



“Intelligent Classroom Management System”

A Thesis submitted to the Dept. of Electrical & Electronic Engineering, BRAC University in partial fulfillment of the requirements for the Bachelor of Science degree in Electrical & Electronic Engineering

Md. Nahal Islam

Farah Tabassum

Gourab Kumar Sarker

Dhrubashish Sen

December 2013

Declaration

We do hereby declare that the thesis titled “Intelligent Classroom Management System” is submitted to the Department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelor of Science in Electrical and Electronics Engineering. This is our original work and was not submitted elsewhere for the award of any other degree or any other publication.

Date:

Dr. Md. Mosaddequr Rahman
Thesis Supervisor

Mohammed Nahal Islam
Student ID: 10121082

Farah Tabassum
Student ID: 10121089

Gourab Kumar Sarker
Student ID: 11121112

Dhrubashish Sen
Student ID: 10121044

Acknowledgement

We would like to take this opportunity to express our gratitude to Dr. Mosaddequr Rahman for his guidance, and instruction that he has given us from the time we have been doing thesis under his supervision through the completion of our under graduate program. The skills we have been able to learn from him during our Thesis work have benefited us immensely and will continue to do so throughout our future endeavors.

Abstract

Modern day class rooms are equipped with electronic devices that have supporting software to improve and facilitate teaching methods. However, it is often seen that significant class time is wasted on taking attendance, or the class may face interruption due to late entries of students and disturbances such as the manual control of fan and light. Therefore, to overcome those problems a feasible system is created in this thesis project that will have no physical intervention from teachers, students or floor attendance. Thus, the system will facilitate the smooth running of the scheduled classes at our university, and minimize time loss.

Contents

Section		Page
Chapter 1: Introduction		
1.1	Objective	2
1.2	Motivation	2
1.3	Project Overview	3
1.4	Thesis Overview	3
Chapter2: Automatic Attendance System		
2.1	Introduction	5
2.2	Understanding Excel, Visual Basic, Macros and shortcut	8
2.3	Setting up the Attendance Sheet	9
2.4	Freeze Panning	11
2.5	Renaming Tabs	12
2.6	Saving as Macro Enabled File	15
2.7	Removing Ribbon	15
2.8	Setting Up the Macros using the Developer's tab	16
2.9	Programming the Attendance	17
2.10	Programming Automatic Absent	21
2.11	Programming the Password	23
2.12	Programming Total number of Late, Present and Absent	25
Chapter3: Automated Fan		
3.1	Introduction	27
3.2	Outline of the automated fan	27
3.3	Understanding the components	28
3.4	Setting up the automated fan and Coding Using IDE interface	32

Chapter4: Automated Light Control

4.1	Introduction	37
4.2	The Mechanism of the PIR sensor	37
4.3	Components of a PIR sensor	39
4.4	Setup of the Circuit	42
4.5	Coding	47

Chapter5: Conclusion and Future research 49

Reference

Chapter 1

Introduction

1.1 Objective

The Intelligent Classroom management system has three main objectives that will be explained in details in the further chapters. The first objective of the system is to automatically take attendance from students using their ID card. Student ID cards contains barcode data, and if the data is utilized and transmitted to a computer via scanner, the attendance can be recorded with greater efficiency.

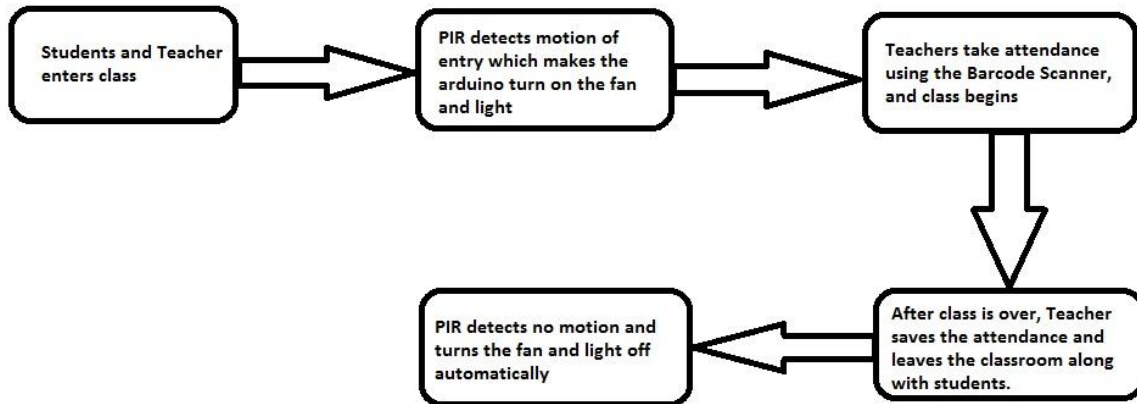
The second objective is to automatically control the fan and lights. The idea is to plant several sensors around the classroom and give a calculated feedback to the response these sensors receive. An example would be having temperature sensors around the class room that would detect the temperature and respond to any change by altering the fan speed. For light control, there will be proximity sensors that would detect student's presence and cause the lights to turn on if students are near that area.

The third objective is to be overall cost effective. While the idea to improve classroom is the priority of this thesis work, importance is made to ensure that this system is affordable to all those who need it. The setup cost may initially be a bit high, but in the long run it is expected that there will be a reduction in both electricity and paper cost.

1.2 Motivation

An ideal classroom is an environment in which teachers are able to focus solely on their lectures and the students are able to concentrate on the information they are being given. Unfortunately, this does not happen in Bangladesh. During class hours, time is usually wasted in many ways such as manually recording student's attendance one after another. Other disruptions also occur throughout class time such as temperature and light variation. These problems cause affected students to wander around the class guessing for the right switch and adjusting it to equilibrate the environment back to satisfying conditions. This causes disturbances for both teachers and other students, and so to eliminate these irritations an intelligent classroom system is created that allows the classroom to become more efficient, and eliminate any human assistance.

1.5 Project Overview



1.4 Thesis Overview

This thesis is comprised of an additional 4 chapters, where Chapter 2 gives a detailed explanation of the setup of the automatic attendance sheet. The equipments required for setup are stated and explained as well as the codes that are needed to be written in the software.

Chapter 3 and 4 discusses the automatic fan control system, and light control system respectively. These chapters will include schematics, diagrams, and the function of each component required for both the systems. Explanations of the setups are also given, as well as the final output results.

Chapter 5 is the concluding chapter, where a discussion about the results of the work and scopes of future research on this topic is present.

Chapter 2

Automatic Attendance System

2.1: Introduction

During class hours, attendance is usually recorded in two ways. The first method would be the teacher calling out the student's name one after another and would record the attendance to those who responds. The other method would have the attendance sheet passed around the class for the students to sign in with their signature. The problem with both of these methods is that it can either be a time consuming and tedious job for the teacher, or it could run the risk of having "false attendances" where a student signs in on behalf of another friend who was not present. There are also other problems that are faced with these methods such as not being able to hear the teacher's voice during name call, or the sheet not being properly passed around during class. To eliminate these hassles an automatic attendance sheet is made which not only significantly reduce the time taken to record attendance and avoid false attendances, but it can also allow the teachers and students to focus more on the lectures than anything else.

The automatic attendance system that is being implemented requires two components which are the Barcode Scanner, and Microsoft Excel (version 2003 or higher). The barcode scanner will be the hardware, which would scan the barcode found in the student's ID cards, and it would feed this information to Excel. The Excel, which is the software component, would take that information from the scanner and give a desired specific output automatically.

There can be many platforms used to create the software portion of the attendance sheet, such as using Java Coding, C++, html, and etc. However, the idea is to have a user interface system that appears user-friendly and easy to use, so that anyone can operate it with minimum struggle. Hence, Microsoft Excel was used as the main platform, as most users are somewhat familiar with the basics of how to operate it.

In Microsoft Excel, the coding language used is Visual Basic (VBA). Emphasis will be made in this language to make the attendance sheet automatic and simple for the users. The programming would consist of the following features:

1. A "sign in" function that would prompt the student's barcode ID as input, match it with the ID in its database and then automatically record and display the date and time of input in the student's respective row.
2. An "Absent" function that would allow a teacher to mark all those who are absent within a click of a button.
3. A function that would calculate the total number of absents, total number of late, and total number of Present automatically.
4. A "password" function to protect the file from unauthorized access (optional).

The outline of how the program would function is shown in the next page.

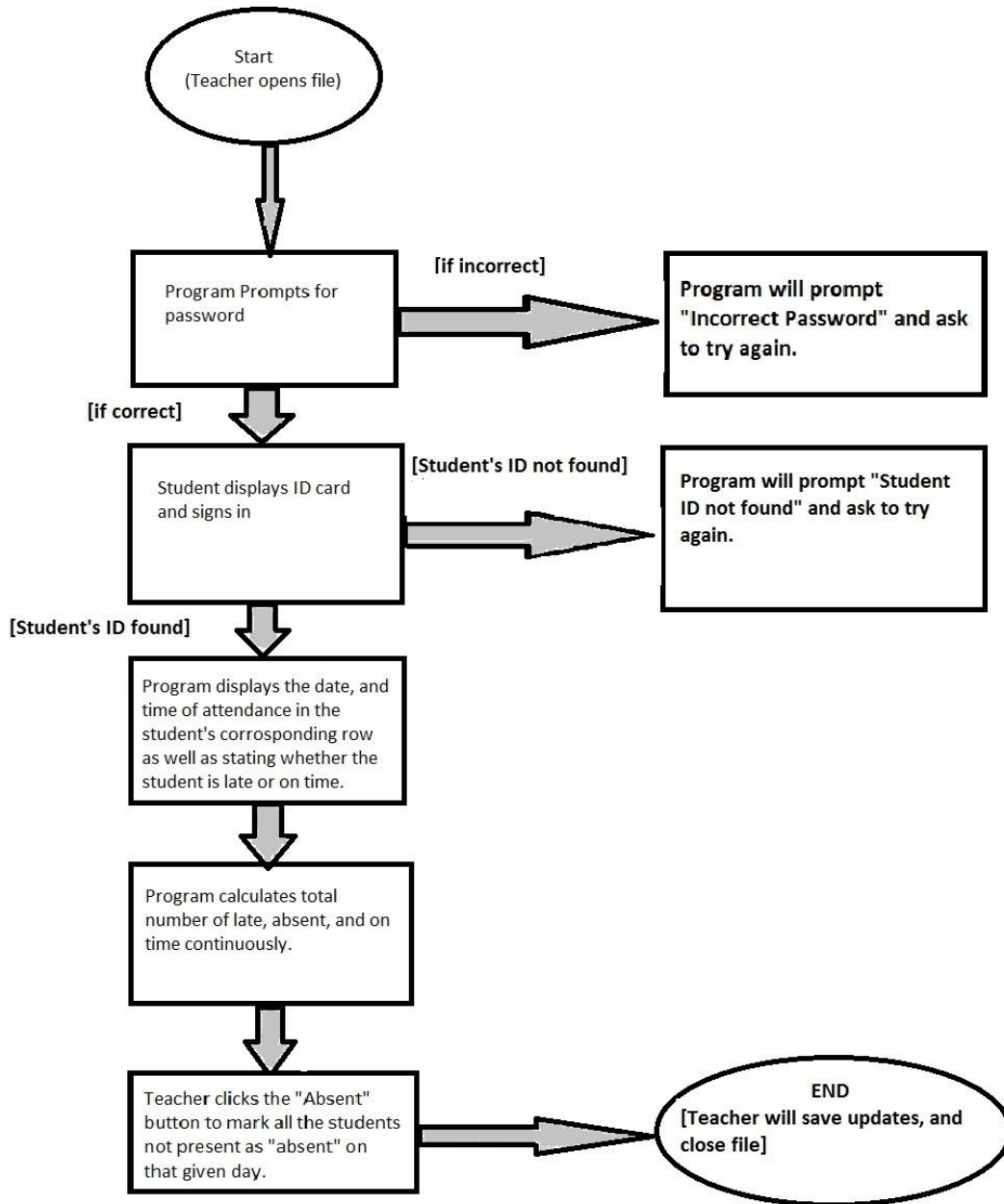


Figure 2.1: Flow diagram of Attendance system

2.2: Understanding Excel, Visual Basic, Macros and shortcuts

Before getting into details about the coding used in the attendance sheet, it is best for one to understand a few fundamentals of Excel before proceeding.

An Excel spreadsheet is a grid of cells arranged in numbered rows, and lettered columns. The x-axis would be numbered from 1 to 5064, and the y axis would be labeled from A to ZZZ. This is very important information when setting up the attendance sheet, as the program will be depending on this grid system when allocating a student's row and giving outputs at specific cells.

Visual Basic (VBA) is the language this attendance sheet will be programmed with. For every program (also called "Macro") created, it can run either automatically, or after being prompted to do. The VBA language is similar to languages like Java, and Matlab as they have functions like "loop", "if", "or", and "and" as well as many other function such as creating arrays and calling functions. However, the key terms when using these functions are different which will be explained later on.

