



ELECTRONICS & COMMUNICATION ENGINEERING

Under the Supervision of:

Mr. Sarwar Jahan

Assistant Professor

Name of Project: **Design and Implementation of Password Protected Home Security System Based On Arduino**

A Project Submitted by:

Zibon Kumer ID: **2012-2-55-045**

Md. Faridul Hsan ID: **2012-2-55-030**

DECLARATION

This declaration is to clarify that all of the submitted contents of this project work entitled “**Design and Implementation of Password Protected Home Security System Based On Arduino**” are original except for source material which have been admitted specifically in the references. All content of this project have been submitted as a part of partial fulfillment of Bachelor of Electronic & Telecommunication Engineering under the supervision of Mr. Sarwar Jahan, Assistant Professor, Department of Electronics and Communication Engineering, East West University.

Countersigned

Signature of Students

.....

.....

Mr. Sarwar Jahan
Assistant Professor
Department of Electronics & Communications
Engineering, East West University
Dhaka, Bangladesh

Zibon Kumer
ID: 2012-2-55-045

.....
MD Faridul Hasan
ID: 2012-2-55-030

APPROVAL

This project report titled “**Design and Implementation of Password Protected Home Security System Based On Arduino**” has been prepared and submitted by **Zibon Kumer (ID: 2012-2-55-045)** and **MD Faridul Hasan (ID: 2012-2-55-030)**. This project has been submitted in term of scope, quality and presentation as partial fulfillment of the requirement for the Bachelor of Science in Electronic & Telecommunication Engineering, East West University.

Checked and Approved By

.....

Mr. Sarwar Jahan
Assistant Professor
Department of Electronics and Communications Engineering
East West University

.....

Dr. M. Mofazzal Hossain
Professor & Chairperson
Department of ECE
East West University

ACKNOWLEDGEMENT

We are very thankful to our almighty ALLAH, giving us the opportunity to complete this project which is titled as “**Design and Implementation of Password Protected Home Security System Based On Arduino**”. Without his help we couldn't complete it. This final year project is prepared for leads the degree of Bachelor of Electronic & Telecommunication Engineering under the department of Electronics & Communication Engineering, East West University. We would like to express our deepest thanks to **Mr. Sarwar Jahan, Assistant Professor, Department of ECE** as our supervisor who had guided be a lot of task during two semester session. We also want to thanks the professors, lecturers and staffs of Department of ECE for their valuable information, suggestions and guidance and as well as co-operation during the project that really helped us.

and Finally, we would like to thanks our parents, family and friends for their support during this project.

ABSTRACT

Home security system is needed for convenience and safety. This system invented to keep home safe from intruder. In this work, we present the design and implementation of a GSM based wireless home security system, which take a very less power. In addition there was a need to automate home so that user can take advantage of the technological advancement in such a way that a person getting off the office does not get melted with the hot climate.

Index

Chapter 01: INTRODUCTION.....	07
1.1 What is Arduino.....	07
1.2 Why Arduino.....	08
1.3 Importance of Arduino.....	10
1.4 History of Arduino.....	10
1.5 Types of Arduino.....	12
1.6 Advantages and Disadvantages of Using Arduino.....	15
1.7 How to Started with Arduino	17
Chapter 02: HARDWARE DESCRIPTION	20
2.1 Introduction	20
2.2 PIR Sensor	20
2.3 5V - 2amp power adapter	26
2.4 Wires.....	27
2.5 Light Emitting Diode	28
2.6 Arduino Writing Sketches	30
Chapter 03: SYSTEM ARCHITECTURE	30
3.1 Introduction.....	- 30
3.2 Components Used.....	30
3.3 Components Circuit Diagram.....	31
3.4 Block Diagram.....	34
3.5 Block Diagram of Full components	36

Chapter 04: Implementation and Result.....	37
4.1 Implementation and Result	37
4.2 System in Idle Mode (Circuit design).....	37
4.3 Practical set-up and Result.....	38
Chapter 05: CONCLUSION.....	39
Chapter 06: FUTURE SCOPE.....	39
REFERENCES.....	40
APPENDIXES	42

Chapter-01

INTRODUCTION

1.1 What is Arduino:

Arduino is a micro-controller that's great for hardware prototyping. A micro-controller is kind of like your brain — it processes inputs and sends out outputs. A micro-controller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. With a micro-controller like Arduino, you can control circuits, LED's, and so much more. Microcontrollers are notorious for being difficult to program; the goal of Arduino is to create an accessible way for software developers to enter the world of microcontroller programming. Arduino is a microcontroller interface built around an Atmel ATmega processor, coupled with a language and programming environment for creating logic on the chip.

Arduino is open source, both in its software and hardware specification, so that hobbyists can assemble the simplest Arduino modules themselves by hand. More sophisticated pre-assembled Arduino modules can be purchased and are modestly priced. The hardware comes in many format specifications, from a small wearable device, to larger surface mounted modules. The primary mode of computer connection is via USB, though Bluetooth, serial and Ethernet form factors also exist.

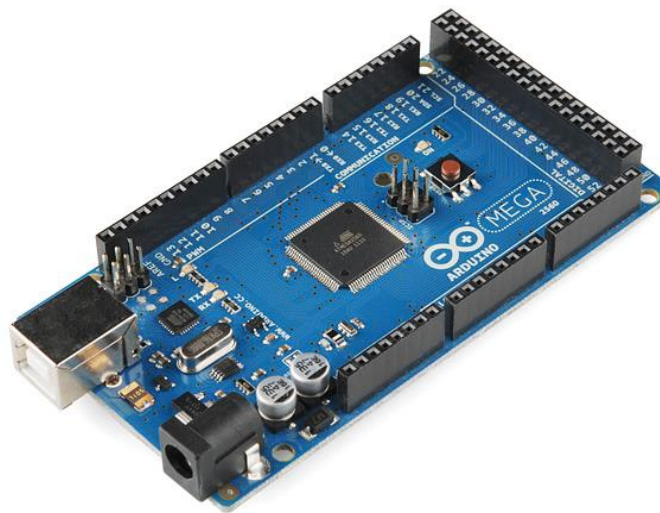


Figure 1.1: Arduino Mega ^[1]

The Arduino software is free and open source. The programming platform is based on the popular Wiring language. The IDE is based on Processing, which is a well-known language among designers and prototypers. Unlike most microcontroller interfaces, Arduino is cross-platform; it can be run on Windows, Linux and Macintosh OS X.

Arduino comes in a variety of different boards. Arduino boards are available commercially in pre-assembled form, or as do-it-yourself kits. The hardware design specifications are openly available, allowing the Arduino boards to be produced by anyone. In mid-2011, it was estimated that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

1.2 Why Arduino:

Arduino allows users a simple pathway to create interactive objects that can take input from switches and sensors, and control physical outputs like lights, motors or actuators. Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available in market for physical computing. Such as Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, BeagleBoard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, artists, tinkerers, and interested amateurs over other systems:

Cross-platform - The Arduino drivers and software runs on Mac, Windows, and Linux operating systems and they are completely open source. Most microcontroller systems are limited to Windows.

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and

even the pre-assembled Arduino modules cost less than \$50 that comes with free authoring software.

Simple board - Many microcontroller boards are historically enormously complex with a lot of added-on parts like LCDs, buttons, LEDs, 7-segments, etc. showing everything it can do. Arduino has the bare minimum. Want more? Get a shield. There are hundreds of Arduino shields, from LCD to Wi-Fi, but it's up to the user to add that. Shields add extra functionality easily.

Simple programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. There are libraries to do simple things, like twiddle pins or debounce buttons and tons of object-wrapped libraries to do complex things, like writing to SD cards, LCD screens, parsing GPS. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino. Arduino is a simple system designed for creative people with little or "no prior knowledge of electronics. Above all, it has a very welcoming attitude towards beginners and tries not to scare them too much."

Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Sensor Data - The Arduino really took off because it has Analog-to-Digital input, in other words, you can take in sensor data like light, temperature, sound, or whatever using the low-cost sensors already on the market and get that into the Arduino easily. It also has ready-to-go SPI and I2C for digital sensors. This covers 99% of sensors on the market. You can't easily do this with other platforms.

1.3 Importance of Arduino:

While some of these Arduino projects may seem frivolous, the technology actually taps into a number of trends that will make it a potentially important force in the industry. "The Internet of Things" is a popular phrase used in the tech community to describe everyday items that are connected to the Internet and able to share information. Smart energy meters are an often-used example, which could regulate appliance usage to save money on energy. Many consider the Internet of things to be an important part of the loosely defined phenomenon called Web 3.0

Also, the concept of ubiquitous computing is fast becoming a cultural norm. Public perception and comfort level is shifting towards integrating technology into the fabric of everyday life. The small form factor of Arduino allows it to be applied in all kinds of everyday objects. In fact, the Arduino LilyPad form factor allows for wearable Arduino devices.

