iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.36086

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Automatic Pantry System in Theatres using Embedded Systems

G. Naveen Kumar[1], T. Vignesh[2], Dr. Y. Srinivasulu[3]
[1, 2, 3]*UG Students,* [4]*Associate Professor, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana*

Abstract: ***One of the problems faced by the audience in theatres is getting food in break time standing in huge queue makes a person discomfort. Suppose a customer wants to order any food or breakfast, he can't access it immediately. Instead of that he will have to wait for service provider. Unless and until service provider take order, your order can't be posted. And after posting the order the passenger or customer will have to wait for delivery and billing purpose. so, we have an alternative where he can order the food from his seating place itself. These problems can be solved by using the AUTOMATED PANTRY ORDER SYSTEMS. This will provide the total automation technology so that customer can order at any moment of time. And the problems because of the fast delivery and billing can be solved. It is easy to handle and manage. It is also a customer-oriented service***

I. INTRODUCTION

A. Introduction of Embedded System

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used everyday, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner.

This is in direct contrast to the personal computer in the family room. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function rather; it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer
will do wish it. One customer may use it for a network file server another may use it exclusively for playing games, and a third may use it to write the next great American novel.

Frequently, an embedded system is a component within some larger system. For example, modern cars and trucks contain many embedded systems. One embedded system controls the anti-lock brakes, other monitors and controls the vehicle's emissions, and a third displays information on the dashboard. In some cases, these embedded systems are connected by some sort of a communication network, but that is certainly not a requirement.

At the possible risk of confusing you, it is important to point out that a general-purpose computer is itself made up of numerous embedded systems. For example, my computer consists of a keyboard, mouse, video card, modem, hard drive, floppy drive, and sound card-each of which is an embedded system. Each of these devices contains a processor and software and is designed to perform a specific function. For example, the modem is designed to send and receive digital data over analog telephone line. That's it and all of the other devices can be summarized in a single sentence as well.

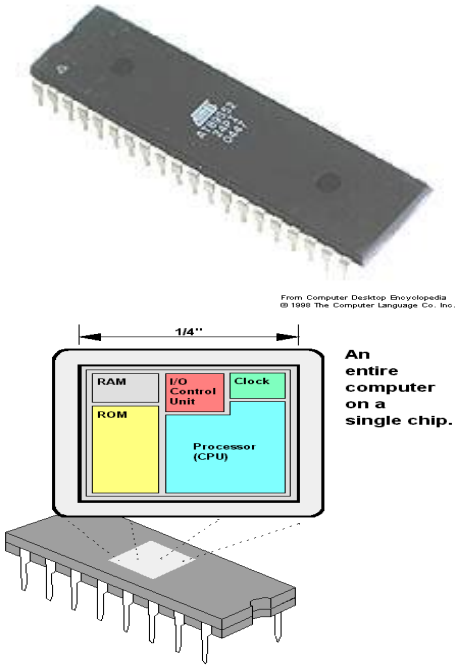
If an embedded system is designed well, the existence of the processor and software could be completely unnoticed by the user of the device. Such is the case for a microwave oven, VCR, or alarm clock. In some cases, it would even be possible to build an equivalent device that does not contain the processor and software. This could be done by replacing the combination with a custom integrated circuit that performs the same functions in hardware. However, a lot of flexibility is lost when a design is hard-cooled in this way. It is mush easier, and cheaper, to change a few lines of software than to redesign a piece of custom hardware.

B. A Brief History OF 8051

In 1981, Intel Corporation introduced an 8 bit microcontroller called 8051. This microcontroller had 128 bytes of RAM, 4K bytes of chip ROM, two timers, one serial port, and four ports all on a single chip. At the time it was also referred as "A SYSTEM ON A CHIP"

C. AT89S52

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory pro-grammer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller, which provides a highly flexible and cost-effective solution to many, embedded control applications. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM con-tents but freezes the oscillator, disabling all other chip functions until the next interrupt



1) 8031 has 128 bytes of RAM, two timers and 6 interrupts.
2) 8051 has 4K ROM, 128 bytes of RAM, two timers and 6 interrupts.
3) 8052 has 8K ROM, 256 bytes of RAM, three timers and 8 interrupts.