Finally, shortcuts will be needed when programming in the "Developer's tab". The developer's tab will be the area where all the macros would be created, or modified. Rather than manually selecting it over and over again while programming, a key feature to remember is the shortcuts "Alt+F11" and "Alt+F8".

"Alt+F11" would automatically take the user to the developer's tab, and "Alt+F8" would allow them to select the macros to run. These shortcuts become extremely helpful the more the software is being coded.

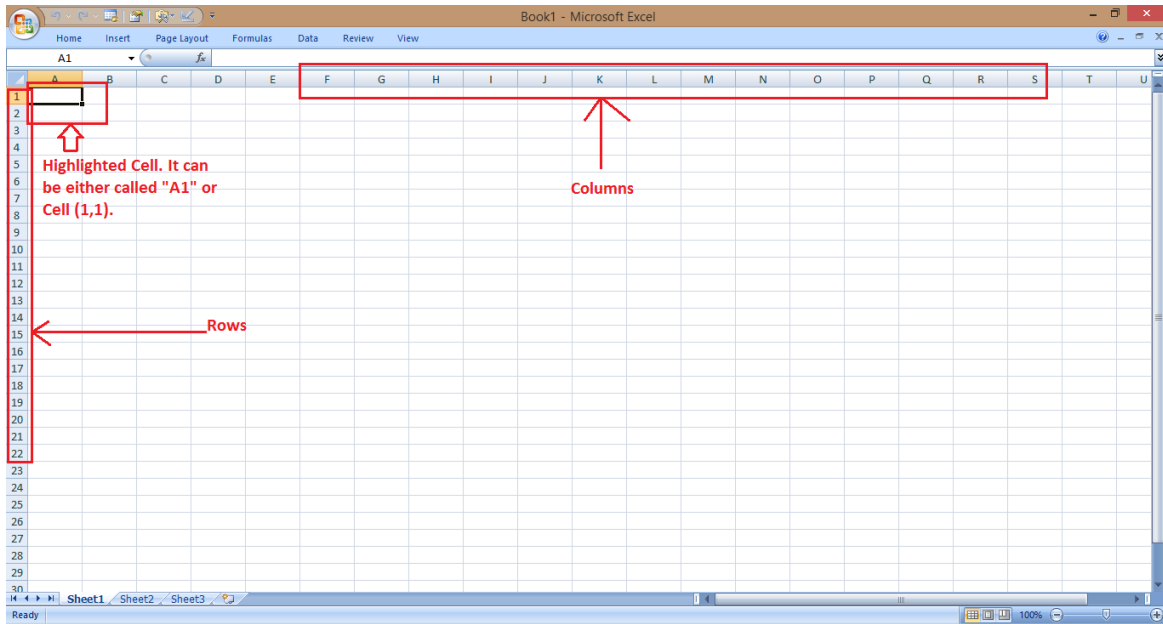
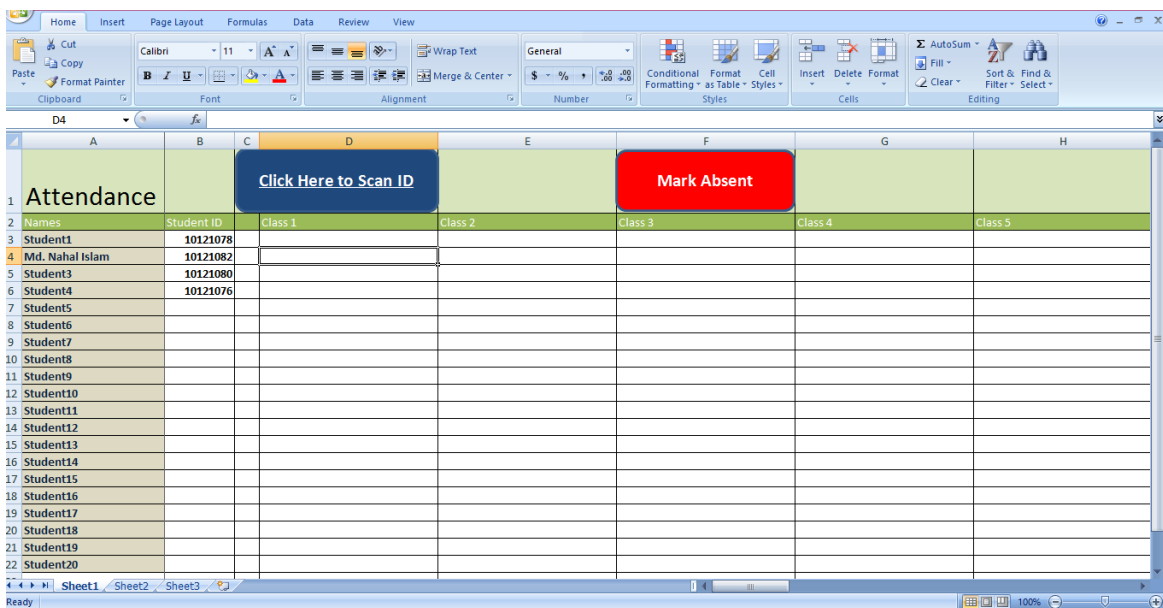


Figure 2.2: Basic Structure of an Excel Spreadsheet.

2.3: Setting Up the Attendance Sheet

There are many ways to set up the attendance sheet, and so there are no specific rules or conditions on how to make one. However, the one thing that must be noted is the cell's position when designing the sheet as they play a crucial role on how the macro would be programmed. The diagram below shows the attendance sheet being used for this thesis.



It can be seen that the Student's name occupy Column A, their student ID number in column B, and the class number would continue from column D till AB (or however long the class number is).

Two macros will be assigned to the two buttons found in the 1st row, which are the “Click here to Scan ID” and the “Mark Absent”. These are what the teacher or student would be mainly clicking on when operating the attendance sheet.

To set up these buttons click on “insert” and then select shapes, and select the shape you desire.

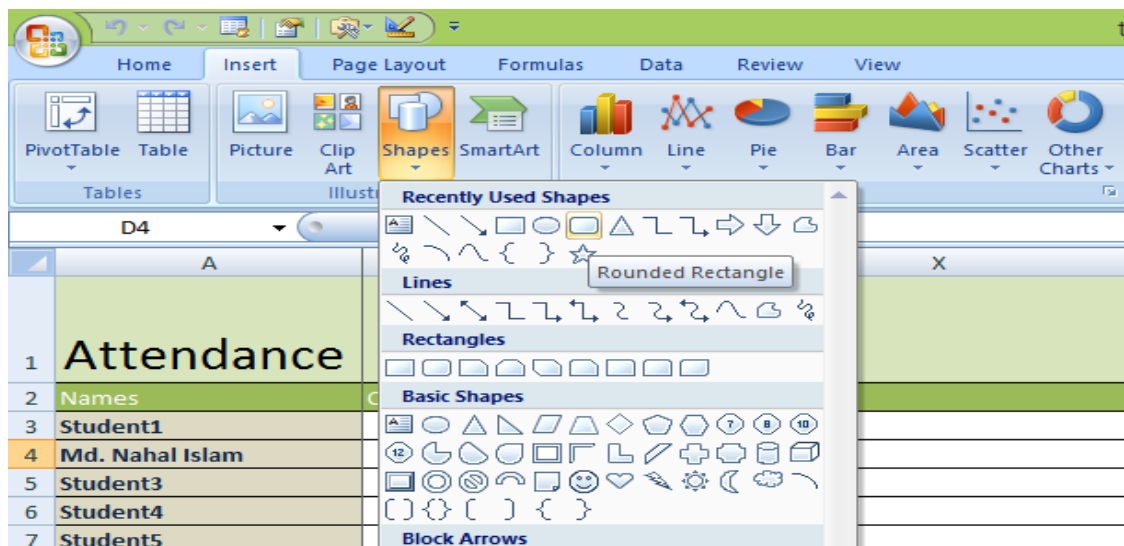
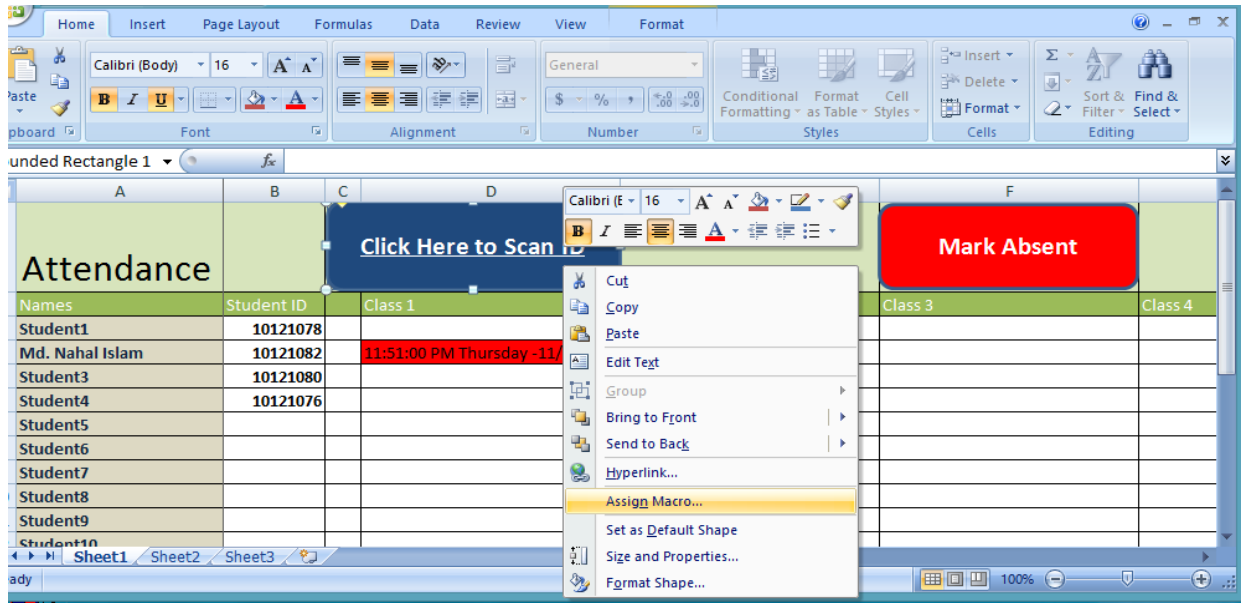


Figure 2.3: Selecting Shapes for Macros

Finally, after customizing the buttons, right click the given shape and select “assign macro” to give the button its function that it would perform (this will be done after the macros are completed)



2.4 Freeze Panning

Freeze Panning is a technique used that Isolates specific rows or columns. This is helpful for student or teachers as they can move across the screen without having to move back to confirm which student missed which class. An example is shown below. To Freeze Panel click on View, Select Freeze Panels and then select “Freeze First Column”. The attendance can now be scrolled across the screen and column A would remain fixed as it is being scrolled.