1.4 History of Arduino:

Hernando Barragán created the development platform Wiring as his Master's thesis project in 2004 at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. Massimo Banzi and Casey Reas (known for his work on Processing) were supervisors for his thesis. The goal was to create low cost, simple tools for non-engineers to create digital projects. The Wiring platform consisted of a hardware PCB with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller

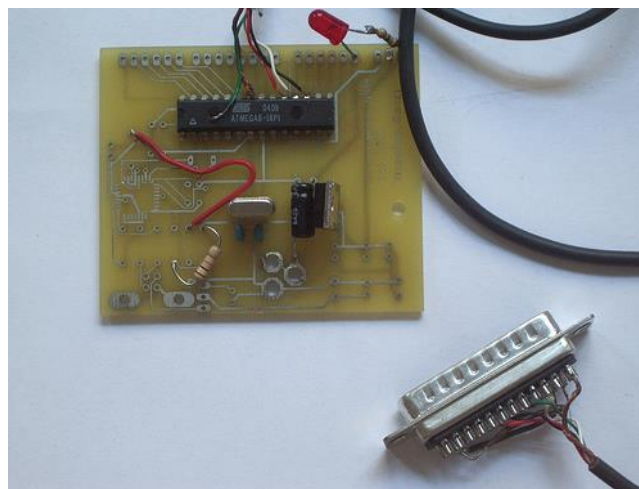


Figure 1.2: Old Arduino Board [2]

In 2005, in Ivrea, Italy, a project was initiated to make a device for controlling student-built interactive design projects that was less expensive than other prototyping systems available at the time. One of the cofounders, Massimo Banzi, named this piece of hardware Arduino in honor of Bar di Re Arduino (In 1002, King Arduino became the ruler of the Italy. Today, the Bar di Re Arduino, a pub on a cobblestoned street in town, honors his memory), and began producing boards in a small factory located in the same region as the computer company Olivetti.

The Arduino project is a fork of the open source Wiring platform and is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment (IDE).

Arduino was built around the Wiring project of Hernando Barragan. Wiring was Hernando's thesis project at the Interaction Design Institute Ivrea. It was intended to be an electronic version of Processing that used our programming environment and was patterned after the Processing syntax. Arduino would not exist without Wiring and Wiring would not exist without Processing.

Currently, there are more than 200 distributors of Arduino products around the world. About 80 percent of people who buy this product are from the United States and Europe. The interest in this product is rising in the China, India, and South America markets. Over the years, new designs of the Arduino have been created. The original design is called the Arduino Uno. Some of the Arduino designs are the Arduino Mega, Arduino Nano, LilyPad Arduino, and Arduino Ethernet. This past year, the Arduino gained publicity by partnering with Google. Google released the Android ADK, or Accessory Development Kit, which is based on the Arduino. A person can build an Android app that uses the Phone's camera, motion sensors, touch screen, and internet connectivity. It looks as though the Arduino is creating a new, cheaper way of programming. It does not seem to be going away any time soon; it is only getting more popular.

1.5 Types of Arduino:

For over ten years, Arduino has held onto its popularity as “that small dev-board aimed to get both artists and electronics enthusiasts excited about physical computing.” Along the way, it’s found a corner in college courses, one-off burning man rigs, and countless projects that have landed here. Without a doubt, the Arduino has a cushy home among hobbyists, but it also lives elsewhere. Arduino lives in engineering design labs as consumer products move from feature iterations into user testing. It’s in the chem labs when scientists need to get some sensor data into their pc in a pinch. Despite the frowns we’ll see when someone blinks an LED with an Arduino and puts it into a project box, Arduino is here to stay. I thought I’d dig a little bit deeper into why both artists and engineers keep revisiting this board so much.

Table: different Types of Arduino ^[1]:

ENTRY LEVEL	ARDUINO UNO ARDUINO LEONARDO ARDUINO 101 ARDUINO ROBOT ARDUINO ESPORA
	ARDUINO MICRO ARDUINO NANO ARDUINO MINI ARDUINO MKR2LINO ADAPTER
	ARDUINO STARTER KIT ARDUINO BASIC KIT ARDUINO LCD SCREEN
ENHANCED FEATURES	ARDUINO MEGA ARDUINO ZERO ARDUINO DUE ARDUINO MEGA ADK ARDUINO PRO
	ARDUINO PRO MINI ARDUINO MOTOR SHIELD ARDUINO USB HOST SHIELD PROTO SHIELD
	MKR PROTO SHIELD MKR PROTO LARGE SHIELD ARDUINO ISP ARDUINO USB 2-SERIAL MICRO
	ARDUINO MINI USB ADAPTER
INTERNET OF THINGS	ARDUINO YÙN ARDUINO ETHERNET ARDUINO MKR1000 ARDUINO WIFI SHIELD
	ARDUINO WIFI 101 SHIELD ARDUINO YÙN SHIELD ARDUINO WIRELESS SD SHIELD
	ARDUINO WIRELESS PROTO SHIELD ARDUINO ETHERNET SHIELD V2 ARDUINO GSM SHIELD V2
	MKR1000 BUNDLE
WEARABLE	ARDUINO CEMMA LILYPAD ARDUINO USB LILYPAD ARDUINO MAIN BOARD
	LILYPAD ARDUINO SIMPLE LILYPAD ARDUINO SIMPLE SNAP
3D PRINTING	MATERIA 101

■ BOARDS
 ■ MODULES
 ■ SHIELDS
 ■ KITS
 ■ ACCESSORIES
 ■ COMING NEXT

1. The Arduino Uno

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins ,6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

2. The Arduino Due

he Arduino Due is the first Arduino with an ARM-based microcontroller. It is based on the Atmel SAM3X8E ARM Cortex-M3 CPU that enables a higher level of performance compared to Arduinos with 8-bit microcontrollers. The Arduino Due has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), a 84 MHz clock, a USB OTG capable connection, 2 DAC (digital to analog converters), 2 TWI, a power jack, an SPI header, a JTAG header, and reset and erase buttons. The Due's processor is faster, has more memory, and more I/O ports. It does not support many shields. Because of the faster CPU, the Arduino Due runs on a lower voltage: 3.3V over the Uno's 5V. This means it cannot always support the same devices.



Figure 1.3 : Different Types of Arduino [3]

3. The Arduino Mega

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

The Arduino Mega comes in two types, the Mega 2560 and the MEGA ADK. The ADK is similar to the 2560; however, it also has a programmable USB host chip installed. It uses the same 5V power supply as the Uno, so many of the Arduino shields are also compatible with the Mega; however, because of the placement of some of the pins, not all of them are usable.

4. The Arduino Leonardo

The Leonardo is not a common board, but has similar features to the Uno, including the 5V power supply and the processing power. It is a good board for those who need more input and output ports than the Arduino Uno, but do not need the horsepower or size of the Due. It uses a micro-USB adapter instead of the Uno's full-size USB port.

Table: Different Types of Arduino:

Features	Arduino Uno	Arduino Due	Arduino Mega	Arduino Leonardo
Processor	16Mhz ATmega328	84MHz AT91SAM3X8E	16MHz ATmega2560	16MHz ATmega32u4
Memory	2KB SRAM, 32KB flash	96KB SRAM, 512KB flash	8KB SRAM, 256KB flash	2.5KB SRAM, 32KB flash
Digital I/O	14	54	54	20
Analogue I/O	6 input, 0 output	12 input, 2 output	16 input, 0 output	12 input, 0 output

1.6 Advantages and Disadvantages of Using Arduino:

Arduino increases exponentially due to its readability and easiness. But the point to think is whether the use of Arduino is in favor of engineers or not. First of all, we will look upon the advantages of Arduino and later we will discuss the disadvantages. Yes there might be some disadvantages of Mighty Arduino as well. Let's start with advantages.

Advantages:

> Ready to Use:

The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5V regulator, a burner, an oscillator, a micro-controller, serial communication interface, LED and headers for the connections. You don't have to think about programmer connections for programming or any other interface. Just plug it into USB port of your computer and that's it. Your revolutionary idea is going to change the world after just few words of coding.

> Examples of codes:

Another big advantage of Arduino is its library of examples present inside the software of Arduino. I'll explain this advantage using an example of voltage measurement. For example if you want to measure voltage using ATmega8 micro-controller and want to display the output on computer screen then you have to go through the whole process. The process will start from learning the ADC's of micro-controller for measurement, went through the learning of serial communication for display and will end at USB – Serial converters. If you want to check this whole process click on the link below. [DC voltage measurement using Atmel AVR micro-controller](#). On the other hand, if you want to measure the voltage using Arduino. Just plug in your Arduino and open the Read Analog Voltage example as shown in the figure 1. The project is ready after putting some reasonable resistors and zener diode. You can easily see the voltage on the Serial terminal of Arduino.

