ADVANCE ADAPTIVE SECURITY SYSTEM

Project Report submitted in partial fulfillment of the requirement
for the degree of

Bachelor of Technology.

In

Electronics and Communication Engineering

Under the Supervision of

Dr. Vinay Kumar

&

Dr. Rohit Sharma

By

Karandeep Singh (061059) **Nikhil Pandey (061081)**

Rahul Garg (061104) **Rohit Singhal (061113)**

To

JAYPEE UNIVERSITY OF
INFORMATION TECHNOLOGY

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that the work entitled, "ADVANCED ADAPTIVE SECURITY SYSTEM" submitted by KARANDEEP SINGH, NIKHIL PANDEY, RAHUL GARG, and ROHIT SINGHAL in partial fulfillment for the award of degree of Bachelor of Technology (B.Tech) in ELECTRONICS AND COMMUNICATION ENGG. of Jaypee University of Information Technology has been carried out under my supervision.

This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Date: 24-05-10

SUPERVISORS:

DR. VINAY KUMAR

(Asst. Professor)

DR. ROHIT SHARMA

(Asst. Professor)

Acknowledgment

We wish to express our earnest gratitude to **DR. ROHIT SHARMA** and **DR. VINAY KUMAR,** for providing us invaluable guidance and suggestions, which allowed us to present our project in its final form.

We would also like to thank all the staff members of Electronics and Communication Engineering Department of Jaypee University of Information Technology, Waknaghat, H.P. (INDIA), especially Mr. Pramod Kumar and Mr. Manoj Pandey, for providing us with all the facilities required for the completion of our project.

Date:

Name of Students:

Karandeep Singh
061059
ECE

Nikhil Pandey
061081
ECE

Rahul Garg
061104
ECE

Rahul Garg

Rohit Singhal
061113
ECE

Rohit

Table of Content

List of Figures

List of Snapshots

List of Abbreviations

GUI	- Graphic User Interface
CCTVs	- Closed Circuit Televisions
LASERs	- Light Amplification by Stimulated Emission of Radiation
TV	- Television
IC	- Integrated Circuit
SOIC	- Small-Outline Integrated Circuits
PDIP	- Plastic Dual-Inline Package
BJTs	- Bi-Junction Transistors
LEDs	- Light Emitting Diode

List of Units

MHz	- Mega Hertz	(Frequency)
V	- Volt	(Voltage)
°	- Degree	(Angle)
A	- Ampere	(Current)

Abstract

Security has always been an issue for banks and owners of highly valued, sought after commodities. As time progresses public accessibility also increases, thus security has to be compromised for comfort. This in turn results in a security with some loopholes, hence ways to bypass the security are always found in spite of stringent measures taken. The proposed system accounts for a non-contact based security system which deploys a low level electric field and detects any disturbance in the same. The Adaptive Nature of the system ensures security levels to be flexible, which changes from time to time, thereby reducing the predictability of the system and maximizing reliability. The system when used in conjunction with pre-existing systems can be deemed as a considerable step up in security. In addition the GUI in the system allows for easy handling at the user end, owing to the fact that it does not require skilled personnel to operate the system, thereby reducing the dependability of the system on a particular user.

CHAPTER 1

Introduction

Need for security is as old as human civilization. As soon as humans began settling in small groups and began owning things, the need to protect them from other people or animals or even from weather arose.

In this manner the concept of security was conceived. The earliest forms of security were the creation of houses, where humans stored their belongings. As they began to own animals, barns came into being. Slowly, the simple houses gave way to more sophisticated houses with secret rooms or basements.

As civilization progressed, historical and legendary artifacts became collector's items and were greatly sought after. Some of these artifacts were also passed down in families and were zealously guarded. The methods varied from secret locations, armed guards, hidden rooms or fake copies being developed.

As the accessibility to these artifacts increased for general public, other means were developed. For storing and securing money, jewels, artifacts seldom used and heirlooms, banks were used.

Slowly museums and art galleries came into being. Earlier security measures included armed guards and frisking while entering and departing the said places. Then came into being, hidden traps, which were placed near artifacts and at strategic places.

With further advancements in technology, surveillance cameras and CCTVs were used. Later on electronic triggers for traps were used, e.g. when lifting the artifact from its designated place, a trap is triggered and the area is cordoned off.

There are many other modes of security that have been introduced due to high technology that is available. All of these measures have one thing in common; they can only be utilized to their full potential only when isolated from human contact.

The measures include LASERs moving randomly in the area, noise detector, password protected areas, temperature detector, weight detector, and biometric password protected areas to name a few.

All these measures could be implemented to some degree for publically accessed areas, but could not be used to their full potential due to human interference.

This project proposes another non-touch security system. The system produces a low level electric field and checks for disturbance in the same.

The premise is that, to go unnoticed by any of the systems mentioned previously, another device has to be introduced to the system to cause interference or malfunctioning of the system. Doing so for this system would cause problems as by virtue of introduction of a foreign material in the electric field, the field density changes. The change varies according to the type of substance used (conductor, insulator, semi-conductor, dielectric, ground, etc.)

Secondly, by setting the alert level or sensitivity of detection of changes in electric field, the system can be theoretically used in areas with high human interference.

To further improve the system, the main detection scheme will be supplemented by security measures which can help remove any blind spots, if any. The supplements used for this system are some of the existing systems which have been implemented prior to this attempt, either independently or in conjunction to others, for instance: temperature sensors and ultrasonic sensors.

CHAPTER 2

Need for the System

As stated earlier every system is fallible. When a study is done on how these systems have been fooled, a lot of information is available. Every system used for security uses some basic principle as its lynchpin. This system, for instance, uses the concept of electric fields. The basic concept is that any interference introduced in the field lines affects the field lines in one way or another. The effect may be increased or decreased electric flux, thereby changing the electric field, which is the basis for detection for this system. Similarly, taking the example of a LASER system, the system depends upon the property of reflection of light. The system, however, does not take into account where the reflected rays are coming from; instead, the system detects intrusion by virtue of obstacle in the path of LASERs. Therefore, theoretically, it is possible to reflect the light in a manner that it only goes from its source to its destination, thereby, bypassing the system.

Similarly, the other systems can be bypassed if the property that system is monitoring can be found out. Temperature sensors can be fooled by introducing liquid nitrogen which is pressurized when stored. When the nitrogen comes out from the nozzle it changes into gaseous form thereby reducing the temperature of the room and hence fooling the sensors.

Therefore a system was required such that any tampering to it would lead to detection. This system is particularly suited for this purpose. As the IC used for the production and detection of the low level electric field is relatively new, not much is known about it. There is also the fact that manipulation of electric field will be very difficult, this is because of the thresholds set up are such that the margin for error is very low, also due to its adaptive property these margins are changed by randomizing the thresholds on the grid. Thereby this system can theoretically be not bypassed easily. To remove any blind spots in the system, the temperature sensors are used.

CHAPTER 3

Conception of the Idea

We came upon this concept after reading about the many legendary robberies that have taken upon the previously proclaimed invincible security. The idea came to us about designing a better security system which is harder to break. This desire was also helped by watching robbery movies and the way the security was bypassed. It came to us that every security as previously mentioned works on a basic principle, which if known, can be broken by proper methods.

Our first idea was to use biometric sensors to make access limited to people on the database of biometric platform. This system, as we found out is very shaky, as once the outer layers are breached, i.e. the biometric platform is breached; then rest of the security is overridden. Besides, this type of security is only possible for areas with limited access to public, i.e. low number of people in biometric database.

While watching an episode of Prison Break which is a TV program series centered on breaching areas with high security, we came upon idea of a non-contact based security system which identifies intrusion. While searching the internet for any non-contact based sensor, we chanced upon a new IC made in the year 2005 [MC33794]. This IC was being used in some automated cars, where it detected when the driver was about to sit on the seat and then activating some features provided by the car manufacturer.

By reading the specifications of the IC, we found an alternate way to use the IC in our design for non-contact based security system. This IC has also been used in a project named **Kasubana**[1] which was directing an artificial flower to bloom when any intrusion in the field was identified.

CHAPTER 4

Components Used

1. **MC33794**

   Manufacturer: Freescale Semiconductors.

   Package Type: SOIC (small-outline IC).

   Brief Description: This IC generates a low level electric field and detects change in the said field.

2. **DS1621**

   Manufacturer: Dallas Semiconductors.

   Package Type: 8 pin, PDIP (Plastic Dual-Inline Package).

   Brief Description: This IC provides user with digital temperature levels.

3. **ATMEGA 32**

   Manufacturer: Atmel

   Package Type: 40 pin, PDIP (Plastic Dual-Inline Package).

   Brief Description: This IC acts as a brain for the system. It is a 32-bit Microprocessor.

4. **MAX232**

   Manufacturer: Phoenix Technology CO. LTD.

   Package Type: 16 pin, PDIP (Plastic Dual-Inline Package).

   Brief Description: This IC is used for serial communication between the computer and the system.

5. **ULN2003**

   Manufacturer:

   Package Type: 16 pin, PDIP (Plastic Dual-Inline Package)

   Brief Description: This IC is used to drive the stepper motor interfaced with the microcontroller.

6. **7805 and 7812**

   Manufacturer:

   Package Type: 3 pin, PDIP (Plastic Dual-Inline Package)

   Brief Description: This IC is used for voltage regulation for 5V and 12V respectively.

7. **Stepper Motor**

   Brief Description: The motor is interfaced with a microcontroller. Step Angle is 7.5°. A camera is mounted on the stepper motor.

8. **Resistors and Capacitors.**

9. **Diode IN4007**

10. **BC548**

    Brief Description: N-P-N Transistor used for switching purposes.

11. **Crystal Oscillator**

    Brief Description: 12 MHz oscillator used as a clock for microcontroller.

12. **Reset Switch**

13. **Coaxial Cables & Crocodile Clips**

14. **Aluminum Plates**

    Dimensions: 10x5cm^2

15. **Male/Female DB9 Connectors**

    Brief Description: Used for serial communication between computer and microcontroller.

16. **Web-Camera**

    Manufacturer: Creative
    Model: PD1100

17. **LEDs**

    Color: Red

18. **Molex Connectors (Male/Female Connectors)**

CHAPTER 5

Security System

5.1 Phase I: Formulation of Algorithm

Similar to the project, the algorithm was formulated in phases. As such, there are 4 algorithms in total for this system.

5.1.1 Algorithm for MC33794:

MC33794 is a newly developed IC, as such its uses and programming is not well charted. The IC in itself is a microprocessor and processes many instructions, it can also be programmed. The programming part had already been dealt with the help of ATMEGA32 processor, therefore a flowchart was required to be formulated for the proper working of the system.

The flowchart shown below depicts the basic functioning of the sensors present in MC33794 and the process of detection of intrusion in the electric field produced.

First the pins are calibrated, and then the shield for MC33794 is set. After this the references by which the detection is measured is set. The detection is done by comparing the values received at the electrodes to the permissible values which are set by fixing the references.

If the values do not match the permissible levels, then the alarm is raised.

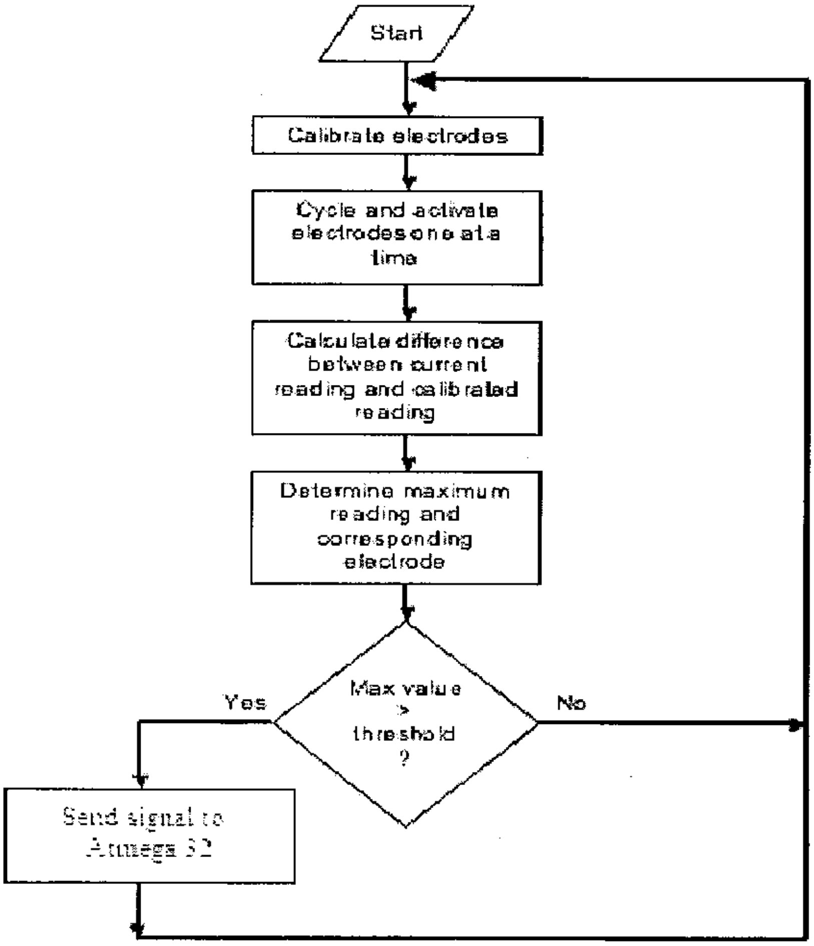

Start
Calibrate electrodes
Cycle and activate electrodes one at a time
Calculate difference between current reading and calibrated reading
Determine maximum reading and corresponding electrode
Max value > threshold ?
Yes
No
Send signal to Atmega 32

Fig5.1: Flow Chart for the working of MC33794

5.1.2 Algorithm for ATMEGA32:

ATMEGA32 is the main processor IC for this circuit, therefore its programming was a vital part of the system. An algorithm was developed to account for all the interfaced devices as well as the internal programming.

The flowchart provided in the figure no 5.2 depicts the basic working of the microcontroller ATMEGA32. Controller takes input from the three devices MC33794 IC which is used as the electric field sensing device, this device is connected to the sensor plates and sense any change in the electric field if there is any disturbance in the field.

Another input is from the GUI which also responds to the output which is provided by the controller. GUI provides many testing procedures such as Serial Test in which it is checked if the connection is established or not. In Camera test the camera is tested whether it is working or not and is responding correctly to the left and right command when send by the GUI. Change Alert Level shows different alert levels which can be selected and corresponding array of permissible limit values is selected. In grid testing the controller selects each grid sequentially and checks the value at it and tells whether the grid is present or not to the GUI.Third input is taken from the Temperature sensing IC (DS1621). In this the digital value which is provided to the controller is processed and if the value exceeds the normal temperature than an alarming signal is send to the GUI which in turn raises the alarm as it does in the case of the sensors.

In the figure 5.3 the flow chart is of the Serial testing .Serial testing is done by receiving the character **Q** from the GUI after which any character which is sent is returned back and the GUI performs the corresponding function.

In the flowchart of camera test (figure no 5.4) a particular value is serially sent to the controller which performs the corresponding function. In this case if the value **R** is received than the controller drives the motor in **right** direction and if **L** is sent than the motor moves **left**. As the camera is mounted on the motor so our camera moves in the same direction as of the motor.

In the flow chart of changing the Alert level(Figure 5.5) the alert levels are changed corresponding to the serially received values. If M is received than another character is asked for as per the received character 1, 2, 3 corresponding changes are made in the controller. For each value there is a corresponding array of values which define the limit of the sensors and thus defining the alert type of the system.

The Figure 5.6 is the flow chart of Grid Testing. In the grid testing procedure the character S is sent by the GUI after which the controller sends the value to select the different sensor plates and then the electric field value corresponding to this plate is sent back by the MC33794. This is received by the inbuilt ADC of the ATMEGA32 which converts this analog value to the digital value. Then this value is returned to the GUI which is then printed.

The Figure 5.7 is of Flowchart for Intrusion Detection. The values of the sensor are read and saved in the same way as explained in the grid testing procedure. In this if there is any change in the values of the plate which is beyond the permissible limit of the sensors than the alarming signal is sent to the GUI and the camera is positioned to the same plate so as to focus the reason of the intrusion.

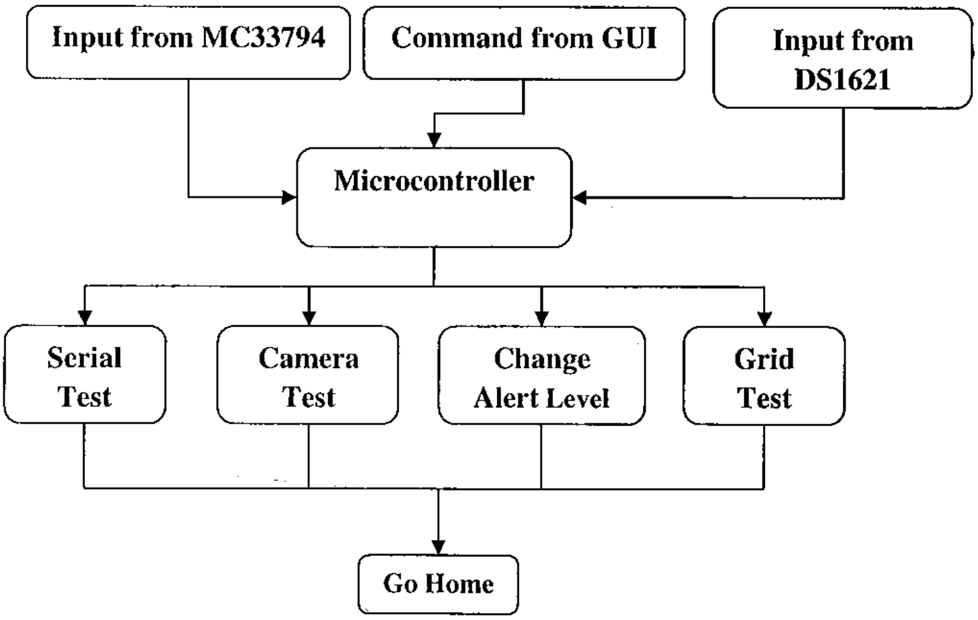


Fig5.2: Basic Working of Microcontroller.