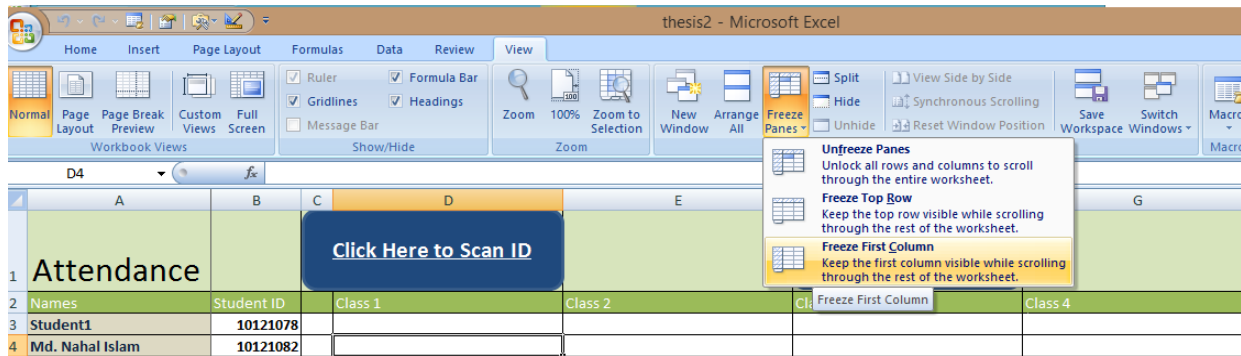


Figure 2.4: Freezing Panels

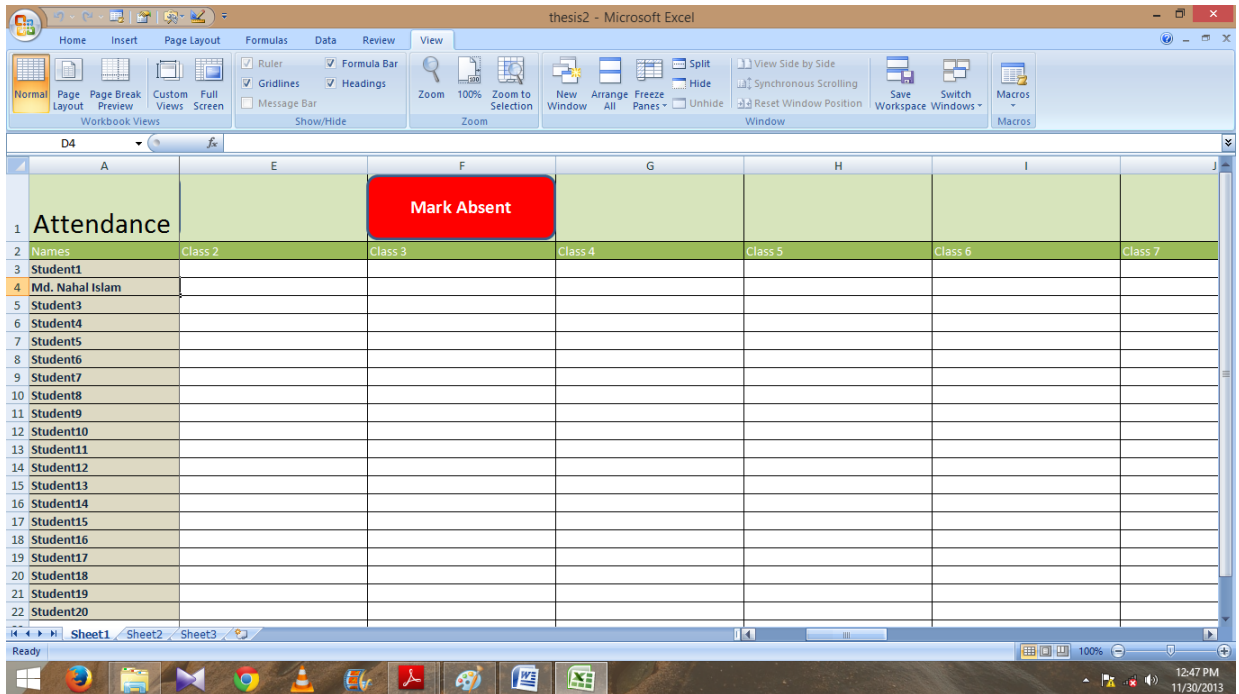


Figure 2.5: Sliding across Panels, with Column A remaining fixed.

2.5 Renaming Tabs

Sheets can be renamed for the convenience of the teacher if he or she teaches multiple classes. To do so simply right-click the tab called “Sheet1” found near the bottom of the tab and select “rename”. It is important to remember what you named your sheet as it plays a crucial role when programming the macro. For example if the tab is written “EEE207” then when writing the line for that specific sheet, instead of writing “*ThisWorkbook.Sheets(“Sheet1”)*” it would be written as “*ThisWorkbook.Sheets(“EEE207”)*”.

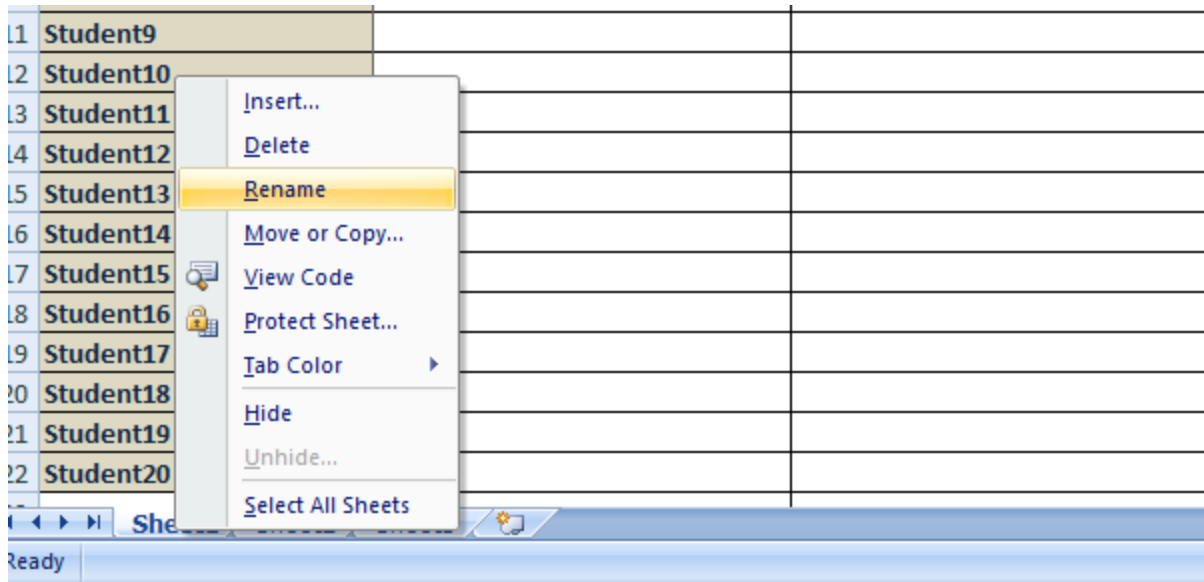


Figure 2.6: Renaming Tabs

Protecting Worksheet

If the teacher wishes the attendance sheet to be viewed by others, but not allow any changes to be made than he or she can protect the worksheet by clicking on the “Review” Tab found on the ribbon and select “Protect Workbook”. This will use a password system, and hence no changes can be made till the password is verified.

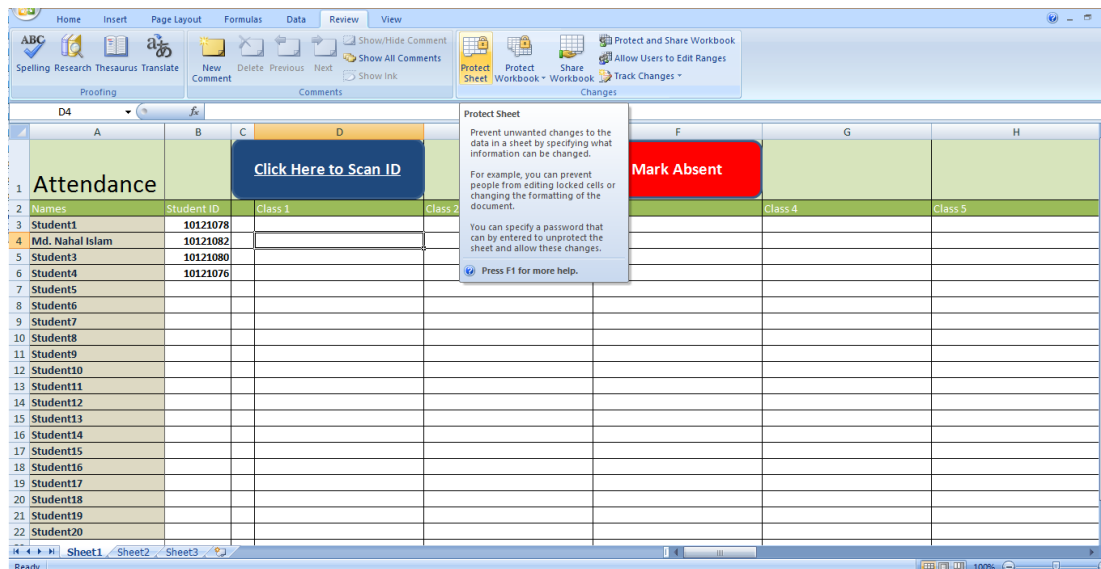
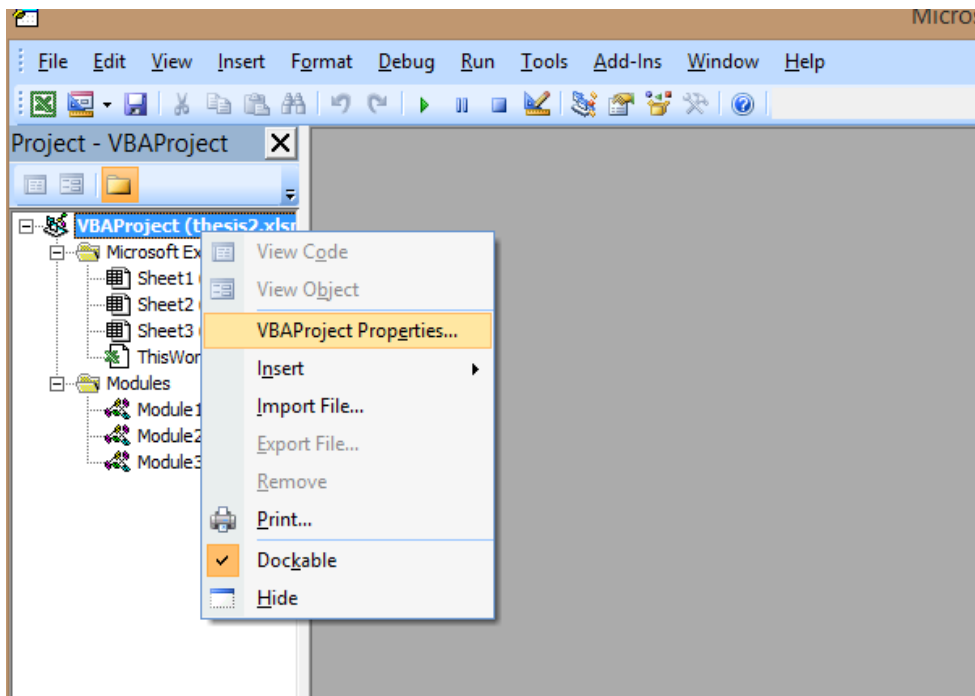


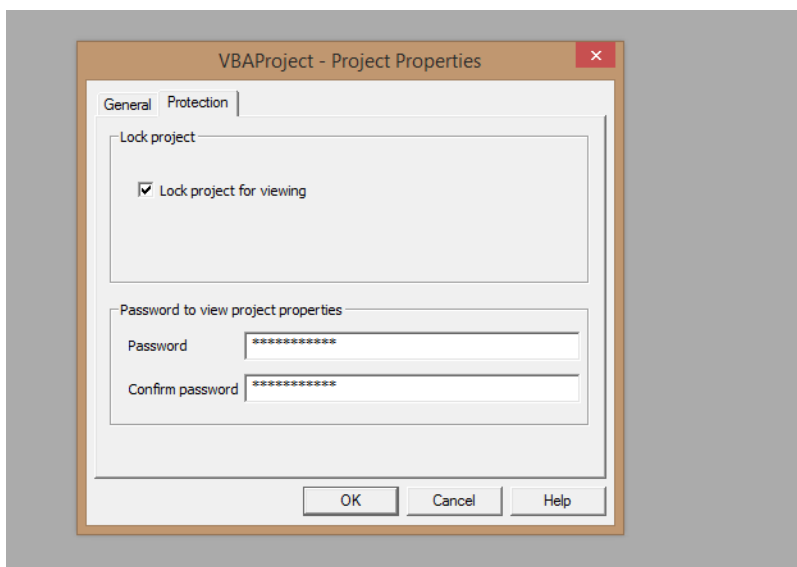
Figure 2.7: Protecting Spreadsheet

Protecting Macro

To protect the macro so that no unauthorized user can change the codes in it, it can be protected by opening the developer's tab (or press Ctrl+F11) and then right-click VBA project and select "VBA Project Properties"

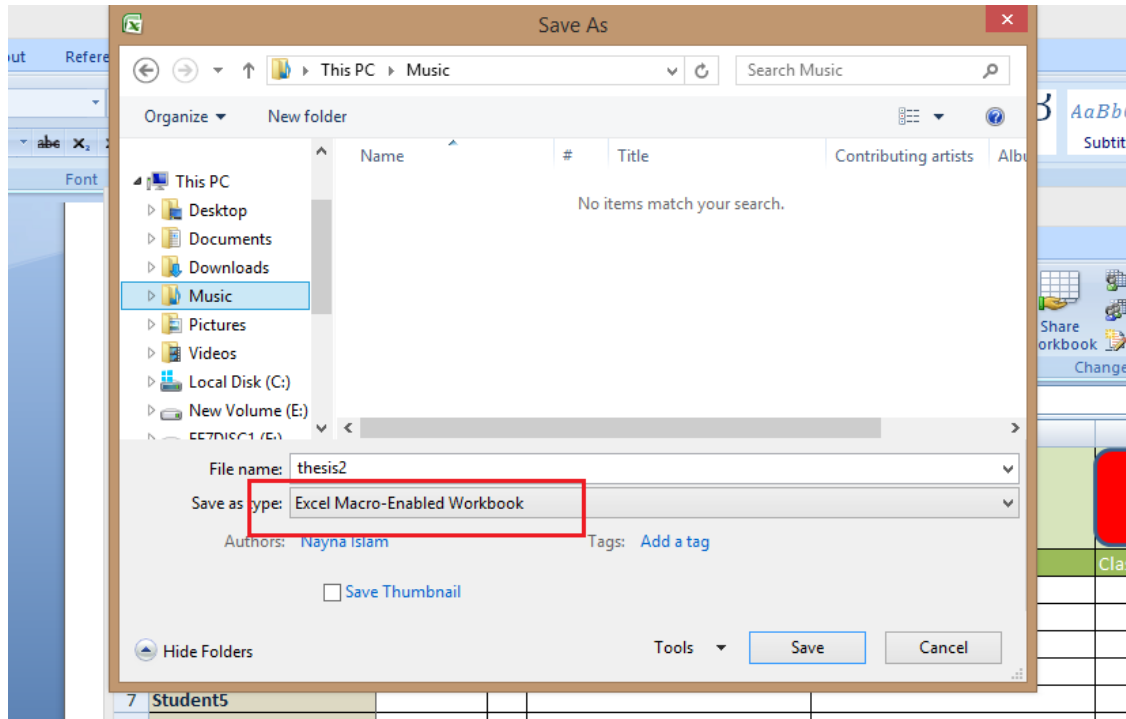


Here a window pops up, and we select protection and create a password for the protected Macro



2.6 Saving as a Macro Enabled File

This attendance Sheet runs on Macros so when saving the file, select “Macro Enabled workbook” otherwise the Programs will not work.



