a system. In system, we are suggesting, microcontroller is MCU and we have to start project by it.

2.1.1 Choice of Microcontroller

Choice of a microcontroller is always a big deal for students while selecting a project and developing its different scenarios. Microcontroller has to do everything in mostly project; it depends upon your selection that can make your project simpler or complex. There are different categories of microcontrollers which are subcategorized into further devices.

2.1.1.1 80XX series

Intel introduce 80xx series three to four decades ago as a concept of microcontroller, actual motive of introducing series was to familiarize the world with new term microcontroller. Word was only familiar with the term microprocessor at that time, introducing a device which is itself a microprocessor but has its own in built RAM and ROM was a new idea for tech world.

Figure 3 (chapter 2) 8051 microcontroller[9]

Intel's series had 8051 (which is the most commonly known microcontroller) as its first member. Everybody, even in this era, who wants to give a start to microcontrollers, start it from 8051. 8051 is a start for every newbie but some people takes it as an easy option so they go for 8051 as a lifetime option which makes things worse. 8051 was a basic microcontroller with no features installed. Other member of 80xx series has more memory RAM, ROM, processing speed, but none of them came with inbuilt sensors, convertors, etc.

With no upgrade in features list makes 80xx series not a good option to be opted as MCU in any of project as it will involve development of every feature externally, which tends to make a circuit much bigger than it should be, and obviously a complexity factor is increased. For larger projects where space, budget and space matters, 80xx is never a choice, not even in worst case scenario.

For this project 80xx could never be an option, as synchronization of RFID with 80xx had a lot of difficulties and real time clock issues.

2.1.1.2 PIC Microcontrollers

Pic microcontrollers are introduced by microchip for real time advancement in technology. As mentioned earlier 80xx series was increasing complexity and price when complex systems were designed, which tend Microcontroller fabricators to think about a series of controllers which have some of these external features built in, that will involve complex programming skills but will reduce space of circuit and complexity in it.

Figure 4(Chapter 2)PIC microcontroller[12]

Microchip introduced PIC microcontroller as a series in which microcontrollers were categorized in different ways. Concept of only 40 pin microcontroller was overruled by microchip, they introduced different sized controllers. Less than 40 pin packages were introduced in the circuits where space is the main concern in whole project (obviously number of ports in microcontroller was compromised). Microchip introduced different protocols as inbuilt function in PIC microcontrollers, unlike 80xx series you don't need to develop an external circuit, you just have to read data sheet, program it for special feature and it will itself perform inner conversions and calculations.

PIC microcontrollers are much more convenient in terms of speed and memory and they have many exceptions for specific long compilation scenarios.

PIC microcontrollers are one of the most trending microcontrollers, most of the projects are PIC based, which revolutionized embedded systems industry again.

2.1.1.3 AVR Microcontrollers

Atmel introduce a new chain of microcontrollers in contrary to 80xx series, to retain their position in industry they had to introduce more efficient series of microcontrollers. Atmel introduced AVR microcontrollers which were more efficient in terms of performance, and are advanced with a lot of built-in features.

AVRs have built in synchronizations for voltage regulators, so as far as voltage regulations are concerned AVR is the best option in town. AVRs are much faster as compared to 80xx series and PIC microcontrollers.

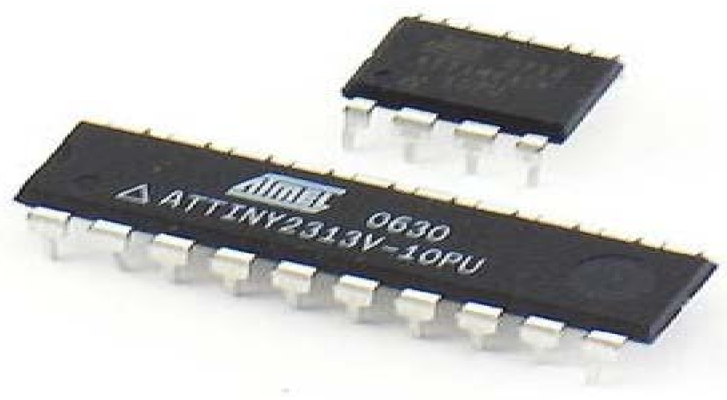


Figure 5(Chapter2) AVR microcontroller[13]

AVRs are more compatible for flash protocols and synchronous communication protocols as its computation speed is more than any other microcontrollers, which makes it a best option for any complex system.

2.1.1.4 ASICs

Abbreviated from “Application Specific Integrated Circuits” which are special type of microcontrollers that are not available in market and they can’t be categorized under one tag. Every member of family is meant for different purpose.

Embedded systems market was introduce with this different kind of series, when developing of a same product at a large level concerned, they thought of manufacturing a separate chip for that specific system. ASICs are specially designed for specific systems to improve speed, computation, efficiency and to avoid using chips with additional features that are never used in the system. ASICs have no particular size, pins package or anything they are just designed as per requirements of system.

ASICs have some disadvantages, they are so expensive that they can only be used in industrial projects; individual products can’t be controlled through it. ASICs don’t allow any R&D in system, once it’s designed it works as it is. To introduce any change in system you need to design a new IC for system. So ASIC will never be an option for this project.

So we had to opt out one of PIC microcontrollers and AVR microcontrollers, AVR are much faster and compatible for this system but to make a cost efficient system PIC microcontrollers are cheaper. PIC microcontrollers were chosen as to make a cost effective and a faster system.

2.1.2 Microcontroller Programming

Selection of microcontroller leads to programming of microcontroller. Writing an algorithm for whole project needs to be done properly. Programming of microcontroller involves few things; Selection of programming language is first on the list.

2.1.2.1 C or Assembly

You need to opt out a programming language out of all available options. Assembly is a deep level programming, also known as gate level coding. Assembly can help you in understanding developed algorithm at hardware level and to troubleshoot written algorithm at gate level. But when complex algorithms are involved, assembly is not a good option because coder will never understand where code leads to. Complex algorithms are always coded in C language.

After selection of programming language, compiler for programming microcontroller was selected. Compilers that can be used to program a PIC microcontroller are:

- Mikro C
- Hi Tech C Compilers
- MPLab

2.1.2.2 Mikro C

Mikro C is an advanced compiler, designed by MikroElectronica for programming PIC microcontrollers and AVRs. Mikro C has an advantage over all other compilers in list, it is a multi-language compiler i.e. it supports both C and assembly (though assembly can’t be codded directly, but you can convert C code into assembly for considerations which can be edited according to requirements). Code is later on molded into hex file which is then programmed on microcontroller.