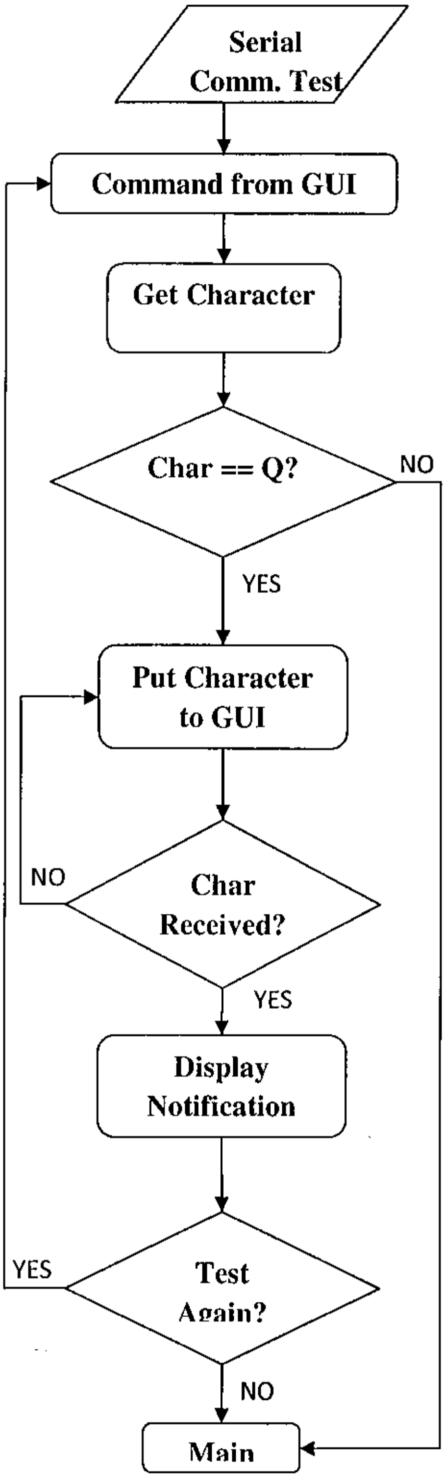


Fig5.3: Test Page Flow Diagram.

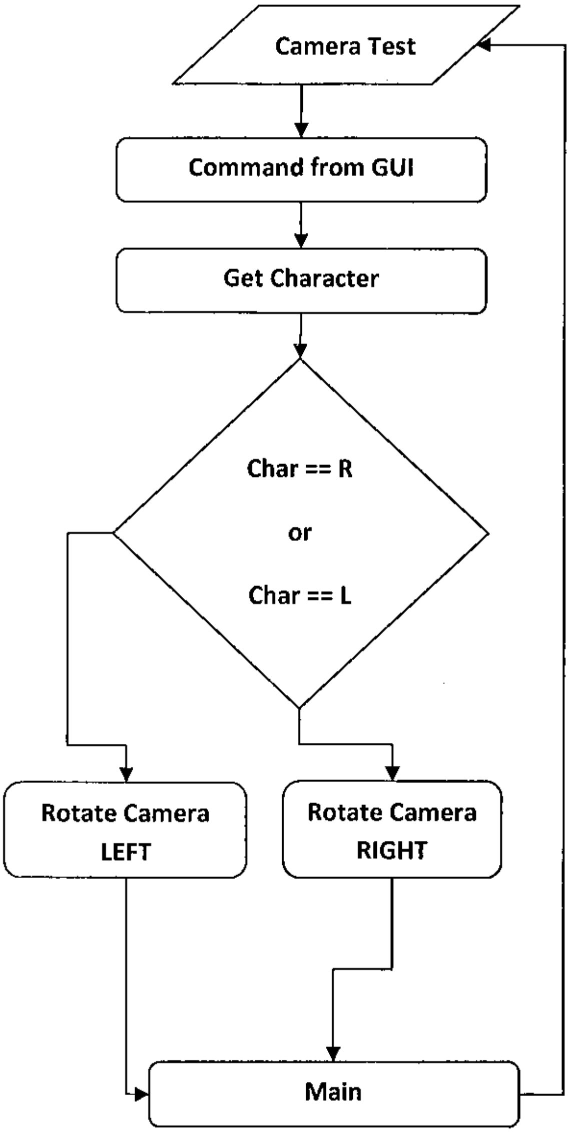


Fig5.4: Camera Testing.

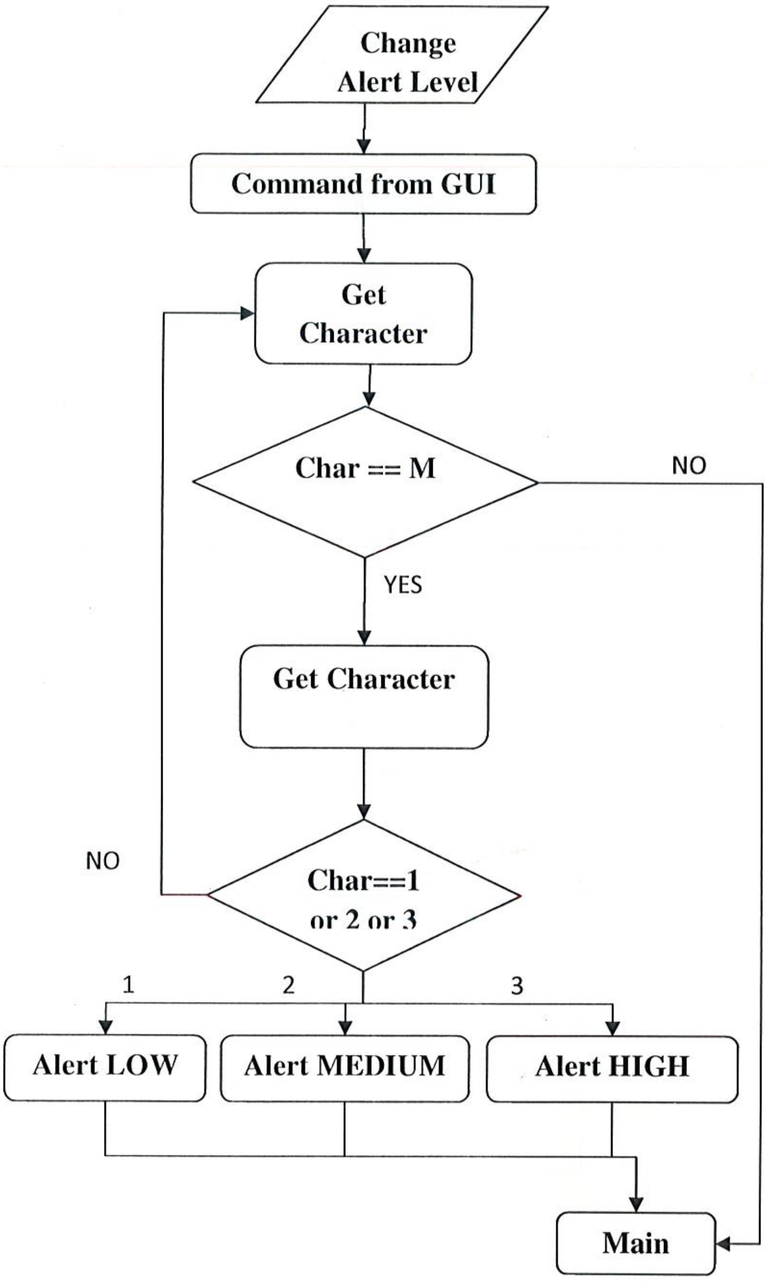


Fig5.5: Alert Level Change

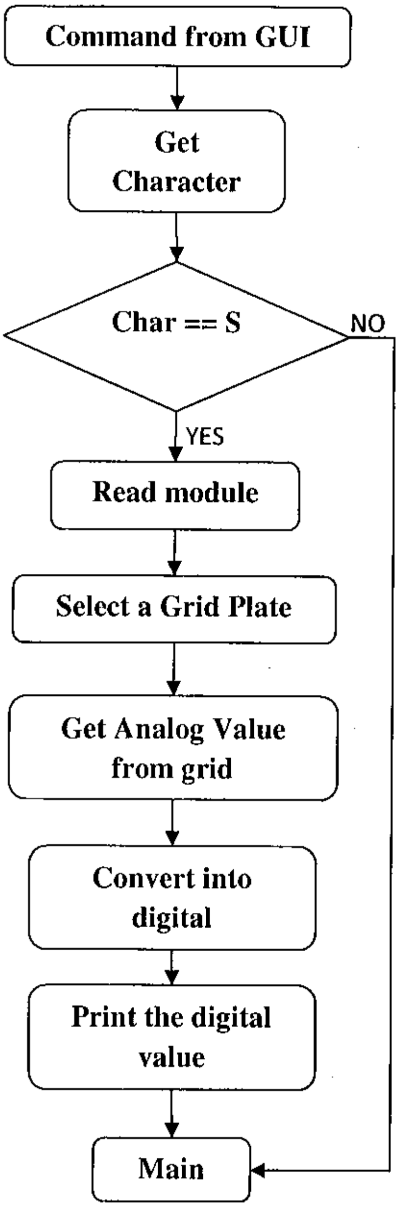


Fig5.6: GRID TESTING

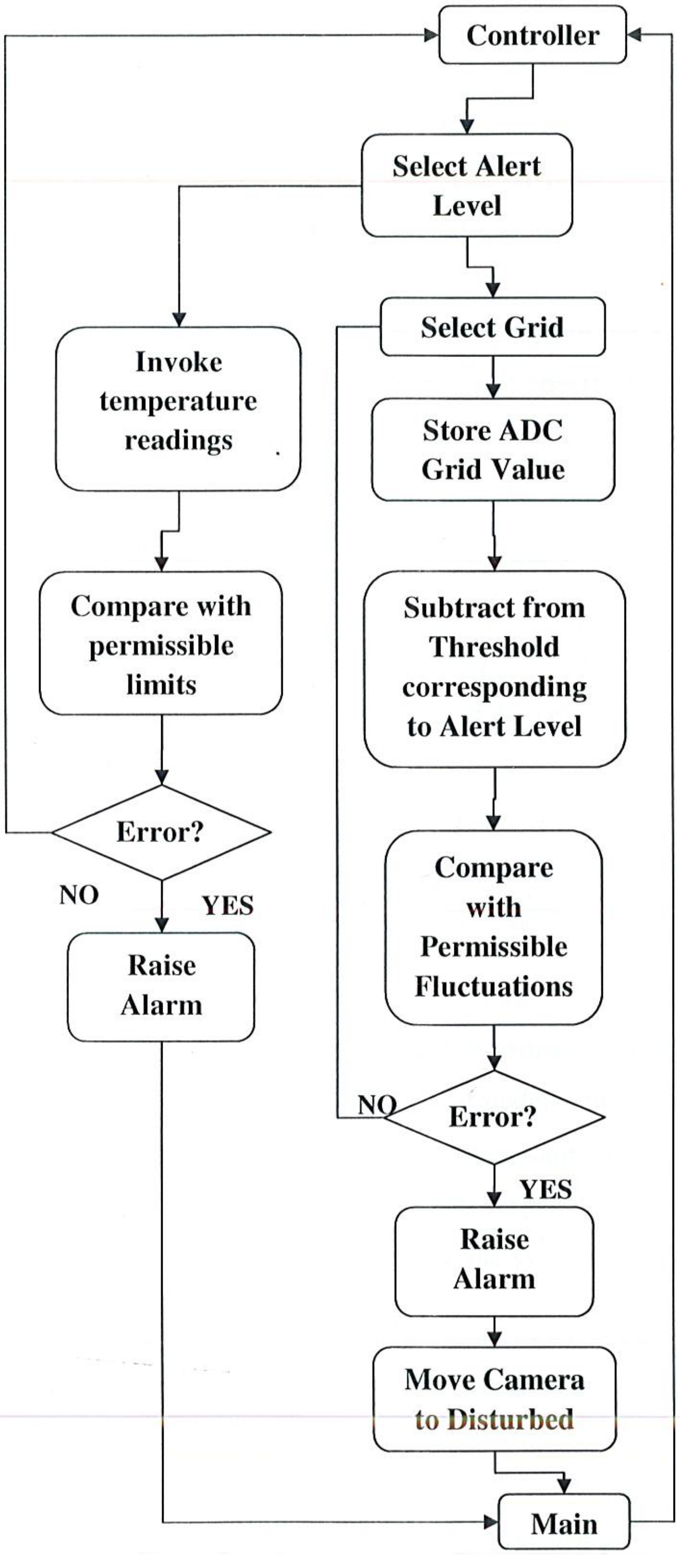

Controller
Select Alert Level
Select Grid
Invoke temperature readings
Store ADC Grid Value
Compare with permissible limits
Subtract from Threshold corresponding to Alert Level
Error?
NO
YES
Raise Alarm
Compare with Permissible Fluctuations
NO
Error?
YES
Raise Alarm
Move Camera to Disturbed
Main

Fig5.7: Flowchart for Intrusion Detection

5.1.3 Algorithm for Visual Basic:

Using Visual Basic a GUI (Graphic User Interface) was developed, once the product is finished this GUI will be the only part of the system on which there is any interaction with the security management. The GUI has several functions, such as, setting of alert level either manually or automated, camera test, grid test to name a few.

Figure 5.8 is a flowchart which explains the working of the VB GUI, it consists of one main page, where user can access menu and observe results. By accessing menu, user can test the device, change alert level & control type or view the camera.

By observing results, user can determine the intrusion level, grid report, temperature level and alert level.

Figure 5.9 is a flowchart which explains the working of the test page. Here, first the user selects the test type out of the provided options then once user starts testing, connection is established between the device and computer, if failed, a notification is sent to the user else, the device computes results in accordance to the test type and displays results. If the user has completed the testing, he can leave the test page by clicking "Go Home" or else he may test device for other tests.

Figure 5.10 is a flowchart which explains working of the main page, this is the most complicated part of the GUI, here a real-time connection is being established within the system which constantly monitors and updates the security status. Here, three timers work continuously, namely, alarm timer, value timer and control timer, which fire at an interval of 200ms, 2s and 500ms, respectively. Alarm timer checks the alarm status, if the device calls for an alarm, the system generates a beep which can be stopped by the user.

Control timer is activated if the control type is automatic. In this procedure we check the clock of the pc and update the profiles of the system accordingly, namely between medium, low and high alert profiles. Between, 8am – 1pm we use the medium profile, for 1pm - 2pm we use the low profile, 2-4 pm we use the medium profile and for 4pm – 8am, we use the high alert profile.

Value timer is used for requesting values from the device at a very short interval, so as to update the status of the main page. The fire interval is to be calibrated with the device performance so that a real-time connection between the two could be established.

Figure 5.11 and 5.12, explain how the user can change alert levels and control type respectively, for both the procedures, first, the current status is displayed then if the user necessitates to change the present configuration, he may select any of the available option and hit the "Update" button.

Figure 5.13 is a flowchart to explain the procedure of viewing camera, first, when a request is generated to view the camera by user, the GUI checks if the camera is present or not, if the camera is not present then it declines the request, else if present, then the GUI establishes link between the camera and itself. Now, the user is able to observe the camera, here he can also change the video format, by clicking the "Video Format" he may open a dialog box where he can pick pre-defined resolution and pixel depth and update the same later.

Figure 5.14 explains the flow of the control as the GUI proceeds, first, a splash screen is displayed providing user with details of project and team members, then an authentication check is placed so as to avoid use by unauthorized users. If the user passes through the authentication check, the GUI starts loading itself, where it also checks if the device present or not. If present, GUI proceeds further else it will display notification to re-connect device and try again. Once the GUI is loaded, a connection is established between the GUI and device, where both send and receive instructions from each other.

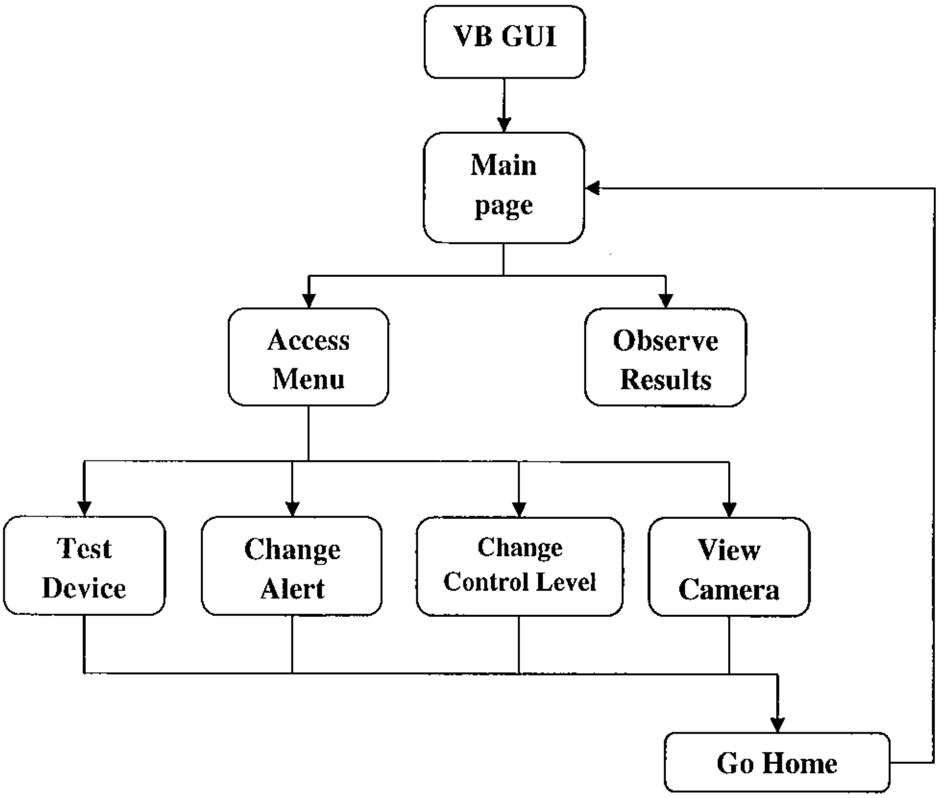


Fig5.8: Tree Diagram for GUI.