2.7 Removing Ribbon

Finally, to remove any unwanted clusters on the top screen of the Excel Spread Sheet simple right-click the top part of the sheet, and select “Remove Ribbon”

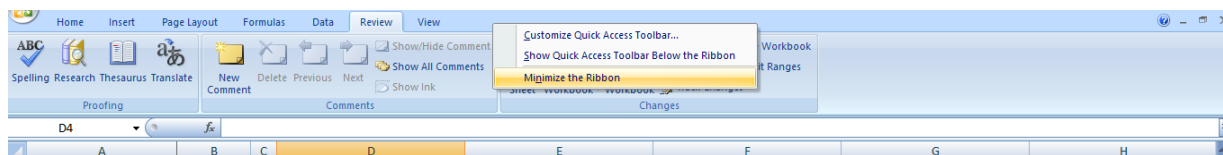


Figure 2.9: Removing ribbon

This would make the final attendance sheet look similar to the following.

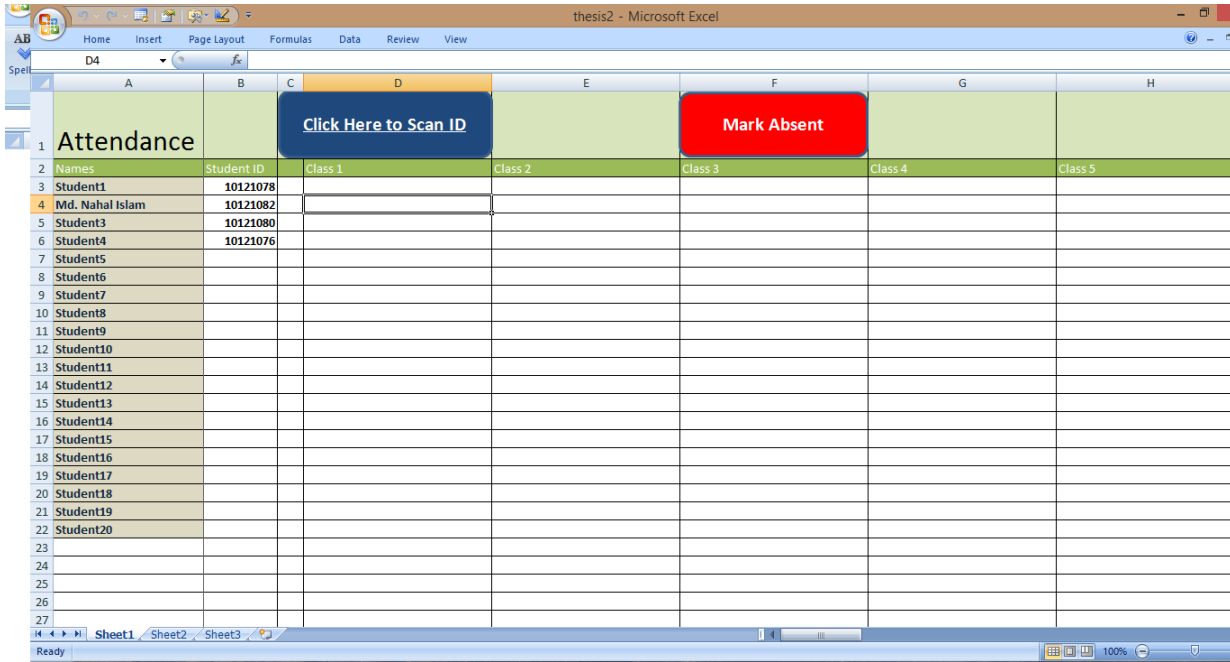


Figure 2.10: Final Look of attendance sheet

The focus can now be made on the Attendance Sheet alone and not anywhere else for users when using the sheet.

2.8: Setting Up the Macros using the Developer's tab

The developer's tab will be the area where all the coding will be programmed; as mentioned earlier the tab can be accessed by pressing "Alt+F11". When the Developer's Tab screen shows up, right click "Sheet1" on the left side of the screen, highlight "Insert" and then select "New Module". We use a new module for every program we create.

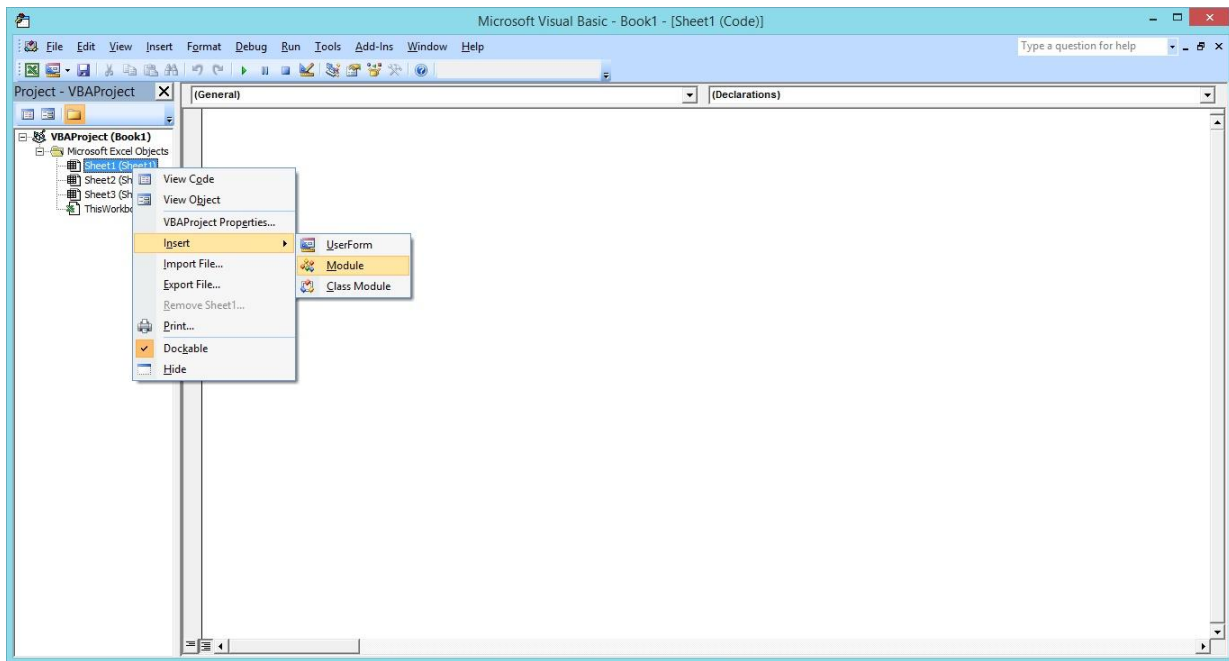


Figure 2.10: Accessing the developer’s Tab, and inserting new module.

2.9 Programming the Attendance

The main function of the Automatic Attendance sheet is to be able to take the input from the student’s ID card, scan and match that input with the stored data it contains and then give an output to the student’s corresponding row.

To do this we have to create an Array that will store all the number of student’s ID. This can be done by typing the following code in the module:

Dim Students(5) as String

The “(5)” specifies the number of students present in the class. For this example we will only use 5. We then store the Student’s ID into these variables which is shown in the figure below

```
Dim Student(5) As String

Student(0) = "0010121082"

Student(1) = "0010121080"

Student(2) = "0010121078"

Student(3) = "0010121076"

Student(4) = "0010121074"

Dim StudentID As String
Dim x As Integer

x = 0

StudentID = InputBox("Please scan your ID card")
```

It is important that when storing 5 Strings, the String will start from 0-4 when being called upon, and not 1-5. The program would then prompt the Students to scan their ID cards using the “input Box” function and their inputs would be stored onto the “StudentID” string.

The “x” variable will be initialized and act as a counter that will be used to go through the array, which will be made much more clear in the next figure.


```

Do While Student(x) <> StudentID
x = x + 1
If Student(x) = Student(5) Then
x = 0
MsgBox ("Student not found, please try again")

StudentID = InputBox("Give ID")

Else

End If

Loop

Dim c As Integer

c = 4
Dim y As Date

y = Format(Time, "hh:mm")

```

The “Do While Student(x) <> StudentID” means that if the user’s input “(StudentID)” does not match with the given String “Student(x)”, then it will go to the next string by incrementing the value of x. So if the ID is not matched in Student(1) it would then go to Student(2) and so on and so forth. This will go on until Student(x) equals to Student(5). If this occurs, then it means it had scanned through the entire array, and did not find the ID that the user has given. In this situation the program will tell the student “Student ID not found, please try again”, and prompt a new Input box.

If the input matches with one of the strings in the array, it would end the loop, and follow the next lines of code which would be to initialize another counter labeled “c” and store the current time in a variable name “y”. The reason why we are storing the current time is to understand whether the student is late or on time. So the “y” variable would then be compared with the time stored on the program, and specific outputs would be given. If the value of y is less or greater than the time programmed, it will mention so in the output, and give different results.

The “c” variable is initialized at c=4 for a particular reason. From the attendance sheet, it can be seen that the class attendance is recorded starting from the “D” column till the “AA” column. Using the cell property in VBA, the “D” column can also be written as “4”, so if we wish to locate a cell such as “D5” we would have to write it as Cell (5, 4). This is important when programming because we want to ensure that the attendance will be taken from one column to the next by incrementing the “c” variable as we go on, hence we use the “c” as a counter and initialize it as an integer.

```
If StudentID = Student(1) Then
Do While ThisWorkbook.Sheets("Sheet1").Cells(5, c).Value <> ""
c = c + 1
Loop

If y > #8:56:00 AM# Then

ThisWorkbook.Sheets("Sheet1").Cells(5, c).Interior.Color = 255
ThisWorkbook.Sheets("Sheet1").Cells(5, c).Value = y & " " & Format(Date, "dddd -mm/dd/yyyy")
MsgBox ("Attendance recorded. You were late")

Else

ThisWorkbook.Sheets("Sheet1").Cells(5, c).Value = y & " " & Format(Date, "dddd -mm/dd/yyyy")
MsgBox ("Attendance Recorded. Thank you")

End If

Else

End If
```

The first thing this set of codes does is confirm whether the Student’s ID card matches with the String stored in Student(1). If it does not, then it would move on to the next Student, otherwise it would continue the next line of code.

The line:

```
Do While ThisWorkbook.Sheets("Sheet1").Cells(5, c).Value <> ""
```

```
c = c + 1
```

```
19
```

Loop

Means that it will check the 5th row and see if it is not an empty cell, this is done by comparing it with an "" sign which indicates empty. If it is not empty, then it means a value is already stored in that particular cell, and so it would continue jumping from one column to the next till it finds an empty cell. The reason why there is a "5" in cell(5,c) is because we assigned Student(1)'s row in row 5, so depending on which row the student is in, the cell's coordinate would also be changed corresponding to that.

After finding an empty cell, the program would then compare the time the student gave its input (y) with the programmed time. If the student's input is greater, the program would change the color of the cell

using "Interior.Color=255" and then fill the cell with the "y" variable. It would then give a message box stating the student being late. If the student is not late then it would skip this line, and start from the "Else" command, and fill the cell with the "y" variable without changing the cell's color.

This process can be repeated for any number of students with only one small change needed to be made, which is to understand what row the student is in when assigning their codes.

2.10 Programming Automatic Absent

At the end of the class, when students present have signed in to the attendance sheet, the teacher would then mark those who are absent. Rather than manually scanning for each individual student, and typing "absent" in their corresponding cell, we can be able to automatically mark them absent by simply clicking on the "Mark Absent" Button, and let the program do that work instantaneously.

To do this we go back to the developers tab and open up another module in Sheet1. From there we would initialize three variables as integers “x”, “y” and “z”, and then initialize another variable “a” that would store the current time of input.