Of the three microcontrollers, 8051 is the most preferable. Microcontroller supports both serial and parallel communication.

In the concerned project 8052 microcontroller is used. Here microcontroller used is AT89S52, which is manufactured by ATMEL laboratories.

The 8051 is the name of a big family of microcontrollers. The device which we are going to use along this tutorial is the *'AT89S52'* which is a typical 8051 microcontroller manufactured by Atmel™. Note that this part doesn't aim to explain the functioning of the different components of a 89S52 microcontroller, but rather to give you a general idea of the organization of the chip and the available features, which shall be explained in detail along this tutorial.

The block diagram provided by Atmel™ in their datasheet showing the architecture the 89S52 device can seem very complicated, and since we are going to use the C high level language to program it, a simpler architecture can be represented as the figure 1.2. A. This figure shows the main features and components that the designer can interact with. You can notice that the 89S52 has 4 different ports, each one having 8 Input/output lines providing a total of 32 I/O lines. Those ports can be used to output DATA and orders do other devices, or to read the state of a sensor, or a switch. Most of the ports of the 89S52 have 'dual function' meaning that they can be used for two different functions: the fist one is to perform input/output operations and the second one is used to implement special features of the microcontroller like counting external pulses, interrupting the execution of the program according to external events, performing serial data transfer or connecting the chip to a computer to update the software.

D. Necessity Of Microcontrollers

Microprocessors brought the concept of programmable devices and made many applications of intelligent equipment. Most applications, which do not need large amount of data and program memory, tended to be costly.

The microprocessor system had to satisfy the data and program requirements so, sufficient RAM and ROM are used to satisfy most applications. The peripheral control equipment also had to be satisfied. Therefore, almost all-peripheral chips were used in the design. Because of these additional peripherals cost will be comparatively high.

1) An Example

2) 8085 chip Needs

An Address latch for separating address from multiplex address and data.32-KB RAM and 32-KB ROM to be able to satisfy most applications. As also Timer / Counter, Parallel programmable port, Serial port, and Interrupt controller are needed for its efficient applications.

In comparison a typical Micro controller 8051 chip has all that the 8051 board has except a reduced memory as follows.

4K bytes of ROM as compared to 32-KB, 128 Bytes of RAM as compared to 32-KB.

II. CIRCUITS AND THEIR OPERATION

A. Circuit Diagram

1) Menu ordering system

2) Operation of the circuit

In the order section,we are having power supply,keypad,microcontroller , LCD , RF transceiver.Power supply provides +5v to the microcontroller to switch on the microcontroller.Liquid cristal display ,displays the list of food items that are entered from the keyboard.Keypad is attached to the seat of the customer,so that he can order the list of food items.Keypad gives the input to the microcontroller and microcontroller the process input and produces the output.The output the microcontroller is given as input to the LCD.A 16x2 LCD has two registers.They are Command Register ,

Data Register.When command register stores the command instructions given to the LCD.when RS=0 it selects command register and RS=1 it selects data register.16x2 means 16characters perline ,we have two lines in LCD.RF Transcevier perform both transmission and reception it is wireless serial communication medium.it can transmit upto 100 metresand operating frequency is 433MHz. In the recevier section we are having power supply, buzzer ,PC ,LCD , RF recevier.The Rf trascevier transmits thr serial data and the RF receiver recevies the data and gives as a input to the LCD.Power supply is to switch on the circuit.Personal computer stores the data gicen by the microcontroller.Microcontroller the data and it is display on the lcd.

III. SOFTWARE DEVELOPMENT

A. Introduction

In this chapter the software used and the language in which the program code is defined is mentioned and the program code dumping tools are explained. The chapter also documents the development of the program for the application. This program has been termed as “Source code”. Before we look at the source code, we define the two header files that we have used in the code.

B. Tools Used

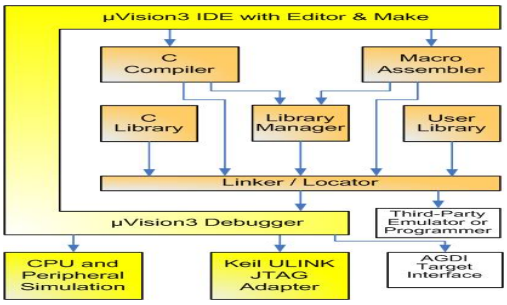


Figure 5.1 Keil Software- internal stages

Keil development tools for the 8051 Microcontroller Architecture support every level of software developer from the professional applications

C. C51 Compiler & A51 Macro Assembler

Source files are created by the µVision IDE and are passed to the C51 Compiler or A51 Macro Assembler. The compiler and assembler process source files and create replaceable object files.