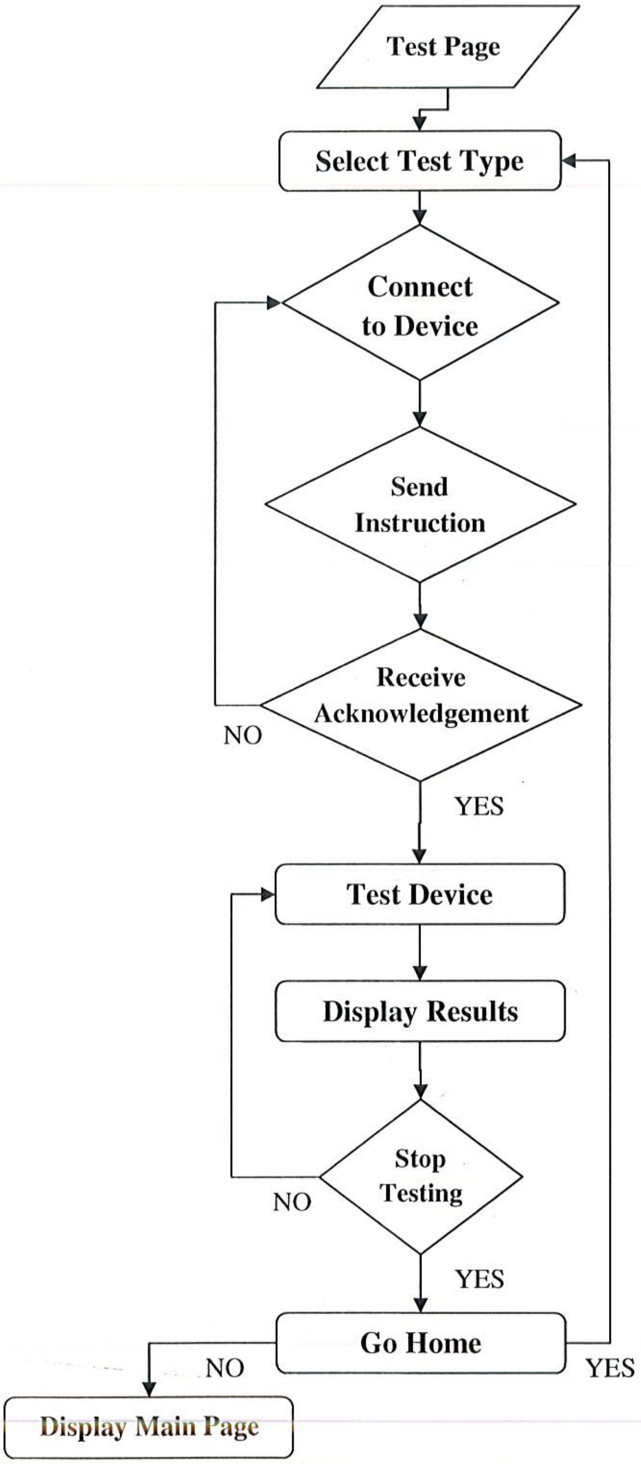


Fig5.9: Test Page Flow Diagram

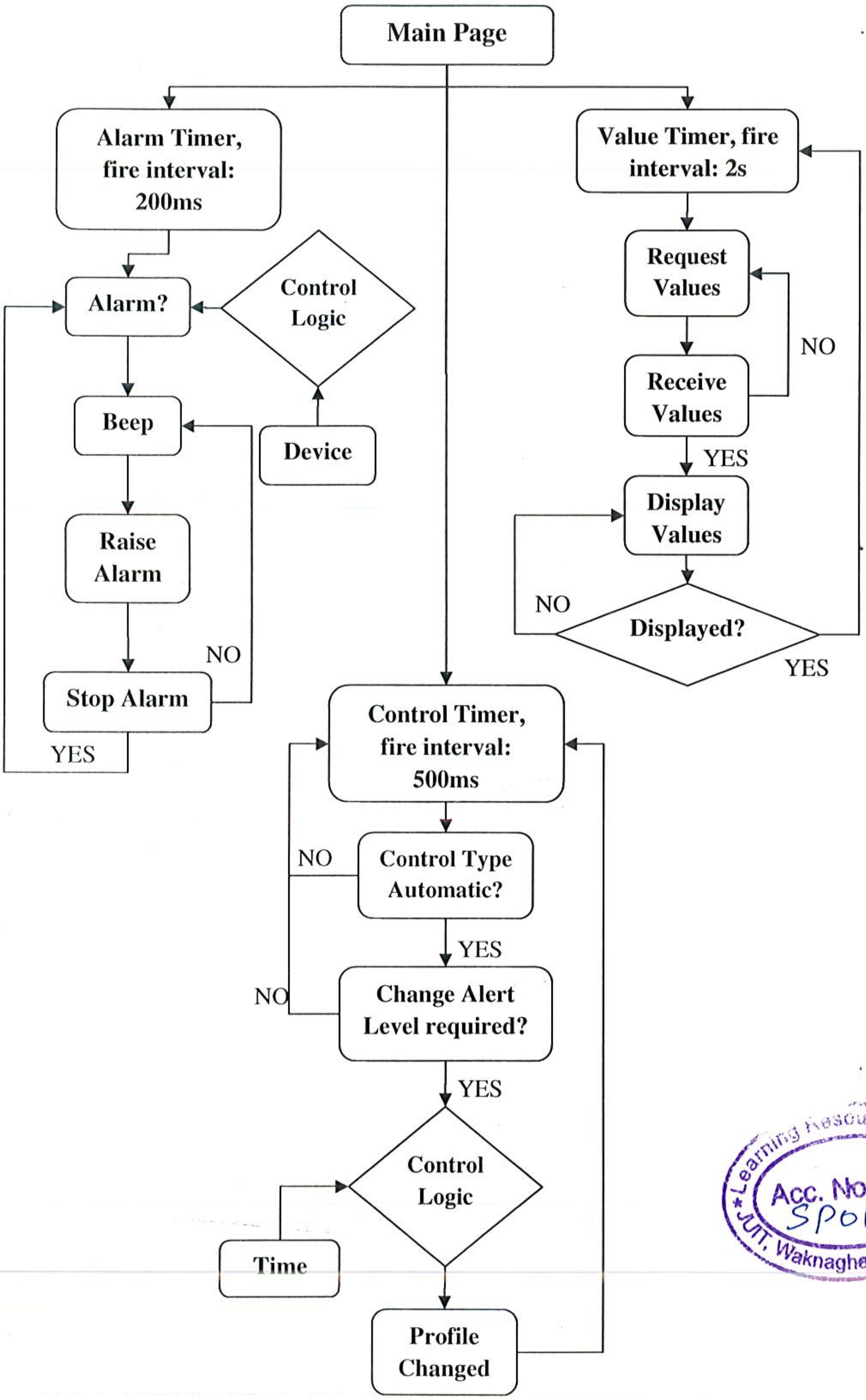


Fig5.10: Main Page Flow chart

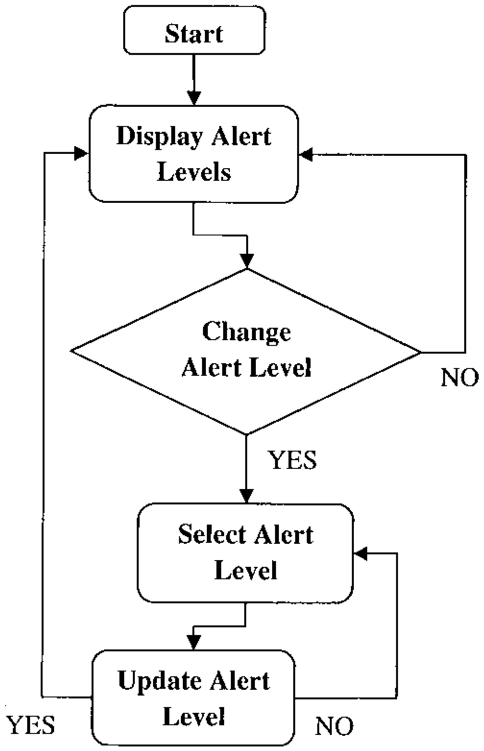


Fig5.11: Change Alert Level Flowchart.

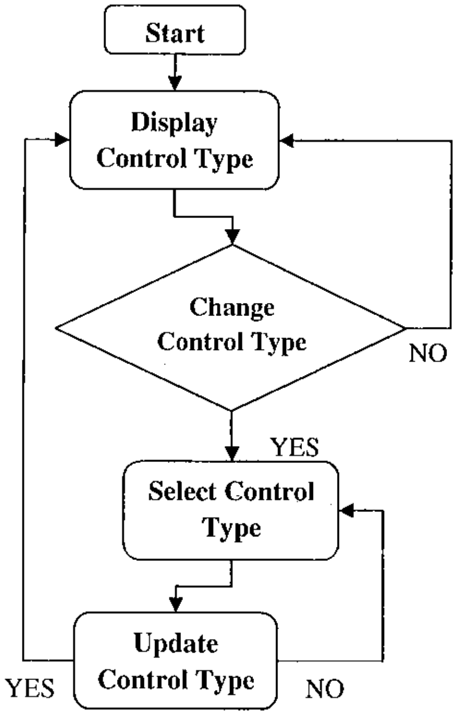


Fig5.12: Change Control type Flowchart

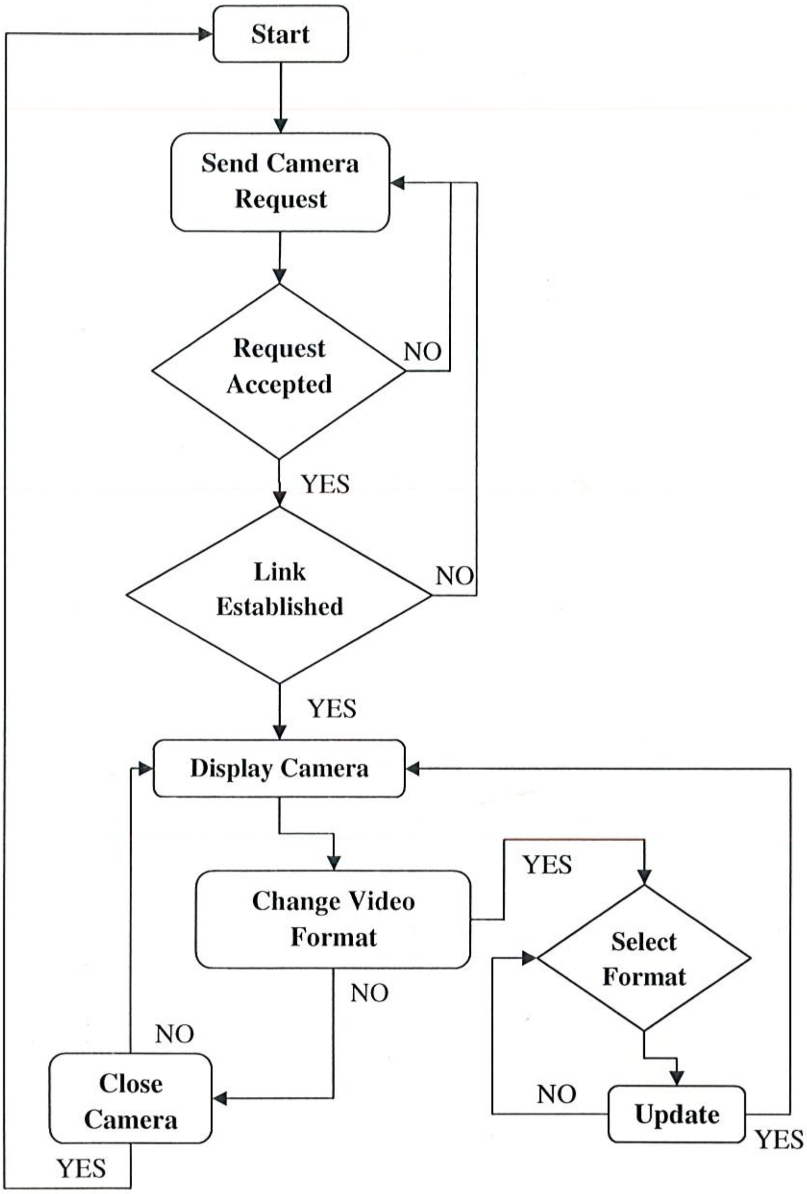


Fig5.13: Display Camera Flow chart.

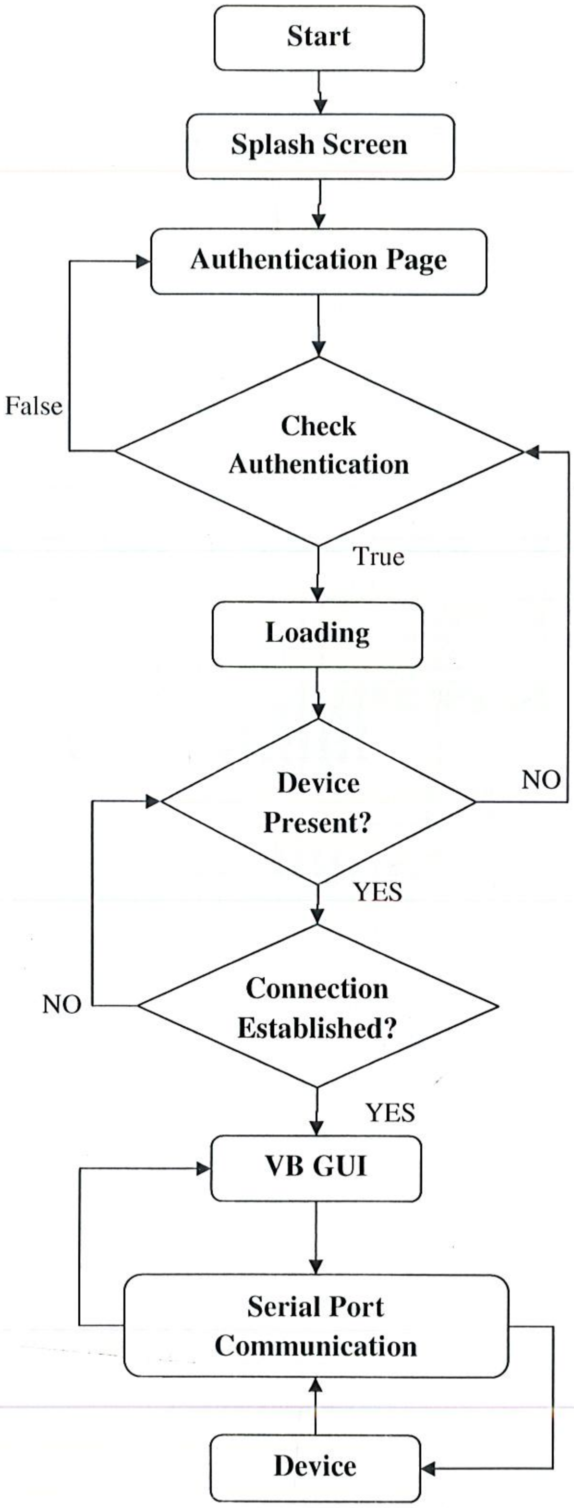


Fig5.14. Executed VB Flow chart.

5.2 Phase II: Finalization & Analysis of Circuitry

The circuit schematic is as follows:

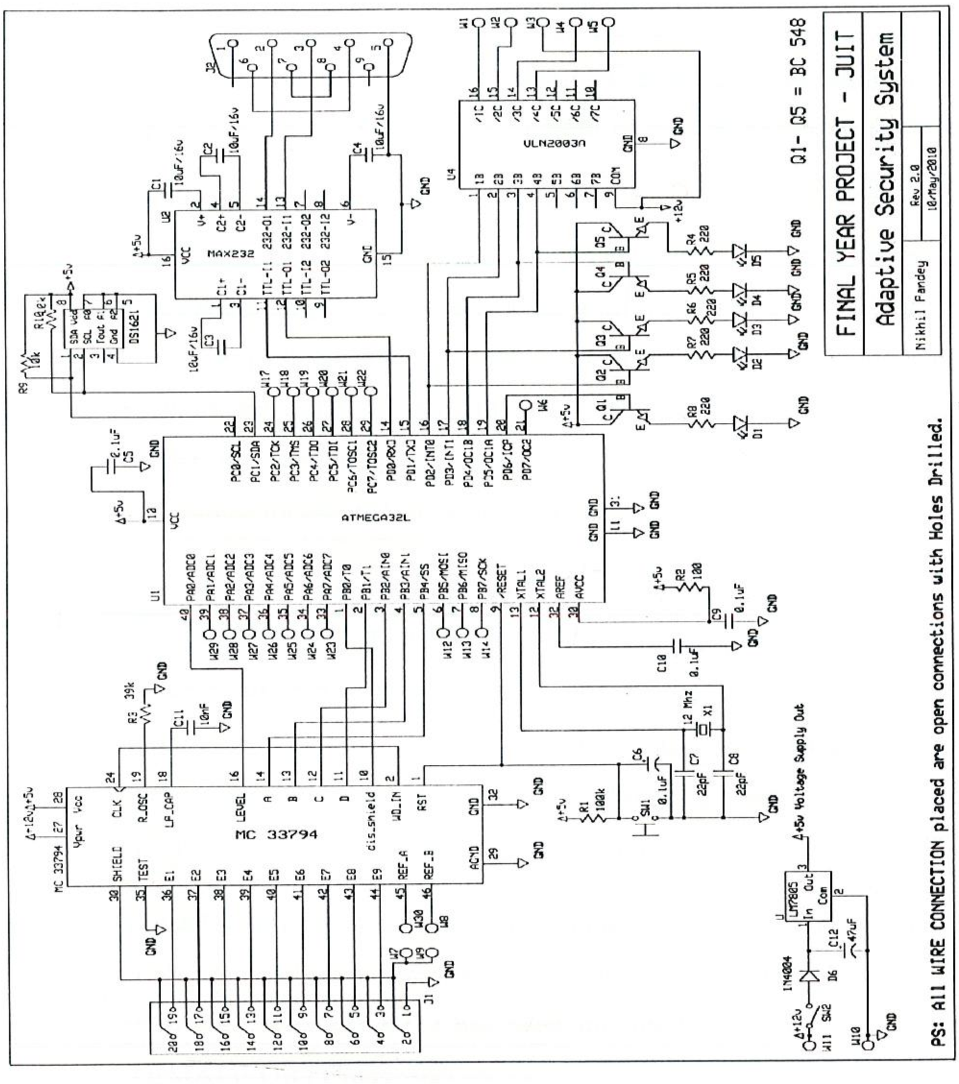


Fig5.15: Finalized Schematic

As can be seen in the circuit design, there are 4 processors. The first processor that is used is the MC33794. The MC33794 has 9 electrode pins, these pins generate electric field. In this circuit these pins are connected to metal plates, hence the surface area covered by the electric field is increased. A shield is also connected to the electrodes. Then the MC33794 is interfaced with ATMEGA32. The reference pins have to be set as shown in the diagram, and the input driving voltages for MC33794 are 12V and 9V.

The ATMEGA 32 processor is the brain of the system; it is interfaced with all the other devices, where it either drives them into giving some output or takes their output values as input. One port of the processor is provided for external connections (Port A).

Port A is internally multiplexed with Analog to Digital convertor which converts the analog signal which is sent to the Microcontroller by the MC33794 using its Signal pin. This port is entirely responsible for the data which is being received by the microcontroller from the MC33794.

Port B acts as the output port and is interfaced with MC33794 and the values which are sent to the port help in selecting the sensor whose value is to be read. After the selection of the sensor the value is read by the MC33794 and the analog value is sent to the microcontroller as previously explained. The crystal oscillator used for the microcontroller to adjust its internal clock, has a frequency of 12MHz.

Port D is used to drive MAX232 and ULN2003. The pins driving ULN2003 are also connected to the BJTs BC548, which are N-P-N Transistors. The output of these transistors has been connected to LEDs. This is done so that it can be physically seen that the ULN2003 is being driven. The input driving voltages for ATMEGA32 are 9V and 5V. ULN2003 is used to drive Stepper Motor. The driving voltage for ULN2003 is 12V. A camera is mounted on the Stepper Motor driven by ULN2003.

MAX232 is used in this circuit for serial communication. It plays a vital role for the GUI developed, as it is used to interface the system with the computer. MAX232 has a driving voltage of 9V. The MAX232 has been provided with output pins through which MALE/FEMALE DB9 Connectors can be plugged in.

A separate circuitry is added to supply a constant voltage of 5V and 12V by using IC 7805 and 7812 respectively. The regulated voltages (5V, 12V) are used to drive the other ICs on the circuit board.