> Effortless functions:

During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability. You can say that during debugging you don't have to worry about the units conversions. Just use your all force on the main parts of your projects. You don't have to worry about side problems.

> Large community:

There are many forums present on the internet in which people are talking about the Arduino. Engineers, hobbyists and professionals are making their projects through Arduino. You can easily find help about everything. Moreover the Arduino website itself explains each and every functions of Arduino.

So, We should conclude the advantage of Arduino by saying that during working on different projects you just have to worry about your innovative idea. The remaining will handle by Arduino itself.

Disadvantages:

> Structure:

Yes, the structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible. But with the big structures of Arduino we have to stick with big sized PCB's. If you are working on a small micro-controller like ATmega8 you can easily make your PCB as small as possible.

> Cost:

The most important factor which you cannot deny is cost. This is the problem which every hobbyist, Engineer or Professional has to face. Now, we must consider that the Arduino is cost effective or not.

1.7 How to Started with Arduino:

we can start on our projects, we first need to get your Arduino talking to your computer. We need to do this so you can compile and send code for your Arduino to execute.

Step 1: Download the Arduino software:

Go to <http://arduino.cc/en/Main/Software> and download the Arduino Software for your Windows.

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

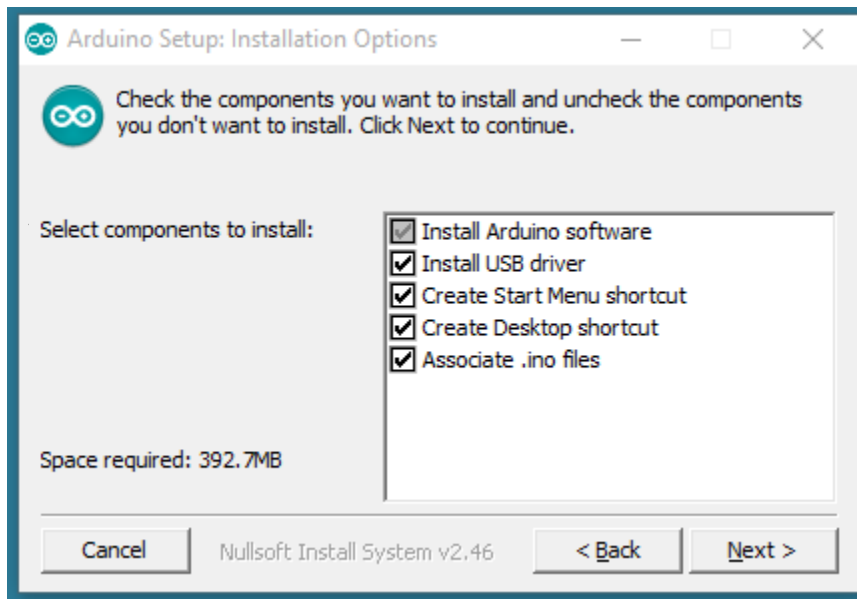


Figure 1.4: Screenshot of Arduino Software installation

Step 2: Install the software

Install the Drivers:

- Plug in your board and wait for Windows to begin its driver installation process. After a few moments, the process will fail, despite its best efforts.
- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named “Arduino UNO (COMxx)”.
- Right click on the “Arduino UNO (COMxx)” port and choose the “Update Driver Software” option.
- Next, choose the “Browse my computer for Driver software” option.
- Finally, navigate to and select the Uno’s driver file, named “ArduinoUNO.inf”, located in the “Drivers” folder of the Arduino Software download.
- Windows will finish up the driver installation from there.

Choose the components to install

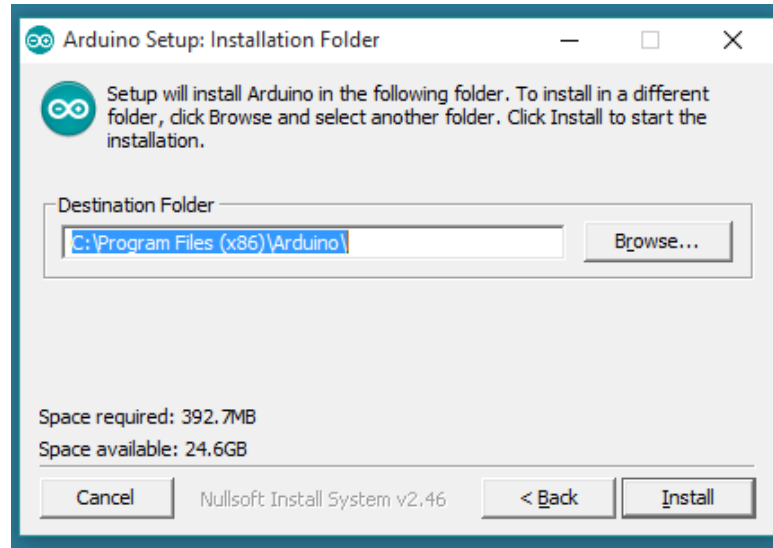


Figure 1.5: Screenshot of Arduino Software installation

Choose the installation directory (we suggest to keep the default one)

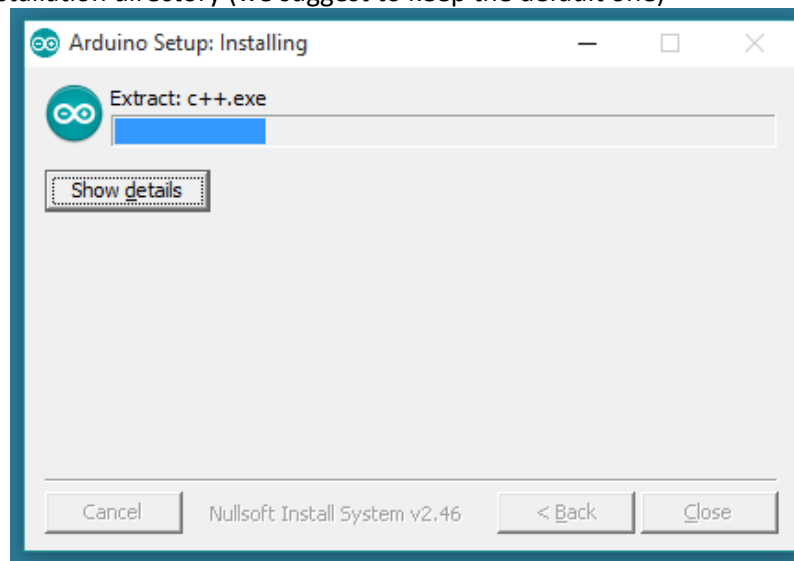


Figure 1.6: Screenshot of Arduino Software installation

The process will extract and install all the required files to execute properly the Arduino Software (IDE)

Chapter-02

HARDWARE DESCRIPTION

2.1 Introduction:

This designed system consists of Arduino mega 2560, Servo motor, LED 5mm, Wires(male to female and male to male), 4×4 keypad, breadboard, 220 ohm resistances, 5 volt 2A adopter, PIR sensor and POT. Here we are going to elaborate the functions and characteristics of all the elements that we used in this design.

2.2 PIR Sensor:

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.



Figure 2.1: PIR sensor ^[4]

Fig: PIR Motion Sensor

- a) With Variable &
- b) Without Variable

2.2.1 Overview:

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in

homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors. The IR energy of an object depends on different parameters such as its temperature, color, and texture. This energy is not visible with the human eye, but PIR sensors can detect it.

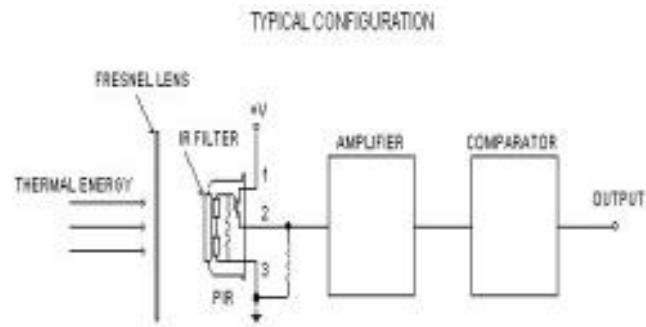


Figure 2.2: Block Diagram of PIR Sensor [5]

2.2.2 Operating principle of PIR Sensor:

All objects with a temperature above absolute zero emit heat energy in the form of radiation. Usually this radiation isn't visible to the human eye because it radiates at infrared wavelengths, but it can be detected by electronic devices designed for such a purpose.

The term passive in this instance refers to the fact that PIR devices do not generate or radiate any energy for detection purposes. They work entirely by detecting the energy given off by other objects. PIR sensors don't detect or measure "heat"; instead they detect the infrared radiation emitted or reflected from an object.