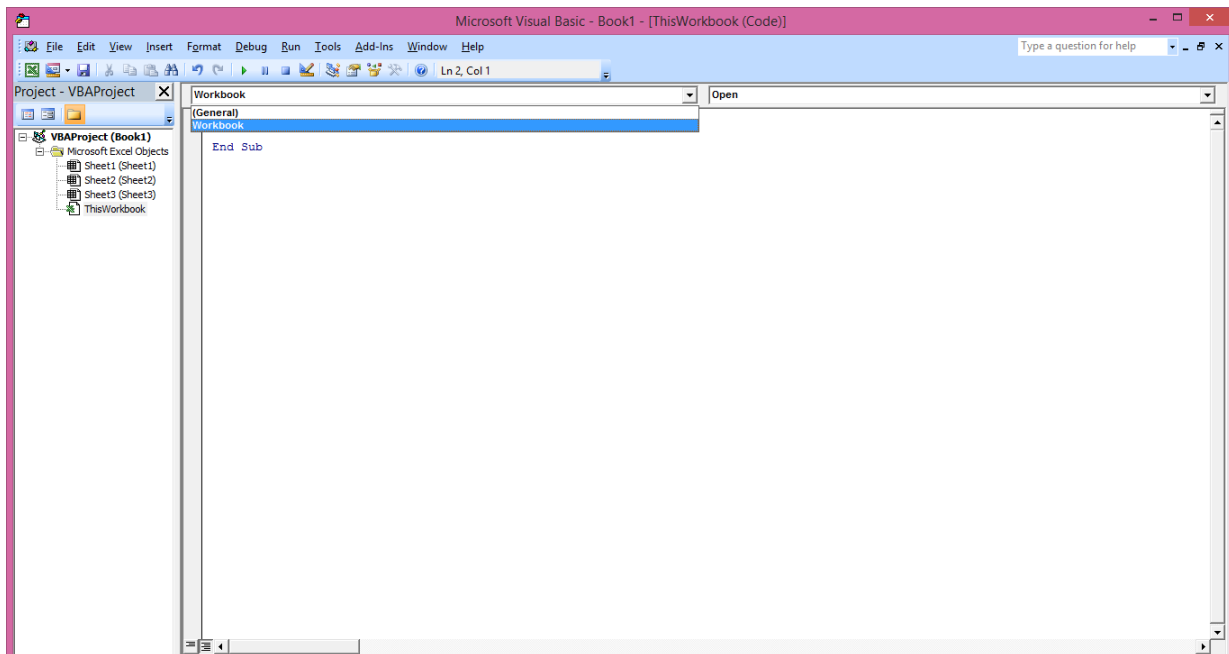
The program would then compare the time in “a” with the time stored, and determine whether it would continue further action. If the time is less, no action would be made. If the time is accepted, then it would begin scanning for any marked absent per column. If there is an “Absent” found, then it would move on to the next column by incrementing the “y” variable, and restart the “x” value to “1”. This is because the given row has already been marked previously. If a column is found that contains no “Absent” in any cell then the program would start filling all the empty cells with a red color using the “Interior.Color” function, and marking an “Absent” in that cell. The “z” variable is used as a counter to move down the column when the selected column is being marked.

```
Sub Absent()  
  
Dim a As Date  
a = Format(Time, "hh:mm")  
  
If a > #2:32:00 AM# Then  
Dim x, y, z As Integer  
x = 1  
y = 4  
z = 1  
For Row = x To 20  
  
If ThisWorkbook.Sheets("Sheet1").Cells(Row, y).Value = "Absent" Then  
y = y + 1  
x = 1  
End If  
  
Next  
  
Do While z < 20  
If ThisWorkbook.Sheets("Sheet1").Cells(z, y).Value = "" Then  
  
ThisWorkbook.Sheets("Sheet1").Cells(z, y).Value = "Absent"  
ThisWorkbook.Sheets("Sheet1").Cells(z, y).Interior.Color = "700"  
  
z = z + 1  
Else  
z = z + 1  
End If  
  
Loop  
  
End If  
  
End Sub
```

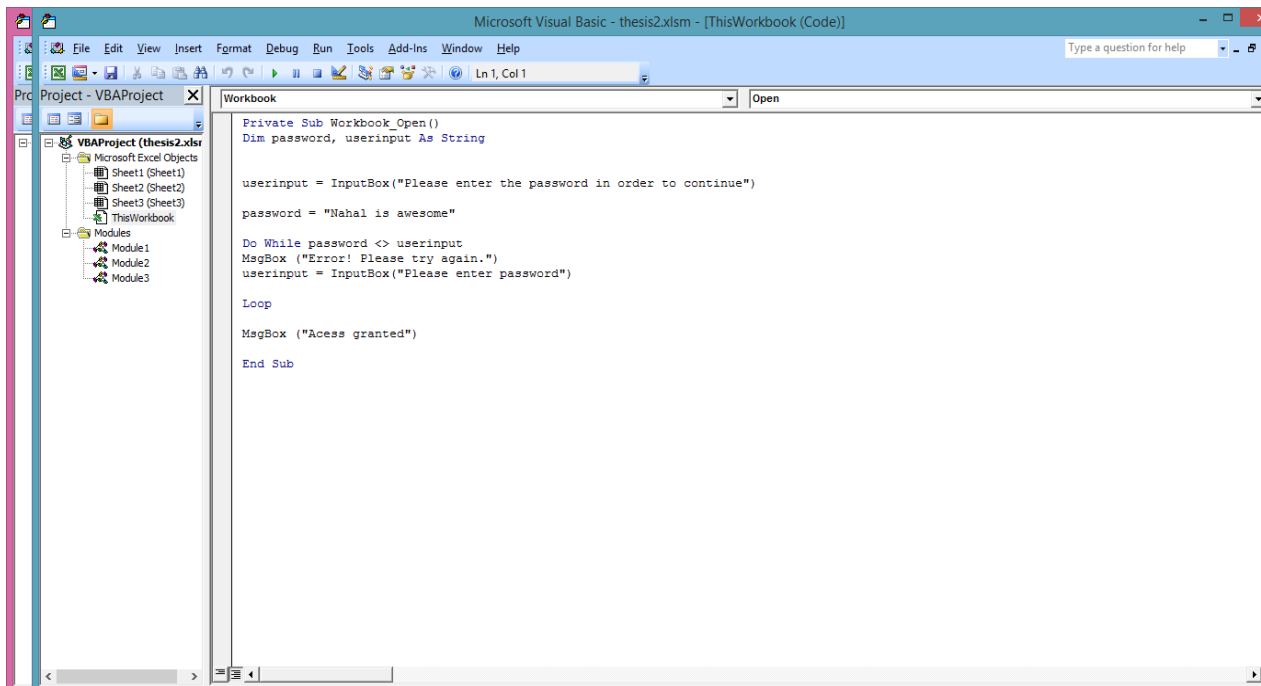
2.11 Programming the Password

The idea behind the password system is to prevent unauthorized viewers from accessing or viewing the file. If the user wishes to allow others to view the data without the privilege to manipulate values then an alternate method can be used which will be explained later. This function is optional, and depends on teacher's preference if he or she wishes to use it.

To start press "Alt+F11" to open the developer's tab and double click "Thisworkbook" which is found underneath "Sheet3". From there, click on "(General)" which is found on the top area of the tab, and then select "Workbook".



In the module, Microsoft Excel would automatically write the lines "Private Sub Workbook_Open ()" and "End Sub". The password program would be written in-between these two statements, which is given in the next diagram.



“Dim” is a term Visual basic uses to initialize character(s) to act as specific function, such as an Integer, a String, and so on. So we type

Dim password, userinput As String

This means that Visual Basic will take the terms “Password” and “Userinput”, and initialize them as strings which we will use during this program.

For the next 2 lines:

userinput = InputBox("Please enter the password in order to continue")

password = "Nahal is awesome"

The term “InputBox” is a function that will prompt the user a message when opening the file, and it will ask to input a value. The string “userinput” would store the value that the user would write.

The “password” would equal to whatever the original user wants to save it as.

The following lines after that would check whether the user’s input matches with password stored, if it is not the same, then a message box “Error! Please try again” would appear and would prompt the user to try again. Otherwise a message box saying “Access granted” would appear, and the user would finally have access to the file.

Calculating Total Attendance

At the end of the semester, the teacher may wish to see the total attendance of a student during the semester. Instead of manually counting the number of Present, Late, and Absents a program is made that would calculate it all automatically. The program is as followed:

```
Sub Total()  
  
Dim a, x, y, z As Integer  
a = 2  
z = 0  
x = 0  
y = 0  
  
Do While a <> 20  
a = a + 1  
  
If ThisWorkbook.Sheets("Sheet1").Cells(4, a).Value = "Absent" Then  
x = x + 1  
ElseIf ThisWorkbook.Sheets("Sheet1").Cells(4, a).Interior.Color = 255 Then  
y = y + 1  
ElseIf ThisWorkbook.Sheets("Sheet1").Cells(4, a).Value <> "" Then  
z = z + 1  
Else  
End If  
  
Loop  
  
ThisWorkbook.Sheets("Sheet1").Cells(4, 28).Value = x  
ThisWorkbook.Sheets("Sheet1").Cells(4, 29).Value = y  
ThisWorkbook.Sheets("Sheet1").Cells(4, 30).Value = z  
  
End Sub
```

Four variables would be declared as integer. The “a” variable would be used as a way to move from cell to cell while scanning. The “x”, “y”, and “z” variable will increment whenever they spot a “Present”, “Late”, and “Absent” respectively. Finally, after scanning till the end of the attendance sheet it would display the total number of present, absent, and late in their corresponding cells.

Chapter 3

Automated Fan Control

3.1 Introduction

The automated fan is the second part of the Intelligent Management System. The idea behind the system is to be able to detect the surrounding temperature in the classroom, and then adjust the fan speed in correspondence with the temperature it is detecting. While this may seem like any other thermostat that is available in the market, the automated fan system created here will be much more cost effective than any other thermostat around (\$20 for the completion of the setup of the system compared to \$200 for a thermostat available in the market). This chapter will give an in depth analysis of the components being used and how they operate together to achieve this output. Diagrams of the setup and flow charts are also given to give a better clarity of how the system is functioning. The software used and the codes involved in the system are also given at the end of the chapter as well as the explanation behind the codes

3.2 Outline of the automated fan

The following figure shows the basic block diagram of how the fan will operate

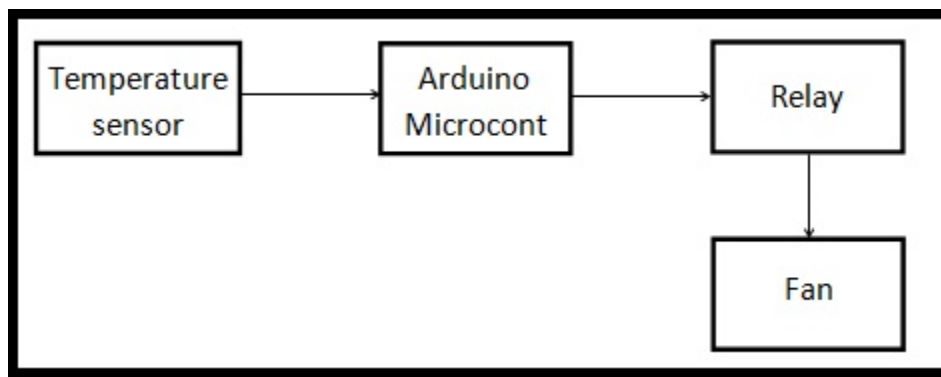


Figure 3.1 Block Diagram of Automated Fan setup

The Temperature sensor that is used in the fan control is the LM35. Whenever the LM35 detects temperature, it generates a specific voltage as output in relation to the detected temperature. We can then calibrate this voltage in Celsius or Fahrenheit. The Arduino microcontroller will act as the “brain” of the control system; it would receive the voltage from the LM35, and then give specific instructions to the fan so that it can function the way it is commanded to be. The relay component acts as a voltage converter, and it is used to generate higher voltage so that the Arduino can supply sufficient voltage to the fan.

3.3 Understanding the components

LM35 (Temperature Sensor)

The LM35 is an IC temperature sensor with its output voltage proportional to the input temperature (in C). The sensor circuitry of the LM35 that is used to detect the temperature is sealed. While temperature can be measured using a thermistor, an LM35 is used instead because of its ability to measure temperature with a greater degree of accuracy. It also possesses low self heating and does not cause more than 0.1 C temperature rise or fall when operating.

The operating temperature of the LM35 ranges from -55°C to 150°C . One feature of the LM35 that must be noted is that its output voltage can vary by 10mV in response to every C rise/fall. *i.e.*, its scale factor is 0.01V/ C. For example, if the temperature of the surrounding environment was 23 C, the output voltage is expected to be 23mV, but due to this variation from the LM35 the voltage detected could be anywhere from 13mV to 33mV. Therefore the variation of the output voltage must be noted when the system is being set up.

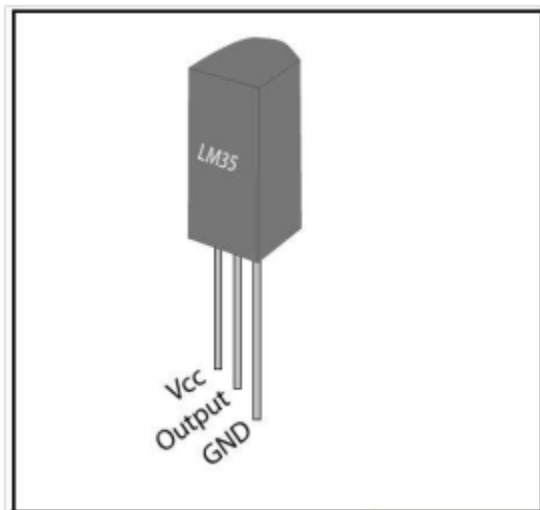


Figure 3.2 Diagram of the LM35

Pin 1 is the supply voltage which is labeled as “Vcc”, and 5V will be supplied to it during setup. Pin 2 is the Output Voltage where the Voltage would be 6V, and finally Pin 3 is the ground pin for the LM35.