5.3 Phase III: Testing of MC33794

Test circuit which is used to determine reference values for thresholds:

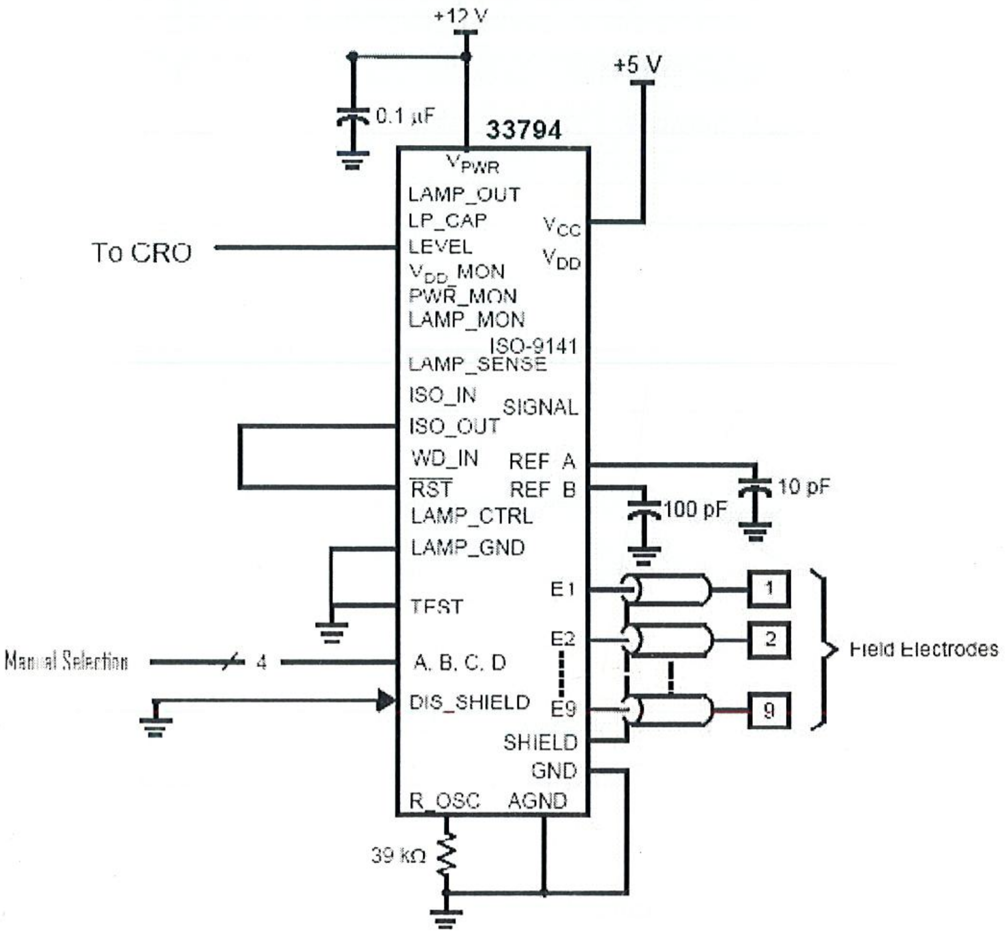


Fig5.16: Test Circuit for MC33794

This test circuit was used and following results were tabulated using different classes of materials for interference in electric field:

Interference: Human

Serial no.	Maximum (V)	Distance from plate
1	2.02	Without touching
2	2.00	3cm from plate
3	1.87	1cm from plate
4	1.69	1point of contact
5	1.12	Full contact

Fig 5.17

As can be seen from the results displayed above, there are different variations in the electric field with change in altitude from the plates. To widen the scope of the test, different types of material are used to cover the plate.

From the values obtained, thresholds are determined, which in turn give us the permitted safeguard band for electric field.

5.4 Phase IV: Interfacing of Stepper Motor

A **stepper motor** (or **step motor**) is a brushless, synchronous electric motor that can divide a full rotation into a large number of steps. The motor's position can be controlled precisely without any feedback mechanism (see Open-loop controller), as long as the motor is carefully sized to the application. Stepper motors are similar to switched reluctance motors (which are very large stepping motors with a reduced pole count, and generally are closed-loop commutated.).

Stepper motors operate differently from DC brush motors, which rotate when voltage is applied to their terminals. Stepper motors, on the other hand, effectively have multiple "toothed" electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external control circuit, such as a microcontroller. To make the motor shaft turn, first one electromagnet is given power, which makes the gear's teeth magnetically attracted to the electromagnet's teeth. When the gear's teeth are thus aligned to the first electromagnet, they are slightly offset from the next electromagnet. So when the next electromagnet is turned on and the first is turned off; the gear rotates slightly to align with the next one, and from there the process is repeated. Each of those slight rotations is called a "step," with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle. There are two basic winding arrangements for the electromagnetic coils in a two phase stepper motor: bipolar and unipolar. This project uses UNIPOLAR PHASE Stepper motor.

A unipolar stepper motor has two windings per phase, one for each direction of magnetic field. Since in this arrangement a magnetic pole can be reversed without switching the direction of current, the commutation circuit can be made very simple (e.g. a single transistor) for each winding. Typically, given a phase, one end of each winding is made common: giving three leads per phase and six leads for a typical two phase motor. Often, these two phase commons are internally joined, so the motor has only five leads. A microcontroller or stepper motor controller can be used to activate the drive transistors in the right order, and this ease of operation makes unipolar motors popular

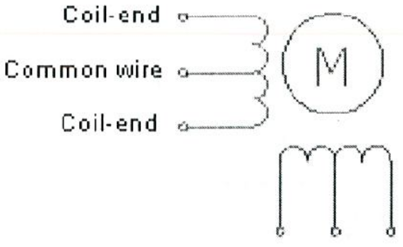


Fig5.18: Unipolar Stepper motor coils

Stepper motor performance is strongly dependent on the drive circuit. Torque curves may be extended to greater speeds if the stator poles can be reversed more quickly, the limiting factor being the winding inductance. To overcome the inductance and switch the windings quickly, one must increase the drive voltage. This leads further to the necessity of limiting the current that these high voltages may otherwise induce.

So as to provide the drive circuit the ULN2003 Stepper motor driving IC is used. The ULN2003 contains seven Darlington transistors. The ULN2003 can pass up to 500mA per channel and has an internal voltage drop of about 1V when on. It also contains internal clamp diodes to dissipate voltage spikes when driving inductive loads. The circuit for driving stepper motor using ULN2003 is shown below.

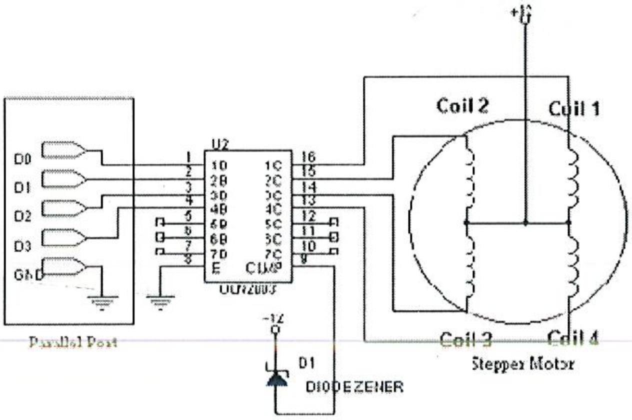


Fig5.19: Connection of Stepper Motor with ULN2003

The Stepper Motor is used in this system as a mount for a camera which is used to keep a visual on the area being protected. The Stepper Motor used in the system has a step angle of 7.5°. The Stepper Motor is driven by an IC ULN2003, which in turn takes its input from Port D of ATMEGA32.

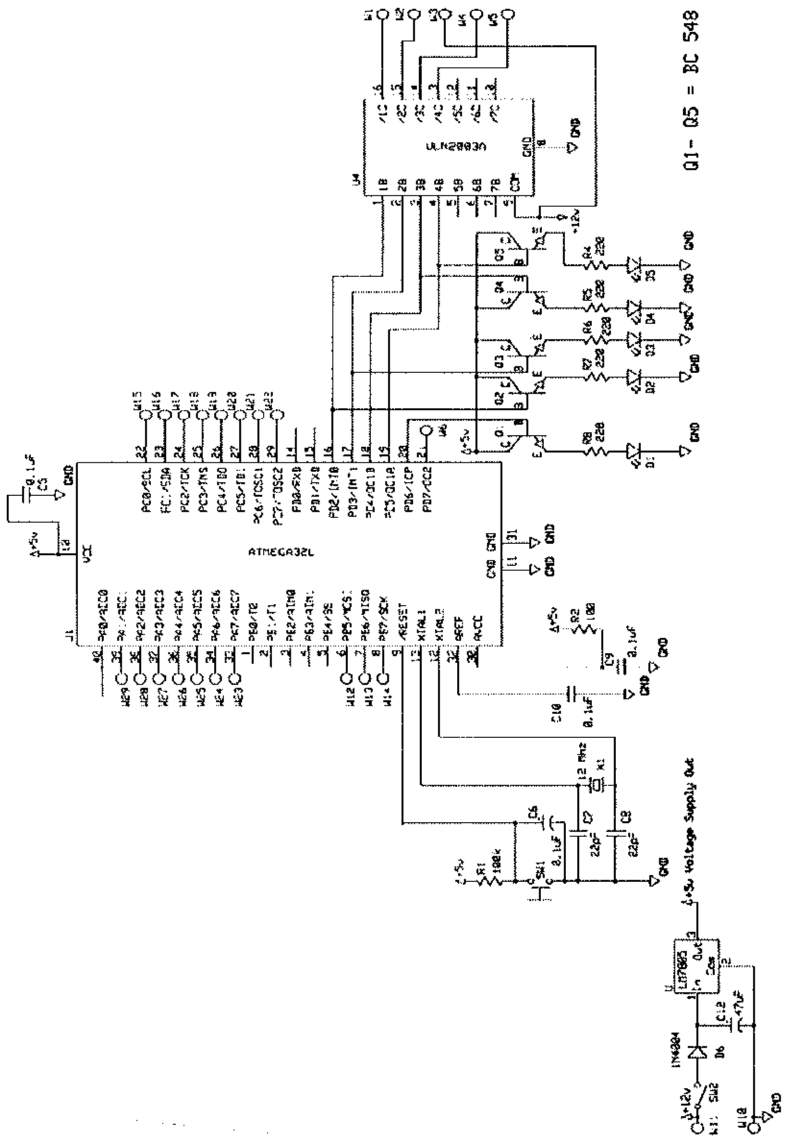


Fig5.20: Stepper Motor Interfacing

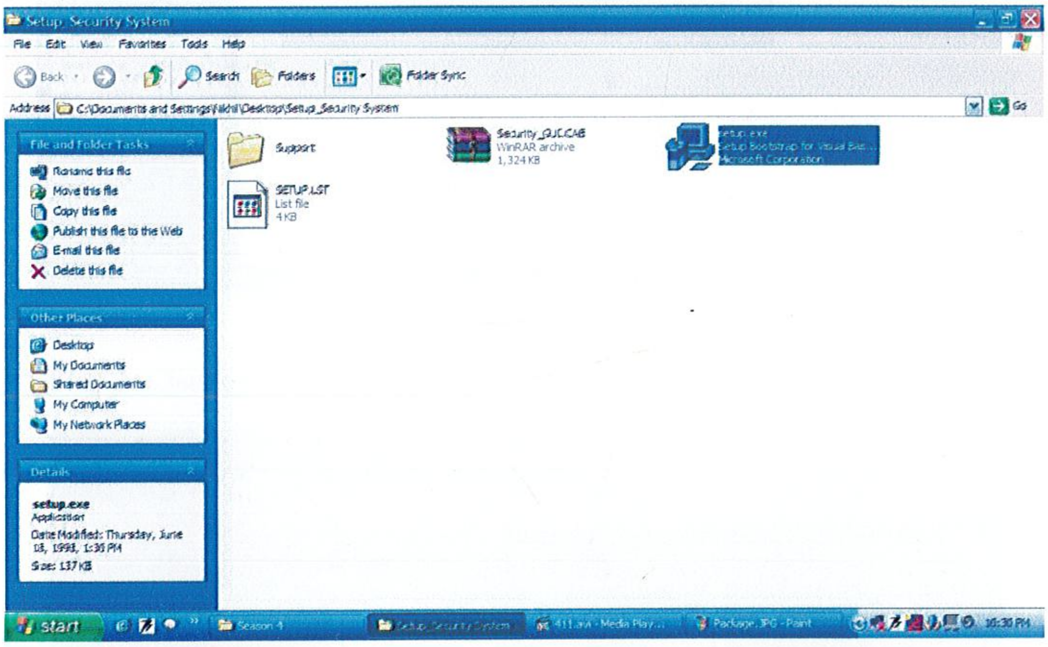
The Stepper motor is connected to the pins W1 to W5.

5.5 Phase V: Graphic User Interface (GUI) Development

This project provides a Package file which acts as a setup for the system. First, copy the folder "Package" found on the compact disk provided with the report.

Snapshot 5.1

On opening the Package folder, user can find the setup (Setup.exe) as shown below,



Address: C:\Documents and Settings\Nikhil\Desktop\Setup_Security System

Support

SETUP.LST
List file
4 KB

Security_GUI.CAB
WinRAR archive
1,324 KB

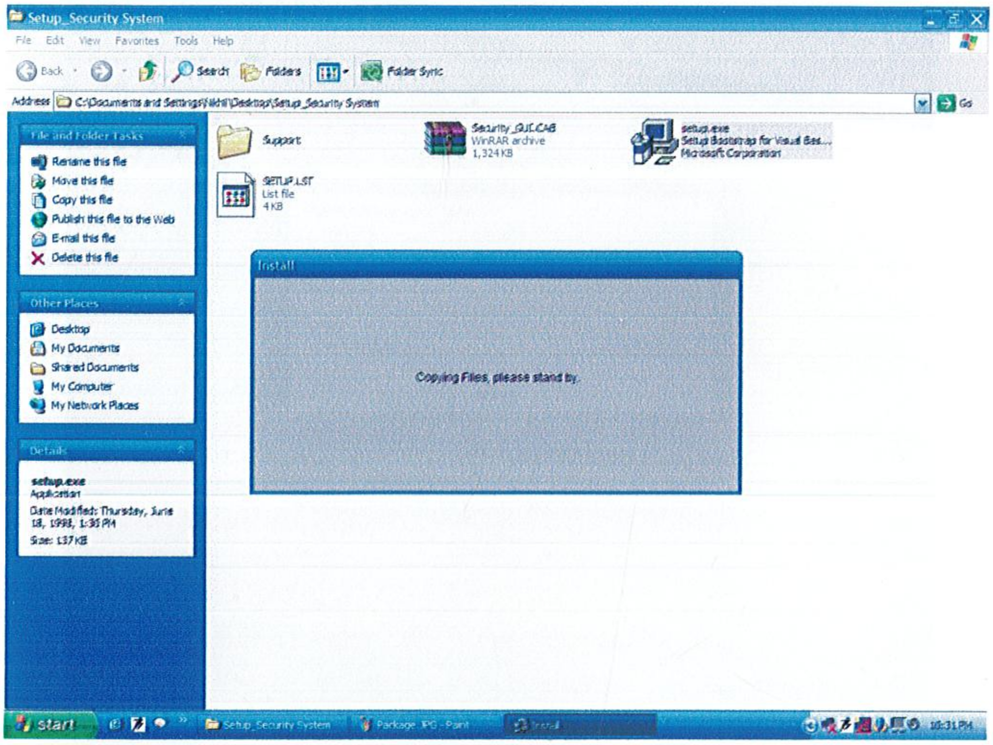
setup.exe
Setup Bootstrap for Visual Bas...
Microsoft Corporation

setup.exe
Application
Date Modified: Thursday, June 18, 1998, 1:30 PM
Size: 137 KB

Snapshot 5.2

When the in package is opened the setup.exe is opened. The following procedure is to be followed while installing the package provided in the hard disk with this project.

Step 1: Please stand-by for the processing of the files to be executed.

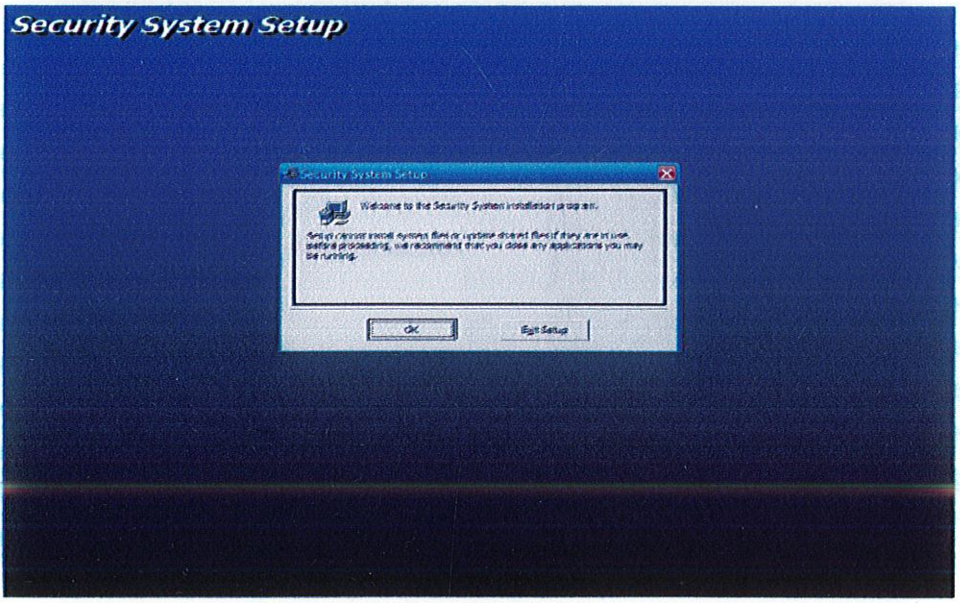


setup.exe
Application
Date Modified: Thursday, June 18, 1998, 1:35 PM
Size: 137 KB

Copying Files, please stand by.

Snapshot 5.3

Step 2: Click OK to continue.

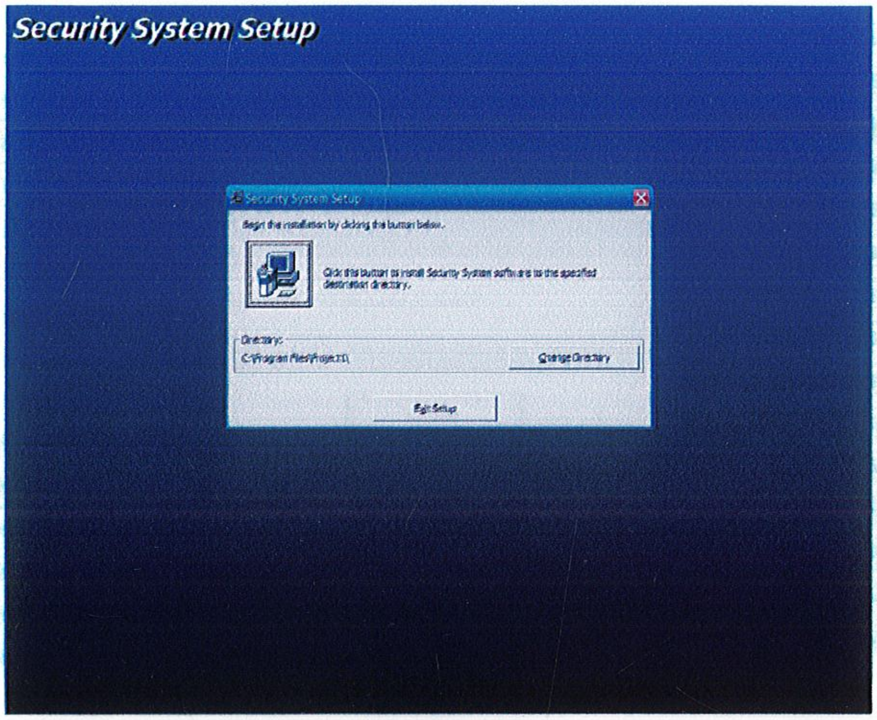


Welcome to the Security System installation program.