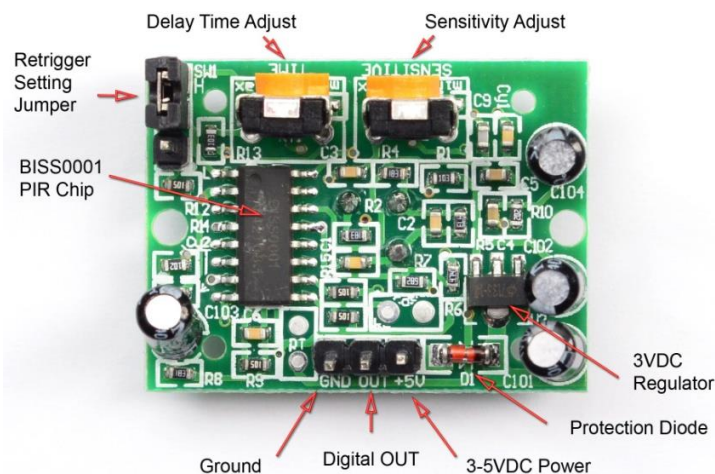


Figure 2.3: Principle of PIR Sensor [6]

2.2.3 How PIR Motion Works

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 really doing much and so we see 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 like a human or animal passes by, it first 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.

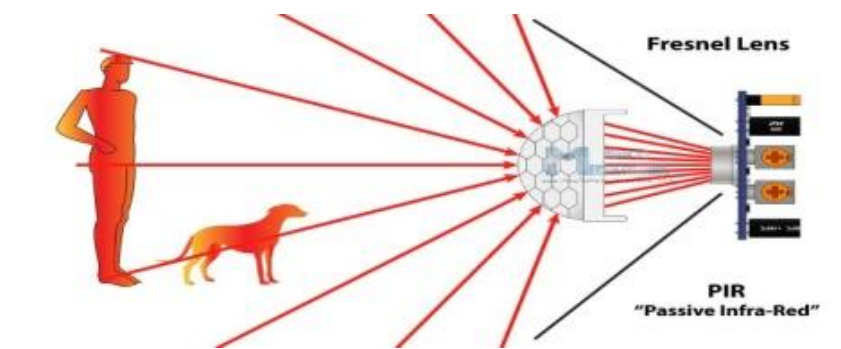


Figure 2.4: PIR Sensor Working Procedure [7]

The aim of the PIR sensor is to detect motion in a specific area. As seen in Figure: "PIR sensor", these sensors contain two parts that are sensitive to IR. Since it is motion that is important, the signal delivered by the sensor reflects different amounts of IR detected by each part. If both parts "see" the same amount of IR, the sensor detects nothing. But, if one of the two parts "sees" more or less IR than the other part, the sensor detects something and the output voltage of the sensor varies.

The area of the two rectangles that is IR sensitive is small ($\approx 2 \text{ mm}^2$ for each rectangle). Clearly, it would be better to increase the IR sensitive area. This can be done in two ways:

- By increasing the number of sensors
- By using a Fresnel lens.

The latter method is highly recommended as it is inexpensive and improves IR reception by the sensor. It multiplies the area visible to the sensor and it focuses reception of IR by the sensor.

2.2.4 The sensor signal

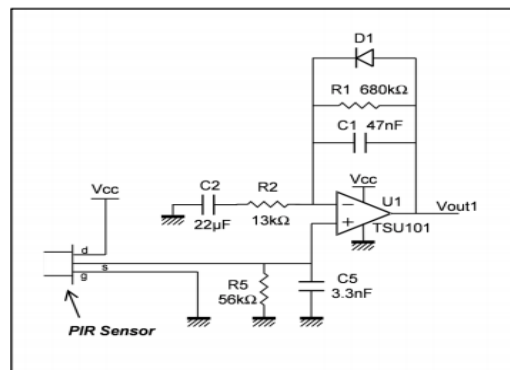
When a body, with a temperature different to ambient, moves into the detection field of a PIR sensor, the sensor provides a small AC signal. This signal is in the range of 1 mVpp. This small voltage is around a DC signal that may significantly vary from one sensor to another. For this reason, it is mandatory to cancel the DC part of the signal and to amplify the AC part. As the signal is disturbed by the environment, a noise filter is useful.

2.2.5 Three-stage architecture for detecting motion

To detect motion, a two-stage architecture based on op-amps is used to condition the signal of a PIR sensor. This architecture is explained below.

Stage 1

The first architectural stage amplifies the signal. It cancels the DC part of the signal and filters the high frequency noise that could lead to false detections



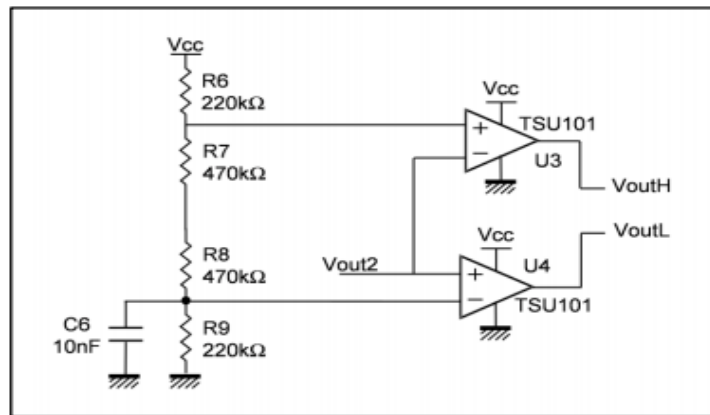
1. C = capacitor, D = diode, R = resistor, U = op-amp

Figure 2.5: The first architectural stage amplifies the signal [8]

Figure shows that noise is filtered thanks to the components R1 and C1. The cut-off frequency is 5 Hz ($f_{high1} = 1 / (2 \times \pi \times R1 \times C1)$). This application does not need to work at higher frequencies because usually we are detecting human motion. The second filter is used to reject the DC part of the signal. R2 and C2 perform a high pass filter that has a cut-off frequency of 0.6 Hz ($f_{low1} = 1 / (2 \times \pi \times R2 \times C2)$). Since we do not amplify the DC part of the signal, the V_{io} , which is the input offset voltage of the op-amp, has no importance in this application. The stage gain is 53.3 ($Gain1 = 1 + R1/R2$). This amplification allows a usable signal that is higher than the noise level. The DC bias of the first architectural stage is determined by the sensor. To avoid saturation, the gain must not be too big because amplification is made around this common mode voltage (which is not $VCC/2$). STMicroelectronics' TSU101 op-amp is a perfect fit for this architectural stage. As well as having very low consumption, the TSU101 operates with a typical current consumption of only 600 nA at $VCC = 3.3$ V.

Stage2

The second and last architectural stage allows the user to perform a window comparator. As a consequence, the signal is perfectly conditioned to go to a microcontroller. Figure shows the schematic of this stage.



1. C = capacitor, R = resistor, U = op-amp

Figure 2.6: The Second architectural stage amplifies the signal ^[9]

When a heat source is detected, the output of the op-amp U3 and/or U4 is in low state. The divider bridge, composed of the resistors R6, R7, R8, and R9, is used to set the voltage reference of these devices. The TSU101 op-amp is an input/output rail to rail op-amp, and there is no constraint on the input common mode voltage. Thus, while the voltage references of U3

and U4 are within the Vcc range the window comparator works. For the following example, the reference of U3 is set to $0.84 * V_{cc}$:

$$\frac{R_7 + R_8 + R_9}{R_6 + R_7 + R_8 + R_9} \times V_{cc} = 0.84 \times V_{cc} \dots\dots\dots(1)$$

When the signal (Vout2) is above this reference (2.77 V if Vcc = 3.3 V), the output of U3 is in low state, close to ground. U4 is used to detect when the signal is below a certain reference. In this example, the voltage reference is set to 530 mV thanks to the divider bridge composed of R6, R7, R8, and R9. The voltage reference calculation is:

$$\frac{R_9}{R_6 + R_7 + R_8 + R_9} \times V_{cc} = 0.16 \times 3.3 = 0.53V \dots\dots\dots(2)$$

Thus, when the signal (Vout2) is lower than 0.53 V, the output voltage of U4 is in low state (see Table). If only one digital output is required, the user may use a NAND gate connected to the outputs of U3 and U4.

Table 2: NAND Truth Table ^[2]

Vout2	Output U ₃	Output U ₄	Output NAND gate
Vout2 < 0.16 * Vcc	High	Low	High
0.16 * Vcc < Vout2 < 0.84 * Vcc	High	High	Low
Vout2 > 0.84 * Vcc	Low	High	High

2.2.6 Conclusion

PIR sensors are widely used and require op-amps to amplify and filter the signal they generate which is noisy and small in amplitude. Op-amps or comparators can be used to compare the amplified signal with threshold voltages before it goes into the I/O of a microcontroller, there is no need analog to digital converter.