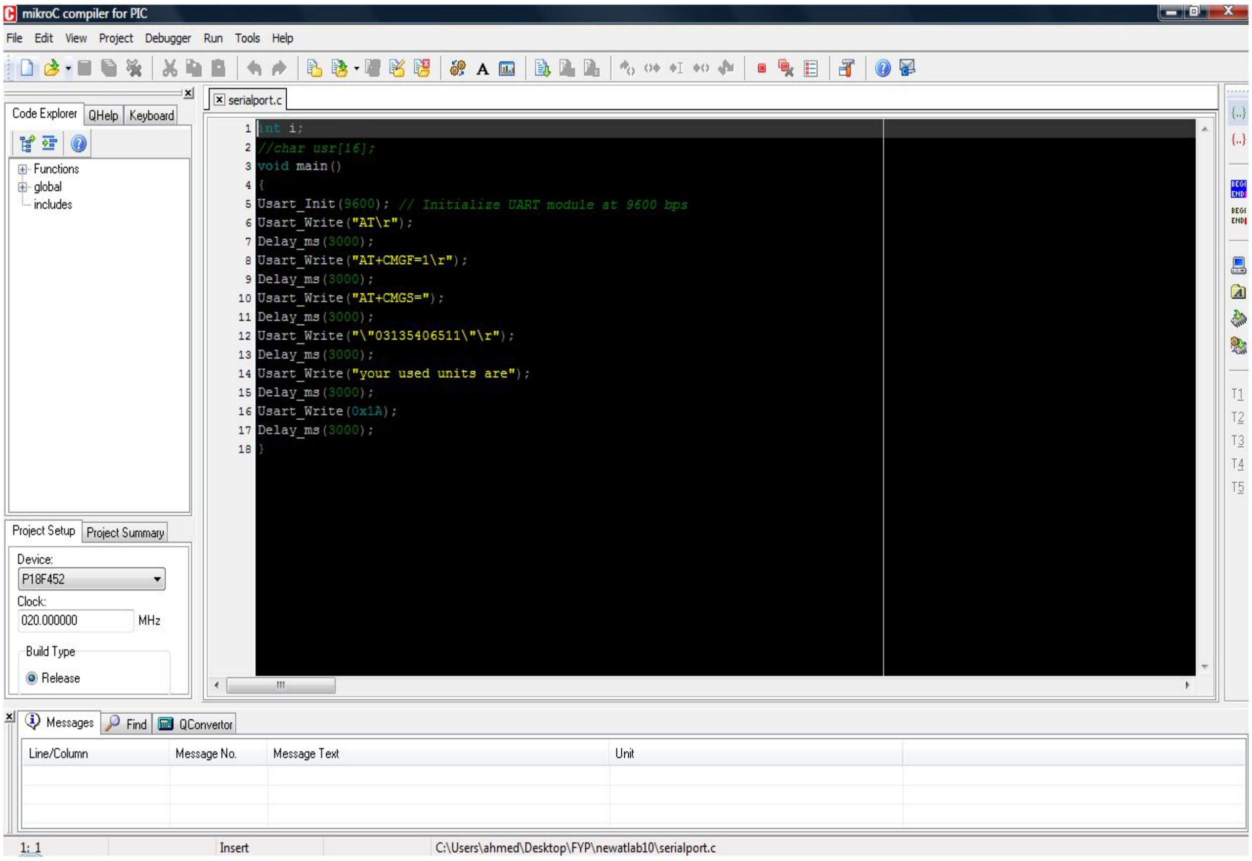


Figure 6 (Chapter2)Mikro C

Mikro C is a different kind of compiler with built in libraries which makes it user friendly and easy to handle for newbies. Programming via Mikro C is much easier as you can add a protocol from library with its hardware configuration and can easily use it. You don't need to write specific libraries for protocols.Mikro C also provides hardware synchronization configurations, i.e. pin configurations and their connectivity with other hardware devices.

Mikro C has disadvantages, as a user can never understand basic working of a specific protocol and its interfacing at hardware level, so any problem while operating a protocol will never be able to be solved.

2.1.2.3 MPLab

MPLab is second on list compiler to develop a source code algorithm for microcontroller. MPLab has one benefit that makes it prior, it has been itself developed by Microchip. MPLab is single language compiler i.e. C, compatible for only C programming. MPLab doesn't have any built-in library so you have to write every bit of code yourself.

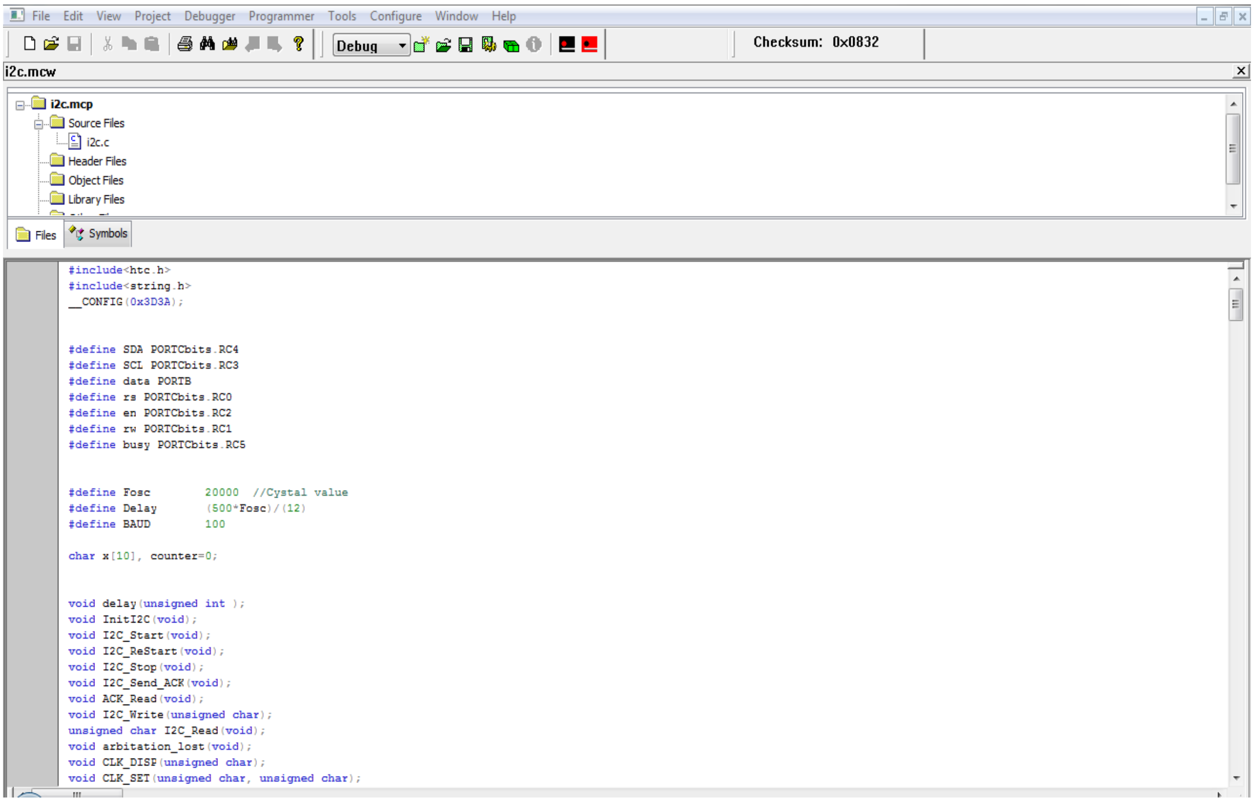


Figure 7 (Chapter2) MP Lab

Programming via MPLab is much harder in comparison with Mikro C but it has a plus point, it provides a better understanding of every protocol you're writing for. Second but one of important factors it helps you troubleshooting hardware at gate level that helps you in solving real time difficulties easily. MPLab has a graphical user mode which develops a graph or state diagram of code that helps in easy navigation through the circuit. MPLab is the most recommended software for writing an algorithm for PIC microcontrollers.