Arduino

The **Arduino** is one of the main components of the automated fan system. It is a single-board microcontroller that uses electronics to make projects more accessible. The hardware has an “open-source” hardware board which is made with an 8-bit Atmel AVR microcontroller. The Arduino uses a programming language called the “Integrated Development Environment” or in short “IDE”. The IDE, which this system will use, is a program written in C++ and it allows users to create instructions for the microcontroller. These instructions are then sent using a boot loader which would execute the given commands to the microcontroller and then coordinate the fan on how fast it would rotate.

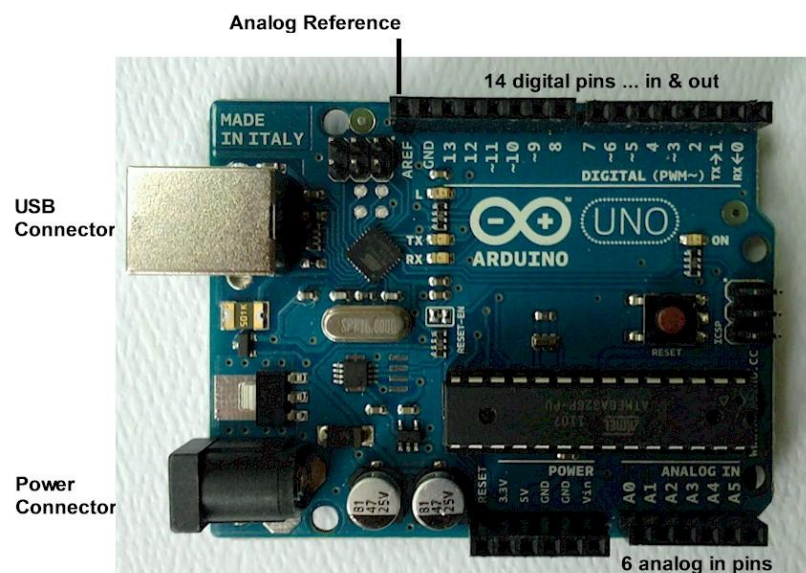


Figure 3.3 Hardware portion of the Arduino Uno



```
Arduino IDE - Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards.
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}
```

Figure 3.4 Software Portion of the Arduino Uno using the IDE interface

Relay



Figure 3.5 Actual Model of a relay

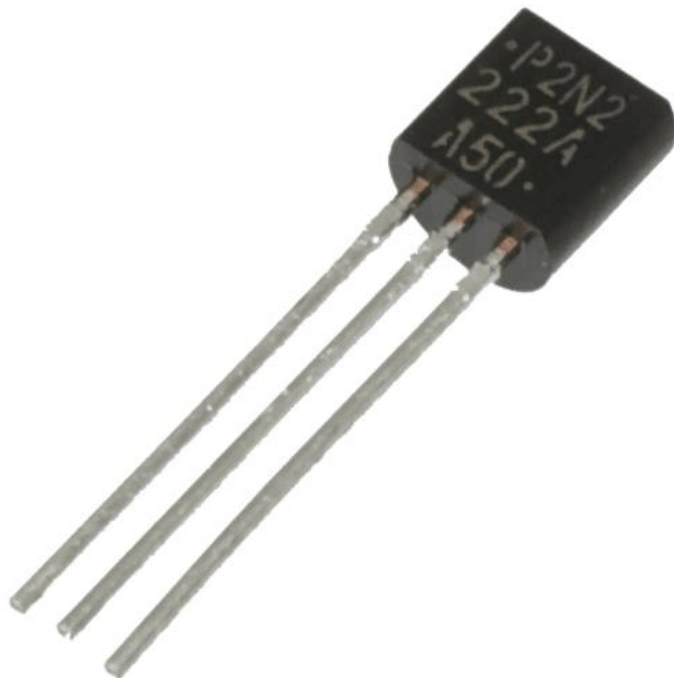
Relays are switches that open and close circuits electromechanically or electronically. It controls one electrical circuit by opening and closing contacts in another circuit. When a relay contact is

normally open (NO), there is an open contact when the relay is not energized. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized. Relays are generally used to switch smaller currents in a control circuit and do not usually control power consuming devices except for small motors and Solenoids that draw low amps. Moreover, relays can "control" larger voltages and amperes by having an amplifying effect.

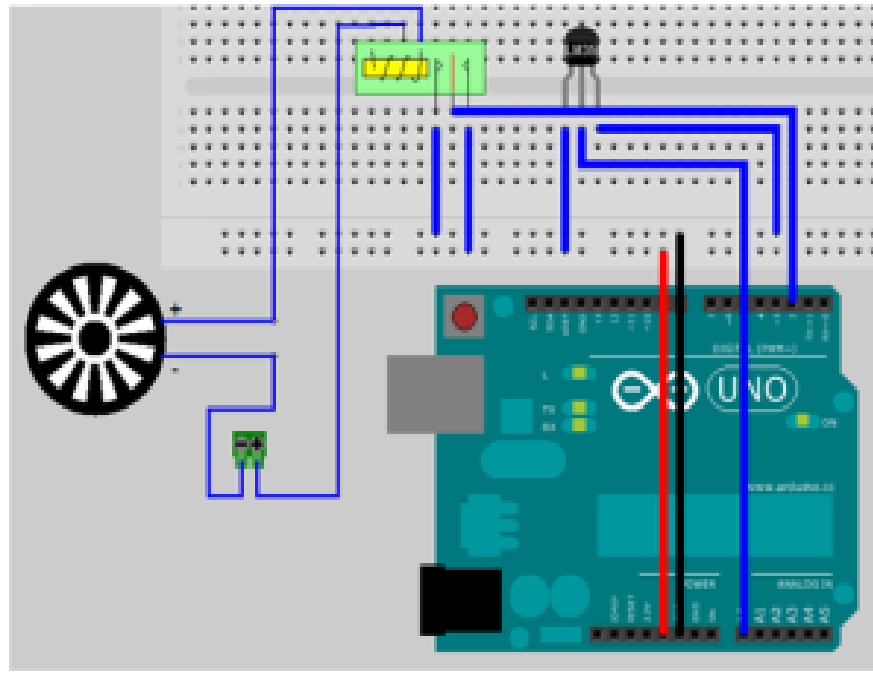
Protective relays can prevent equipment damage by detecting electrical abnormalities, including over current, undercurrent, overloads and reverse currents. In addition, relays are also widely used to switch starting coils, heating elements, pilot lights and audible alarms

NPN transistor

Transistors are used to control the coiled current of the relay. For the fan, we are using an NPN transistor because when NPN transistor is used then the relay will be energized and would turn on, when control voltage V_{in} is equal to +12. The relay would turn off if the V_{in} is 0V. The opposite would occur if the PNP used.



3.4 (a) Setting up the automated fan



The temperature sensor, LM35 is taken which has 3 pins. Two of these pins, Pin 1 and Pin 3 are for the 5V and ground supply respectively, and the middle pin is used as output. Caution must be taken when connecting the LM35 as the sensor would get very hot or even destroyed if connected improperly. The Arduino Uno is then taken and is supplied with 5V and 3.3V from the breadboard. The two grounds of the microcontroller are then joined together on both sides of the breadboard. The Arduino and the data or output voltage pin is hooked to any analog pin on the Arduino. The figure below gives the final look of the set up.

3.4 (b) Coding Using the IDE interface

The following code should be written in the IDE program, and then finally compiled to the Arduino for it to store the commands that is programmed.

```
//declare variabes
float tempC;
int tempPin=0;
int ledPin=13;
int fan1=5; //fan1 is connected to pin5
//Write setup function
void setup()
{
Serial.begin(9600); //open serial port to communicate with the temp sensor
pinMode(ledPin,OUTPUT);
pinMode(fan1,OUTPUT);
}
//Writing down the loop that will control what we want the arduino to do with the sensor read
out
void loop()
{

tempC=analogRead(tempPin);//taking the temp Pin reading and setting it=tempC variable
tempC=(5.0*tempC*100.0)/1024; //will convert analog input to a temp in C
Serial.print((byte)tempC); //will output the converted temp
if( tempC>26)
{
digitalWrite(ledPin,HIGH);
digitalWrite(fan1,HIGH);
}
else
{
digitalWrite(ledPin,LOW);
digitalWrite(fan1,LOW);
```

```
}  
delay(1000);  
}
```

The variables that are being used for the code are declared initially. Only 2 variables are needed for the temperature sensor. The first variable is a float data type and it will be labeled as “temp C”. The other variable is an “int” data type and it will be named as “tempPin” which is set to equal 0 [the reason is that temperature sensor is plugged to analog pin 0]. The setup function of the Arduino starts with the void setup. For this, a serial port is opened which allows the Arduino to communicate via usb to the computer so that the temperature recorded by the sensor can be seen. To see that temperature, “Serial.begin[]” is written followed by using a loop. An equation is then created to convert the sensor’s output to the Arduino into an actual temperature that would be analyzed from the computer. This is the conversion of analog input to a temperature in Celsius. The equation created is “tempC=(5.0*tempC*100.0)/1024.0”. To know the reading of the sensor, “Serial.println((byte))tempC.TempC” is written because everything is converted and stored in the variable in temp C.

A delay is given to the program, otherwise readings will be given on the temperature every time the sensor switches. The Arduino is then connected to the computer so that the code can be uploaded.

For connecting an LED light (which acts as an indicator for LM35), a new variable is created and it is made an integer and is labeled as “int led Pin=13”[since it is plugged in the digital Pin 13]. The LED is connected with Pin 13 because it has already registered in line with that pin. Pin 13 is set as a output by writing “InMode(ledPin,output)”. An “if” statement is used for temperature range. For example, if the temperature is greater than 15 the the light would be turned on, else it would remain off. To check the code the temperature of the sensor, a finger can be used to hold the sensor to increase its temperature.

A PC fan is then taken and connected to the Arduino .The ground wire of the fan is connected to the ground of the bread board and the power of the fan is connected to the digital 29

Pin5 of the Arduino. This is a 12V fan that will run at 5V. For the fan to work, (Fan1, High) is written inside the loop function, and in the *else* statement “low” is written. The rest of the codes remain the same .The sensor is then heated and it is seen that the fan and led are working together. Using 5V the fan would run, but very slowly so a relay and an NPN transistor is used for the fan to run at higher voltage. A transistor consists of 3 pins. The leftmost pin is called the emitter, the middle pin is the base and the rightmost pin is known as the collector. The base of the transistor is connected to pin 5 of the Arduino, the collector connected to the negative portion of the fan and emitter connected to the ground of the breadboard as a result whenever a signal is given out from the Arduino to turn the fan ON it will flip the switch of the transistor and allow connections between the ground wires to complete the circuit.

For running the fan at higher voltage the power of the relay is connected at the same position as the power of the fan on the breadboard and the ground is connected to the ground of the board. Thus the fan will run to our desired value.

Chapter 4

Automated Light Control

4.1 Introduction

Electricity is one of the most important energy source in human life. The usage of electricity is increasing worldwide mainly due to the increased population, but also due to the lack of awareness of how much electricity is being consumed. After class hours it is found at times that the lights would remain on, and unnecessary amount of electricity is being wasted. Thus to overcome this problem, a “light energy saving” system will be made that would automatically control the usage of light with the help of a PIR sensor. A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. Since this device is a passive measuring device it is called “Passive Infrared”, "Pyroelectric", or "IR motion" sensor. The PIR sensor will detect any object emitting IR radiation, such as human activities, etc and give a voltage output as a response. This voltage response would be detected by the Arduino which would then perform specific instructions to either turn the light on or off. The sensor itself has a short range of approximately 1m, but using a lens that focuses the IR radiation on the sensor we can increase the sensing range of up to 30m. By incorporating this sensor with the Arduino we can effectively save more energy and stop unnecessary use of electricity.

4.2 The mechanism of the PIR sensor

PIR sensors are more complicated than many of the other sensors (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output. The PIR sensor itself has two slots in it; each slot is made of a special material that is sensitive to IR. The lens used here is not doing much and so it can be seen that the two slots can 'see' out past some distance (basically the sensitivity of the sensor).