Setup cannot install system files or update shared files if they are in use. Before proceeding, we recommend that you close any applications you may be running.

Snapshot 5.4

Step 3: Click on the icon highlighted.



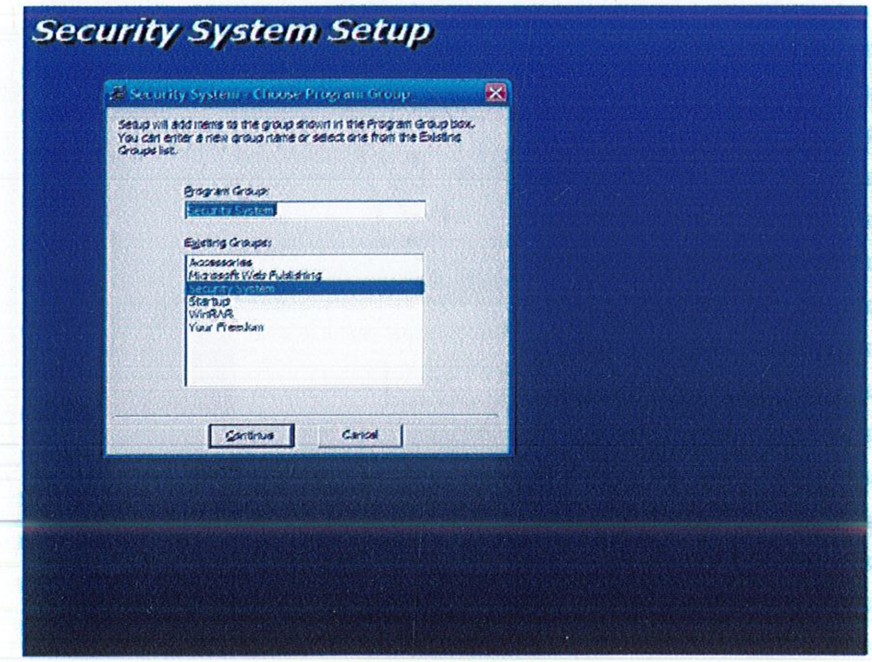
Begin the installation by clicking the button below.

Click this button to install Security System software to the specified destination directory.

Directory:
C:\Program Files\Project1\

Snapshot 5.5

Step 4: Click on the CONTINUE button to proceed further.



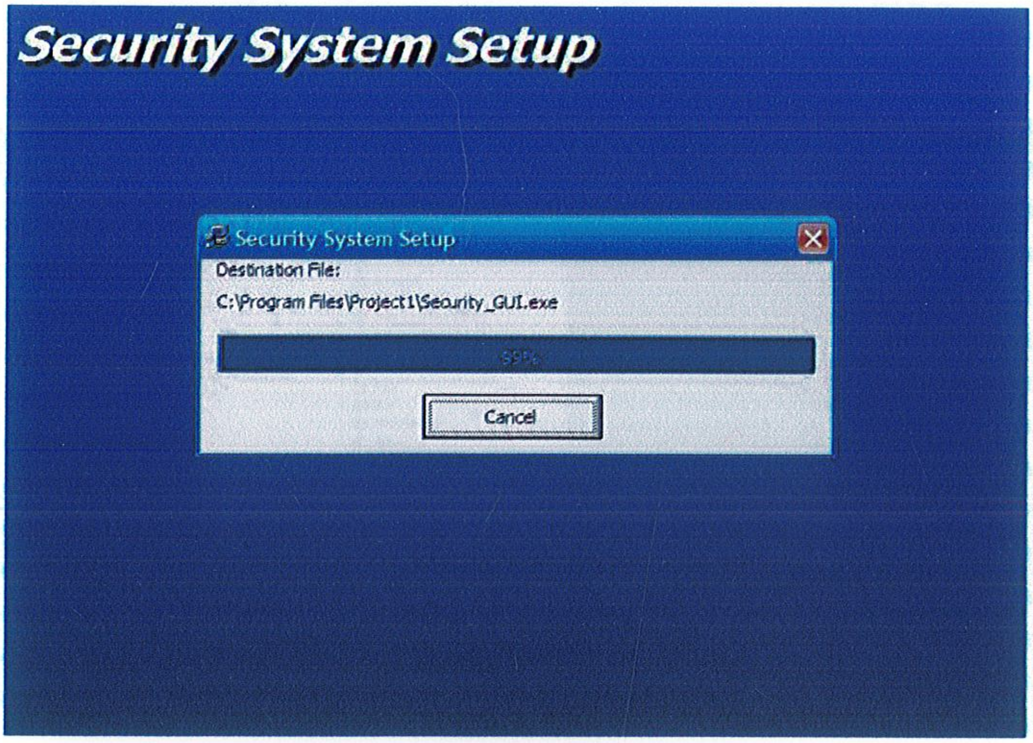
Setup will add items to the group shown in the Program Group box. You can enter a new group name or select one from the Existing Groups list.

Program Group:
Security System

Existing Groups:
Accessories
Microsoft Web Publishing
Security System
Startup
WinRAR
Your Freedom

Snapshot 5.6

Step 5: Please stand-by for the processing of the files to be executed.

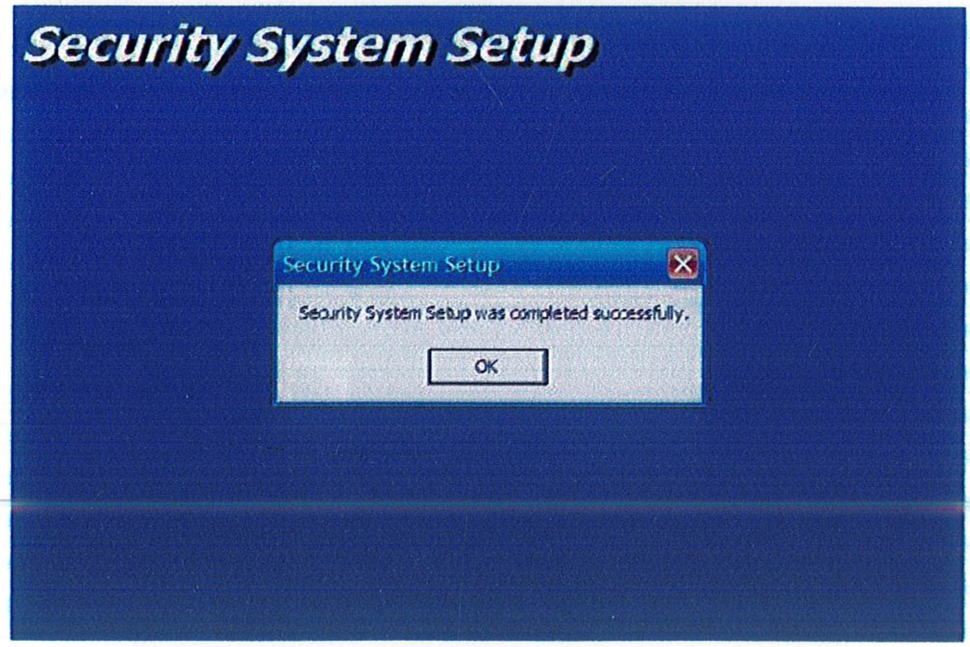


Destination File:

C:\Program Files\Project1\Security_GUI.exe

Snapshot 5.7

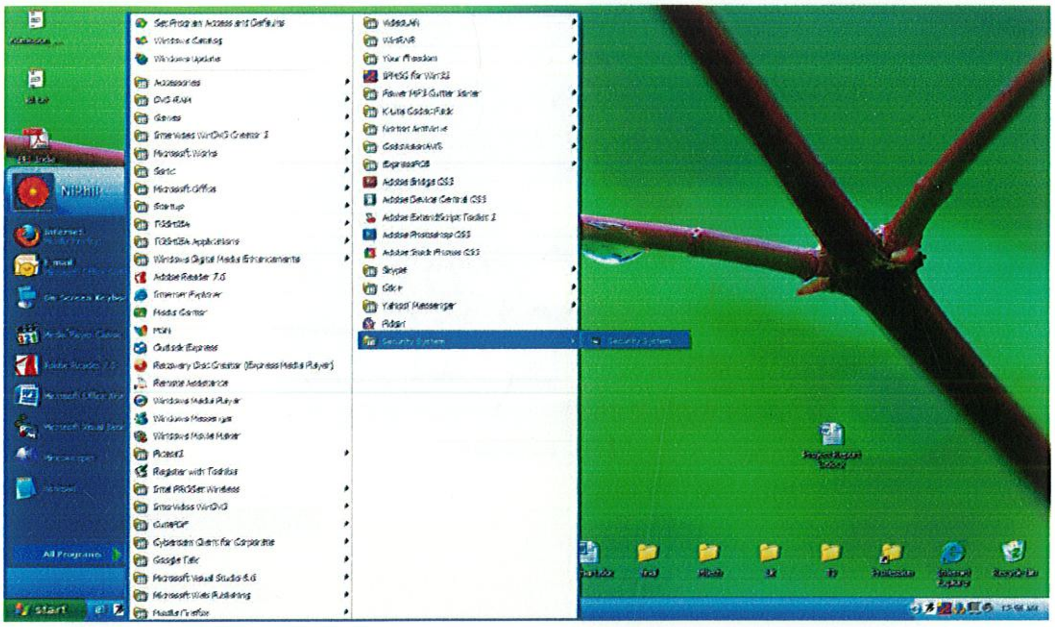
Step 6: Congratulations, setup is complete, please click OK to continue.



Security System Setup was completed successfully.

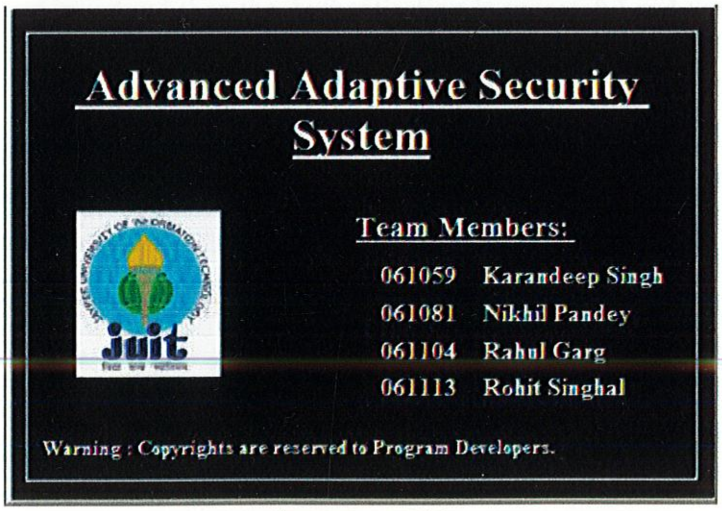
Snapshot 5.8

Please follow the following instructions to start working with the GUI. Click on the "Start" button provided on the lower left corner of your screen, go to "All Programs" and find "Security System" and then click on "Security System" application.



Snapshot 5.9

Following are the snapshots for the GUI to control and monitor the hardware; first, we start with a splash screen which provides the basic introduction to the user with the project title and team members.



Advanced Adaptive Security System

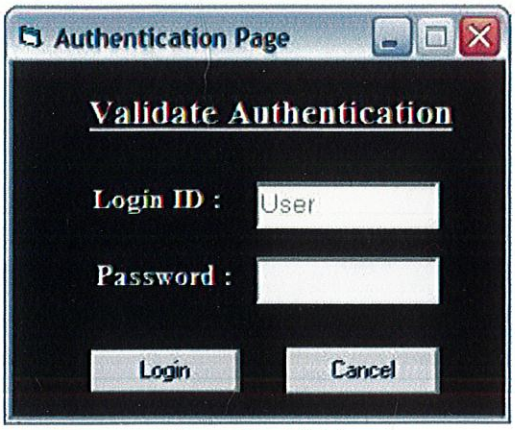
Team Members:

061059 Karandeep Singh
061081 Nikhil Pandey
061104 Rahul Garg
061113 Rohit Singhal

Warning : Copyrights are reserved to Program Developers.

Snapshot 5.10

This splash screen unloads automatically after 5 seconds or on the click by user within or on the splash screen. Once this happens, splash screen is replaced by the authentication page shown below,



Login ID : User

Password :

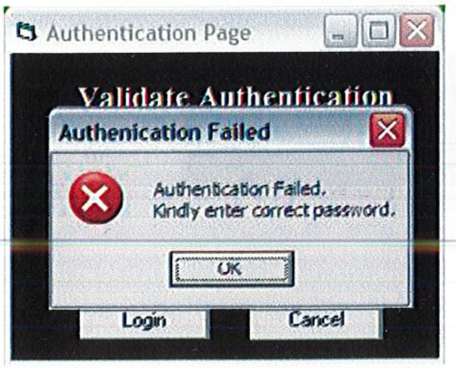
Snapshot 5.11

Valid Authorization:

User name: User

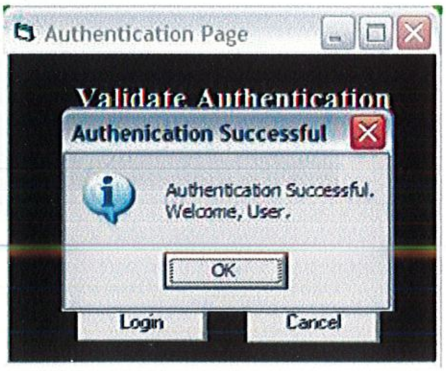
Password: User

The Authentication allows for only the authorized users to work on the GUI ahead. Authentication is checked by asking for a password from the user. If wrong Password is entered, then Authentication fails and the user is redirected to the login page, if the correct password is entered then the Authentication is successful and then connection is tried to establish between the hardware and GUI.



Authenication Failed

Authentication Failed.
Kindly enter correct password.



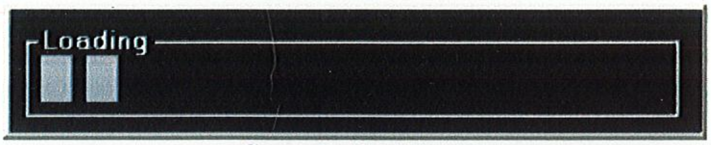
Authenication Successful

Authentication Successful.
Welcome, User.

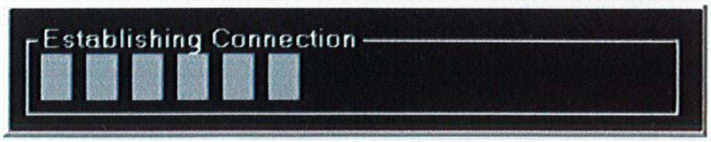
Snapshot 5.12

Snapshot 5.13

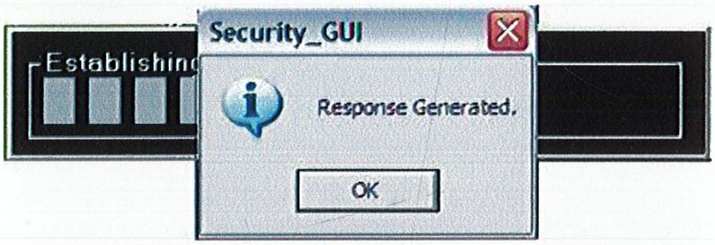
After this step, GUI begins to load itself and establishes a connection in between the hardware provided and computer to control and monitor the system, through serial port communication.



Snapshot 5.14

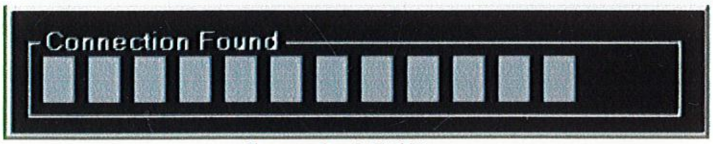


Snapshot 5.15

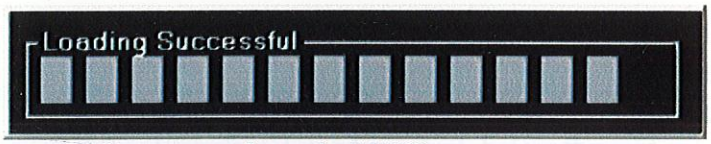


Response Generated.

Snapshot 5.16



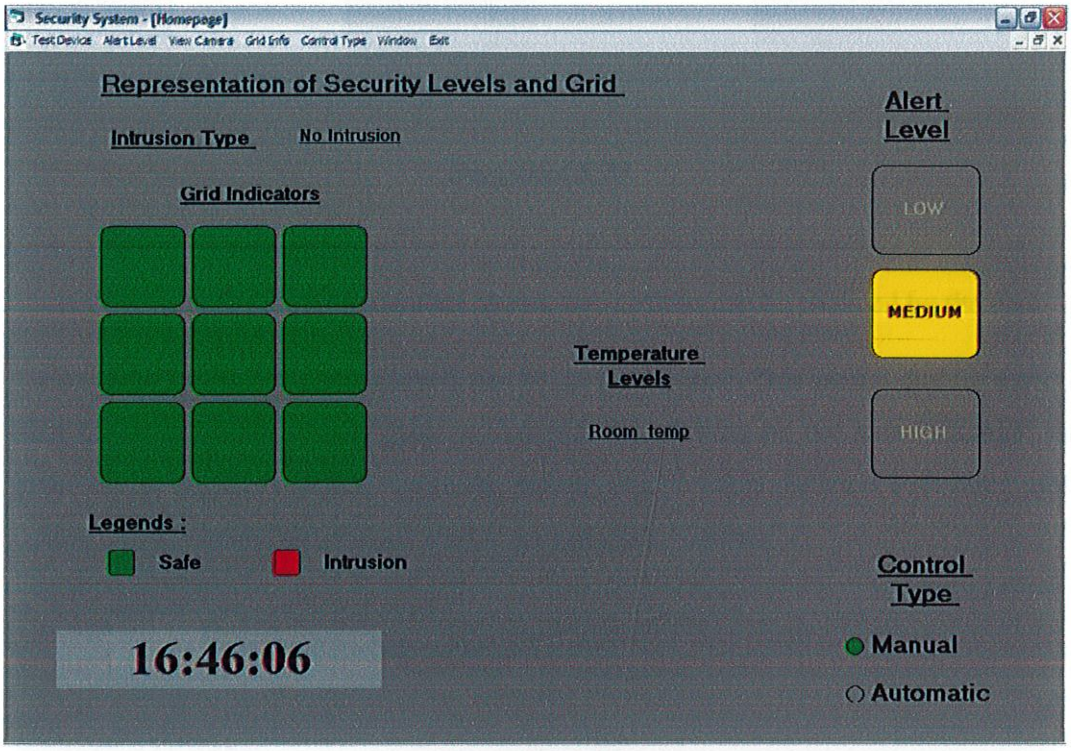
Snapshot 5.17



Snapshot 5.18

With the successful loading, a Main page for GUI is opened with the default configuration settings as:

- Alert Level: Medium
- Control Type: Manual



Intrusion Type: No Intrusion

Temperature Levels: Room temp

Alert Level: MEDIUM

Legends :

🟩 Safe 🟥 Intrusion

16:46:06

Control Type

● Manual

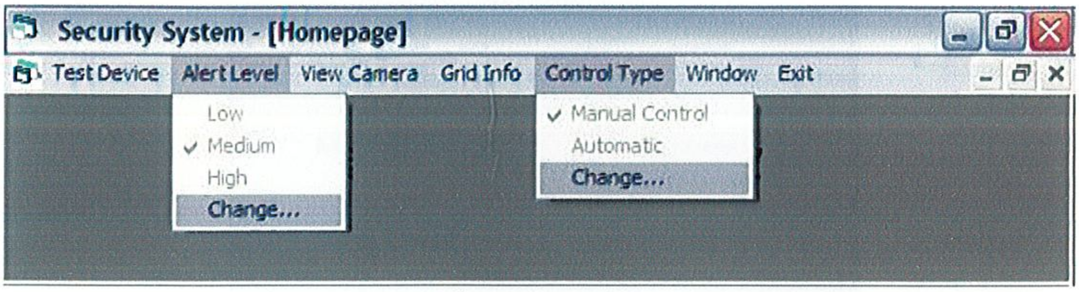
○ Automatic

Snapshot 5.19

Main page of the system is the extremely talented work of the programmers which summarizes all the operations in one single screen. Here, the green squares represent the grid plates, the color green specifies that all the plates are safe and no intrusion has been detected.