2.3 5V - 2amp power adapter

For additional power we need to use 5V-2amp power adapter.



Figure 2.7: 5V-2amp Power Adapter ^[10]

2.4 Wires

Wire Types –

- Arduino Cable
- USB 2.0 cable type A/B
- Male to Male
- Male to Female
- Adapter power plug and play wire etc.

Figure 2.8: Arduino Cable ^[11]Figure 2.9: USB 2.0 Cable type A/B^[11]Figure 2.10: M2M and M2F Cable^[11]

2.5 Light Emitting Diode

2.5.1 What is LED?

a light-emitting diode (LED) is a semiconductor device that emits light when an electric current is passed through it. Light is produced when the particles that carry the current (known as electrons and holes) combine together within the semiconductor material. The output from an LED can range from red (at a wavelength of approximately 700 nanometers) to blue-violet (about 400 nanometers). Some LEDs emit infrared (IR) energy (830 nanometers or longer); such a device is known as an *infrared-emitting diode* (IRED).

2.5.2 How LED Light Works:

A light-emitting diode (LED) is a semiconductor device that creates light using solid-state electronics. A diode is composed of a layer of electron rich material separated by a layer of electron deficient material which forms a junction. Power applied to this junction excites the electrons in the electron rich material leading to photon emission and the creation of light. Depending on the chemical composition of the semiconductor layers, the color of light emission will vary within the electromagnetic spectrum.

The individual diodes are grouped together to form a traffic signal where depending on the individual LED size up to several hundred “lamps” are packaged into an array to form a traffic signal head. Two styles of LED traffic signals known as the diffused and pixilated style are commonly used. In the pixilated style the actual LED array is visible within the traffic signal head while the diffused lens provides a uniform appearance similar to an incandescent signal.

LED's are much more energy efficient than their incandescent counterparts for several reasons. LED's produce uniform light dispersion and light output is dispersed evenly over the lens which make them brighter than incandescent lamps. LED's are very energy efficient producing up to 90 percent light output with very little heat while incandescent bulbs use up to 90 percent of their energy generating heat. Incandescent lamps only produce white light which must be filtered for traffic signal use, and this leads to an additional loss in energy. LEDs, on the other hand, produce colored light that does not need to be filtered out – all of the energy is concentrated around one color band and none is “wasted” on undesired colors.

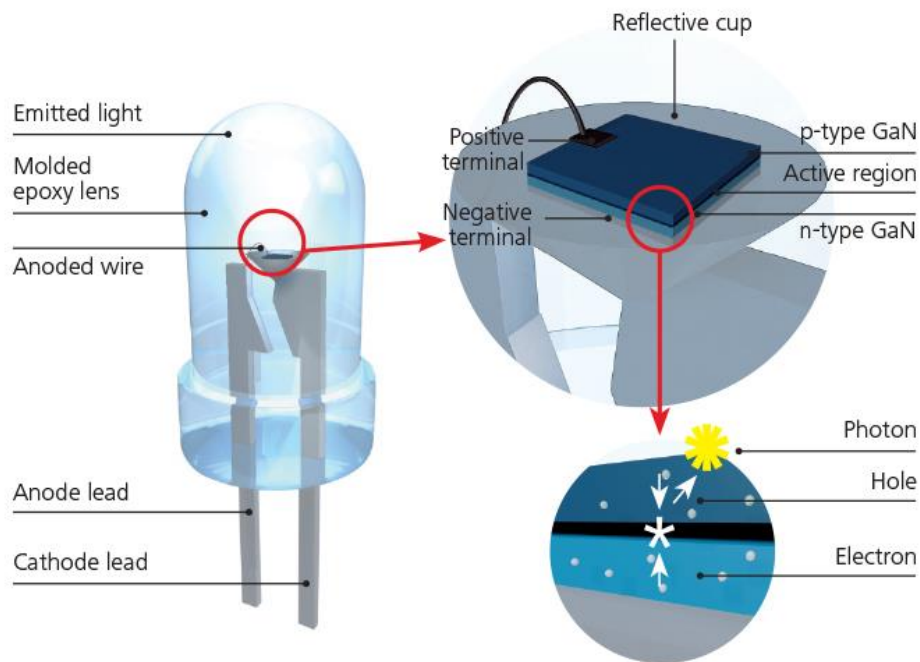


Figure 2.11: LED light ^[12]

2.6 Arduino Writing Sketches

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension `.ino`. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Chapter-03

SYSTEM ARCHITECTURE

3.1 Introduction

This project deals with the design & development of a theft control system for home, which is being used to prevent/control any theft attempt. The developed system makes use of an embedded system based on PIR sensor. The designed & developed system can be installed in the home. An interfacing intrusion-detector unit is also connected to the microcontroller-based security system.

3.2 Components Used

The designed system consists of the following components-

- Arduino Mega 2560
- Adjust IR Pyroelectric Infrared IR PIR Motion Sensor
- 5V Four Channel Relay Module With opt coupler
- 5V 1A Adapter,
- LED Lights,
- Arduino Sketch Software ,
- Active Buzzer Alarm Module Sensor Beep
- Display,
- 4 x 4 Matrix Array 12 Key Membrane Switch Keypad,
- Servo Motor,
- Male to Male Connectors,
- Male to Female Connectors.
- 4.7k Potential meter.
- Real Time Clock DS1307 I2C AT24C32 Module SainSmart LCD Module For Arduino 20 X 4
- SMD RGB LED
- Breadboard

3.3.3 Servo Motor

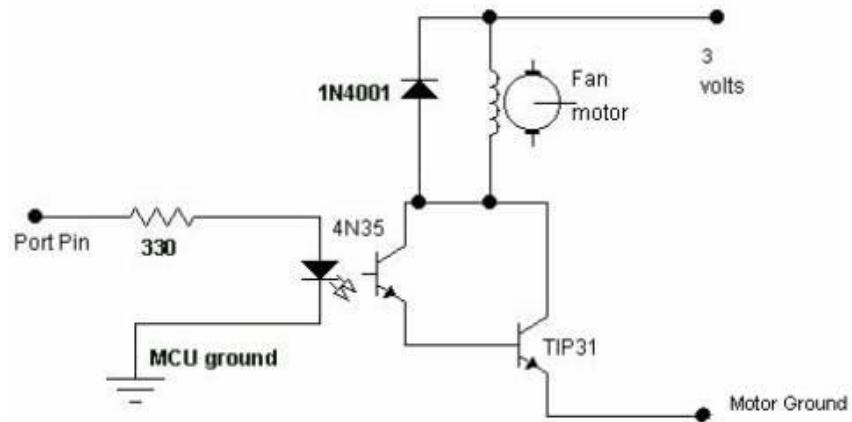


Figure 3.3: Circuite Diagram of Servo Motor [15]

3.3.4 LCD Display Pin diagram:

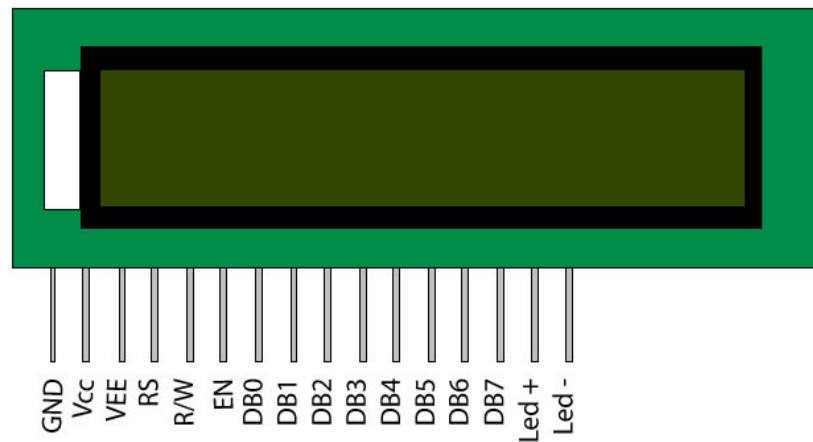


Figure 3.4: LCD Display Pin Diagram [16]

3.3.6 Arduino Sketch Software



Figure 3.5: Screenshot of Arduino Software

3.4 Block Diagram

3.4.1 Block diagram of PIR Sensor Connection

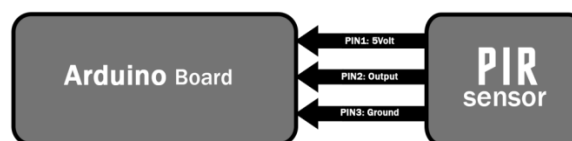


Figure 3.6: Block Diagram of PIR Sensor

3.4.2 Block diagram of Display Connections:

In here we used a PIR sensor and it is directly connected to the Arduino. In the PIR sensor there are 3 pins. First one is connected to the 5 volt of the Arduino Microcontroller second pin is connected to the pin 2 and 3rd one is connected to the ground of the Arduino Microcontroller.