When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body (such as a human passes by) it intercepts one half of the PIR sensor which causes a *positive differential* change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

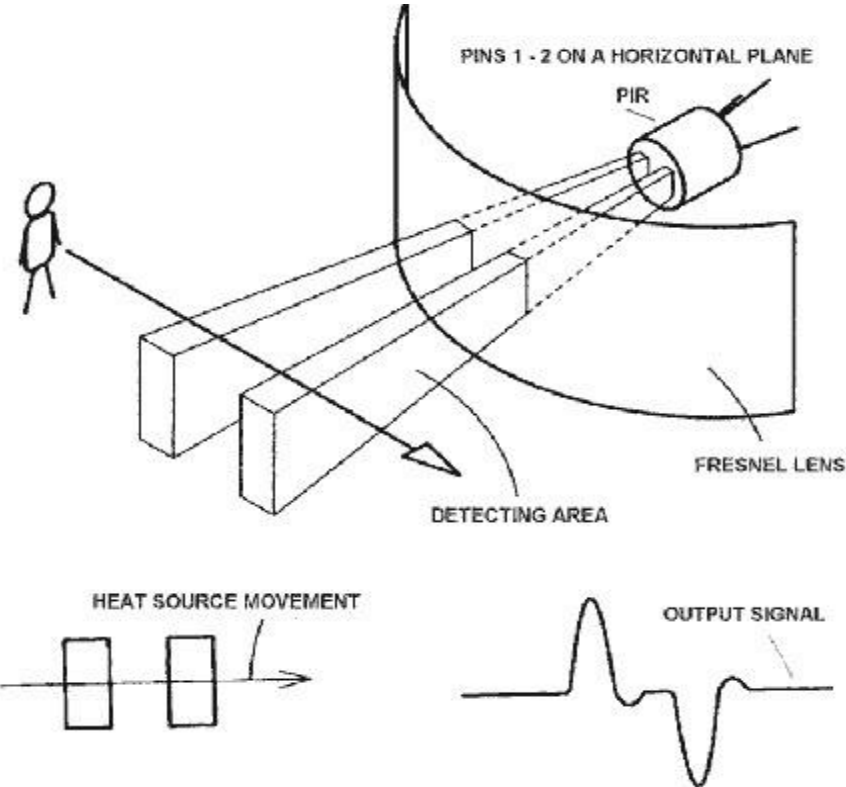
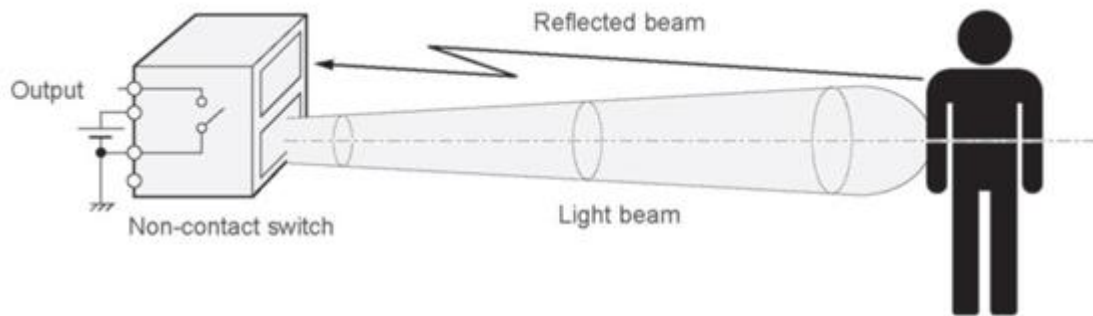


Fig: 4.1 An example drawing of PIR response to a moving body



The sensor sends out a ray of light toward the human body, then uses the reflected light to measure the distance and determine whether there is a person within a given distance of the sensor. If the sensor decides that there is a person within the given distance, it sets an output non-contact switch to ON.

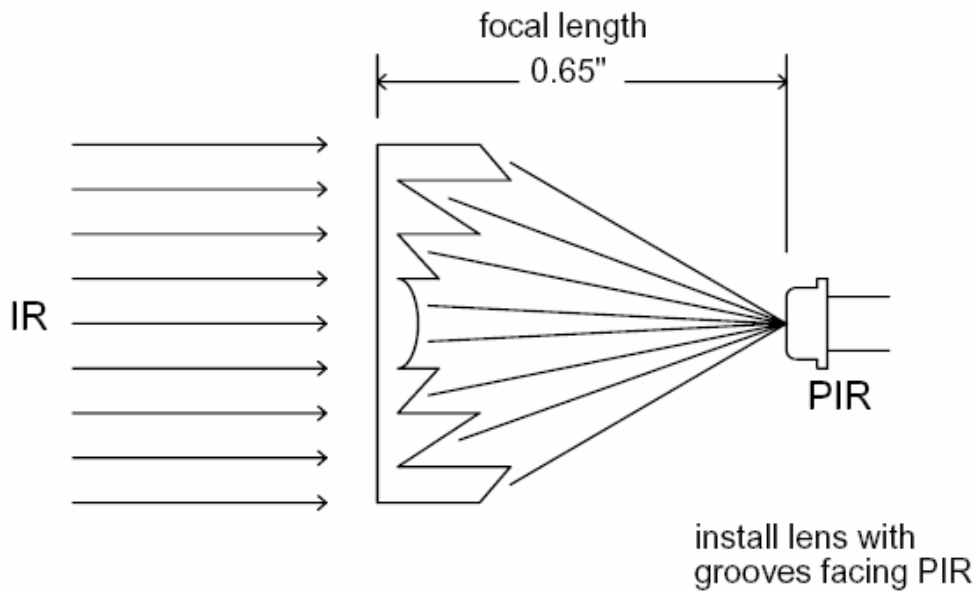
4.3 Components of a PIR sensor



Fig: 4.2 Passive Infrared Sensor (PIR)

- **Fresnel Lens**

A Fresnel lens is a Plano Convex lens that has been collapsed on itself as in figure 5 to form a flat lens that retains its optical characteristics but is much smaller in thickness and therefore has less absorption losses. The FL65 Fresnel lens is made of an infrared transmitting material that has an IR transmission range of 8 to 14 μm that is most sensitive to human body radiation. It is designed to have its grooves facing the IR sensing element so that a smooth surface is presented to the subject side of the lens which is usually the outside of an enclosure that houses the sensor. The lens element is round with a diameter of 1 inch and has a flange that is 1.5 inches square. This flange is used for mounting the lens in a suitable frame or enclosure. Mounting can best and most easily be done with strips of Scotch tape. Silicone rubber adhesive can also be used to form a more waterproof seal. The FL65 has a focal length of 0.65 inches from the lens to the sensing element. It has been determined by experiment to have a field of view of approximately 10 degrees when used with a PIR325 Pyro-electric sensor. This is shown in Fig: 4.2



FL65 FRESNEL LENS

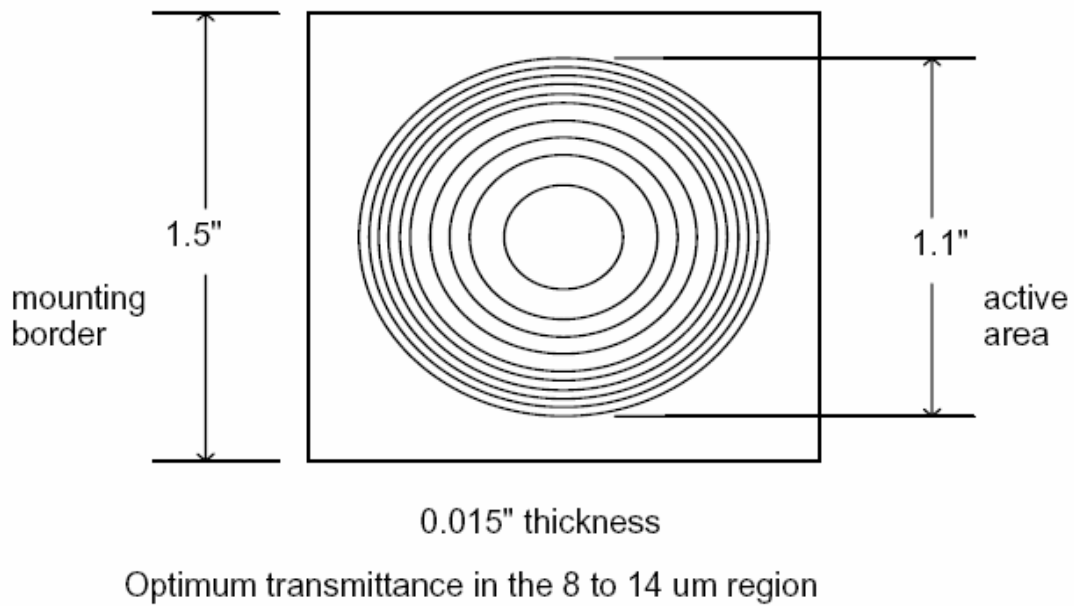


Fig: 4.3 Fresnel Lens

4.4 Set Up of the Circuit

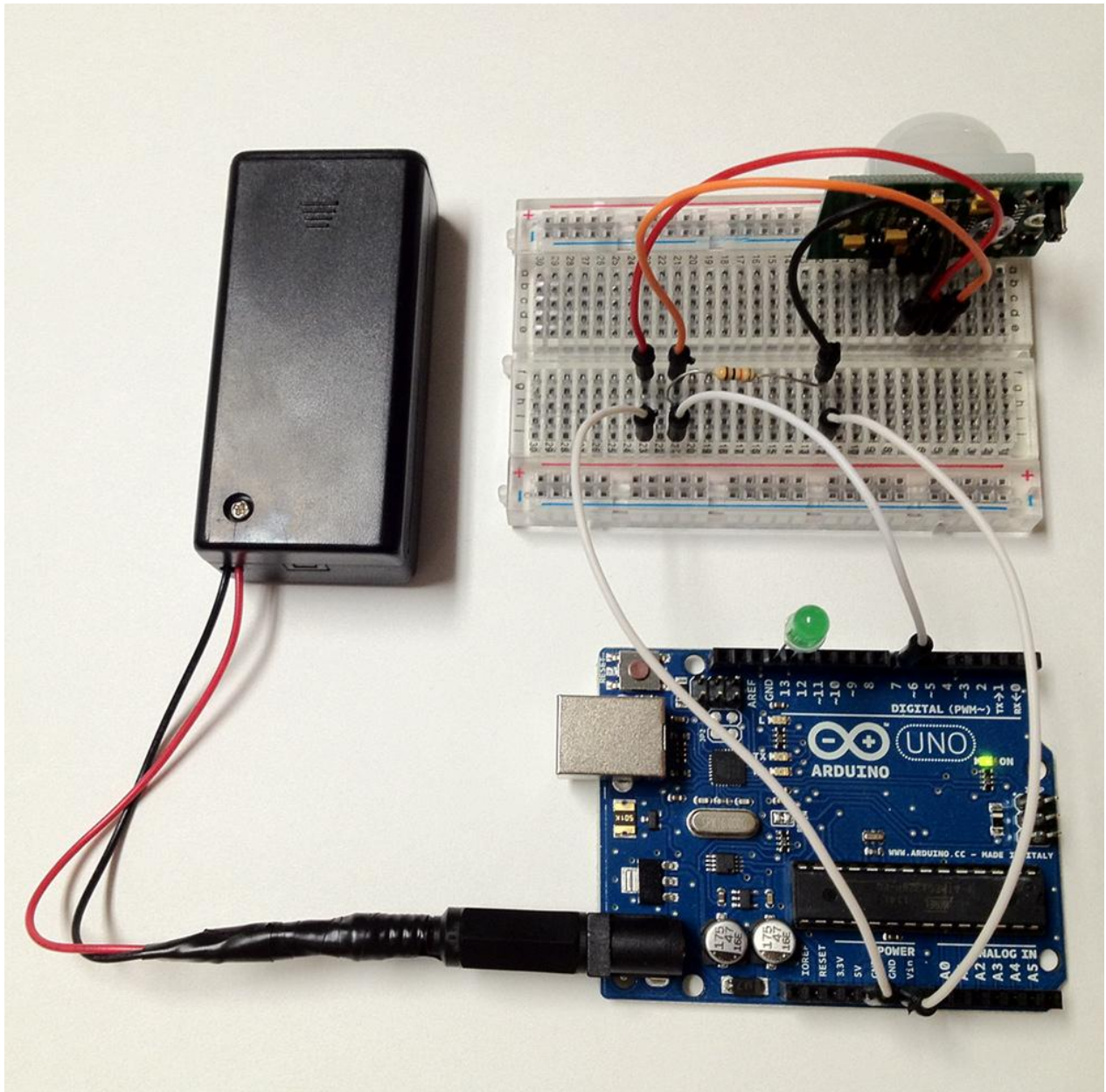


Fig 4.4 Light Sensing device using PIR motion sensor

In the set up for the light sensor an Arduino has been used because of the following reasons:

- It is an **open-source project**, software/hardware is extremely **accessible** and very flexible to be customized and extended
- It is **flexible**, offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs
- It is **easy to use**, connects to computer via USB and It is **inexpensive**

The PIR sensor usually consists of three pins: - out + as shown in the figure below

The “+” connects to Arduino +5V

The “-” connects to Arduino Ground

The “out” connects to Arduino digital Pin 2

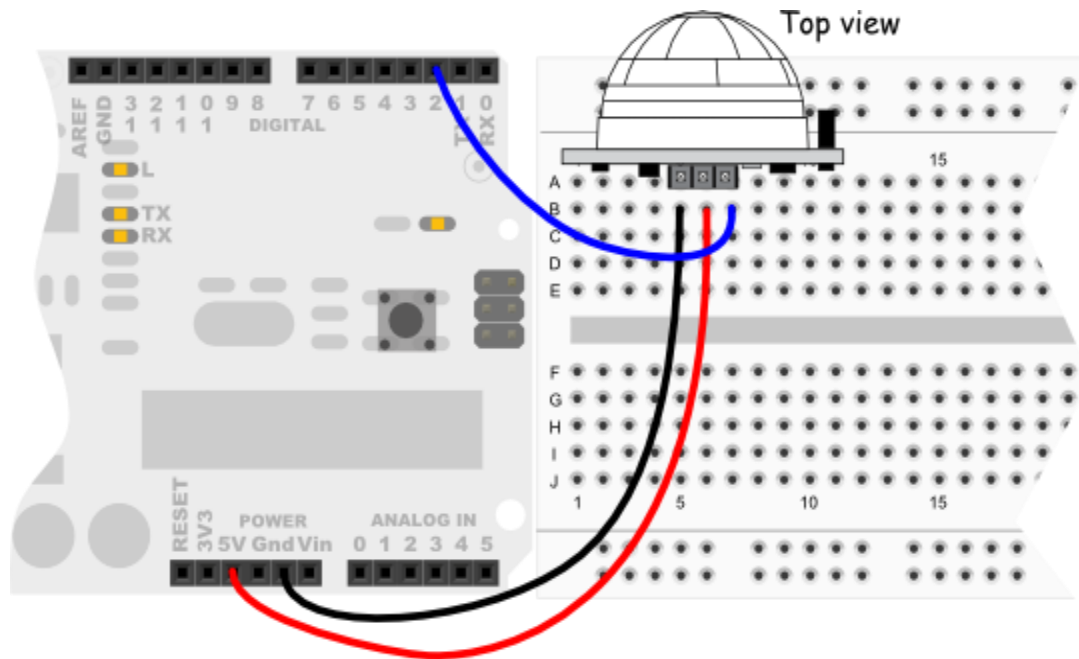


Fig: 4.4(a) PIR sensor connected to Arduino

Then we take an LED and connect the positive end of the LED to Pin 13 of the arduino and the negative end with the ground.

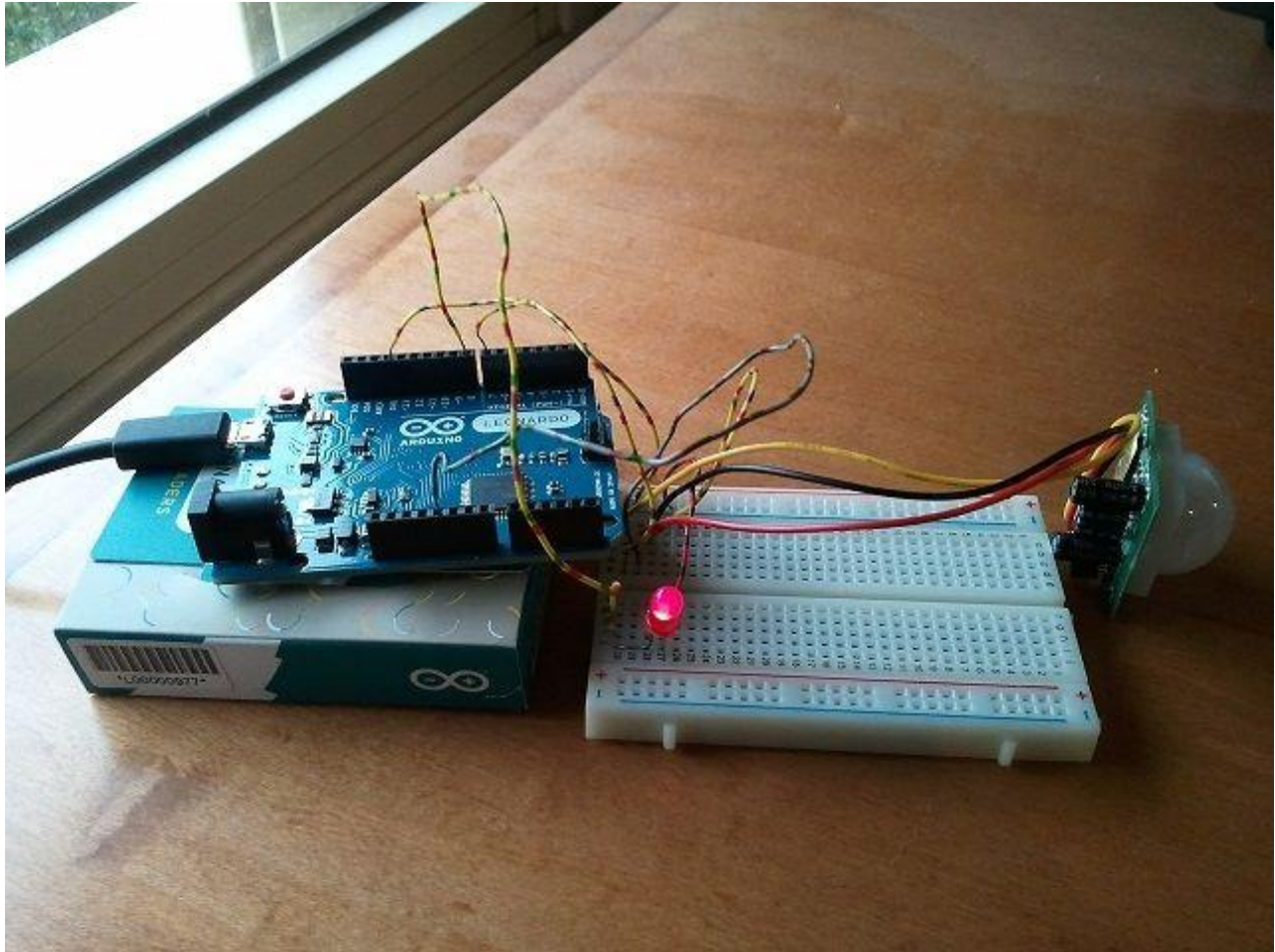


Fig: 4.4(b) Complete set up of the circuit

After running down the entire coding if the PIR senses any motion within its defined area the LED turns ON automatically. If there is no motion then the LED turns OFF. Since class time is usually 1 hr 20 minute so that is equal to 80 minutes. So during this 80 minutes interval the first student who enters the class his/her motion will be detected in the PIR sensor and from then on the light will be turned on and after 80 minutes the light will be turned off automatically. Thus this process will continue for the rest of the classes. A PIR sensor can be connected to a relay (perhaps with a transistor buffer) such as NPN when working with 220V DC supply.

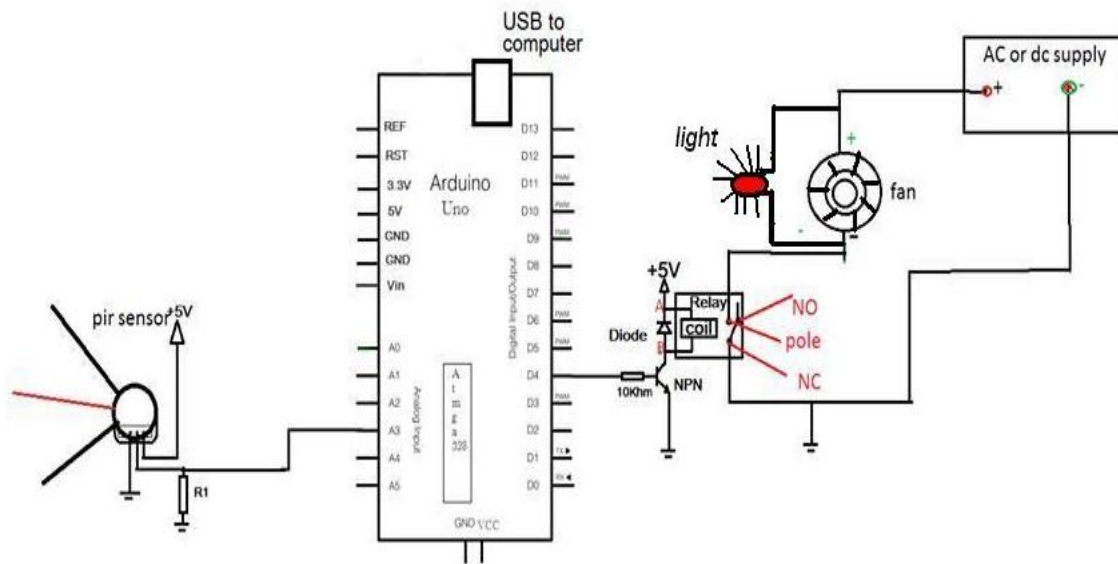


Fig: 4.4(c) Circuit set up using a relay when connected with 220V ac supply

4.5 Coding:

```
int ledPin = 13;      // choose the pin for the LED

int inputPin = 2;    // choose the input pin (for PIR sensor)

Int pirState= HIGH   // we assume motion detected

int pirState = LOW;  // we start, assuming no motion detected

int val = 0;         // variable for reading the pin status

Int fan1=5           // fan 1 is connected to pin 5

void setup() {

  pinMode(ledPin, OUTPUT); // declare LED as output

  pinMode(inputPin, INPUT); // declare sensor as input

  Serial.begin(9600);

}

val = digitalRead(inputPin); // read input value

void loop(){

  if (val == HIGH) {      // check if the input is HIGH

    digitalWrite(ledPin, HIGH); // turn LED ON

    digitalWrite(fan1, HIGH); // turn Fan ON

    if (pirState ==HIGH) { // we have just turned on

      Serial.println("Motion detected!");

      delay(600000); // after every 10 minutes it will detect motion (10*60*1000)

    }

  }

}
```

```
delay(10000); // after every 10 minutes it will detect motion  
  
    }  
  
} else {  
  
    digitalWrite(ledPin, LOW); // turn LED OFF  
  
    digitalWrite(fan1,LOW); // turn Fan OFF  
  
    if (pirState ==LOW){  
  
        // we have just turned off  
  
        Serial.println("Motion ended!");  
  
    delay(600000); //after 10 minutes it will try to detect motion  
  
    }  
  
}  
  
}
```

Chapter 5:

CONCLUSION

&

FUTURE RESEARCH

This thesis presents the prototype design of the classroom management system. The automated attendance system was able to achieve its goal and take the attendance in a way that was both user-friendly and efficient. The lights were able to turn on in response to any interruptions within its vicinity, and was also able to turn off within the given time delay. The fan however, was only able to turn on within the given temperature set, but it would not automatically adjust its fan speed as planned. Along with fan control, there are also few areas of the classroom management system that may need further research or improvement in the future.

The fan's inability to adjust its speed is mainly due to lack of programming time. We were unable to derive the code that would send different signals to the fan in relation to the temperature. This should be easily fixable to those who plan to work on the fan in the future, and this knowledge from the fan control system could also be applied on Air Conditioners which would make them become automatic as well. This would help in conserving electricity further as there are plenty of situations where the AC is left running in unattended classrooms.

The attendance system, although easy to use, requires a bit of coding knowledge when configuring the class's attendance. This means that the teacher has to be able to understand and modify the codes that stores student's ID, the timings of the class, and any other attendance related. A future scope could be to create an additional program to the attendance sheet that would prompt the teacher for the information required, and then automatically configure the attendance system with the information obtained. This would make it easier for teachers, rather than having them manually scan the code, and understand which string stores what information. Another improvement that could be made is to find a more cost effective way to scan ID cards. The barcode scanner happens to be the most expensive component within the intelligent management system, and if a cheaper alternative can be found then it would contribute immensely in making the system being cost effective.

Finally, the PIR sensor used for the light control had a drawback during practical setup. One of the problems faced was that it was a delicate hardware that could burn out if higher voltage was applied to it. Thus precaution should be made when handling the sensor. For future improvement, the light's intensity in the classroom could automatically adjust itself throughout the day using an LDR. As the lights may not be required to be at full strength throughout the day, this can help in improving the classroom's environment.

References:

1. National Semiconductor , November 2000, *LM35 Precision Centigrade Temperature Sensors*, Retrieved from http://www.ece.usu.edu/ece_store/spec/lm35dt-3p.pdf
2. Simon Monk, *Programming Arduino* (2011), Getting started with Sketches (McGraw-Hill)
3. Bullock, Madeline. "How Relays Work" 01 April 2000. HowStuffWorks.com, Retrieved from <http://electronics.howstuffworks.com/relay.htm> 13 January 2014.
4. John Walkenvach, *Microsoft Excel 2010 Excel Programming with VBA*, John Wiley & Sons,inc
5. Shpater, Pinhas, \Passive infrared motion detector and method", US Patent 6,215,399, November 10,1997.
6. D203B, Pyroelectric Infrared Radial Sensor Datasheet, <http://www.micropik.com/PDF/D203B-e.pdf>,
PIR SENSOR CO.,LTD..
7. Kyoung Nam Ha , Kyung Chang Lee, Suk Lee \Development of PIR sensor based indoor location detection system for smart home", in proceedings of the SICE-ICASE International Joint Conference, pp. 2162{2167, Oct. 18-21, 2006.