On the left side of the screen, is the summarization of the control and alert levels which can be changed through another procedure. And, below the grid representation is the legends followed by the digital clock, which reminds the user of the time.

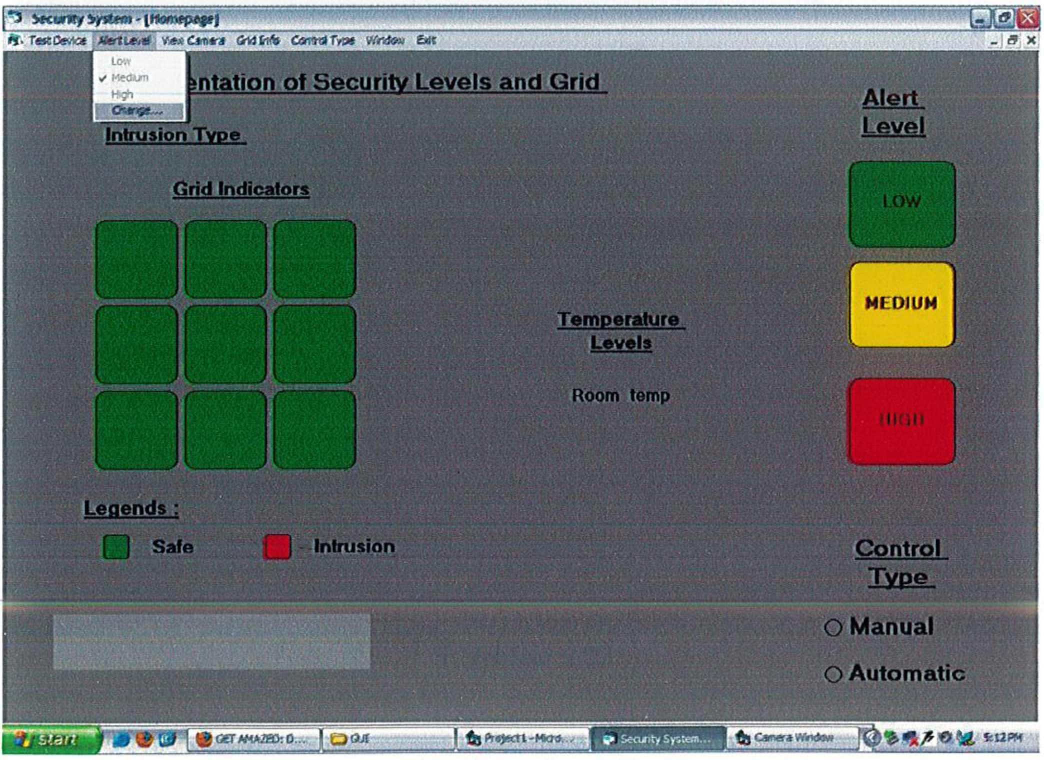
With the help of options provided in GUI menu, there are several operations available to the user. The operations available are:



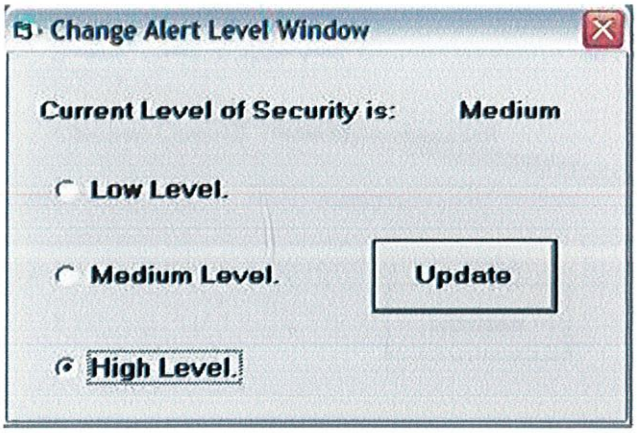
Snapshot 5.20

As can be seen from the snapshot above, two options can be changed by the user.

1. **Alert Level:** Different levels can be set by the user. This means that for various alert levels threshold for the grids is changed. By clicking on the icon "Change" the given below form is opened, where the present alert level is depicted with three other option buttons, select the desired alert level and click "Update".



Snapshot 5.21



Current Level of Security is: Medium

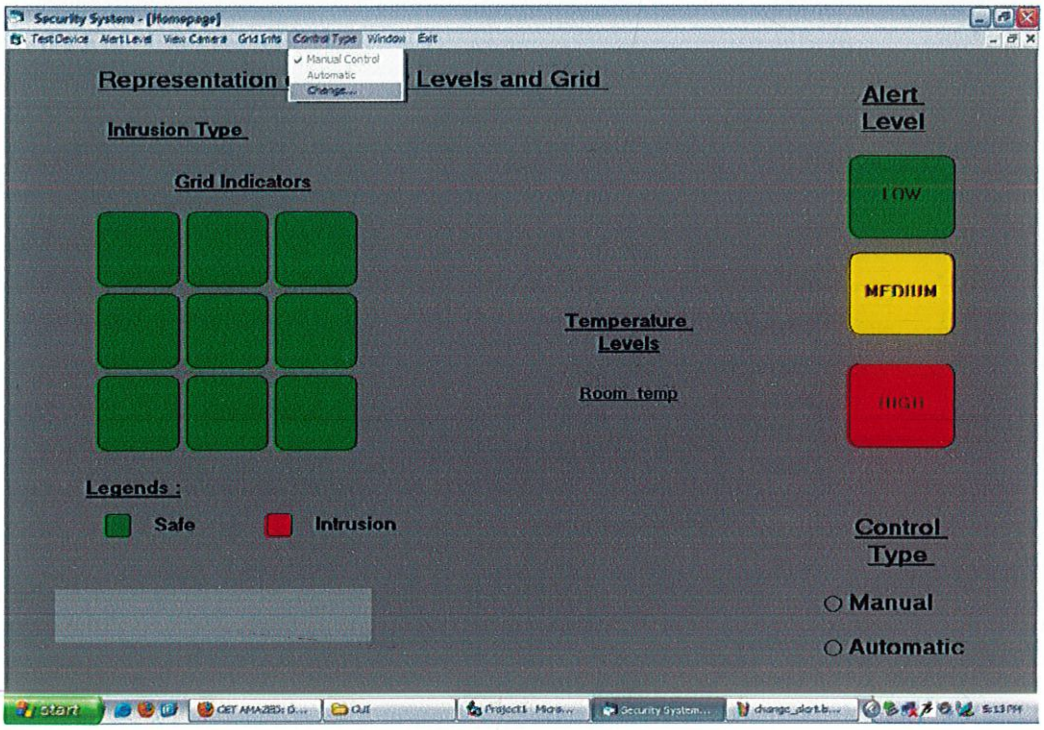
○ Low Level.

○ Medium Level.

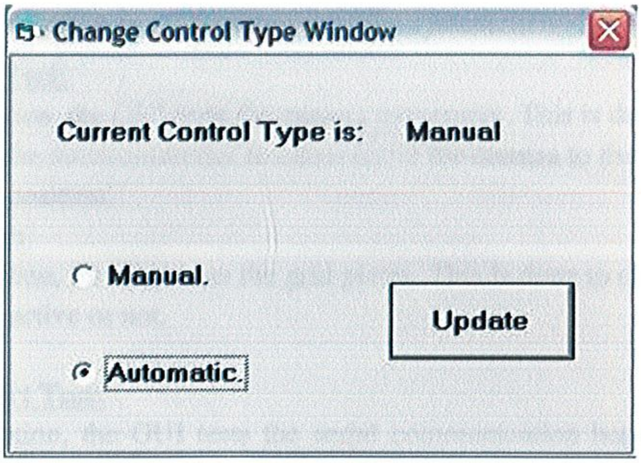
● High Level.

Snapshot 5.22

2. **Control Type:** The user can change control of the security from Automatic to Manual. By clicking on the icon "Change" the given below form is opened, where the present control type is depicted with two other option buttons, select the desired control type and click "Update".



Snapshot 5.23



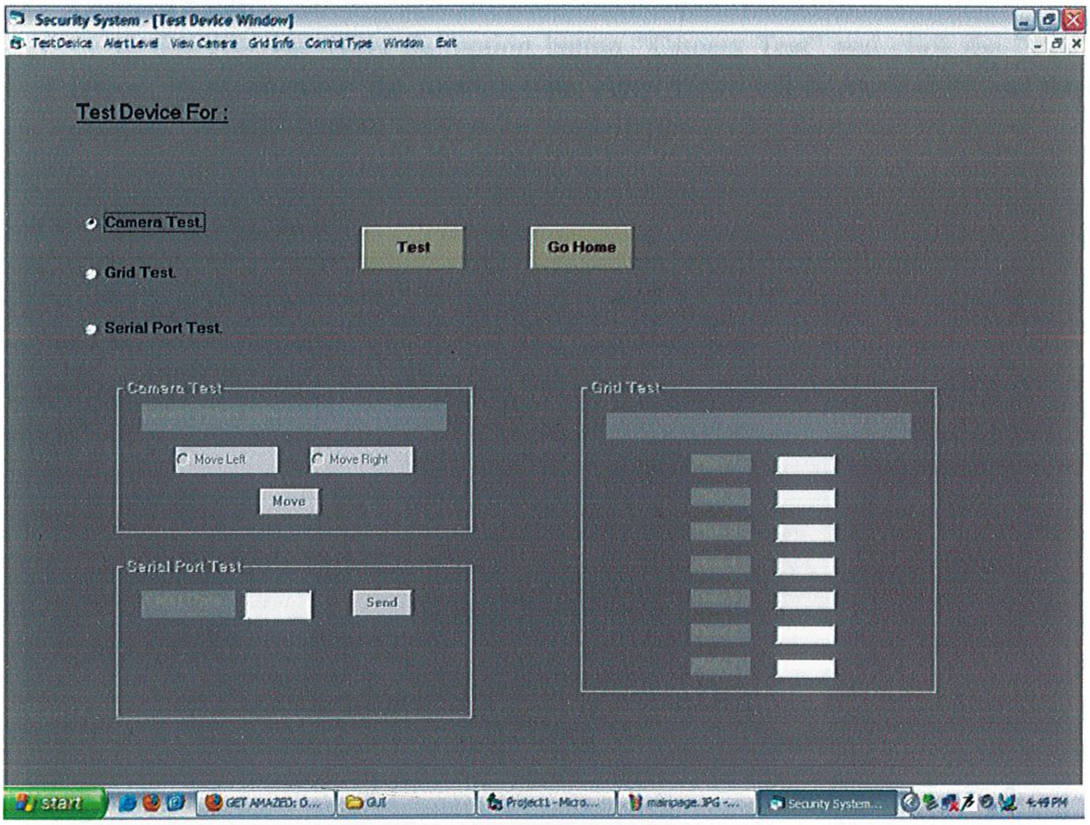
Current Control Type is: Manual

○ Manual.

● Automatic.

Snapshot 5.24

Another option available to the user in the GUI interface is Test Device:



Test Device For :

● Camera Test

○ Grid Test

○ Serial Port Test

Snapshot 5.25

In this section we can test three devices of the system.

1. **Camera Test:**

   In this option, the GUI tests the camera movement. This is done by sending a command to the microcontroller to either move the camera to the left or to the right of its current position.

2. **Grid Test:**

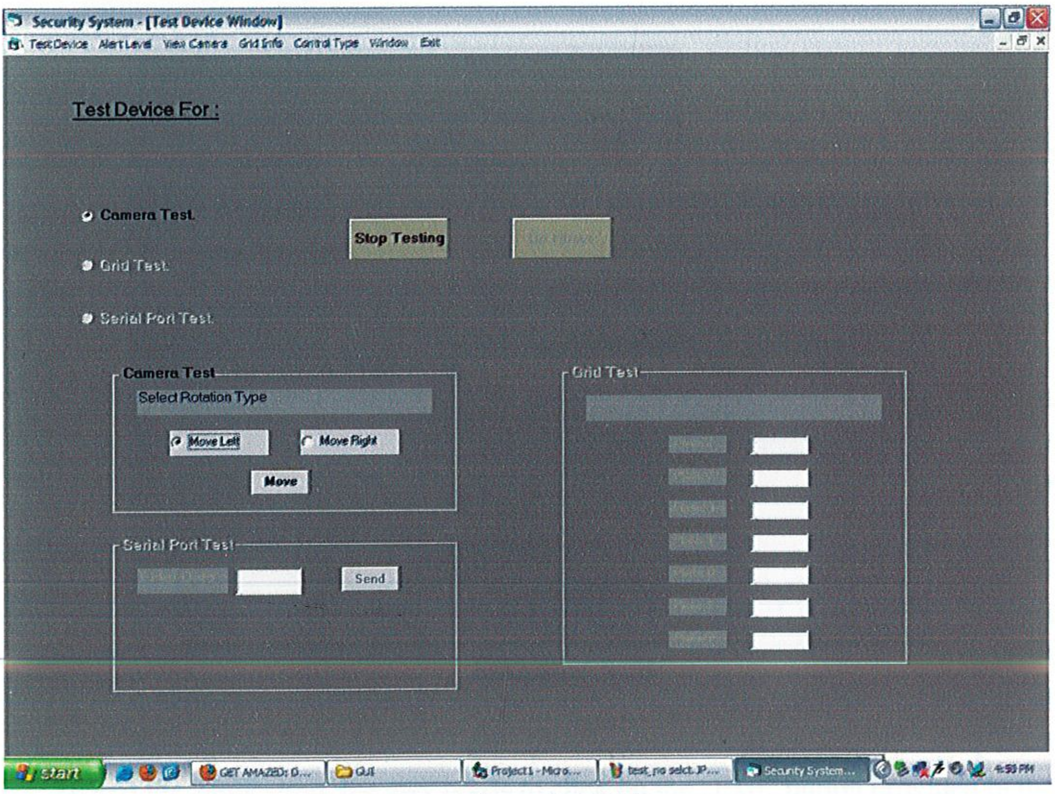
   In this option, the GUI tests the grid plates. This is done to check if all the grids are detecting active or not.

3. **Serial Port Test:**

   In this option, the GUI tests the serial communication between the computer terminal and the hardware of the system. This is done by sending a character across the port, which if received back, indicates the established and error-free connection.

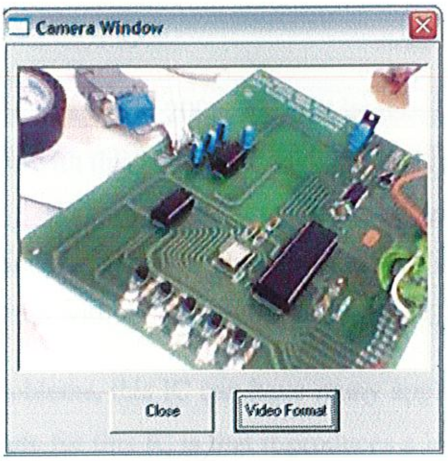
Camera Test:

In this test, we select the option button "Camera Test" and click the "Test" button. Now, we select the rotation type, either move left or move right and then click on "MOVE" button.



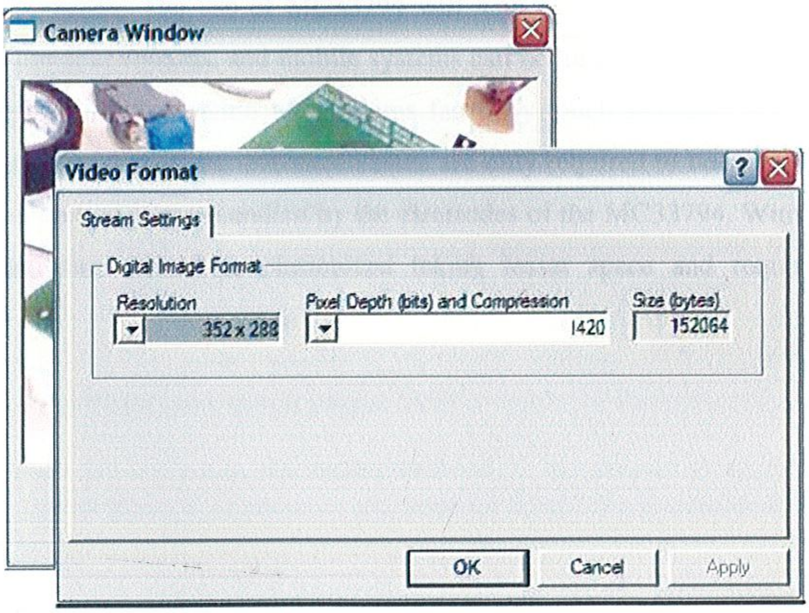
Snapshot 5.26

Apart from the testing, GUI provides user with an option to view camera in the computer as well, camera can be accessed by clicking on the "View Camera" icon, as shown below.



Snapshot 5.27

This camera window can be closed by clicking on the button "Close", also, we have provided user with the option to adjust the video format of the camera, this can be accessed by clicking on "Video Format", where user can change the resolution and pixel depth of the video, as shown below.



Digital Image Format

Resolution: 352 x 288

Pixel Depth (bits) and Compression: I420

Size (bytes): 152064

Snapshot 5.28

To exit the application, the user needs to click on the "Exit" button provided in the top menu, in the right corner.

CHAPTER 6
Application of MC33794

MC33794 is an IC developed in the year 2005. As such its uses have been very limited up till now. Part of the problem with this is that although the IC is a processor, meaning that it can be programmed, it's packaging (SOIC) creates a problem in the development of easily available program burners. There is also the fact that, due to its packaging the circuit cannot be easily realized without the use of fabricated PCBs.

Despite all the above problems, this IC can have many applications in various fields. As the basic working principle for this IC is that it produces a low-level electric field and detects any fluctuations in the said field, many contact based systems can be converted into non-contact based systems. An example can be the series of **'Touch'** mobiles, keyboards and various other utility items. It is found that if left without calibration for some time, the **touch** systems require a harder touch. Also these systems are very fragile and are easily broken. The use of MC33794 will lead to **'virtual touch'** concept. For example, the touch keyboards, and mobile systems can be turned into non-touch systems. This will lead to the elimination of problems faced by touch systems. The calibration problem is easily solved as the reference values are only required to be set up for such a system, the calibration part is handled by the electrodes of the MC33794. With the use of MC33794 hardware can be revolutionized taking lesser space and requiring lesser contact.