Figure 3.7: Block Diagram of Display Connections

We used a display and it is also directly connected to the Arduino. In the display, 1 to 5 pins are connected with Arduino's 11 to 15 pins. First one is connected to the pin 13 of the Arduino Microcontroller and the 2nd one is connected to the ground of the Arduino Microcontroller.

3.4.3 Block diagram of LED Connection:

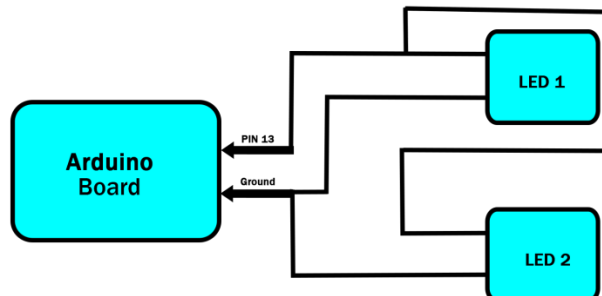


Figure 3.8: Block Diagram of 2 LED connected with Arduino

In this connection both of the Leds pin 1 is connected to the pin 13 of the Arduino and both of the Leds pin 2 is connected to ground of the Arduino Microcontroller.

3.4.4 Block diagram of 5V 2A Adapter:

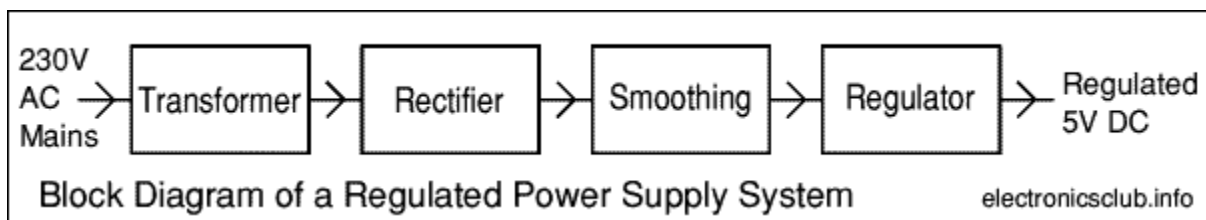


Figure 3.10: Block diagram of 5V 1A Adapter

5V 1A adapter is used for extra power supply for Arduino and Display so that the program run and display successfully.

3.4.5 Block Diagram of Servo Motor:

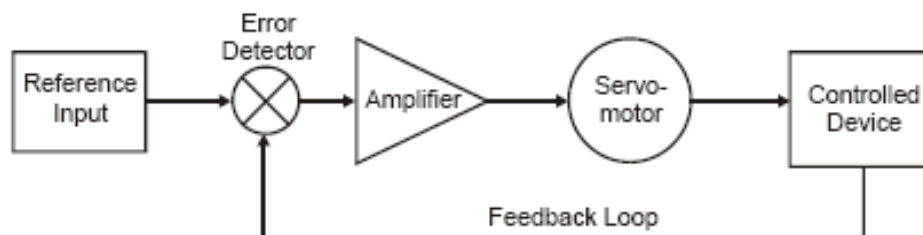


Figure 3-44. Typical servo system block diagram

Figure 3.11: Block diagram of Servo Motor ^[17]

3.5 Block Diagram of Full components

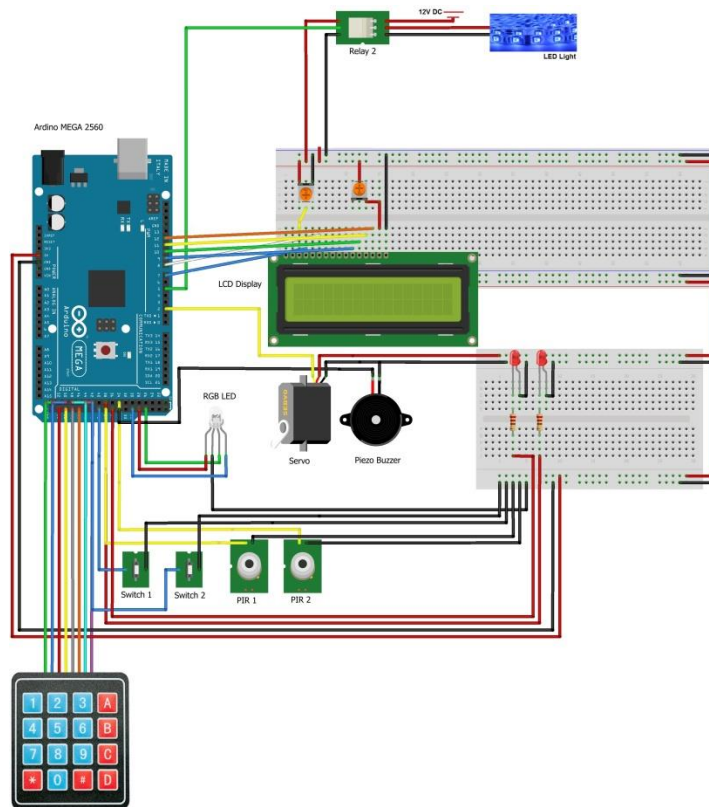


Figure 3.12: Block Diagram of full component connection [18]

To make up the system the separate components are combine together. we used a pir sensor and it is directly connected to the Arduino. In the PIR sensor there are 3 pin. First one is connected to the 5 volt of the Arduino Microcontroller second pin is connected to the pin 2 and 3rd one is connected to the ground of the Arduino Microcontroller. A servo motor and it is also directly connected to the Arduino. In the Servo Motor there are 2 pin. First one is connected to the pin 13 of the Arduino Microcontroller and the 2nd one is connected to the ground of the Arduino Microcontroller. We use two LED light, and both of the Leds pin 1 is connected to the pin 13 of the Arduino and both of the Leds pin 2 is connected to ground of the Arduino Microcontroller. In the port of relay NO and pole is connected the selfie stick switch and port A and B is connected to the pin 12 and ground respectively of the ArduinoMicrocontroller.A selfie stick supported smart phone is use for captured the picture of moving object. It is connected to the Relay.

Chapter-4

Implementation and Result

4.1 Implementation and Result

After all the parts connect as circuit design. Then we connect Arduino to the computer with data cable and upload the programming code as we compiled in the Arduino and we get positive result. At last we see that it works properly according to our design.