The Keil C51 Compiler is a full ANSI implementation of the C programming language that supports all standard features of the C language. In addition, numerous features for direct support of the 8051 architectures have been added.

D. Start µvision

What's New in µVision3?

µVision3 adds many new features to the Editor like Text Templates, Quick Function Navigation, and Syntax Coloring with brace high lighting Configuration Wizard for dialog based startup and debugger setup. µVision3 is fully compatible to µVision2 and can be used in parallel with µVision2.

1) What is µVision3?

µVision3 is an IDE (Integrated Development Environment) that helps you write, compile, and debug embedded programs. It encapsulates the following components:

- A project manager.
- A make facility.
- Tool configuration.
- Editor.
- A powerful debugger.

  To help you get started, several example programs (located in the \C51\Examples, \C251\Examples, \C166\Examples, and \ARM\...\Examples) are provided.
- **HELLO** is a simple program that prints the string "Hello World" using the Serial Interface.
- **MEASURE** is a data acquisition system for analog and digital systems.
- **TRAFFIC** is a traffic light controller with the RTX Tiny operating system.
- **SIEVE** is the SIEVE Benchmark.
- **DHRY** is the Dhrystone Benchmark.
- **WHETS** is the Single-Precision Whetstone Benchmark.

  Additional example programs not listed here are provided for each device architecture.

2) Building an Application in µvision

To build (compile, assemble, and link) an application in µVision2, you must:

a) Select Project –

b) (forexample, 166\EXAMPLES\HELLO\HELLO.UV2)

c) Select Project - Rebuild all target files or Build target.

µVision2 compiles, assembles, and links the files in your project.

Creating Your Own Application in µVision2

To create a new project in µVision2, you must:

- Select Project - New Project.
- Select a directory and enter the name of the project file.
- Select Project - Select Device and select an 8051, 251, or C16x/ST10 device from the Device Database™.
- Create source files to add to the project.
- Select Project - Targets, Groups, Files. Add/Files, select Source Group1, and add the source files to the project.
- Select Project - Options and set the tool options. Note when you select the target device from the Device Database™ all special options are set automatically. You typically only need to configure the memory map of your target hardware. Default memory model settings are optimal for most applications.
- Select Project - Rebuild all target files or Build target.

3) Debugging an Application in µVision2

To debug an application created using µVision2, you must:

a) Select Debug - Start/Stop Debug Session.

b) Use the Step toolbar buttons to single-step through your program. You may enter **G, main** in the Output Window to execute to the main C function.

c) Open the Serial Window using the **Serial #1** button on the toolbar.

Debug your program using standard options like Step, Go, Break, and so on.

4) Starting µVision2 and Creating a Project

µVision2 is a standard Windows application and started by clicking on the program icon. To create a new project file, select from the µVision2 menu

a) *Project:* New Project…. This opens a standard Windows dialog that asks you for the new project file name. We suggest that you use a separate folder for each project. You can simply use the icon Create New Folder in this dialog to get a new empty folder. Then select this folder and enter the file name for the new project, i.e., Project1. µVision2 creates a new project file with the name PROJECT1.UV2 which contains a default target and file group name. You can see these names in the Project

b) *Window – Files:* Now use from the menu Project – Select Device for Target and select a CPU for your project. The Select Device dialog box shows the µVision2 device database. Just select the microcontroller you use. We are using for our examples the Philips 80C51RD+ CPU. This selection sets necessary tool options for the 80C51RD+ device and simplifies in this way the tool Configuration

5) Building Projects and Creating a HEX Files

Typical, the tool settings under Options – Target are all you need to start a new application. You may translate all source files and line the application with a click on the Build Target toolbar icon. When you build an application with syntax errors, µVision2 will display errors and warning messages in the Output
Window – Build page. A double click on a message line opens the source file
on the correct location in a µVision2 editor window. Once you have successfully generated your application you can start debugging.