2.1.2.4 Hi Tech C Compilers

Hi-Tech C Compilers are a product of Hi-Tech software for programming PIC microcontrollers and AVRs. As its name suggest it only supports C as a programming language. Hi-Tech C Compilers have more libraries as compared to MPLab, so they support more functionality as compared to MPLab. These compilers have built in generic libraries which help in writing generic codes which almost work on every microcontroller regardless of its category or kind.

Hi Tech C Compilers are mostly used in industries and industrial level projects where having a compiler like Hi-Tech C compilers becomes handy for user.

MPLab is used for writing algorithm for project.

2.1.3 Writing on Microcontroller

After compilation and computation of algorithm, next step in the way is writing the same code on microcontroller. A compiler compiles a file to hex form which is actually a machine language that a controller understands. Burning a file (writing) can be done in following ways.

- On board Programmer
- External Programmer

2.1.3.1 On Board Programmer

On board programmer can easily be built with few steps. You only need to build a two wire jumper that is connected at receiver and transmitter pins of microcontroller. That's also known as J tagging. Most of the PC involves J tagging as they can't remove microcontrollers so they leave a spot to be used for programming microcontroller later.

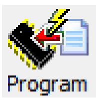
For a simple circuitry, it has its own benefits as you don't need to remove microcontroller again and again to program it, but when serial communication is performed in circuit, J tagging can be a hurdle in the system.

2.1.3.2 External Programmer

The other way around for programming a microcontroller is by using an external programmer. There are a lot of programmers available in market for different pin packages. There is a whole list of programmers for different pin packages which can be used according to microcontroller you're using. We used a Universal programmer for programming pic.

Programmer had drivers to be operated, known as Dataman Pro. You need to install Dataman Pro if you are tending to use Universal Programmer.

Step by step programming of microcontroller can be seen as follows.

1. Make a feasible connection of Dataman pro programmer with a computer and execute programmer's drivers.
2. Opt out specific microcontroller you're using in project.
3. Load compiled hex file of algorithm.
4. Search for Program icon 

as you need to configure clock and other basic microcontroller functionality manually.
5. A following window will pop up in your display.

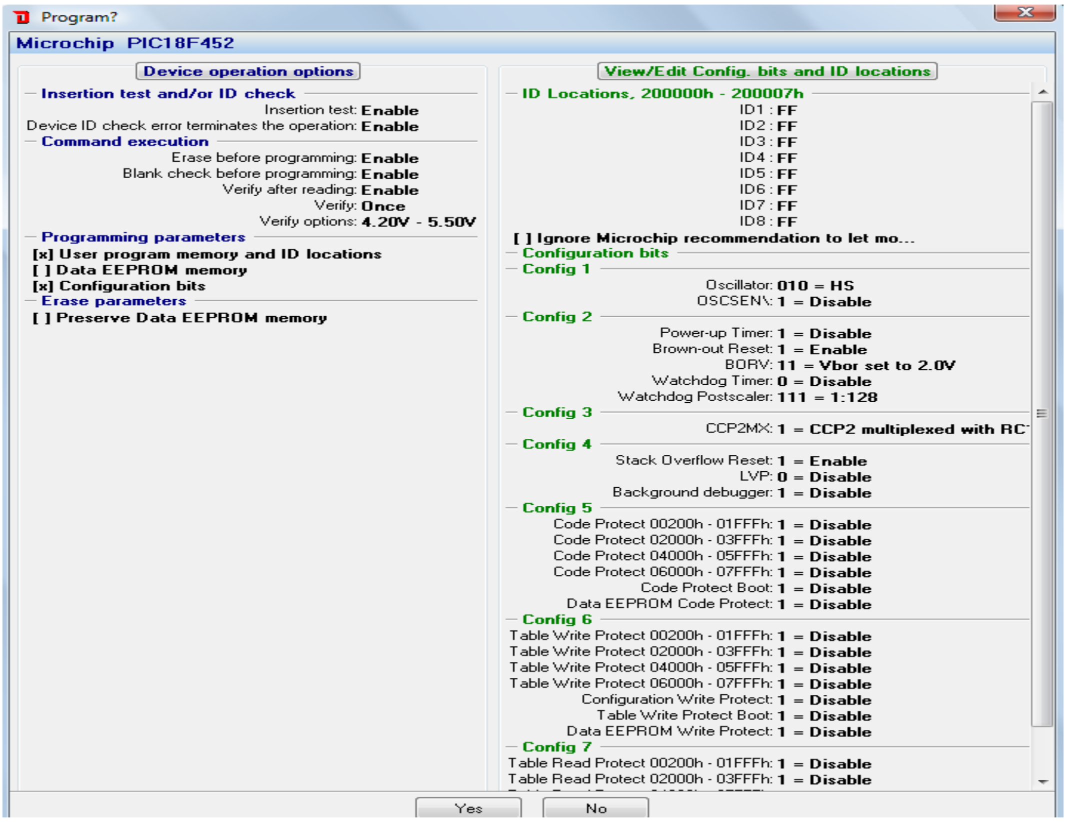


Figure 8 (chapter 2) Data man pro config.1

6. Click "View/Edit Config. bits and ID Locations" button at the top.
7. Configure settings manually according to your installed oscillator and other basic functionalities.

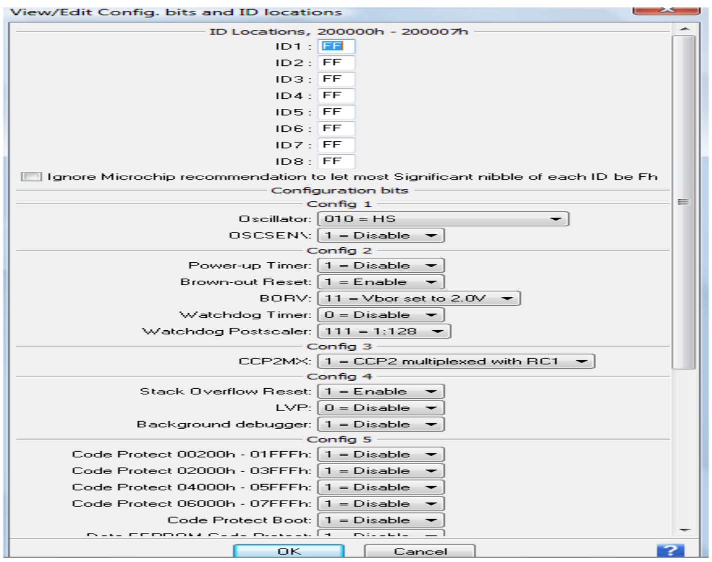


Figure 9 (chapter 2) Data man pro config.2

8. Click OK to save settings. And then 'Yes' at next popped up window.

You're done with programming microcontroller.

2.1.4 Choice of Oscillator

Post microcontroller selection lies Oscillator selection, which is one of the most important points while developing hardware, having a microcontroller as a master mind. Selection of a perfect oscillator is what you need. Most of the students opt out for a wrong one which leads them to failure of hardware and they don't understand how to solve the problem they are facing. Oscillators can be categorized in different manner.

2.1.4.1 Simple Oscillator

Simple oscillator seems somehow as follows:

Figure 10 (chapter 2) simple oscillator

We can categorize it as a simple oscillator because it's an ordinary oscillator with no special features and most commonly available in market. People usually opt out for this simple oscillator because it's easily available and cost efficient.

Simple oscillator works properly as far as duties of oscillator are concerned, but where it lags is its high sensitivity and accuracy. So what happens is, a little negligence leads to failure of oscillator circuitry as this oscillator provide you very little error margin. You can use this oscillator, as much as you want to but you need to be very careful with clock circuitry, because a little heating up while soldering or while running system, this circuitry will tend to fail.

2.1.4.2 High Speed Oscillator

A better solution for an oscillator circuit for newbies is building a clock using high speed oscillator. A high speed oscillator is an oscillator with same frequency but with larger margin of error in oscillations.

Simple Oscillator has to couple up with capacitors to complete circuitry for microcontroller clock. Using external capacitor involves a lot of problems for newbies, they handle it carelessly resulting in reduction of capacitance ultimately tends to failure of clock circuit. All you will end up into a failed circuit with no clue.

High speed oscillator comes up with 4 legged packages with built in capacitors, so there is no need of external capacitors, and there will be no chance of capacitance failure due to negligence. High speed oscillators are grounded and connected to circuit, and oscillator circuit runs perfectly.

Figure 11 (chapter 2) High speed oscillator

High speed oscillator has some disadvantages too, as they cover more space as compared to simple ones. They are costly as compared to simple oscillators. They are not easily available in market. So, the ones with practice will never go for high speed oscillator.

2.2 RFID

RFID is abbreviated from Radio frequency identification. RFID is a new emerging technology in the field of electronics that supports the remote controlling of devices from larger distances. RFID is a system that helps you operating at higher frequencies with the best part that RFID tags (normally called as RFID Cards) don't need any power supply or battery to be detected. RFID tags work at shorter distance while being powered and detected by electromagnetic field. RFID are better for used as security purposes as compared to other Infra-red remotes and wireless controls.

RFIDs are of different types categorized as passive ones and active ones. Some of tags need battery empowerment to transmit signal, or you can say to be detected. RFID has different operating frequency ranges depend upon the functionalities you're using it for and depend upon battery empowerments and cards configurations. RFID operates using electromagnetic field induction, with tags embedded with electronic produce codes.

RFIDs are used worldwide for security protocols and for many other long distanced and small distanced monitoring.

2.2.1 RFID reader

A RFID reader is a monitoring device which catechizes frequencies to read RFID tags. As mentioned earlier means of communication between reader and tag are wireless, so sightline between reader and tag is not obligatory. A reader is an RF module functionally, a module which has to transmit signal to detect cards and when cards are detected, module acts as a receiver for receiving bounced back signals. Transmitter consists of oscillator, modulator and amplifier. Oscillator generates career frequency, on which signal is to be modulated. Modulator intrudes data commands on carrier signal, so that data can be modulated. Amplifier acts as a booster, it boosts signal to such level that tag can be awaken.

Receiver, inbuilt in RFID reader, has a demodulator and amplifier. Demodulator demodulates the received signal to extract data from it. Amplifier boosts the data to such a level that it can easily be manipulated.

A microprocessor is installed to control whole working mechanism and forms control unit. Microprocessor holds operating system and applies data filters and stores data in memory. This explains how RFID reader works:

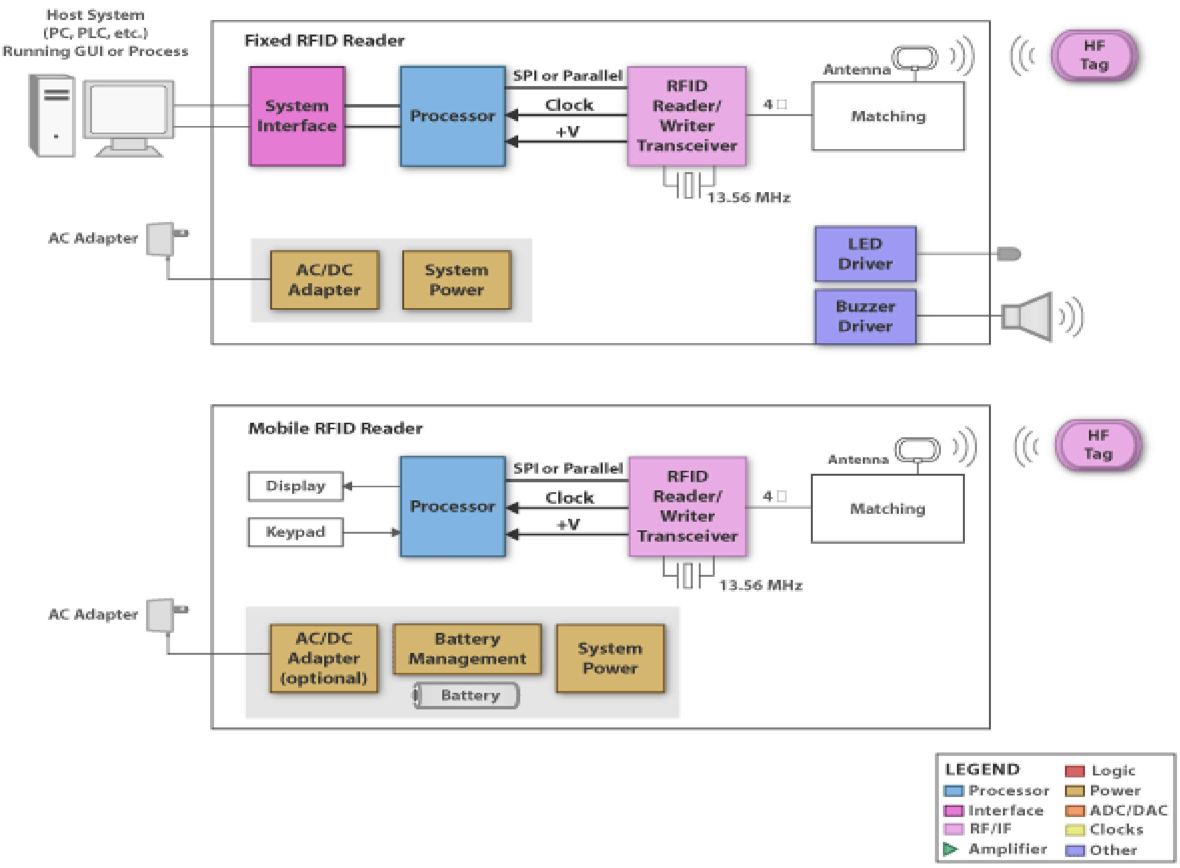


Figure 12 (chapter 2) RFID block diagram[10]

2.2.2 RFID Tag

RFIDtag is a microchip pooled with antenna in compact package. RFID tag is designed so that it can be attached to any object that is needed to be monitored. RFID tags are of various sizes starting from a size equal to rice grain to the size of small paperback book.

RFID tag works as an electromagnetic device that picks RF signals from reader via antenna, and then returns signal back loaded with some extra information like a serial number or data that is supposed to be contained by tag. Tags are further categorized as active and passive cards.

2.2.2.1 Active RFID Tags

Active RFID tags are the ones which are equipped with battery because they need battery empowerments to operate antenna and rest of tag's circuitry. They can't work without battery empowerments.

2.2.2.2 Passive RFID Tags

Passive RFID tags are the ones which are not equipped with any battery because they don't need battery empowerments. They operate only on the signal received by reader, which provides enough power for these cards.

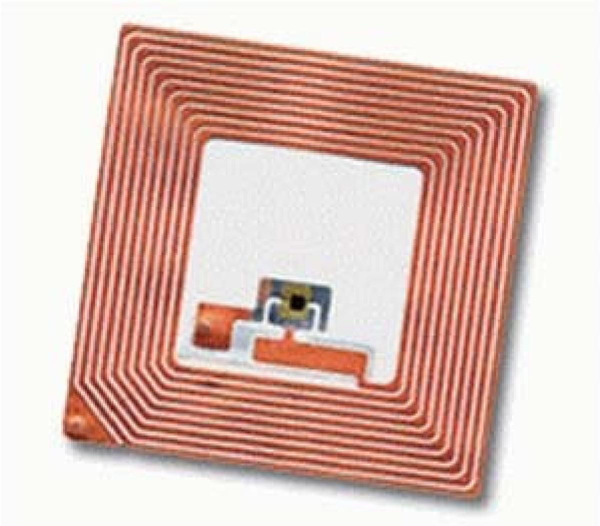


Figure 13(chapter 2) RFID tag[6]

3 System Methodology

Project methodology helps in understanding working mechanism of project, the way different devices inter-communicate in the circuit and perform their functionalities. A project can never be done in one go, or you can say a project is never planned as you start from one end and run to the other end of project. One always have to divide project in different modules, first these modules are separately made and tested, and then by the end they are interlinked to form one device, that's the best approach that can be made while doing project.

We divided our project in different modules; we will be talking about in this chapter, and then interlinked them. Our major modules are

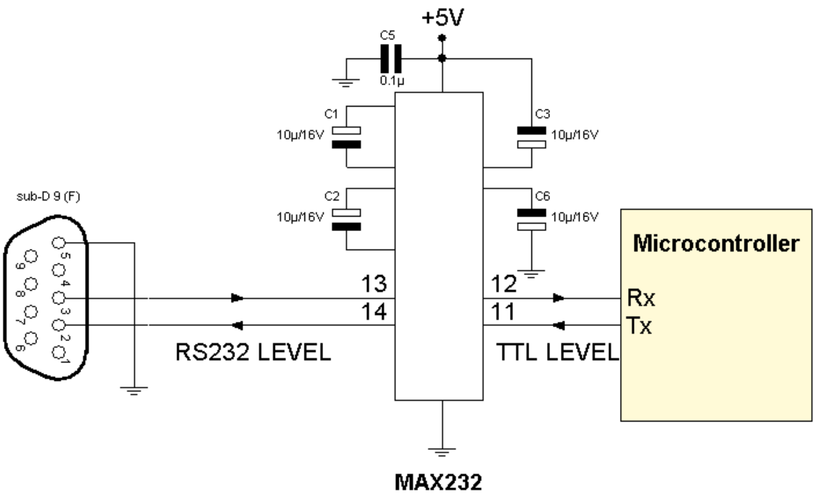
- RFID module
- Input unit (keypad)
- Switching unit
- Control unit

3.1 RFID module

RFID module consists of RFID reader, RFID tag and serial circuit. RFID module is purchased from market with developed RFID reader and tag, but they are operated serially, so a driver circuit was needed to be designed. Driver circuit for a RFID module is a serial communication port.

3.1.1 Serial Port

Serial port is designed using max 232 IC connected with DB9 connector to be connected with RFID module. Pin connections of max232 with DB9 connector are:



Serial communication made by Microcontroller is used to control RFID module in the circuit for using RFID for unlocking vehicles.

RFID module reader generates a signal which is reflected back by RFID tag when we swap it across reader. RFID tag bounces back signal with data that carries specific information, used to identify that special tag. The whole process of generating signal and reading signal is interpreted via serial communication.

3.2 Input Unit

Input unit in our project comprises of a keypad only that is used to send commands using operating keys, and used for entering password, which is used to lock engine access from unauthorized persons.

Keypad used is a simple telephonic keypad, with 16 keys, connected with an eight pin connecter cable. Keypad doesn't need any driver circuit or driving IC. It directly connects with control unit and operated directly. Keypad provides password to system, which on matching unlocks engine. Key 1 is pressed from keypad that will ignite the engine to start it.

3.3 Switching Unit

Switching unit is named to this part because it performs all the controlling in circuit, igniting, unlocking and all changes are performed by this unit of project. Switching unit is comprised of 3 relays used for 3 different purposes. These relays are directly connected to control unit and operated directly.

First relay is used for unlocking/locking doors. When switching unit is given a signal of tag matching, relay is switched which unlocks the car. Second relay is switched when control units order it to unlock engine for further operations. Third relay is switched when control unit is asked for igniting engine.

3.4 Control Unit

Control unit consists of microcontroller only that acts as brain of whole system that controls the whole system. It reads from RFID module and Input unit and manipulate readings and order switching unit accordingly. Control unit is operating all the modules at the same time while operating RFID serially.

3.5 Project Mechanism

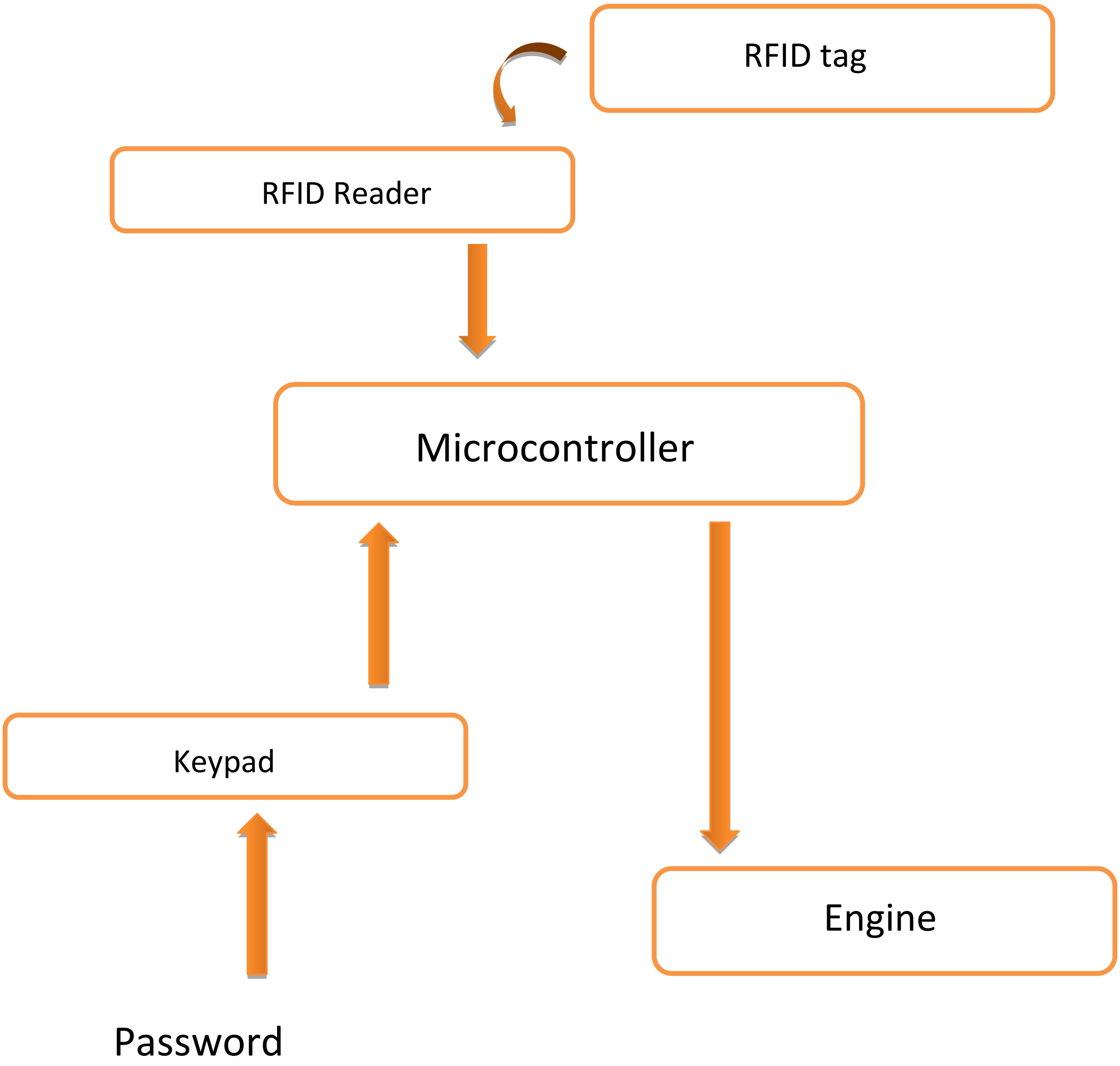
Whole project when interlinked and all units are integrated; it forms RFID car security system. It works as microcontroller is serially reading data from RFID reader, when a tag is detected that matches the one required by the system, microcontroller switches first relay, unlocking all the doors. After unlocking doors, microcontroller reads from keypad where password is entered. If password matches exactly with the one, already stored in database, microcontroller switches on the second relay, that unlock other privileges of engine controlled by system. Now all locks are opened. Now if you press a key (key 1) microcontroller reads it and sees if the same key has been assigned for igniting engine, it will switch on third relay for the same time, that key is pressed.

This is the way a smart security system is established that controls car security.

4 System Implementation

4.1 Flow Chart

The flow illustration of the entire scheme is as follows;



4.2 Simulations

Simulations of whole system, performed on Proteus are:

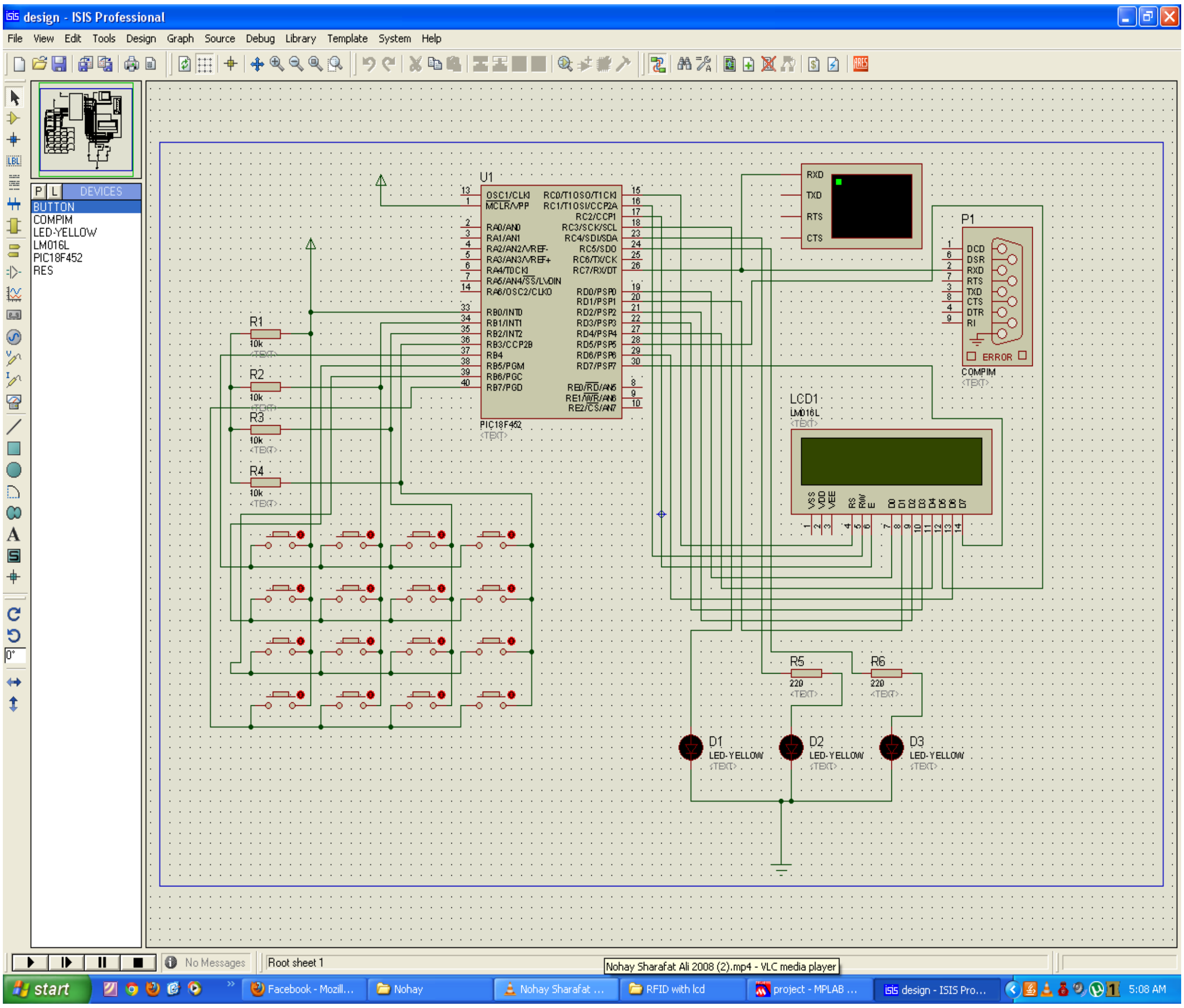


Figure 14 (chapter 4) Simulations

In simulations we used LCD for testing different modules. All the relays are replaced by LEDs that actually represent switching of relays.

4.3 Coding algorithms

Coded algorithm for whole protocol is written in appendix A.

5 System Requirements

RFID car security system requires few pre-installed features in the subject car. These features are mandatory to perform specific duties assigned to security system.

A central locking system must be installed in car. Central locking system is a system in which all locks are monitored by one switch, i.e. locks are operated by motors and there connection is appointed to one key, if key is pressed car is unlocked and vice versa. Central locking is required because system can never lock car itself, but it will control unlocking via that central unlocking/locking switch.

A security feature must be installed such that, until it is properly unlocked, no one can start car. Such features are available in remote control security systems; they should be present in car, as to ensure that car doesn't start until and unless password protection is unlocked. System will control engine via that engine locking feature, ensuring that car is password protected, you can never start car even if you have key.

Car's engine must be in good condition i.e. it can start only on igniting it, no manual conditions needed for it. As some cars need acceleration or some needs some other manual things to start them. Car installed with RFID car security system must be tuned for starting via signal ignition provided by the system.

System must be installed some place closer to door as system needs to read RFID tag so if system is installed in some place closer to door, it will be easier for system read tag at larger distances.

For better security, system must be installed at a place which is not prominent, i.e. system installed in some hidden place is more productive. If someone breaches car security, helping himself in unlocking car, he will still not able to manage to start car, or it will not be possible for him to find and breach security offered by this system.

Installation should be done by some experienced person while making proper electric connections, because short circuiting or lose connections might harm system performance.

6 Conclusion

Conclusion that can be originated from whole report and project is summarized in few words below. A system is considered as best options which are smart enough to be capable of taking care of it. Every system has its own benefits and drawbacks; you need to maintain trade-off between benefits and drawbacks to establish a better system.

A security system is considered good when it has its own control on everything, and there should be no way of compromising security in the system. A need of smart security system design lays there, as already designed systems are not smart enough as there always exist a way that can compromise whole system.

Use of RFID as a security system makes some sense that we are trying to generate a system which is hard to compromise, compromising a RFID reader is almost impossible until and unless you are aware with frequency of RFID tag used by the system. In surveying simple security systems, it is concluded that every simple security system can be compromised until and unless it is monitored by a smart device that is hard to be compromised.

Use of a keypad as to use password protection is a good idea to maintain security. So until and unless password is not provided, car can't be started or engine ignition is blocked. For remote control devices, and other devices there is still a way that can be used to avoid security loop.

Avoiding a key, and using a keypad for igniting engine is the best idea, as no key is involved then there is no short cut possible or can be interpreted as there is no chance of avoiding that security loop. Benefits of designing this system a controller base system were a controller can never be compromised. One must reprogram it to gain access to car, and that makes a car theft hard thing to do.

Losing keys, running out of remote battery were some common problems concluded in survey, a system designed using RFID and keypad, you need only to keep RFID tags and that's not a big thing to be contained. Keypad is attached fixed inside in the panel, so no one needs to be worried about it.

So overall conclusion based on technical and non-technical review of project is; a system based on RFID and keypad for security purposes is beneficial to be used, it costs less than any other security protocol. It has very less flaws as compared to any other car security system in the market already.

6.1 Future Enhancements

Some of the enhancements that can be made to this system for improved performance and credibility are:

Introduction of a touch system will improve user interface of system and make things easier in control.

Profile systems with multi-password are good to be introduced, as in building a data base of 2-3 passwords, every password for a specific person. That means if your car is driven by more than one person, then everybody has his own password.

RFID tags embedded in user’s watches or key chains can be made, that will help in carrying and protection of RFID tags.

Use of trackers controlled by microcontrollers’ can be helpful in a way, if car is to be monitored or, parents want to control car’s movement out of the area, they set as a limit for children.

7 References

1. http://image.made-in-china.com/2f0j00SMTEqsyISFbO/Car-Alarm-Remote-Control-CT-CN8003-.jpg (15-Februray,2013) (12:34a.m)
2. http://www.simocosystem.net/images/product-img2.gif (13-Februray,2013) (1:22a.m)
3. http://en.wikipedia.org/wiki/Radio-frequency_identification (12-Februray,2013) (5:34a.m)
4. http://www.ti.com/graphics/blockdiagram/blockdiagram_images/6283.gif (14-Februray,2013) (4:22a.m)
5. http://www.rfidreader.info/ (13-Februray,2013) (12:34a.m)
6. http://www.ictrfid.com/images/rfid-tag.jpg (12-Februray,2013) (12:30a.m)
7. http://www.technovelgy.com/ct/technology-article.asp?artnum=50 (12-Februray,2013) (12:35p.m)
8. http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=21 (13-Februray,2013) (12:46a.m)
9. http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=47 (15-Februray,2013) (12:55a.m)
10. http://smallbiztrends.com/wp-content/uploads/2011/02/rfid-large.jpg (15-Februray,2013) (1:00a.m)
11. http://3.bp.blogspot.com/-4nZFFeU7lPo/TfU13tfVFsI/AAAAAAAAARs/9TfJQ5hY2Zs/s1600/RS232%2BMenggunakan%2BMAX232.gif (14-Februray,2013) (12:00a.m)
12. http://upload.wikimedia.org/wikipedia/commons/thumb/1/18/PIC18F8720.jpg/220px-PIC18F8720.jpg (12-Februray,2013) (12:04p.m)
13. http://www.electronics-lab.com/blog/wp-content/uploads/2008/02/attiny45-2313.jpg (13-Februray,2013) (12:41p.m)
14. http://4.bp.blogspot.com/-MdSsugMcQm8/TXkMUetjk0I/AAAAAAAAAB4/uU-AlL5cvvw/s1600/intel%2B8051%2B.jpg (13-Februray,2013) (1:59 p.m)

8 Appendix

8.1 Appendix A

```
# include<htc.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

# include "delay.h"



//macros definition

#define C1              RB0

#define C2              RB1

#define C3              RB2

#define C4              RB3

#define R1              RB4

#define R2              RB5

#define R3              RB6

#define R4              RB7

#define ROW1() ((R1=0),(R2=1),(R3=1),(R4=1))

#define ROW2() ((R1=1),(R2=0),(R3=1),(R4=1))

#define ROW3() ((R1=1),(R2=1),(R3=0),(R4=1))

#define ROW4() ((R1=1),(R2=1),(R3=1),(R4=0))

#define LCD_RS                  RC0

#define LCD_RW                  RC1

#define LCD_EN                  RC2

#define rfidLED     RC3

#define keypadLED   RC4

#define s                       RC5
```

```
#define LCD_DATA        PORTD

#define BAUD            9600

#define _XTAL_FREQ      20000000



//microcontroller configuration

__CONFIG(1, OSCSDIS & HS);

__CONFIG(2,BOREN & BORV42 & PWRTEN & WDTDIS);

__CONFIG(3, CCP2RC1);

__CONFIG(4, DEBUGDIS & LVPDIS & STVRDIS);

__CONFIG(5, UNPROTECT);                  // Chnage to CPA

__CONFIG(6, WPB);

__CONFIG(7, TRU);



unsignedint check=0,i,k_rfidMatch,k_keypadMatch,index=0,x=0;

voidchk_isr(void);

voidlcd_cmd(unsigned char item);

voidlcd_data(unsigned char item);

voidlcd_string(unsigned char *str);

intread_keypad();

voidpassword_compare(unsigned char *pass,unsigned char *recieved);

intmatch_password(unsigned char keypad_pass[],unsigned char read[]);

unsigned char rfid[14]={'\0'};

unsigned char password[12]="42008A9081D9";

unsignedintkeypad_password[4];

unsignedintstr[4];
```

```
intkeypad,self_check=0;



//Interrupt driven function for serial port
static void interrupt isr(void)
{
if(RCIF==1)
   {
rfid[check++]=RCREG;     //Read RCREG

            }
}



void main()
{
TRISB=0x0F;
PORTB=0;
TRISD=0;
PORTD=0;
TRISC0=0;
TRISC1=0;
TRISC2=0;
TRISC3=0;
TRISC4=0;
TRISC5=0;
TRISC6=0;
```

```
TRISC7=1;

PORTC=0;

ADCON1=0b00000110;

INTCON=0b11000000;

PIE1=0b00100000;                              //Enable RX interrupt

TXSTA=0b10100000;

RCSTA=0b10010000;

SPBRG=(int)(_XTAL_FREQ/(64.0*BAUD)-1);

lcd_cmd(0x38); //clear lcd

DelayMs(10);

lcd_cmd(0x0e); //force cursor to the beginning of frst line

DelayMs(10);

lcd_string(" System On");

DelayMs(10);

keypad_password[0]=2;

keypad_password[1]=2;

keypad_password[2]=2;

keypad_password[3]=2;




while(1)

{

if(self_check==1)

        {

                if(RB1==0)

                {
```

```
                s=1;

                while(RB1!=1);

                s=0;
            }
        }

        if(check==12)
        {
            self_check=0;
check=0;
    CREN=0;
DelayS(1);
            lcd_cmd(0x38); //clear lcd

            DelayMs(10);
lcd_cmd(0x01);
            DelayMs(10);
            lcd_cmd(0x0e); //force cursor to the beginning of frst line
            DelayMs(10);
            password_compare(password,rfid);
            if(k_rfidMatch==0)
            {
                lcd_string(rfid);
                rfidLED=1;
                DelayS(2);
```

```
rfidLED=0;
lcd_cmd(0xC0); //force cursor to the beginning of frst line
DelayMs(10);
while(index<4)
{
keypad=read_keypad();
        if(keypad!=0)
        {
                lcd_data('*');
                if(keypad !=keypad_password[index])
                {
                x=1;
                }
                else
                {
                x=0;
                }
                keypad=0;
                index++;
                if(index==4)
                {
                        if(x==0)
                        {
                                keypadLED=1;
                                lcd_cmd(0x38); //clear lcd
                                DelayMs(10);
                lcd_cmd(0x01);
```

```
                                        DelayMs(10);
                                        lcd_cmd(0x0e); //force cursor to the
beginning of frst line
                                        DelayMs(10);
                                        lcd_string("Key Matched");
                                        self_check=1;
                                }
                                if(x==1)
                                {
                                        keypadLED=0;
                                        index=0;
                                        lcd_cmd(0x38); //clear lcd
                                        DelayMs(10);
                        lcd_cmd(0x01);
                                        DelayMs(10);
                                        lcd_cmd(0x0e); //force cursor to the
beginning of frst line
                                        DelayMs(10);
                                        lcd_string("Key not Matched");
                                        self_check=0;
                                }
                        }
                }

        }
        index=0;

    }
```

```
            if(k_rfidMatch!=0)
            {
                  rfidLED=0;
                  lcd_string("Password mismatch!");
                  DelayMs(10);
                  self_check=0;
            }
    CREN=1;
  }
}
}


// LCD COMMAND SENDING FUNCTION
voidlcd_cmd(unsigned char item)
{
  LCD_DATA = item;
  LCD_RS= 0;
  LCD_RW=0;
  LCD_EN=1;
DelayMs(1);
  LCD_EN=0;
return;
}


// LCD DATA SENDING FUNCTION
```

```
voidlcd_data(unsigned char item)

{

   LCD_DATA = item;

   LCD_RS= 1;

   LCD_RW=0;

   LCD_EN=1;

DelayMs(1);

   LCD_EN=0;

return;

}



// LCD STRING SENDING FUNCTION

voidlcd_string(unsigned char *str)

 {

int i=0;

while(str[i]!='\0')

        {

lcd_data(str[i]);

i++;

DelayMs(10);

        }

return;

 }



voidpassword_compare(unsigned char pass[],unsigned char recieved[]) //matching rfid number
```

```
{
    for(unsigned char i=0;i<12;i++)
    {
        if(pass[i]==recieved[i])
        {
            k_rfidMatch=0;
        }
        else
        {
            k_rfidMatch++;
        }
    }
}


intmatch_password(unsigned char keypad_pass[],unsigned char read[])
{
int k;
    for(unsigned char i=0;i<4;i++)
    {
        if(keypad_pass[i] ==read[i])
        {
            k=0;
        }
        else
        {
            k++;
```

```
                    }
          }
return k;
}

intread_keypad()
{int i=0;
ROW1();

DelayMs(10);

if (C1==0)
{
while(C1==0);
i=i+1;
}

if (C2==0)
{
while(C2==0);
i=i+2;
}

if (C3==0)
{
while(C3==0);
i=i+3;
```

```
}

if (C4==0)
{
while(C4==0);
i=i+4;
}

ROW2();


DelayMs(10);

if (C1==0)
{
while(C1==0);
i=i+5;
}
if (C2==0)
{
while(C2==0);
i=i+6;
}

if (C3==0)
{
while(C3==0);
```

```
i=i+7;

}


if (C4==0)

{

while(C4==0);

i=i+8;

}


ROW3();


DelayMs(10);


if (C1==0)

{

while(C1==0);

i=i+9;

}

if (C2==0)

{

while(C2==0);

i=i+10;

}


if (C3==0)

{

while(C3==0);
```

```
i=i+11;
}

if (C4==0)
{
while(C4==0);
i=i+12;

}

ROW4();
DelayMs(10);

if (C1==0)
{
while(C1==0);
i=i+13;
}

if (C2==0)
{
while(C2==0);
i=i+14;
}

if (C3==0)
{
```

```
while(C3==0);

i=i+15;
}

if (C4==0)
{
while(C4==0);
i=i+16;
}

return i;}
```

DELAY.H

```
#ifndef __delay_H
#define __delay_H

#define Crystal_Value           20
#define HalfBitDelay            (500*Crystal_Value)/(12)


voidDelayMs(unsigned int );


#endif
```

DELAY.C

```
 # include<htc.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
# include "delay.h"


voidDelayMS(unsigned int time)
{
unsignedinti,j;
        for (i=0; i<time; i++)
        {
                for (j=0; j<165; j++);
        }
}
```

8.2 Appendix B

Datasheet of 18F452