4.2 System in Idle Mode (Circuit design)

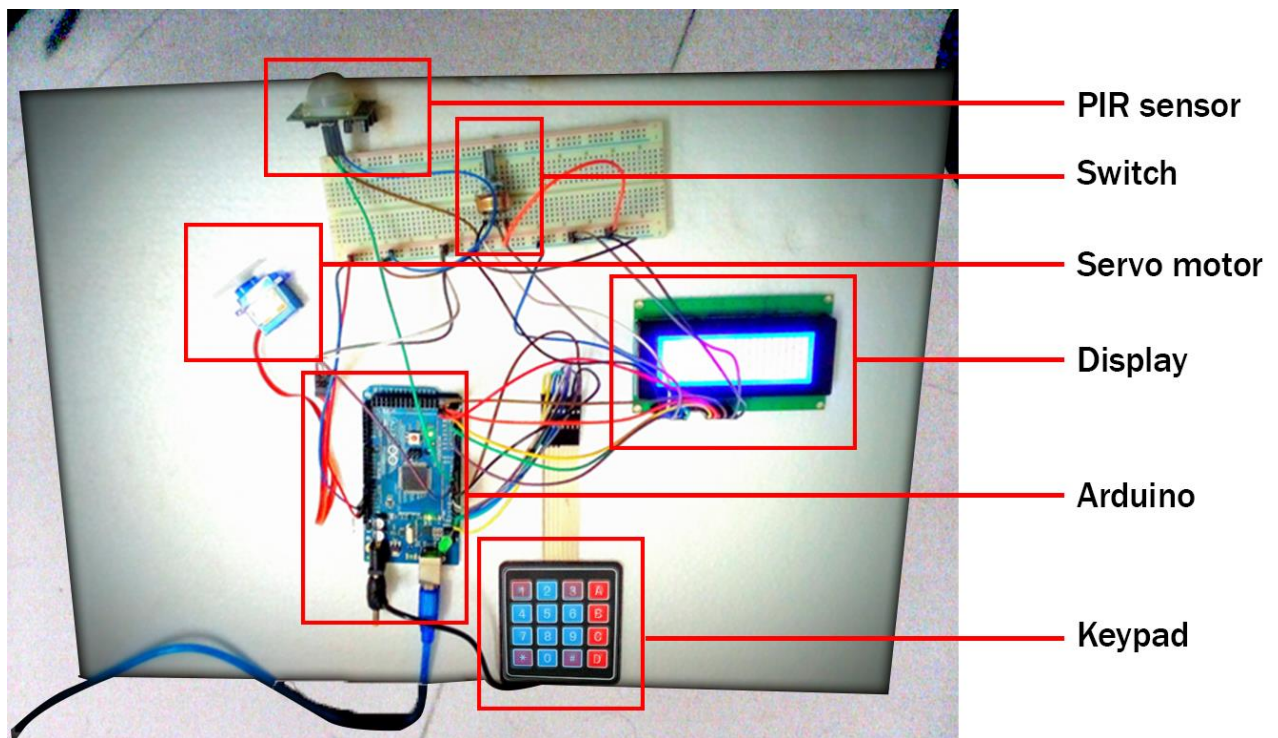


Figure 4.1: Ideal Mode Circuit Design

4.3 Practical set-up and Result

After designing the circuit we still had a lot of challenges ahead. In the above picture it is shown that in a home there is the system is applied for security purpose. When a moving object is moving in the sensor range it take that as system input signal. When getting the input signal the Led lights are lilt and an alarm is generated as well as the phone camera captured that moving object automatically. In the 1st picture there is no moving object and system is in idle mode. If a moving object is passing through the system and the LED are lits. As a result alarm is active. So user can easily find what happen on the home. So it has a great impact in the case of security.

Chapter- 05

CONCLUSION

5. CONCLUSION

Thus, we have designed a home security alarm system using Arduino and PIR motion sensor, which is handy, portable, cost-effective and highly effective as well. Such alarm systems are hugely in demand for security purposes, and thus the given system can be proved useful and effective in view of the above features. The Intelligent Home Security System using Keypad Lock has been designed and developed for making our life more easy and secured. We use 5v Arduino board and 5v-2A power adapter power supply for running the machine. We also use PIR sensor for detecting object receiving. Finally, we have designed and developed the whole control system and tested using keypad. We fix all the problems encountered during the design and testing of the system. The production cost is low. Hence, our product is suitable for commercial use.

Chapter- 06

FUTURE SCOPE

6. FUTURE SCOPE

We can determine the position of the intruder and then send a SMS to the concerned authorities. With home security as the most powerful driver in the adoption of smart home systems, it's no wonder that the world of smart locks are growing. The proposed system of Keypad lock and PIR sensor based home security system to detect the theft or any moving object. The future work of this system will include a centralized control for the group of particular house for current response for any kind unauthorized entry or illegal issues of system.

REFERENCES

Books

- [1] Michael Mcroberts, Beginning Arduino.
- [2] Introduction to Arduino: A piece of cake! by Alan G. Smith

Software

- [3] Arduino IDE 1.0.6 from <https://www.arduino.cc/en/main/software>.

World Wide Web

- [4] www.circuitstoday.com
- [5] www.engineersgarage.com
- [6] www.learn.adafruit.com
- [7] www.instructables.com

- [8] <http://arduino.cc/en/Guide/Introduction> Accessed July 15 2013
- [9] <http://arduino.cc/en/Main/ArduinoBoardUno> Accessed 17 July 2013
- [10] <http://www.atmel.com/Images/doc8161.pdf> Accessed 11 July 2013
- [11] <http://www.activexperts.com/mmserver/cellular/gprsintro/> Accessed 18 July 2013
- [12] <http://wm.sim.com/producten.aspx?id=1019> Accessed 18 July 2013
- [13] <http://www.codeproject.com/Articles/85636/Introduction-to-AT-commands-and-its-uses> Accessed 23 July 2013

- [14] http://www.mc.co.il/Media/Doc/TechnicalInformation/Temp_Measuring1.pdf Accessed 30 July 2013
- [15] www.farnell.com/datasheets/1537854.pdf Accessed 6 August 2013
- [16] http://sensing.honeywell.com/index.php?ci_id=47847 Accessed 7 August 2013.

- [17] <http://www.farnell.com/datasheets/1657826.pdf> Accessed 7 August 2013

Figure References:

- [1] <https://solarbotics.com/product/50450/>
- [2] <https://blog.arduino.cc/2013/07/10/send-in-the-clones/>
- [3] http://skjoy.info/wp-content/uploads/2016/11/arduino_variants.jpg
- [4] http://www.myduino.com/index.php?route=product/product&product_id=58
- [5] <http://www.robotoid.com/appnotes/sensors-passive-infrared.html>
- [6] <http://www.simplelabs.co.in/content/pir-sensor-motion-detection-sensor>
- [7] <http://www.instructables.com/id/A-Burglar-Alarm-Based-on-PIR-Sensor-With-Sms-Alert/>
- [8] <https://wiki.metropolia.fi/display/sensor/Passive+infrared+human+motion+detection+sensor>
- [9] <https://wiki.metropolia.fi/display/sensor/Passive+infrared+human+motion+detection+sensor>
- [10] <http://www.hobbyist.co.nz/?q=power-adapter>
- [11] http://www.electronics.com/store/prototyping-cable-wire-c-42_44.html
- [12] <https://theislandpond.com/2016/06/>
- [13] <http://atmega32-avr.com/arduino-mega-2560/>
- [14] <http://www.electroschematics.com/11972/motion-sensing-security-light/>
- [15] <https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2004/eyc9/BlindBot.htm>
- [16] <https://balajidany.wordpress.com/2015/08/19/16x2-display-i2c-interface-with-arduino/>
- [17] <http://www.globalspec.com/reference/10801/179909/chapter-3-ac-and-dc-motors-servomotors-general-principles-of-operation>
- [18] <http://elcrosst.com/arduino-alarm-system/arduino-home-security-system-youtube-alarm-pdf-maxresde/>

Table References:

- [1] www.arduino.cc/en/Main/Products
- [2] <https://wiki.metropolia.fi/display/sensor/Passive+infrared+human+motion+detection+sensor>

APPENDIXES

```
//=====
//Project Title : Home Security Alarm System
//Student Names : Zibon Kumer & Faridul Hasan
//Student ID NO.: 2012-2-55-045 & 2012-2-55-030
//Supervisor : Mr. Sarwar Jahan
//=====

#include <LiquidCrystal.h>
#include <Wire.h>
#include <Password.h> //http://www.arduino.cc/playground/uploads/Code/Password.zip
#include <Keypad.h> //http://www.arduino.cc/playground/uploads/Code/Keypad.zip
#include <Servo.h>
#include "RTCLib.h"

//Servo
Servo myservo;    // create servo object to control a servo
int pos = 90;    // variable to store the servo position
int passwd_pos = 15; // the position of the password input

//Real Time Clock
RTC_DS1307 RTC;

//Password
```

```
Password password = Password( "1234" );
```

```
const byte ROWS = 4; // Four rows
```

```
const byte COLS = 4; // Four columns
```

```
char keys[ROWS][COLS] = { // Define the Keymap
```

```
{  
  '1','2','3','A'  }
```

```
,
```

```
{  
  '4','5','6','B'  }
```

```
,
```

```
{  
  '7','8','9','C'  }
```

```
,
```

```
{  
  '*', '0', '#', 'D'  }
```

```
};
```

```
byte rowPins[ROWS] = {
```

```
  46, 47, 48, 49}; //connect to the row pinouts of the keypad
```

```
byte colPins[COLS] = {
```

```
  50, 51, 52, 53}; //connect to the column pinouts of the keypad
```

```
// Create the Keypad
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

```
// initialize the library with the numbers of the interface pins
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); // Assignign arduino pins to LCD display module
```

```
//Stroke LED Lights
```

```
int ledDelay = 50; // delay by 50ms
```

```
int redPin = 29;
```

```
int bluePin = 31;
```

```
//constants for LEDs, inputs and outputs
```

```
//int blueLED = 36;
```

```
int greenLED = 37;
```

```
int redLED = 38;
```

```
int pirPin1 = 39;
```

```
int pirPin2 = 34;
```

```
int reedPin1 = 41;
```

```
int reedPin2 = 42;
```

```
int speakerPin = 35;
```

```
//int relay1 = 3; //
```

```
int relay2 = 4; // connected to 12V Blue LED strip
```

```
int relay3 = 5; // Cam
```

```
int relay4 = 6; //
```

```
int alarmStatus = 0;

int zone = 0;

int alarmActive = 0;

void setup(){

  Serial.begin(9600);

  lcd.begin(20, 4);

  //Adding time

  Wire.begin();

  RTC.begin();

  //If we remove the comment from the following line, we will set up the module time and date with the
  computer one

  RTC.adjust(DateTime(__DATE__, __TIME__));

  myservo.attach(2); // attaches the servo on pin 2 to the servo object

  displayCodeEntryScreen();

  //Police LED Lights

  pinMode(redPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

  //setup and turn off both LEDs

  pinMode(redLED, OUTPUT);

  pinMode(greenLED, OUTPUT);

  pinMode(speakerPin, OUTPUT);
```

```
//pinMode(relay1, OUTPUT);  
pinMode(relay2, OUTPUT); //12V Blue LED lighting  
pinMode(relay3, OUTPUT); //camera, 5V external DC supply  
pinMode(relay4, OUTPUT); //  
  
pinMode(pirPin1, INPUT); //Bedroom 2  
pinMode(pirPin2, INPUT); //Garage  
pinMode(reedPin1, INPUT); //Front door  
pinMode(reedPin2, INPUT); //Back door  
  
digitalWrite(redLED, LOW);  
digitalWrite(greenLED, HIGH);  
digitalWrite(speakerPin, LOW);  
  
//digitalWrite(relay1, LOW); //  
digitalWrite(relay2, HIGH); // 12V Blue LED lighting  
digitalWrite(relay3, HIGH); // camera, 5V external DC supply  
digitalWrite(relay4, LOW); //  
  
keypad.addEventListener(keypadEvent); //add an event listener for this keypad  
myservo.write(pos);  
}  
  
void loop(){
```

```
//display time and date
DateTime now = RTC.now();

//DATE
lcd.setCursor(0,1);
lcd.print(now.month(), DEC);
lcd.print('/');
//We print the day
lcd.print(now.day(), DEC);
lcd.print('/');
//We print the year
lcd.print(now.year(), DEC);
lcd.print(' ');

//TIME
lcd.setCursor(13,1);
lcd.print(now.hour(), DEC);
lcd.print(':');
lcd.setCursor(16,1);
lcd.print(now.minute(), DEC);
//lcd.print(':');
//lcd.print(now.second(), DEC);
//delay(1000);

keypad.getKey();
```



```
//Serial.println(digitalRead(reedPin2));  
//Serial.println(digitalRead(pirPin));  
//Serial.println(digitalRead(pirPin2));  
if (alarmActive == 1){  
  if (digitalRead(pirPin1) == HIGH)  
  {  
    zone = 0;  
    alarmTriggered();  
  }  
  if (digitalRead(reedPin1) == LOW)  
  {  
    zone = 1;  
    alarmTriggered();  
  }  
  if (digitalRead(reedPin2) == LOW)  
  {  
    zone = 2;  
    alarmTriggered();  
  }  
  
  if (digitalRead(pirPin2) == HIGH)  
  {  
    zone = 3;  
    alarmTriggered();  
  }  
}
```

```
    }  
}  
  
//////////////////////////////////// Functions //////////////////////////////////////  
  
//take care of some special events  
void keypadEvent(KeypadEvent eKey){  
    switch (keypad.getState()){  
        case PRESSED:  
            if (passwd_pos - 15 >= 5) {  
                return ;  
            }  
            lcd.setCursor((passwd_pos++,0);  
            switch (eKey){  
                case '#':          //# is to validate password  
                    passwd_pos = 15;  
                    checkPassword();  
                    break;  
                case '*':          /* is to reset password attempt  
                    password.reset();  
                    passwd_pos = 15;  
            }  
            // TODO: clear the screen output  
            break;  
        default:  
            password.append(eKey);  
            lcd.print("*");  
    }  
}
```

```
    }  
  }  
}  
  
void alarmTriggered(){  
  int expected_pos;  
  int incr;  
  digitalWrite(speakerPin, HIGH);  
  digitalWrite(relay2, LOW);  
  digitalWrite(relay3, LOW);  
  digitalWrite(redPin, HIGH);  
  
  //  
  password.reset();  
  alarmStatus = 1;  
  // alarmActive = 0;  
  lcd.clear();  
  lcd.setCursor(0,2);  
  lcd.print(" SYSTEM TRIGGERED ");  
  lcd.setCursor(0,4);  
  if (zone == 1)  
  {  
    lcd.print(" Front Door Open ");  
    expected_pos = 65;  
    delay(1000);  
  }  
}
```

```
}  
  
if(zone == 0){  
    expected_pos = 40;  
    lcd.print("Motion in Bedroom 1 ");  
    delay(1000);  
}  
  
else if(zone == 2){  
    expected_pos = 10;  
    lcd.print("Backdoor Open");  
    delay(1000);  
}  
  
else if(zone == 3){  
    expected_pos = 145;  
    lcd.print("Motion in Garage");  
    delay(1000);  
}  
  
if (expected_pos > pos) {  
    incr = 1;  
} else {  
    incr = -1;  
}  
  
for (pos = pos; pos != expected_pos; pos += incr) {  
    myservo.write(pos);          // tell servo to go to position in variable 'pos'
```

```
    delay(5);           // waits 5ms for the servo to reach the position
  }

  /*
  for(pos = 0; pos < angle; pos += 1) // goes from 0 degrees to 180 degrees
  {
      // in steps of 1 degree
    myservo.write(pos);    // tell servo to go to position in variable 'pos'
    delay(20);             // waits 15ms for the servo to reach the position
  }
  for(pos = angle; pos >= 1; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);    // tell servo to go to position in variable 'pos'
    delay(20);             // waits 15ms for the servo to reach the position
  }
  */

  {
    StrokeLight();
  }
}

void StrokeLight(){           //Stroke LED Lights
  digitalWrite(redPin, HIGH); // turn the red light on
  delay(ledDelay); // wait 50 ms
  digitalWrite(redPin, LOW);  // turn the red light off
```

```
delay(ledDelay); // wait 50 ms
digitalWrite(redPin, HIGH); // turn the red light on
delay(ledDelay); // wait 50 ms
digitalWrite(redPin, LOW); // turn the red light off
delay(ledDelay); // wait 50 ms
digitalWrite(redPin, HIGH); // turn the red light on
delay(ledDelay); // wait 50 ms
digitalWrite(redPin, LOW); // turn the red light off
delay(ledDelay); // wait 50 ms
delay(10); // delay midpoint by 100ms
digitalWrite(bluePin, HIGH); // turn the blue light on
delay(ledDelay); // wait 50 ms
digitalWrite(bluePin, LOW); // turn the blue light off
delay(ledDelay); // wait 50 ms
digitalWrite(bluePin, HIGH); // turn the blue light on
delay(ledDelay); // wait 50 ms
digitalWrite(bluePin, LOW); // turn the blue light off
delay(ledDelay); // wait 50 ms
digitalWrite(bluePin, HIGH); // turn the blue light on
delay(ledDelay); // wait 50 ms
digitalWrite(bluePin, LOW); // turn the blue light off
delay(ledDelay); // wait 50 ms
}
```

```
void checkPassword(){ // To check if PIN is corrected, if not, retry!
```

```
if (password.evaluate())
{
  if(alarmActive == 0 && alarmStatus == 0)
  {
    activate();
  }
  else if( alarmActive == 1 || alarmStatus == 1) {
    deactivate();
  } }
else {
  invalidCode();
}}
```

```
void invalidCode() // display message when a invalid is entered
```

```
{
  password.reset();
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("INVALID CODE! LOL!");
  lcd.setCursor(5,2);
  lcd.print("TRY AGAIN!");
  digitalWrite(greenLED, LOW);
  digitalWrite(redLED, HIGH);
  delay(2000);
  digitalWrite(redLED, LOW);
}
```

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

void activate() // Activate the system if correct PIN entered and display message on the screen
{
    if((digitalRead(reedPin1) == HIGH) && (digitalRead(reedPin2) == HIGH)){
        digitalWrite(redLED, HIGH);
        digitalWrite(greenLED, LOW);
        digitalWrite(2, HIGH);
        lcd.setCursor(0,0);
        lcd.print("SYSTEM ACTIVE!");
        alarmActive = 1;
        password.reset();
        delay(2000);
    }
    else{
        deactivate(); // if PIN not corrected, run "deactivate" loop
    }
}

void deactivate()
{
    //digitalWrite(camera, LOW);
    alarmStatus = 0;
    digitalWrite(redLED, LOW);
```



```
digitalWrite(greenLED, HIGH);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" SYSTEM DEACTIVATED!");

digitalWrite(speakerPin, LOW);

alarmActive = 0;

password.reset();

delay(5000);

digitalWrite(relay2, HIGH);

digitalWrite(relay3, HIGH);

displayCodeEntryScreen();
}

void displayCodeEntryScreen() // Displaying start screen for users to enter PIN
{
  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Enter PIN:");

  lcd.setCursor(0,2);

  lcd.print("Home Security System");

  lcd.setCursor(0,3);

  lcd.print("By Jun Peng, TECH169");
}

*/ ----- Code End ----- */
```