After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. µVision2 creates HEX files with each build process when Create HEX files under Options for Target – Output is enabled. You may start your PROM programming utility after the make process when you specify the program under the option Run User Program #1.

6) CPU Simulation

µVision2 simulates up to 16 Mbytes of memory from which areas can be mapped for read, write, or code execution access. The µVision2 simulator traps and reports illegal memory accesses. In addition to memory mapping, the simulator also provides support for the integrated peripherals of the various 8051 derivatives. The on-chip peripherals
of the CPU you have selected are configured from the Device

7) Database selection

You have made when you create your project target. Refer to page 58 for more Information about selecting a device. You may select and display the on-chip peripheral components using the Debug menu. You can also change the aspects of each peripheral using the controls in the dialog boxes.

8) Start Debugging

You start the debug mode of µVision2 with the Debug – Start/Stop Debug Session command. Depending on the Options for Target – Debug Configuration, µVision2 will load the application program and run the startup code µVision2 saves the editor screen layout and restores the screen layout of the last debug session. If the program execution stops, µVision2 opens an editor window with the source text or shows CPU instructions in the disassembly window. The next executable statement is marked with a yellow arrow. During debugging, most editor features are still available.

For example, you can use the find command or correct program errors. Program source text of your application is shown in the same windows. The µVision2 debug mode differs from the edit mode in the following aspects:

⇒ The "Debug Menu and Debug Commands" described below are available. The additional debug windows are discussed in the following.

⇒ The project structure or tool parameters cannot be modified. All build Commands are disabled.

9) Disassembly Window

The Disassembly window shows your target program as mixed source and assembly program or just assembly code. A trace history of previously executed instructions may be displayed with Debug – View Trace Records. To enable the trace history, set Debug – Enable/Disable Trace Recording. If you select the Disassembly Window as the active window all program step commands work on CPU instruction level rather than program source lines. You can select a text line and set or modify code breakpoints using toolbar buttons or the context menu commands. You may use the dialog Debug – Inline Assembly… to modify the CPU instructions. That allows you to correct mistakes or to make temporary changes to the target program you are debugging.

E. Overview Of Keil Uvision Software

1) Proposed Double Threshold Method: The double threshold method proposed in this paper performs re-sensing. When the test statistic lies in between the two thresholds there is need to perform re-sensing. In this paper, this re-sensing is performed by incrementing the value of the number of samples taken in the double threshold methods. The two thresholds are taken as 2% above and below the confusion region. If the value of threshold lies above the threshold then, the primary user is said to be present, if it lies below that statistics then it is said to absent whereas for the re- sensing it should lie between the two thresholds. Algorithm for the proposed method can be explained as Algorithm:

a) Step 1: select P_{fa} = 0.1 and N=500.

b) Step 2: compute the test statistic given by

$$T(y) = N \sum_{n=1}^{N} y(n)^2$$

c) Step 3: calculate the value of two thresholds V_{th0} and V_{th1} .

□ □ V_{th0} : primary user present

□

d) Step 4: if test statistic □□ *V* □*th* 1 V □ *T*: (primary *y*) □ *V th*0 user: Re- absentsensing of samples. For Fig. 3 value of P_{fa} is taken as 0.1 and number of samples are taken to be 500. The graph is plotted for SNR having range between 0 to -20dB and for 1000 monte-Carlo simulations. From the graph between probability of detection and SNR, it can be shown that for SNR value of -12 dB, single threshold detection method is having the value of probability of detection equal to 0.2 approximately. Whereas double threshold method has probability of detection equal to 0.3 which is better than the single threshold method. But our proposed double threshold method gives a value of 0.55 which is better than conventional double threshold method also. Fig.4 shows graph of average number of samples required by the method proposed in this paper.