CHAPTER 7

7.1 Future Prospective

In a security system, there is always a scope for improvement of design, whether be it through the economy of design used or better systems used. This project proposes a novel idea of using the said system along with pre-existing detection systems used. One of the systems that can be used is ultrasonic sensors.

Since the innovation in IC integration and sensor sensitivity is constantly bringing in new technologies, the design can be further improved with the help of these advancements. There is also hope that the biometric detection systems become sufficiently self-reliant so that the project can be used with the biometric system.

With sufficient use of MC33794, its programmable hardware can be easily available in the near future. With this advancement, there is a possibility that the microcontroller, ATMEGA32 will not be required. This will improve the overall design, decrease losses due to interfacing and also increase the processing speed for instructions. With the MC33794 being programmed, a number of advancements and additional features can be added in the system. These systems, along with these advancements, can also further the concept of **"Virtual Touch".**

7.2 Conclusion

This project provides a novel solution to the problem which is being faced by the keepers of the antique and the most valuable artifacts. As most of the sensors or security measures taken are visible to the naked eye and can be fooled such as security personals, CCTV cameras and biometric sensors. Other measures which are not visible can not be used in the areas which are open and have open access, such as LASER and other highly sensitive sensors. Both the constraint that is of open access and sensor visibility are removed in this project as the sensor plates can be carpeted or can be layered with artificial grass so as to make them invisible for the unknown persons.

The constraint regarding the implementation of the project in an isolated or an easily accessible place is removed as the level of sensitivity is can be set by the user and thus empowering it with unpredictability. The interfacing of the system with the temperature sensor makes it a lot more secured device in the closed areas where temperature is nearly constant. This project does not claim to provide the ultimate foolproof and unbreakable security system but it does claim to provide a better method to ensure security where there is an access to the crowd and does it work silently and efficiently.

APPENDIX A

Referenced Datasheets

1. ATMEGA32: www.atmel.com/atmel/acrobat/doc2503.pdf
2. MAX232: www.datasheetcatalog.com/datasheet/M/MAX232.shtml
3. MC33794: www.datasheetarchive.com/MC33794-datasheet.html
4. ULN2003: www.st.com/stonline/books/pdf/docs/5279.pdf
5. Unipolar stepper motor:
   http://www.jameco.com/wcsstore/Jameco/Products/ProdDS/163395.pdf

APPENDIX B

Codes used in the Project

1. Code Burned in ATMEGA32.

```
This program was produced by the
CodeVision AVR V1.25.5 Professional
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project:
Version:
Date    : 5/10/2010
Author: F4CG
Company: F4CG
Comments:


Chip type           : ATmega32
Program type        : Application
Clock frequency     : 1.000000 MHz
Memory model        : Small
External SRAM size: 0
Data Stack size     : 512
*****************************************************/

#include <mega32.h>

// Standard Input/output functions
#include <stdio.h>

#include <delay.h>

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
```

```
//Initializing the variables
int v;
char a;
 int o,z,angle1,x;
 int w ,diff[10],angle[8];
 signed int Efield[10] ;
 int base[10]={67,80,79,68,65,65,168,232};
 int base_diff_low[10];
 int base_diff_med[10];
 int base_diff_high[10];

//Module to calculate difference
void difference_module(void)
 {
  w=1;
   while(w<8)
 {
 diff[w]= Efield[w]- base[w];
 w=w+1;
 }
 }
 void put_value()
 {   g=1;
    while(g<8)
     {
     ef=Efield[g];
     s=ef/8;
     a = s;
     putchar(a);
     putchar(s);
    g=g+1;
     }
 }

 //Module to read the values of sensors
 void read_mod(int l)
    {
   // putchar('n');
    Efield[l]=read_adc(0x00);
    delay_ms(10);
    }
//Module to delay
void delayloop(int x)
    {
   // putchar('m');
    read_mod(x);
    q=q+1;
    }
//Module to select paltes
 void read_module()
 {
  PORTD.3=1;
 //putchar('r');
  //q=getchar();
  q=1;
  if(q== 1)
  {
 // putchar('a');

  delayloop(a);
  }
 if(q== 2)
  {
 //putchar(q);
 //putchar('b');

  delayloop(b);
  }
 if(q== 3)
  {
  //putchar(q);
 // putchar('c');

  delayloop(c);
   }
 if(q==4)
  {
  //putchar(q);
 // putchar('d');

  delayloop(d);
  }
 if(q==5)
  {
  //putchar();
 // putchar('e');

  delayloop(e);
  }
 if(q==6)
  {
  //putchar(q);
 /// putchar('f');

  delayloop(f);
  }
 if(q==7)
  {
  //putchar(q);
 //putchar('g');
  PORTB=0b00011101;
  delayloop(g);
  }
}
//Module to move motor forward
 void forward(void)
    {
    PORTD.2=0;
    PORTD.3=1;
    PORTD.4=1;
    PORTD.5=0;      delay_ms(100);
    PORTD.2=0;
    PORTD.3=0;
    PORTD.4=1;
    PORTD.5=1;
    delay_ms(100);
    PORTD.2=1;
    PORTD.3=1;
    PORTD.4=0;
    PORTD.5=0;
    delay_ms(100);
    PORTD.2=1;
    PORTD.3=0;
    PORTD.4=0;
    PORTD.5=1;
    delay_ms(100);
    }
//Module to move motor backward
void back()
    {
PORTD.2=1;
    PORTD.3=0;
    PORTD.4=0;
    PORTD.5=1;
    delay_ms(100);
    PORTD.2=1;
    PORTD.3=1;
    PORTD.4=0;
    PORTD.5=0;
    delay_ms(100);
    PORTD.2=0;
    PORTD.3=1;
    PORTD.4=1;
    PORTD.5=0;
    delay_ms(100);
    PORTD.2=0;
    PORTD.3=0;
    PORTD.4=1;
    PORTD.5=1;
    delay_ms(100);

    }
//Module to position the camera
void position_camera(int grid,int v)
{
if(v<0)
     {
      while(o<(0-v))
     {
     back();
     o=o+1;
     };
     angle1=angle[grid];
     o=0;
     }

     if(v>0)
```

```
      {
       while(o<v)
      {
      forward();
      o=o+1;
      };
      angle1=angle[grid];
      o=0;
      }


}
//Module to get the camera angle
void move_camera(int grid)
{
      v=angle[grid]-angle1;
      if( v != 0)
      {
       position_camera(grid,v);
      }
}
 //Module to compareb the values of sensors
 void compare(void)
 {
  if(z==1)
  {
   w=0;
   while(w<8)
   {
    if(diff[w]<(0-base_diff_low[w])||diff[w]>base_diff_low[w])
    {
     putchar('D');
     if(b=='z')
     {
     putchar(w);
     }
     move_camera(w);
    }
    w=w+1;
   }
  }
 if(z==2)
  {
   w=0;
   while(w<8)
   {
    if(diff[w]<(0-base_diff_med[w])||diff[w]>base_diff_med[w])
     {
     putchar('D');

     if(b=='z')
     {
     putchar(w);
      }
      move_camera(w);
     }
    w=w+1;
   }
  }
 if(z==3)
  {
   w=0;
   while(w<8)
   {
    if(diff[w]<(0-base_diff_high[w])||diff[w]>base_diff_high[w])
    {
     putchar('D');
     b=getchar();
     if(b=='z')
      {
     putchar(w);
      }
       move_camera(w);
     }
    w=w+1;
   }
  }
 //Module to select alert level and raise alarm
 }
void alert_low(void)
{
 PORTD.6=1;
 x=0;
 while(x<5)
 {
 read_module();
 difference_module();
 z=1;
 compare();
 x=x+1;
 }
 delay_ms(100);
 PORTD.6=0;

}
void alert_medium(void)
{
 PORTD.6=1;
 x=0;
 while(x<5)
 {
 read_module();
 difference_module();
 z=2;
 compare();
 x=x+1;
 }
 delay_ms(100);
 PORTD.6=0;
}
void alert_high(void)
{
```

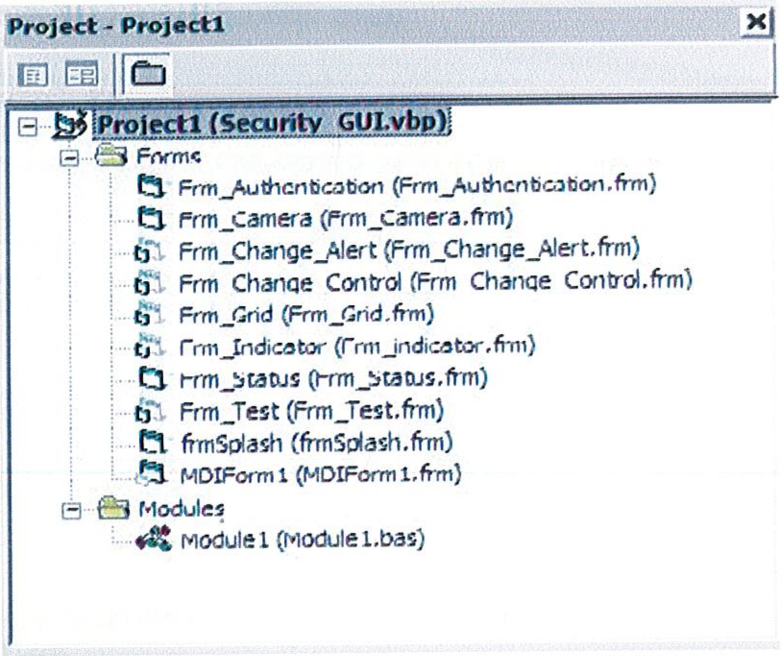
```
PORTD.6=1;
x=0;
while(x<5)
{
read_module();
difference_module();
z=3;
compare();
x=x+1;
}
delay_ms(100);
PORTD.6=0;
}

//The main program
void main(void)
{
char k;
int i;
char r;
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0xff;
DDRA=0xff;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0xff;
DDRB=0x0f;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0xff;
DDRC=0xff;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0xff;
DDRD=0xff;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 4800
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x0C;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: OffSFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 500.000 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE & 0xff;

while (1)
    {
        // Place your code here
     PORTD.6=1;
    // read_module();
     k = getchar();
     angle1=angle[4];
     PORTD.6=0;
     if(k=='Q')                  //Serial module
     {
     q = getchar();
     putchar(q);
     }
     if(k=='R')                  //motor testing
     {
     putchar(k);
     while(i<4)
     {
     forward();
     i=i+1;
     };
     i=0;
     angle1=angle[7];
     main();
     }
     if(k=='L')                  //motor testing
     {
      putchar(k);
     while(i<4)
      {
      back();
      i=i+1;
      };
      i=0;
      angle1=angle[1];
     main();
     }
    if(k=='S')                  //Grid testing
    {
     putchar(k);
     read_module();
     PORTD.3=0;
     put_value();
     main();
    }
    if(k=='A')                  //Connection establish
    {
     putchar(k);
     PORTD.6=0;
```

```
   delay_ms(500);
   PORTD.2=1;
   PORTD.3=1;
   PORTD.4=1;
   PORTD.5=1;
   PORTD.6=1;
   delay_ms(1000);
   PORTD.2=0;
   PORTD.3=0;
   PORTD.4=0;
   PORTD.5=0;
   PORTD.6=0;
   delay_ms(1000);
   main();
  }
   if(k=='M')                    //Alert levels
 {
  r= getchar();
 if(r=='1')
 {

 alert_low();

 }
 if(r=='2')
 {
 alert_medium();
 }
 if(r=='3')
 {
 alert_high();
 }

 }
  };
}
```

2. Code for GUI of Adaptive Security System:

It contains the following forms which when executed, successfully make the GUI.



```
Project - Project1
Project1 (Security GUI.vbp)
  Forms
    Frm_Authentication (Frm_Authentication.frm)
    Frm_Camera (Frm_Camera.frm)
    Frm_Change_Alert (Frm_Change_Alert.frm)
    Frm Change Control (Frm Change Control.frm)
    Frm_Grid (Frm_Grid.frm)
    Frm_Indicator (Frm_indicator.frm)
    Frm_Status (Frm_Status.frm)
    Frm_Test (Frm_Test.frm)
    frmSplash (frmSplash.frm)
    MDIForm1 (MDIForm1.frm)
  Modules
    Module1 (Module1.bas)
```

Frm_Authentication:

```
‘ Code for the authentication page.
Dim pass As String  ‘ defines a string variable.

Private Sub CMD_Cancel_Click()
   End  ‘ends the application.
End Sub

Private Sub CMD_Login_Click()
   pass = Txt_password.Text  ‘ receives the text written in textbox as input password.
   If pass = "user" Then  ‘compares the password for authentication.
      If MsgBox("Authentication Successful." + vbCr + "Welcome, User.", vbInformation,
"Authentication Successful") = vbOK Then  ‘ displays success notification.
      Unload Me  ‘ unloads authentication page for loading GUI.
      Frm_status.Show
      Frm_status.SetFocus ‘sets focus on loading.
      End If
   Else
      If MsgBox("Authentication Failed." + vbCr + "Kindly enter correct password.", vbCritical,
"Authentication Failed") = vbOK Then  ‘ displays failure notification.
      End If
      Txt_password.Text = ""
      Txt_password.SetFocus  ‘ redirects to authentication page.
   End If
End Sub
```

Frm_Camera:

```
Dim hwdc As Long ‘defines a long variable.
Dim startcap As Boolean ‘defines a boolean variable.

Private Sub cmdClose_Click()
Dim temp As Long ‘ defines  a long variable
If startcap = True Then
temp = SendMessage(hwdc, WM_CAP_DRIVER_DISCONNECT, 0&, 0&) ‘disconnects camera_
_from_ GUI.
startcap = False
End If
Unload Me ‘unloads Camera page.
MDIForm1.SetFocus ‘shows main page.
End Sub

Private Sub cmdVideoFormat_Click()
 Dim temp As Long
 If startcap = True Then
  temp = SendMessage(hwdc, WM_CAP_DLG_VIDEOFORMAT, 0&, 0&) ‘opens video format
page.
End If
End Sub

Private Sub Form_Load()‘defines procedure to be loaded at startup of the form.
Dim temp As Long
  hwdc = capCreateCaptureWindow("Adaptive Security System", ws_child Or ws_visible, 0, 0, 320,
240, Picture1.hWnd, 0)
  If (hwdc <> 0) Then
    temp = SendMessage(hwdc, wm_cap_driver_connect, 0, 0)
    temp = SendMessage(hwdc, wm_cap_set_preview, 1, 0)
    temp = SendMessage(hwdc, WM_CAP_SET_PREVIEWRATE, 30, 0)
    startcap = True
    Else
    MsgBox ("No Webcam found") ‘displays notification if no web camera is found.
  End If
End Sub
```

Frm_Change_Alert:

```
  Dim select_alert As Byte ‘defines byte type variable.

Private Sub Cmd_Update_Click()
MSComm1_ALERT.Output = "M" ‘sends output to device through serial port.
    Select Case (select_alert) ‘now we select cases for different values of alert type.

        Case 1 ‘if alert is low
        Alert = select_alert
        Refresh_Alert ‘ refreshes the main page displaying new alert levels.
        MSComm1_ALERT.Output = "1" ‘outputs value 1 to device.

        Case 2 ‘ if alert is medium
        Alert = select_alert
        Refresh_Alert
        MSComm1_ALERT.Output = "2" 'outputs value 1 to device.

        Case 3 'if alert is high
        Alert = select_alert
        Refresh_Alert
        MSComm1_ALERT.Output = "3" 'outputs value 1 to device.

    End Select
    Unload Me 'unloads the form
    Frm_indicator.Show 'displays main page.
    MDIForm1.SetFocus
End Sub

Private Sub Form_Load() 'code to be executed at load-time of form.
    Select Case (Alert) 'checks value for current alert level.
        Case 1
        lbl_status.Caption = "Low" 'displays alert level as low.
        Case 2
        lbl_status.Caption = "Medium" 'displays alert level as medium.
        Case 3
        lbl_status.Caption = "High" 'displays alert level as high.
    End Select
    Port_Open ' opens port for communication.
End Sub

Private Sub Opt_High_Click()
    select_alert = 3 'sets alert variable to high.
End Sub

Private Sub Opt_Low_Click()
    select_alert = 1 'sets alert variable to low.
End Sub
Private Sub Opt_Medium_Click()
    select_alert = 2 'sets alert variable to medium.
End Sub

Private Sub Port_Open()
    MSComm1.CommPort = 5 'select communication port number.
    MSComm1.Settings = "4800,N,8,1" ' sets baud rate to 4800, no parity, 8 data bits and 1 stop bit.
    MSComm1.InputLen = 0 'input length = 0, reads whole buffer at all times.
    MSComm1.PortOpen = True 'opens port for communication.
End Sub
```

Frm_Change_Control:

```
Dim Select_Control As Integer
Private Sub Cmd_Update_Click() 'updates the alert level according to selected level.
    Select Case (Select_Control)
        Case 1 'control type set to manual.
        Control = Select_Control
        Refresh_Control 'refreshes main page for new control type.
        Case 2 'control type set to automatic.
        Control = Select_Control
        Refresh_Control
    End Select
    Unload Me ‘unloads update page.
    Frm_indicator.Show ‘displays main page.
    MDIForm1.SetFocus
End Sub

Private Sub Form_Load() ‘code to be executed at startup of the form.
    Select Case (Control) ‘displays current control type.
        Case 1
        lbl_status.Caption = "Manual"
        Case 2
        lbl_status.Caption = "Automatic"
    End Select
End Sub

Private Sub Opt_auto_Click()
    Select_Control = 2 ‘selects control type to automatic.
End Sub

Private Sub Opt_Manual_Click()
    Select_Control = 1 ‘selects control type to manual.
End Sub
```

Frm_Grid:

```
Private Sub Cmd_Home_Click()
    Unload Me
    Frm_indicator.Show
    MDIForm1.SetFocus
End Sub
```

Frm_Indicator:

```
Dim i As Byte

Private Sub Cmd_Stop_Alarm_Click() ‘turns off the alarm.
Alarm = False
Cmd_Stop_Alarm.Enabled = False
Cmd_Stop_Alarm.Visible = False
End Sub

Private Sub Form_Load() ‘code to be executed when the form loads.
    Refresh_Alert ‘refreshes alert level.
    Refresh_Control ‘refreshes control type.
    Lbl_Intrusion.Caption = "No Intrusion" ‘by default no intrusion is detected.
    Alarm = False ‘alarm set to false.
    Port_Open
End Sub

Private Sub Timer_Alarm_Timer() ‘timer fires every 200ms.
    If Alarm Then
        Frm_indicator.BackColor = vbRed ‘displays a background as red
        i = i + 1
        If i Mod 2 = 0 Then
    Lbl_Intrusion.Caption = "ALARM" ' sets the alarm status label to ALARM.
    Frm_indicator.BackColor = vbRed
      Beep 800, 300 'sends a beep.
      Else
      Frm_indicator.BackColor = &H8000000C 'sets the background to default.
    End If
    Cmd_Stop_Alarm.Enabled = True
    Cmd_Stop_Alarm.Visible = True
  End If
End Sub

Private Sub Timer_clock_Timer() 'fires every 1 second.
Lbl_Time.Caption = Format(Now, "HH:MM:SS") 'displays time in main page.
End Sub

Private Sub Port_Open() 'function that opens port for communication.
  MSComm1.CommPort = 5 'selects communication port.
  MSComm1.Settings = "4800,N,8,1" 'baud rate set to 4800, no parity, 8 data bits and 1 stop bit.
  MSComm1.InputLen = 0 'reads whole buffer at once.
  MSComm1.PortOpen = True 'opens port.
End Sub

Private Sub Timer_Check_Timer() 'fires every 2second, requests device for alarm status.
Dim String_Alert As String
If Control = 1 Then 'if manual control.
String_Alert = Alert
MSComm1.Output = String_Alert
check_values 'requests sensor values.
Check_Alarm 'requests alarm status
End If
If Control = 2 Then 'if control type is automatic.
  If "06:00:00" < Format(Now, "HH:MM:SS") Or "10:00:00" > Format(Now, "HH:MM:SS") Then
  Alert = 2 'sets alert to medium for 6am-10am.
  Refresh_Alert
  MSComm1.Output = "M" 'sends instructions to change alert level.
  MSComm1.Output = "2" 'sends value for alert level.
  End If

  If "10:00:00" < Format(Now, "HH:MM:SS") Or "16:00:00" > Format(Now, "HH:MM:SS") Then
  Alert = 1 ' sets alert to low for 10am-4pm.
  Refresh_Alert
  MSComm1.Output = "M"
  MSComm1.Output = "1"
  End If

  If "16:00:00" < Format(Now, "HH:MM:SS") Or "06:00:00" > Format(Now, "HH:MM:SS") Then
  Alert = 3 'sets alert level to high for 4pm-6am.
  Refresh_Alert
  MSComm1.Output = "M"
  MSComm1.Output = "3"
  End If

  Check_Alarm 'requests alarm status.
  check_values ' requests sensor values.
  End If
  End Sub
```

```
Private Sub check_values() 'function to call sensor values from device.
Dim check_values As Variant
Dim Check_Plate As Variant
Dim i As Integer
MSComm1.Output = "I" 'asks device for intrusion.
Re_Check_Values:
check_values = MSComm1.Input
If check_values = "" Then
   GoTo Re_Check_Values
Else
   Select Case check_values
      Case "C"
         MSComm1.Output = "P"
Re_Check_Plate:
         Check_Plate = MSComm1.Input
         If Check_Plate = "" Then 'if buffer input NULL, loops until there is a value.
            GoTo Re_Check_Plate
         Else
            i = Check_Plate - 1
            All_Grid_Safe
            Grid_Plate(i).BackColor = vbRed 'displays intruded plate.
         End If
          Case "D" ' if no intrusion
       All_Grid_Safe 'displays all grid plates in green.

   End Select
End If
End Sub

Private Sub Check_Alarm() 'calls alarm status from device.
Dim Check_Alarm As Variant
MSComm1.Output = "B"
Re_Check_Alarm:
Check_Alarm = MSComm1.Input
If Check_Alarm = "" Then
GoTo Re_Check_Alarm
Else
   Select Case (Check_Alarm)

      Case "Y" 'alarm.
      Alarm = True

      Case "N" 'no alarm.
      Alarm = False
   End Select
End If
End Sub
```

Frm_Status:

```
Dim i As Byte
Dim time As Byte
Dim transmit As Byte
Dim recieve As Variant
```

```
Option Explicit

Private Sub Form_Load()
   For i = 0 To 13
      Label1(i).Visible = False
   Next i
   time = 0
End Sub

Private Sub Timer1_Timer() 'timer fires at every 200ms.
   If time < 14 Then
      Label1(time).Visible = True
   End If
   time = time + 1
   If time = 5 Then
      Frm_status.Caption = "Establishing Connection"
      Port_Open
      End If
   If time = 10 Then
      MSComm1.Output = "A"
      If MsgBox("Response Generated.", vbInformation) = vbOK Then
Re_recieve:
         recieve = MSComm1.Input 'reads input from device
      If recieve = "" Then
      GoTo Re_recieve
      Else
      If recieve = "A" Then
         Frm_status.Caption = "Connection Found"
      Else
         If MsgBox("Device not Present, Establishment Failed.",vbCritical) = vbOK Then
            Unload Me
            Load Frm_Authentication
            Frm_Authentication. Show
         End If
      End If
      End If
   End If
   End If

   If time = 13 Then
      Frm_status.Caption = "Loading Successful"
   End If
   If time = 16 Then
      Unload Me
      MDIForm1.Show
      Frm_indicator.Show
   End If
End Sub

Private Sub Port_Open()
   MSComm1.CommPort = 5
   MSComm1.Settings = "4800,N,8,1"
   MSComm1.InputLen = 0
   MSComm1.PortOpen = True
   transmit = True
End Sub
```

Frm_Test:

```
Dim test_type As Byte
Dim rotation_type As Byte
Dim click_count As Byte
Dim transmit As Boolean
Dim Plate_count As Byte
Dim a, b, c As Variant

Private Sub Cmd_Home_Click()
   transmit = False 'if transmit is false, no communication will take place.
   Unload Me
   Frm_indicator.Show
   MDIForm1.SetFocus
End Sub

Private Sub Cmd_Rotation_Move_Click() 'moves camera to left or right
If transmit Then
          If Opt_rotation_left.Value = True Then 'if left option is selected.
             MSComm1.Output = "L" 'sends instruction to device to move left.
          Else
          If Opt_rotation_right.Value = True Then
             MSComm1.Output = "R" 'sends instruction to device to move right.
          End If
          End If

End If
Re_Rotation:
       b = MSComm1.Input 'reads input from device.
       If b = "" Then
          GoTo Re_Rotation
       Else
       Print b 'checks for acknowledgment.
          Select Case (b)
                 Case "L"
                 Lbl_Rotation.Caption = "Moved Left" 'sets status label
                 Case "R"
                 Lbl_Rotation.Caption = "Moved Right" 'sets status label
              End Select
       End If
End Sub

Private Sub Cmd_Serial_Send_Click() 'code for testing serial port.
If transmit Then
       MSComm1.Output = Txt_serial_data.Text 'sends data in text box to device.
       Lbl_Serial_transmit.Caption = "Sending: " + Txt_serial_data.Text
   End If
Re_Serial:
  a = MSComm1.Input 'device sends back the same data.
  If a = "" Then
          GoTo Re_Serial
        Else
          lbl_serial_recieve.Caption = "Received : " + a 'if the data is received.
        Lbl_Serial_transmit.Caption = "Data Sent : " + Txt_serial_data.Text 'sets label to data
received.
      End If
End Sub

Private Sub Cmd_Test_Click() 'code for the test button, a test type is selected here.
Dim i As Integer
Dim s As String
  click_count = click_count + 1
  If (click_count Mod 2) = 0 Then
    transmit = False 'sets transmit to false.
    Test_Alloptions_True
    Test_All_False
    Cmd_Test.Caption = "Test"
    Cmd_Home.Enabled = True
    MSComm1.PortOpen = False 'closes port.
    Refresh_Test 'refreshes the test page.
  Else
    Test_Alloptions_False
    Port_Open 'opens port.
    Select Case (test_type)
      Case 2 'camera test selected.
      Opt_Rotation.Enabled = True
      Test_Rotation_Enabled

      Case 3 'grid test selected.
      Opt_Grid.Enabled = True
      Test_Grid_Enabled
      MSComm1.Output = "S"

      For i = 0 To 7 'loops for all plates present.
        s = i + 1
        MSComm1.Output = s
Re_Grid:
        c = MSComm1.Input
            If c = "" Then
        GoTo Re_Grid

        Else

        Select Case (i) 'displays the grid value plate-wise.

        Case 0
        Grid_Txt(i).Text = c
        Case 1
        Grid_Txt(i).Text = c
        Case 2
        Grid_Txt(i).Text = c
        Case 3
        Grid_Txt(i).Text = c
        Case 4
        Grid_Txt(i).Text = c
        Case 5
        Grid_Txt(i).Text = c
        Case 6
        Grid_Txt(i).Text = c
        Case 7
         If c = "E" Then ‘checks for acknowledgement.
         Lbl_Grid.Caption = "Readings Successfully Received." ‘if true, sets caption of label box.
        End If
        End Select

      End If
      Next i
      Case 4
      Opt_Serial.Enabled = True
      Frame_Serial.Enabled = True
      Test_Serial_Enabled

    End Select
    Cmd_Test.Caption = "Stop Testing"
    Cmd_Home.Enabled = False
  End If
End Sub

Private Sub Form_Load()
  Test_All_False ‘disables all other than option buttons.
End Sub

Private Sub Opt_Grid_Click()
  test_type = 3 ‘sets test type to grid test.
End Sub

Private Sub Opt_rotation_Click()
  test_type = 2 ‘sets test type to camera test.
End Sub

Private Sub Opt_rotation_left_Click()
  rotation_type = 1 ‘moves camera to left.
End Sub

Private Sub Opt_rotation_right_Click()
  rotation_type = 2 ‘moves camera to right.
End Sub

Private Sub Opt_Serial_Click()
  test_type = 4 ‘test type set to serial port test.
End Sub

Private Sub Test_Alloptions_True() ‘enables all option buttons.
  Opt_Grid.Enabled = True
  Opt_Rotation.Enabled = True
  Opt_Serial.Enabled = True
End Sub

Private Sub Test_All_False() ‘disables all content on test page.
  Dim i As Integer
  Label9.Enabled = False
  Frame_Serial.Enabled = False
  Txt_serial_data.Enabled = False
  Cmd_Serial_Send.Enabled = False
  Lbl_Serial_transmit.Enabled = False
```

```
    lbl_serial_recieve.Enabled = False
    Frame_Rotation.Enabled = False
    Lbl_Rotation.Enabled = False
    Opt_rotation_right.Enabled = False
    Opt_rotation_left.Enabled = False
    Cmd_Rotation_Move.Enabled = False
    Frame_Grid.Enabled = False
    Lbl_Grid.Enabled = False
    For i = 0 To 6
        Grid_Txt(i).Enabled = False
        Grid_lbl(i).Enabled = False
        Grid_Txt(i) = ""
    Next i
End Sub

Private Sub Test_Serial_Enabled() 'enables serial port frame.
    Label9.Enabled = True
    Txt_serial_data.Enabled = True
    Cmd_Serial_Send.Enabled = True
    Lbl_Serial_transmit.Enabled = True
    lbl_serial_recieve.Enabled = True
End Sub

Private Sub Test_Rotation_Enabled() 'enables rotation frame.
    Frame_Rotation.Enabled = True
    Lbl_Rotation.Enabled = True
    Opt_rotation_right.Enabled = True
    Opt_rotation_left.Enabled = True
    Cmd_Rotation_Move.Enabled = True
End Sub

Private Sub Test_Grid_Enabled() 'enables grid test frame.
    Dim i As Integer
    Frame_Grid.Enabled = True
    Lbl_Grid.Enabled = True
    For i = 0 To 6
        Grid_Txt(i).Enabled = True
        Grid_lbl(i).Enabled = True
    Next i
End Sub

Private Sub Test_Alloptions_False() 'disables all option buttons.
    Opt_Grid.Enabled = False
    Opt_Rotation.Enabled = False
    Opt_Serial.Enabled = False
End Sub

Private Sub Port_Open() 'function to open port for communication.
    MSComm1.CommPort = 5 'selects communication port.
    MSComm1.Settings = "4800,N,8,1" 'sets baud rate to 4800, no parity, 8 data bits and 1 stop bit.
    MSComm1.InputLen = 0 'reads whole buffer.
    MSComm1.PortOpen = True 'opens port.
    transmit = True
End Sub
```

frmSplash:

```
Option Explicit
Dim time As Integer

Private Sub Form_KeyPress(KeyAscii As Integer) ‘ unloads the splash screen in event of any key is pressed.
  Unload Me
  Frm_Authentication.Show
  Frm_Authentication.SetFocus
End Sub

Private Sub Form_Load()
  time = 0
End Sub

Private Sub Frame1_Click() ‘unloads splash screen in case of any click on the frame.
  Unload Me
  Frm_Authentication.Show
  Frm_Authentication.SetFocus
End Sub

Private Sub Image1_Click()
  Unload Me
  Frm_Authentication.Show
  Frm_Authentication.SetFocus
End Sub
Private Sub Timer1_Timer() ‘timer fires every 500ms.
  If time = 5 Then ‘if load time of splash screen is 2.5 seconds, unloads the splash screen.
    Unload Me
    Frm_Authentication.Show
    Frm_Authentication.SetFocus
  End If
  time = time + 1
End Sub
```

MDIForm1:

```
Private Sub MDIForm_Load()
  Alert = 2 ‘sets default value of alert to medium.
  Control = 1 ‘sets default value of control type to manual.
  MDIForm1.mnu_low.Enabled = False ‘disables menu options.
  MDIForm1.mnu_medium.Enabled = False ‘disables menu options.
  MDIForm1.mnu_high.Enabled = False ‘disables menu options.
  MDIForm1.mnu_manual.Enabled = False ‘disables menu options.
  MDIForm1.mnu_auto.Enabled = False ‘disables menu options.
  Refresh_Alert ‘refresh alert levels at main page.
  Refresh_Control ‘refresh control type at main page.
  Frm_indicator.Show
End Sub

Private Sub mnu_cam_Click() ‘on click of this option, show camera.
Load Frm_Camera ‘displays camera form.
Frm_Camera.Show
```

```
End Sub

Private Sub mnu_change_alert_Click() ‘open change alert form.
   Unload_All_Child ‘function to unload all other child forms.
   Load Frm_Change_Alert
   Frm_Change_Alert.Show
End Sub

Private Sub mnu_change_control_Click() ‘open change control form.
   Unload_All_Child ‘unloads all child forms.
   Load Frm_Change_Control
   Frm_Change_Control.Show
End Sub

Private Sub mnu_exit_Click() ‘ends application.
Unload Frm_Camera
End
End Sub

Private Sub mnu_grid_Click() ‘opens grid info page.
   Unload_All_Child
   Load Frm_Grid
   Frm_Grid.Show
   Form1.Show
End Sub

Private Sub mnu_test_Click() ‘opens test page for device.
   Unload_All_Child
   Load Frm_Test
   Frm_Test.Show
End Sub
```

Module1:

```
Public Alert As Byte ‘defines alert as a public byte variable.
Public Alarm As Boolean ‘defines alarm as a public boolean variable.
Public Control As Byte ‘defines control as a public byte variable.

'defines beep as a function to generate frequency for alarm.
Public Declare Function Beep Lib "kernel32" (ByVal soundFrequency As Long, ByVal soundDuration
As Long) As Long

'for Web camera.
Public Const ws_child As Long = &H40000000
Public Const ws_visible As Long = &H10000000
Global Const WM_USER = 1024
Global Const wm_cap_driver_connect = WM_USER + 10
Global Const wm_cap_set_preview = WM_USER + 50
Global Const WM_CAP_SET_PREVIEWRATE = WM_USER + 52
Global Const WM_CAP_DRIVER_DISCONNECT As Long = WM_USER + 11
Public Const WM_CAP_DLG_VIDEOFORMAT As Long = WM_USER + 41
Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hWnd As Long, ByVal
wMsg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA"
(ByVal a As String, ByVal b As Long, ByVal c As Integer, ByVal d As Integer, ByVal e As Integer,
ByVal f As Integer, ByVal g As Long, ByVal h As Integer) As Long
```

```
Public Sub Refresh_Alert() 'function to refresh the alert levels at main page.
  Select Case (Alert) 'selects different alert type.
    Case 1 'for low alert.
    MDIForm1.mnu_low.Checked = True
    MDIForm1.mnu_medium.Checked = False
    MDIForm1.mnu_high.Checked = False
    Frm_indicator.Shp_low.FillStyle = 0
    Frm_indicator.Shp_medium.FillStyle = 1
    Frm_indicator.Shp_High.FillStyle = 1
    Frm_indicator.Lbl_Low.Enabled = True
    Frm_indicator.Lbl_Medium.Enabled = False
    Frm_indicator.Lbl_High.Enabled = False

    Case 2 'for medium alert.
    MDIForm1.mnu_medium.Checked = True
    MDIForm1.mnu_low.Checked = False
    MDIForm1.mnu_high.Checked = False
    Frm_indicator.Shp_medium.FillStyle = 0
    Frm_indicator.Shp_low.FillStyle = 1
    Frm_indicator.Shp_High.FillStyle = 1
    Frm_indicator.Lbl_Medium.Enabled = True
    Frm_indicator.Lbl_Low.Enabled = False
    Frm_indicator.Lbl_High.Enabled = False
    Case 3 'for high alert.
    MDIForm1.mnu_high.Checked = True
    MDIForm1.mnu_low.Checked = False
    MDIForm1.mnu_medium.Checked = False
    Frm_indicator.Shp_High.FillStyle = 0
    Frm_indicator.Shp_low.FillStyle = 1
    Frm_indicator.Shp_medium.FillStyle = 1
    Frm_indicator.Lbl_High.Enabled = True
    Frm_indicator.Lbl_Low.Enabled = False
    Frm_indicator.Lbl_Medium.Enabled = False
  End Select
End Sub

Public Sub Refresh_Control() 'function to refresh the control type at main page.
  Select Case (Control) 'selects different control type.
    Case 1 'manual control.
    MDIForm1.mnu_manual.Checked = True
    MDIForm1.mnu_auto.Checked = False
    Frm_indicator.Shp_manual.FillStyle = 0
    Frm_indicator.Shp_auto.FillStyle = 1
    Case 2 'automatic control.
    MDIForm1.mnu_manual.Checked = False
    MDIForm1.mnu_auto.Checked = True
    Frm_indicator.Shp_auto.FillStyle = 0
    Frm_indicator.Shp_manual.FillStyle = 1

  End Select
End Sub

Public Sub Refresh_Test() 'refreshes all content on test page.
  Frm_Test.Frame_Grid.Enabled = False
  Frm_Test.Frame_Serial.Enabled = False
  Frm_Test.Frame_Rotation.Enabled = False
  Frm_Test.lbl_serial_recieve = ""
  Frm_Test.Lbl_Serial_transmit = ""
  Frm_Test.Txt_serial_data = ""
End Sub


Public Sub Unload_All_Child() 'function to unload all child forms.
  Unload Frm_Change_Alert
  Unload Frm_Change_Control
  Unload Frm_Grid
  Unload Frm_indicator
  Unload Frm_Test
End Sub

Public Sub All_Grid_Safe() 'function to display all grids as safe.
Dim i As Integer
For i = 0 To 8
  Frm_indicator.Grid_Plate(i).BackColor = vbGreen
Next i
End Sub
```

APPENDIX C

Bibliography

1. www.avrfreaks.com
2. www.roboticsindia.com
3. www.wikipedia.com
4. www.howstuffworks.com
5. "Prison Break Season 4: Episode 11" American Television Series, Fox Entertainment Channel.
6. Ming-Zher Poh & Yuk Kee Cheung "kasubana The smart flower."
7. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay "The 8051 Microcontroller and Embedded Systems Using Assembly Language" First Edition.