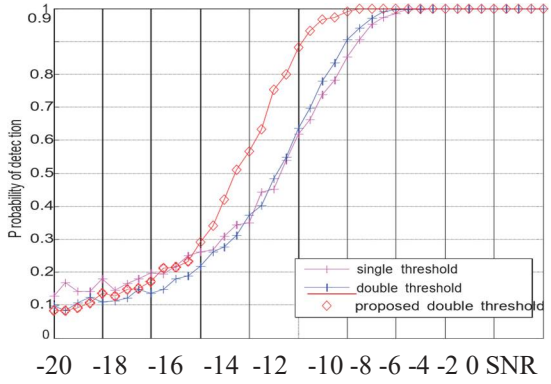
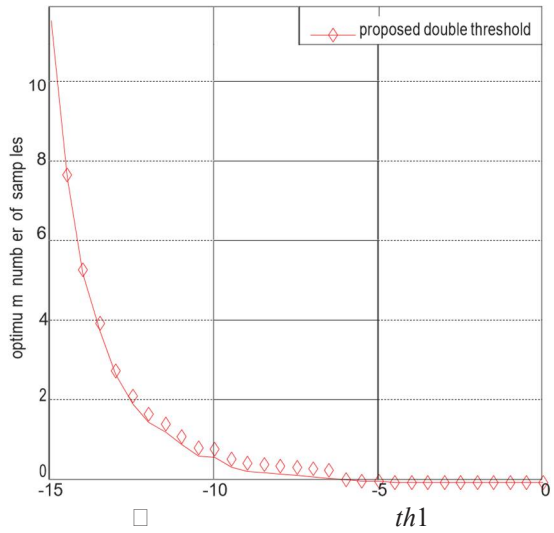


Fig.3. Probability of detection versus SNR x 10^4 12



e) Step 5: for re-sensing increment the value of N and go to step 2.
f) Step 6: repeat from step 2 till a decision is made.

IV. CONLUSIONS

Spectrum sensing techniques can be used to check the availability of the spectrum. Energy detection using single threshold is one of the simplest technique which can be used for spectrum sensing. In this paper, we have analyzed the three methods used to perform spectrum sensing. The two methods are single threshold detection method and double threshold detection method. Conventional energy detection method has certain limitations. So, in order to overcome these limitations double threshold detection method was proposed. The new double threshold method has proposed in this paper has better performance than the other two methods i.e. single threshold method and conventional double threshold method. The methods are analyzed on the basis of parameters such as SNR, probability of detection and number of samples. Simulations show that for different SNR value lying in the range between 0 to -20 dB double threshold detection method performs better than single threshold detection method.

A. Advantages

1) Rate of transmission and reception is fast.
2) User friendly device, easy to access for children as well as senior citizens.
3) Power required for operating the system is low.
4) Less expenditure
5) Low labour cost
6) Easy to handle

B. Applications

1) It is use in pantry service in train.
2) It can be used as E-menu in hotels or restaurants

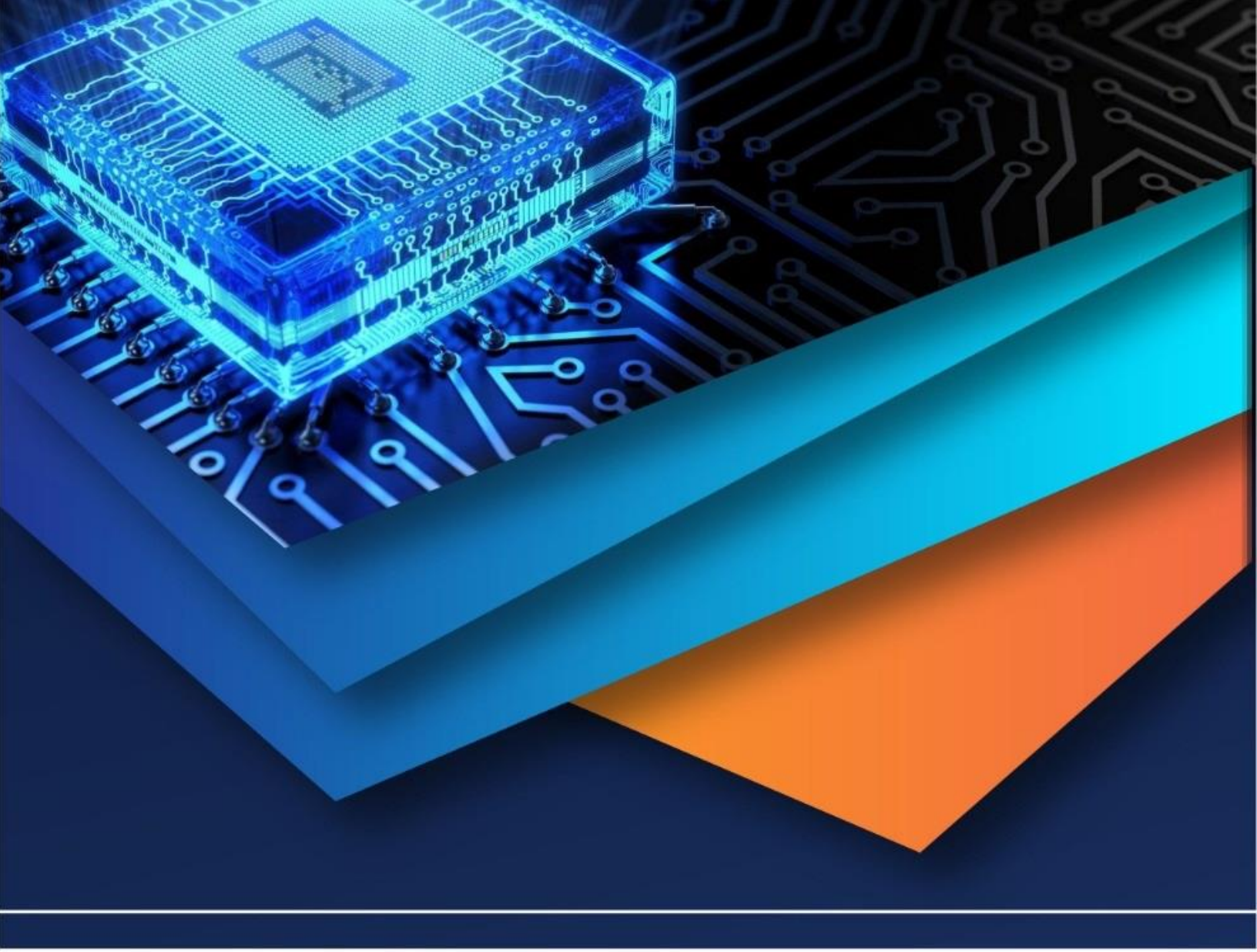
It can be used as a E-notice board where user can type their queries get their answer.

V. CONCLUSION

The project “” has been successfully designed and tested. Integrating features of all the hardware components used have developed it. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced IC’s and with the help of growing technology the project has been successfully implemented.

REFERENCES

[1] M. Thompson and J. K. Fidler, "Fast antenna tunning using transputer based simulated annealing", Electron. Lett.vol. 36, no. 7, pp. 603–604, Mar. 2000.
[2] Markus R Pfaffinger "Higher Vibration Modes in Railway Tracks at their cutoff frequencies", thesis, Swiss federal institute of technology, 2000.
[3] Shen-Haw Ju,Hung-Ta Lin,Jeng-Yuan Huang, "Dominant Frequencies of Train Induced Vibrations", thesis, march 2008.
[4] Shweta Shashikant Tanpure, Priyanka R. Shidankar, Madhura M. Joshi, "Automated Food Ordering System with Real-Time Customer Feedback", IJRCSSE, vol.3, issue 2, Feb.2013.
[5] Kamran Sartipi, "Design of fast food restaurant system using statement tool", thesis, university of Waterloo, Ontario, Canada, N2L, 3G1, Nov.1995.
[6] P.Khanja,S.Wattanasirichaigoon,J.Natwichai,S.Noimanee, L.Ramingwong,"A Web Base System For ECG Data Transfered using ZIGBEE/IEEE Technology", ISBME, 2008.
[7] Stefan Soucek, Gerhard Russ, Clara Tamarit, "The Smart Kitchen Project – An Application of Fieldbus Technology to Domotics", Institute of Computer Technology Gusshausstr, 27-29/384, A-1040, Wien, Austria.
[8] Biming Tian, Song Han, Liu Liu, Saghar Khadem, Sazia Parvin, "Towards enhanced key management in multi-phase ZigBee network architecture", vol.35, pp.579-588, Dec.2011.
[9] Alexander Divinsky, Matthew Donders, Andrew Durfee, Christopher Lesko, "Smart Kitchen Inventory Management System", thesis, ECE 423 – Senior Design, Group 11, July 2010



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY