

Design and Implementation of Smart Home Security System with Automatic Snapshot

Developed by:

Md. Nafiul Hasan

ID: 2011-1-55-029

A. Noman

ID: 2011-1-55-002



EAST WEST UNIVERSITY

Department of Electronics & Communication Engineering

East West University

Project Supervisor:

Dr. Md. Habibur Rahman

Professor,

Dept. of Electrical and Electronics Engineering University of Dhaka

Declaration

We declare that this project was completed under the supervision of our honorable and beloved Professor, **Dr. Md Habibur Rahman**. We declare that this is our original work and no part of it had been submitted elsewhere, either partially or fully, elsewhere for any award or degree. Any material reproduced in this project has been properly acknowledged.

Signature:

.....
Md.Nafiul Hasan

ID: 2011-1-55-029

.....
A. Noman

ID: 2011-1-55-02

Supervisor

Dr. Md. Habibur Rahman

Professor, Dept. of Electrical and Electronics Engineering

University Of Dhaka

APPROVAL

This is to certify that the Project titled as “Design and Implementation of Smart Home Security System with Automatic Snapshot” submitted to the respected members of the Board of Examiners of the Faculty of Engineering for partial fulfillment of the requirements for the degree of Bachelor of Science in Electronics & Telecommunications Engineering by the following students and has been accepted as satisfactory.

Submitted By:

Md. Nafiul Hasan

ID: 2011-1-55-029

A. Noman

ID: 2011-1-55-002

Supervisor
Dr. Md. Habibur Rahman
Professor,
Dept. of Electrical and Electronics Engineering
University of Dhaka

Chairperson
Dr.M. Mofazzal Hossain
Chairperson & Professor,
Dept. of Electronics and Communication
Engineering
East West University

ACKNOWLEDGEMENT

This project is made to stand upright firstly by the help of Almighty God; and secondly everyone, including our parents, teachers, friends and in essence, everyone who helped us in some way or another, be it financially or be it spiritually, in the completion of our work. We would also like to extend our thankfulness towards our advisors and contributors. We would like to express our special gratitude towards **Dr. Md. Habibur Rahman** for his utmost support and encouragement. He offered valuable detailed advices on the components that are used in this project. We would also like to thank the laboratory staff of ECE department for their fast response and cooperation to get the materials we needed for our project. Finally, we would like to say that this report would not come to fruition without the help of all of them.

ABSTRACT

Security has become an important issue everywhere. So the main aim of our project is to make a smart home security system with an automatic snapshot. In this smart home security system, we included some efficient features. When there is any motion detection by PIR sensor, then the camera take automatic snapshot and also the security system give us alarm which turns on the light. All of the control system has been developed using Arduino Uno R3 microcontroller development board. This security system can be suitable for home, office bank or any other place.

TABLE OF CONTENTS

Short Contents	Page
TITLE PAGE	01
DECLARATION	02
APPROVAL	03
ACKNOWLEDGEMENT	04
ABSTRACT	05
TABLE OF CONTENTS	06
LIST OF FIGURE	10
CHAPTER 1	
1.1 INTRODUCTION	11
1.2 Microcontroller	11
1.3 Arduino	12
1.4 History	12
1.5 Development	13
1.6 Evolution	13
CHAPTER 2	
Components Use in the Design	
2.1 Introduction	14
2.2 PIR SENSOR	14
2.2.1 Introduction	14
2.2.2 How PIR works	15
2.2.3 The sensor signal	17
2.2.4 Architecture stage for detecting motion	17
2.2.5 Conclusion	19
2.3 Solar power bank	19

2.3.1 What is power Bank	19
2.3.2 How does power banks work	20
2.3.3 What types of Power Banks are	21
2.3.4 How do we charge a Power Bank	21
2.4 Light Emitting Diode	21
2.4.1 What is LED	21
2.4.2 How does LED light work	22
2.5 One Channel 5v Relay	24
2.5.1 What is Relay	24
2.5.2 How does Relay work	25
2.6 Selfie Stick	27
2.7 Speaker	28
2.8 Development Boards	28
2.8.1 Arduinio Uno R3 Development Board	28
2.8.2 Technical specs	29
2.8.3 Documentation	30
2.8.4 Programming	30
2.8.5 Warnings	31
2.8.6 Differences with others boards	31
2.8.7 Power	31
2.8.7 Memory	32
2.8.8 Input and Output	32
2.8.9 Communication	33
2.8.10 Automatic (Software) Reset	33

Chapter 3

SYSTEM DESIGN AND ANALYSIS

3.1 Introduction	34
3.2 Block diagram	35
3.2.1 Block diagram of PIR Sensor Connection	35
3.2.2 Block diagram of Speaker Connection	35
3.2.3 Block diagram of LED Connection	36
3.2.4 Block diagram of one channel 5V relay	37
3.2.5 Combination of all component	37
3.3 Software Design	39
3.3.1 Installing Arduino	39
3.3.2 Verifying the Hardware	39
3.3.3 Arduino Language	39
3.3.4 Logical Diagram	39
3.3.5 Flow Chart	40

Chapter 4

Implementation and Result

4.1 Implementation and Result	41
4.1.1 System in Idle Mode (Circuit design)	41
4.1.2 System in Active Mode: (Circuit design)	42
4.1.3 Practical set-up and Result	43

Chapter –5

Discussion

5.1 Discuss	44
5.2 For research purpose	44
5.3 For security purpose	45
5.4 For save energy	45

Chapter 6

Conclusion and future work scope

6.1 Suggestion for future work	45
6.2 Difficulties that we faced	46
6.3 Conclusion	46
APPENDIX	47
Reference	50

LIST OF FIGURE

1. Figure 2.1: PIR sensor	14
2. Figure 2.2: Back side of PIR sensor	15
3. Figure 2.3: Principle of the PIR sensor	16
4. Figure 2.4: Architecture stage (1) of PIR sensor	17
4. Figure 2.5: Architecture stage (2) of PIR sensor	18
5. Figure 2.6: Solar power bank	20
7. Figure 2.7: LED light	22
8. Figure 2.8: Mechanism of LED light	23
9. Figure 2.9: Circuit design of a relay	24
10. Figure 2.10: Relay	25
11. Figure 2.11: How does relay work (a)	26
12. Figure 2.12: How does relay work (b)	26
13. Figure 2.13: Selfie stick with mobile phone	27
14. Figure 2.14: Speaker	28
15. Figure 2.15: Arduinio Uno R3	29
16. Figure 3.1: Block Diagram of PIR Sensor	35
17. Figure 3.2: Block Diagram of Speaker	35
18. Figure 3.3 Block Diagram of 2 LED connected with Arduino	36
19. Figure 3.4 Block Diagram of Relay connected with selfie stick & Arduino	37
20. Figure 3.5 Block Diagram of full component connection	38
21. Figure Fig3.6: Flow Chart	40
22. Figure 4.1: In the idle mode connection	41
23. Figure 4.2: In the Active mode connection	42
24 Figure 4.3: system on idle mode	43
25 Figure 4.4 system on Active mode	43

Chapter-1

Introduction

1.1 Introduction

A Smart Home security is one that provides its home owners comfort, security, energy efficiency (low operating cost) and convenience at all times, regardless of whether anyone is home or out side. “Smart Home security” is the term commonly used to define a residence that has appliances, lighting, audio and video or still image systems, camera system etc for security purpose and that are capable of communicating with one another or can be controlled automatically.

1.2 Microcontroller

A microcontroller is a compact microcomputer designed to govern the operation of embedded system in motor vehicles, robots, office machines, complex medical devices, mobile radio transceivers, vending machines, home appliances and various other devices. A typical microcontroller includes a processor, memory, and peripherals. The simplest microcontrollers facilitate the operation of the electromechanical systems found in everyday convenience items. Originally, such use was confined to large machines such as furnaces and automobile engines to optimize efficiency and performance. In recent years, microcontrollers have found their way into common items such as ovens, refrigerators, toaster, clock radios, and lawn watering systems. Microcomputers are also common in office machines such as photocopiers, scanners, fax machines and printers.

The most sophisticated microcontrollers perform critical functions, spacecraft, ocean-going vessels, life-support system and robots of all kinds. Medical technology offers especially promising future roles. For example, a microcontroller might regulate the operation of an artificial heart, artificial kidney, or other artificial body organ. Microcomputers can also function with prosthetic devices (artificial limbs). A few medical-science futurists have suggested that mute patients might someday be able, in effect, to speak out loud by thinking of the words they want to utter, while a microcontroller governs the production of audio signals an amplifier and loudspeaker.

Microcomputers enjoy immense popularity among electronics hobbyists and experimenters. Perhaps the most widely known and used of these devices belong to the PIC family, manufactured by Microchip Technology, Inc. of Chandler, Arizona. All devices in the PIC family come with a wide variety of development tools, are easy to find, remain relatively inexpensive, and have excellent documentation.

1.3 Arduino

Arduino is a software company, project, and user community that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

The project is based on microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can interface to various expansion boards (termed *shields*) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on a programming language named Processing, which also supports the languages C and C++.

The first Arduino was introduced in 2005, aiming to provide a low cost, easy way for novices and professionals to create devices that interact with their environment using sensor and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

1.4 History

Colombian student 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 ATmega128 microcontroller, an integrated development environment (IDE) based on Processing and library functions to easily program the microcontroller.

In 2005, Massimo Banzi, with David Mellis (then an IDII student) and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked (or copied) the Wiring source code and started running it as a separate project, called Arduino.

The Arduino's initial core team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis.

The name *Arduino* comes from a bar in Ivrea, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and king of Italy from 1002 to 1014.

Following the completion of the Wiring platform, its lighter, lower cost versions were created and made available to the open-source community. Associated researchers, including David Cuartielles, promoted the idea.

1.5 Development

Arduino is an open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the GNU General Public License, version 2.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested that the name "Arduino" be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the *Arduino* name by using *-duino* name variants.

1.6 Evolution

The Arduino project grew largely out of the “DIY” climate created by the burgeoning popularity of rapid-prototyping boards like PLC and Wiring, as well as in response to an increasing need of artists and designers to easily prototype interactive works. In fact, the immediate precursor to the Arduino was a custom-made wiring microcontroller created by the artist/ designer Hernando Barragan in 2004 for his masters thesis (at Institute Design Institute Ivrea), intended for use a “non-technical audience” of “artists, designers, and architects”. In 2005, the Arduino team was formed in Ivrea, Italy, consisting of Barragan, Massimo Banzi, David Cuartielles, Dave Mellis, Gianluca Marino, and Nicholas Zambetti. Their goal was to create an electronics prototyping platform which further simplified the wiring platform, making it as accessible as possible for non-technical users in the creative field. As a result, the Arduino incorporated the following characteristics : a programming environment based on Processing language (a programming language conceived by Ben Fry and Casey Reas, also conceived for artists/designers), the ability to program the board via a standard USB connection, and a low price point (starting at about \$35USD). The Arduino achieved rapid success even within its first two years of existence, selling in a quantity of more than 50000 boards. By 2009, it had spawned over 13 different incarnations, each specialized for different applications—for example, the Arduino LilyPad (for wearable technologies projects), the Arduino Mini (miniaturized for use in small interactive objects), and the Arduino BT (with built-in Bluetooth capabilities).

Chapter 2

Components Use in the Design

2.1 Introduction

The designed system consists of Arduino Uno R3 board, PIR sensor, selfie stick, solar power bank, 1 channel 5v relay, Speaker, 2 LED Light, Some Wire, Arduino Sketch Software and Selfie stick supported mobile phone. In this chapter, we are going to discuss briefly about the functions and characteristics of different component in the system.

2.2 PIR SENSOR

2.2.1 Introduction: Pyroelectric passive infrared (PIR) sensors are widely used in daily life. They are a key component in motion detection and can be used for security systems, automatic doors, or automatic light control. They are commonly used to detect humans. For example, when someone is detected in a specified area an alarm may be triggered or a specific room may be lit. Another common use is with regard to pets, for example, cat flaps. Everything on earth has infrared (IR) energy. 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. Passive IR sensors are the opposite of active IR sensors that emit energy, such as ultrasound, light, or microwaves. Active sensors are able to detect a change in the environment when the reflected emitted signal is disturbed. Active sensors consume more energy than passive sensors.

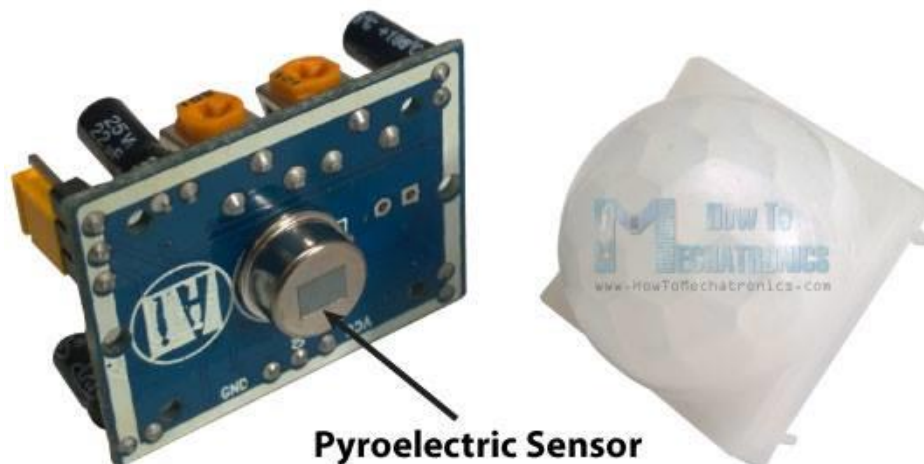


Fig 2.1: PIR sensor

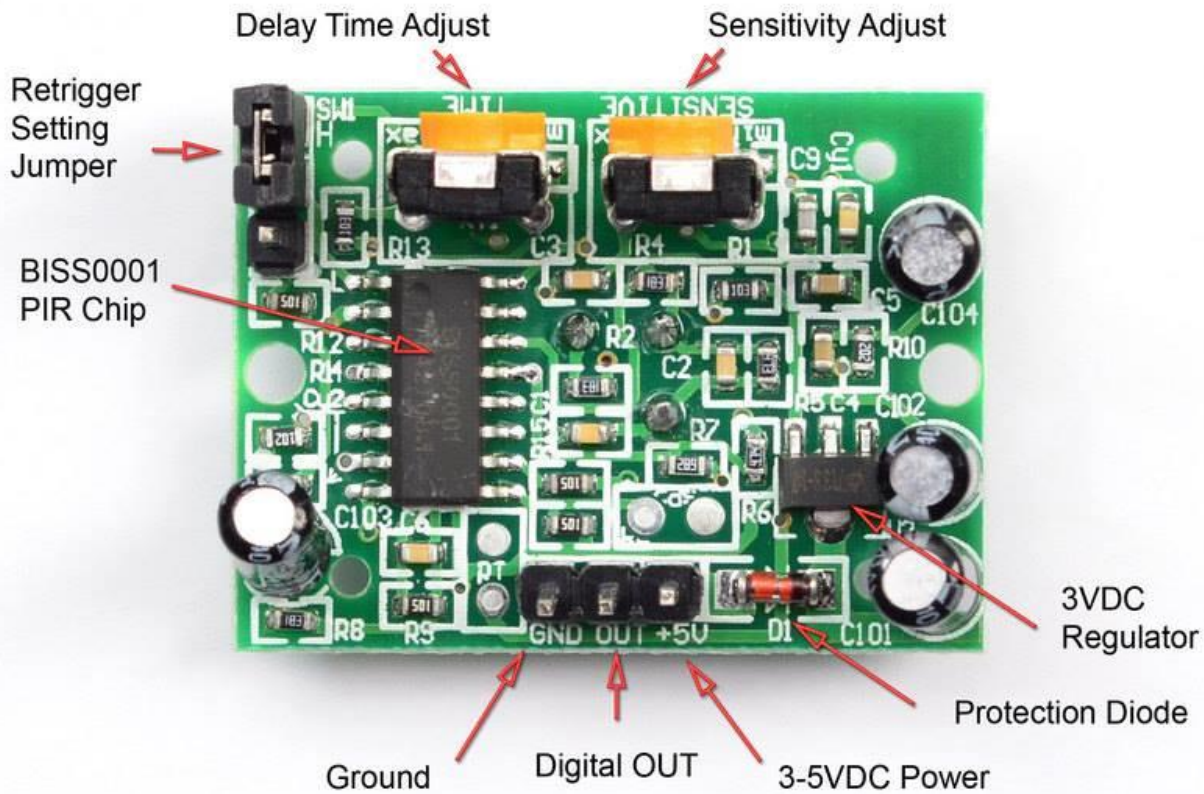


Fig 2.2: Back side of PIR sensor

2.2.2 How PIR Works: 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.

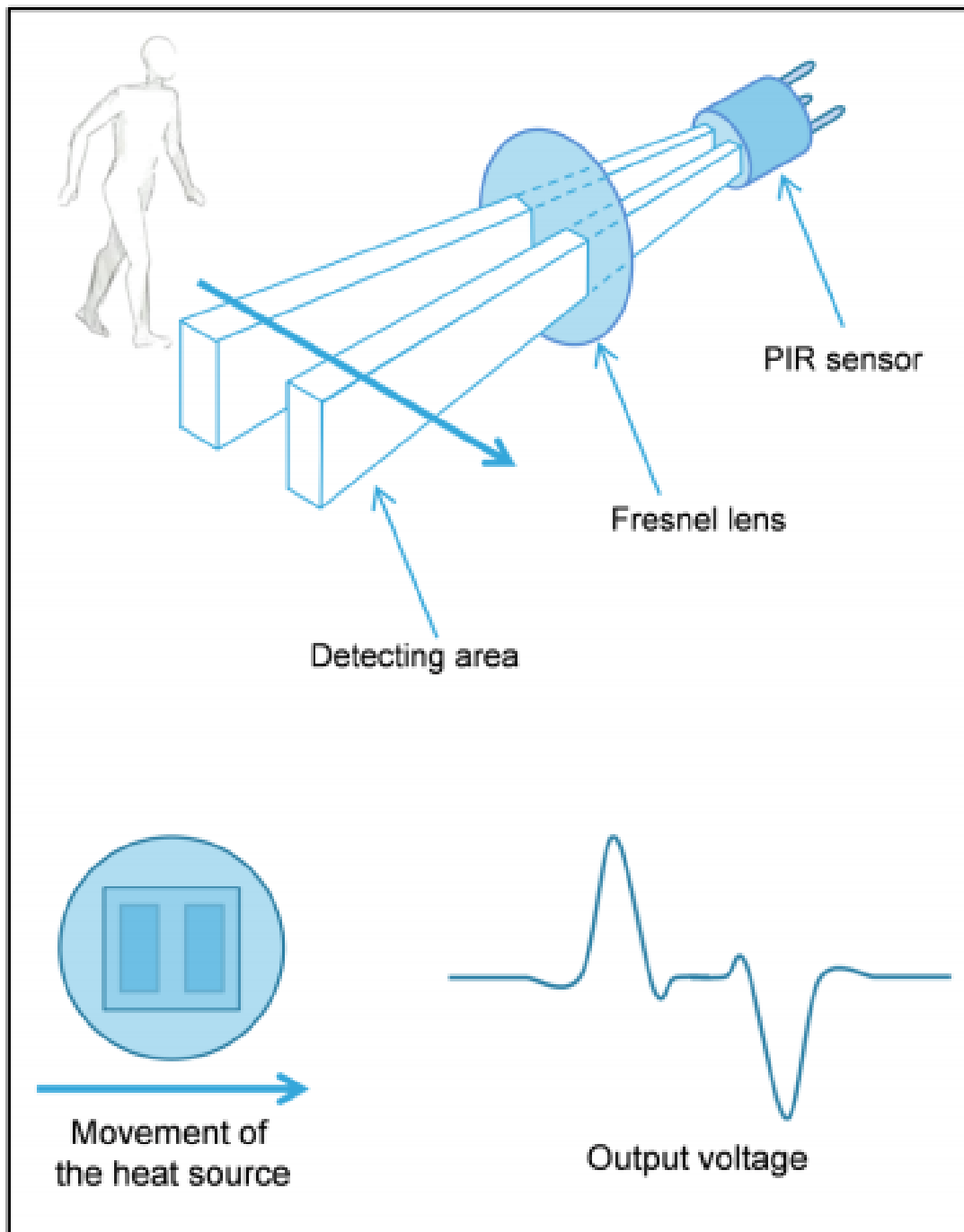


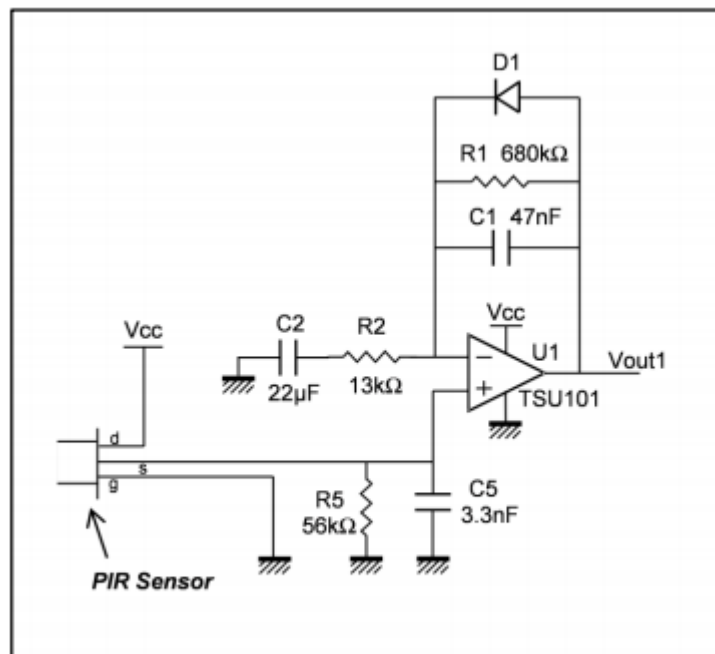
Fig 2.3: Principle of the PIR sensor

2.2.3 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.

2.2.4 Architecture stage 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

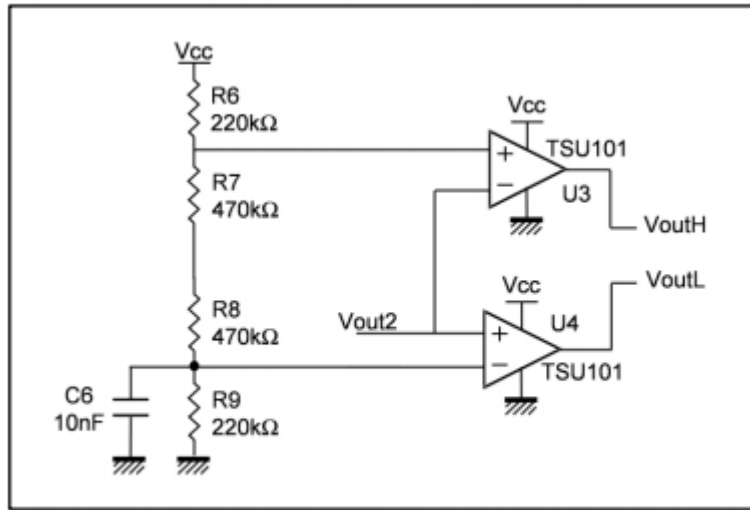
Fig 2.4: Architecture stage (1) of PIR sensor

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 $V_{cc}/2$). 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 $V_{cc} = 3.3$ 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

Fig 2.5: Architecture stage (2) of PIR sensor

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

$$\frac{R_7 + R_8 + R_9}{R_6 + R_7 + R_8 + R_9} \times V_{cc} = 0.84 \times V_{cc}$$

When the signal (V_{out2}) is above this reference (2.77 V if $V_{cc} = 3.3$ 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:

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

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 1: NAND truth table

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

Table 2.1: NAND truth table

2.2.5 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

2.3 Solar power bank

2.3.1 What is power Bank

Power bank is nothing but a simple system that contains a battery that stores energy inside of it and this system has two major portion of circuit. One being the circuit that gets the voltage from the battery, which in most cases is 3.7V-4.2V and bumps it up to 5V as most of these power banks have USB output port which has a standard voltage of 5V. The other portion is the charging circuit, remember most of these power banks use Lithium ion/polymer as their storage and they are pretty tricky to charge, you need to be careful with the current and voltage and also when to change the charging profile hence a sophisticated battery charging circuit is needed. There might be other bits and pieces inside which you can find more about in the link below. A solar power bank is a portable charging device that's equipped with solar panels. These power banks are growing in popularity, and there a variety of brands and types of power banks available on the market.



Fig 2.6: Solar power bank

2.3.2 How does power banks work

Power Banks are all the rage, they came in various shapes and sizes, but what are they for? We explore their potential, and how to choose the right one. What is a Power Bank and what can they charge? Portable Power Banks are comprised of a special battery in a special case with a special circuit to control power flow. They allow you to store electrical energy (deposit it in the bank) and then later use it to charge up a mobile device (withdraw it from the bank). Power Banks have become increasingly popular as the battery life of our beloved phones, tablets and portable media players is outstripped by the amount of time we spend using them each day. By keeping a battery backup close by, you can top-up your device(s) while far from a wall outlet. The Power Banks we're talking about are good for almost any USB-charged devices. Cameras,

Portable speakers, GPS systems, MP3 players, smartphones and even some tablets can be charged from a Power Bank - practically anything that charges from USB at home can be charged from a Power Bank - you just have to remember to keep your Power Bank charged, too! Power Banks may also be known as Power Stations or Battery Banks, too.

2.3.3 What types of Power Banks are

Three major types of Power Bank found on the market today:

1. Universal Power Bank. They come in many sizes and configurations which can be tailored to your device requirements and to your budget.
2. Solar-Charged Power Bank. They have photovoltaic panels which can trickle-charge the internal battery when placed in sunlight. Solar charging isn't fast, so they can usually charge via cable as well.
3. The third type of Power Bank is the older-style battery phone case. While they can be handy, this type of Power Bank has very narrow device compatibility,

2.3.4 How do we charge a Power Bank

Most commonly, a Power Bank will have a dedicated input socket for receiving power. But we used solar-charged power bank. This power bank have photovoltaic panels which can trickle-charge the internal battery when placed in sunlight. Solar charging isn't fast, so they can usually charge via cable as well power can come from a USB socket on your computer, but may charge faster when using a wall socket adapter.

2.4 Light Emitting Diode

2.4.1 What is LED

A light-emitting diode (LED) is a semiconductor device that emits visible light when an electric current passes through it. The light is not particularly bright, but in most LEDs it is monochromatic, occurring at a single wavelength. 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). An LED or IRED consists of two elements of processed material called P-type semiconductors and N-type semiconductors. These two elements are placed in direct contact, forming a region called the P-N junction. In this respect, the LED or IRED resembles most other diode types, but there are important differences. The LED or IRED has a transparent package, allowing visible or IR energy to pass through. Also, the LED or IRED has a large PN-junction area whose shape is tailored to the application.



Fig 2.7 LED light

2.4.2 How does LED Light Work

An LED is a small light bulb that easily fits into an electrical circuit. These light bulbs do not get hot and they are illuminated by the constant movements of the electrons located in the semiconductor material. The average life of an LED is around 50,000 hours, and it surpasses the life of the typical incandescent bulb by several tens of thousands of hours. Nowadays, you can find LED lights in a variety of applications, including LCD TVs. There is a strong reason why people are looking for cost-effective and energy-efficient lighting solutions for their homes: these lighting choices lower the utility bills and also take care of the problem of non-renewable energy sources. In the US alone, almost 90% of the energy relies on non-renewable sources. LEDs are by far the most efficient lights available on the market, and they have become so popular that they can be easily found in home-goods and hardware stores. Unfortunately, these innovative light bulbs are still a mystery to many people, because their mechanism is radically different than the one of standard light bulbs. Here you will find detailed information about how LEDs work and how are they able to generate constant and reliable lights without any maintenance.

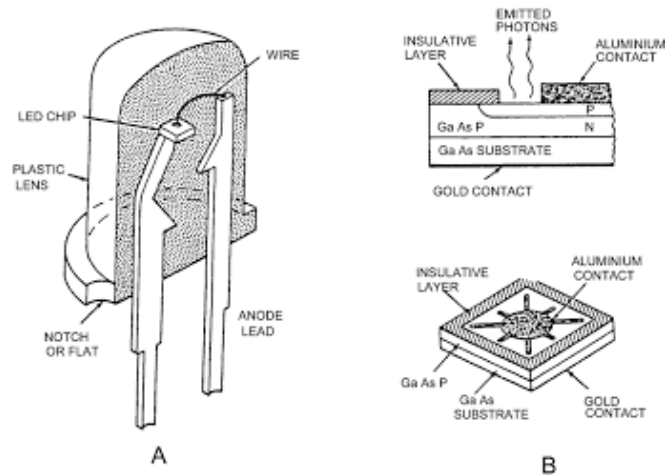


Fig 2.8: Mechanism of LED light

The answer lies in the name of the light bulb: LED stands for “Light Emitting Diodes” and it refers to a special type of diode that is especially created to provide light when electricity passes through it. These diodes are basically semiconductors that are used to control the direction of the electricity flow.

The average light emitting diode does this through what is known as a “p-n junction”, which is formed of a p-type semiconductor (which contains positively charged carriers) and an n-type semiconductor (which contains electrons). The connection between these two types of semiconductors is known as a “p-n junction”. These junctions are the building blocks of every semiconductor electronic device.

Light emitting diodes are modified in order to produce a form of light radiation, and the electrons from n-type semiconductors simply connect to the carriers in the p-type semiconductor, and the electricity flow is unidirectional: it is possible only from n to p, not the other way round. When the junction is formed, electrons located in the diode change their state and start to emit photons: electrons change their orbit by moving from a higher orbit to a lower one – when they do that, they automatically lose energy in the form of light. The more energy is lost, the more powerful the photons of light. This is when the color starts to change and light is being emitted by the LED.

The color that is emitted by the LED is somehow influenced by the material of the bulb as well as the current that runs through it. The semiconductor materials can be designed to limit the amount of energy that flows through them, and this will automatically change the color of the light – this is how you obtain a multi-color LED bulb. When the LED starts to produce light, this phenomenon is known as “electroluminescence”, and this is different from any other kind of light emission because it provides no heat. When the traditional light bulbs emit light, that process is called “incandescence” because they emit heat along with the light (thus the name “incandescent light bulbs”).

Generally speaking, LED lights are designed to emit one color of light, but over the past years manufacturers have come up with bicolor light bulbs, and the most popular colors at the moment are red, green and blue (although there is a wide variety of colored LED bulbs available on the market). In the case of large LED displays, the manufacturers use a combination of LEDs that aim to give a wide range of colors (these displays use a 3-LED pixel).

The improved intensity and the low cost have broadened the applications. The LED contains a pair of leads along with a plastic body that is encased around the leads and the semiconductor – this way, the light is emitted through the bulb plastic. However, most LED bulbs have a round shape that aims to focus the light and focus it in one particular direction. A regular LED bulb does not depend on the voltage and it requires no maintenance over the years.

2.5 One Channel 5V Relay

2.5.1 What is Relay

A **relay** is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

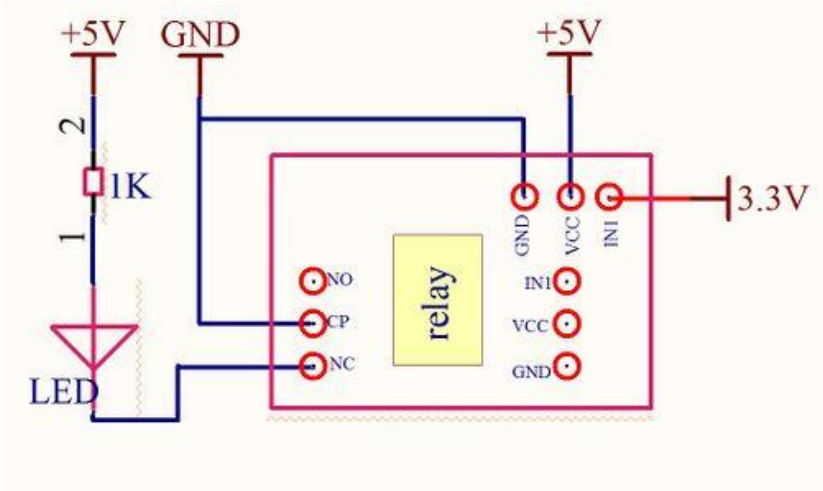


Fig 2.9: Circuit design of a relay

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from

overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

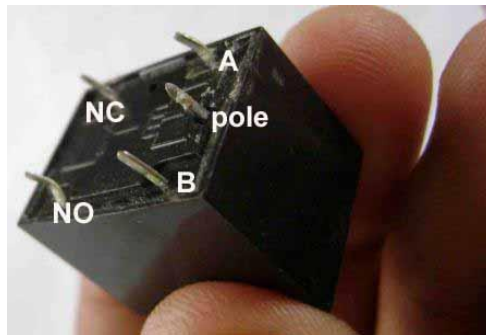


Fig 2.10: Relay

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts.

Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands.

2.5.2 How does Relay work

Before extending to the various types of relays, i will first explain what and how the basic relay operates. Each relay has two mechanical parts inside. The first one is the contact(s) of the relay. The contacts operates similarly to the contacts of a simple switch or pushbutton. You should consider the contacts as a pair of metals like the following diagram:

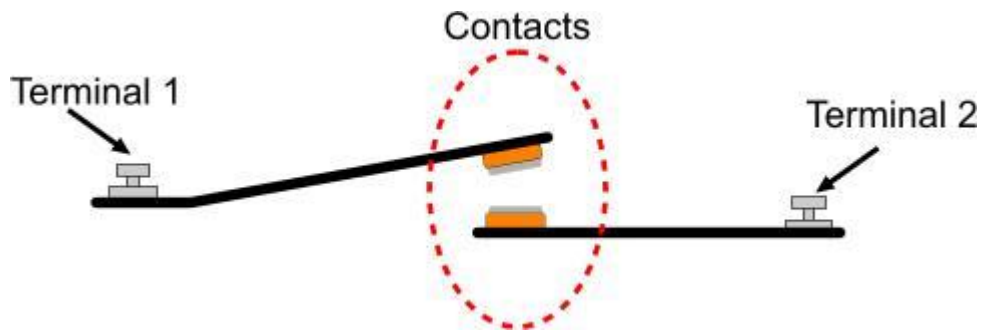


Fig 2.11: How does relay work (a)

The two terminals operates as a switch. When the contacts are 'in contact' then the current flows from Terminal 1 to Terminal 2. There are two types of contacts: the NO and the NC. NO stands for **N**ormal **O**pen contact, while NC stands for **N**ormal **C**losed contact. The Normal Open is a contact like the one showed in the previous illustration. When the contact is still, then no current flows through it (because it is an OPEN circuit). On the other hand, a Normal Closed contact allows the current to flow when the contact is still. Bellow i illustrate both of these contacts:

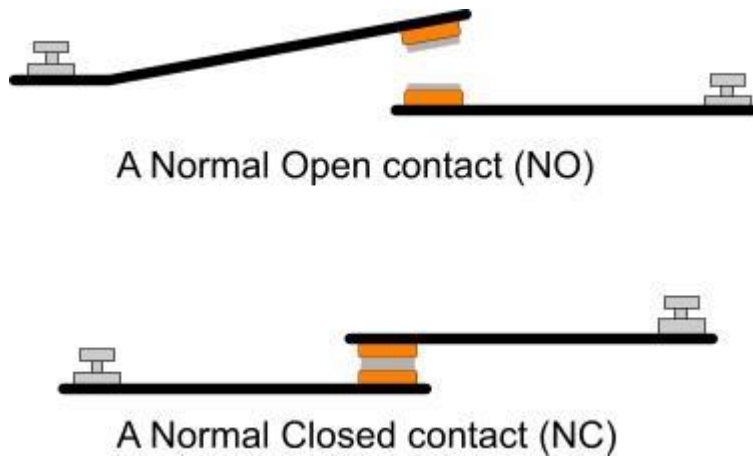


Fig 2.12: How does relay work (b)

You may notice that the NC contact is turned upside-down compared to the NO contact. This is done in purpose. This way, both contacts (NO and NC) will change state if a force is applied to the left metal heading from UP to DOWN. The following animation shows how a NO contact operates by lighting a light bulb:

2.6 Selfie stick

A selfie stick is a monopod used to take selfie photographs by positioning a smartphone or camera beyond the normal range of the arm. The metal sticks are typically extendable, with a handle on one end and an adjustable clamp on the other end to hold a phone in place. Some have remote or Bluetooth controls, letting the user decide when to take the picture, and models designed for cameras have a mirror behind the view screen so that the shot can be lined up. In contrast to a monopod for stabilizing a camera on the ground, a selfie stick's arm is thickest and strongest at the opposite end from the camera in order to provide better grip and balance when held aloft.



Fig 2.13: Selfie stick with mobile phone

2.7 SPEAKER

All sound speakers have a permanent magnet located behind the center of cone, no matter what size or how powerful the speaker is, to produce the sound by interacting with the changing magnetic field, driving the cone. Speaker manufacturers have been searching a long time for better magnet in speaker driver to achieve the best of tone quality in different ranges, such as AlNiCo, Ferrite, Neodymium, SmCo etc.



Fig 2.14: Speaker

2.8 Development Boards

2.8.1 Arduino Uno R3 Development Boards

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 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.

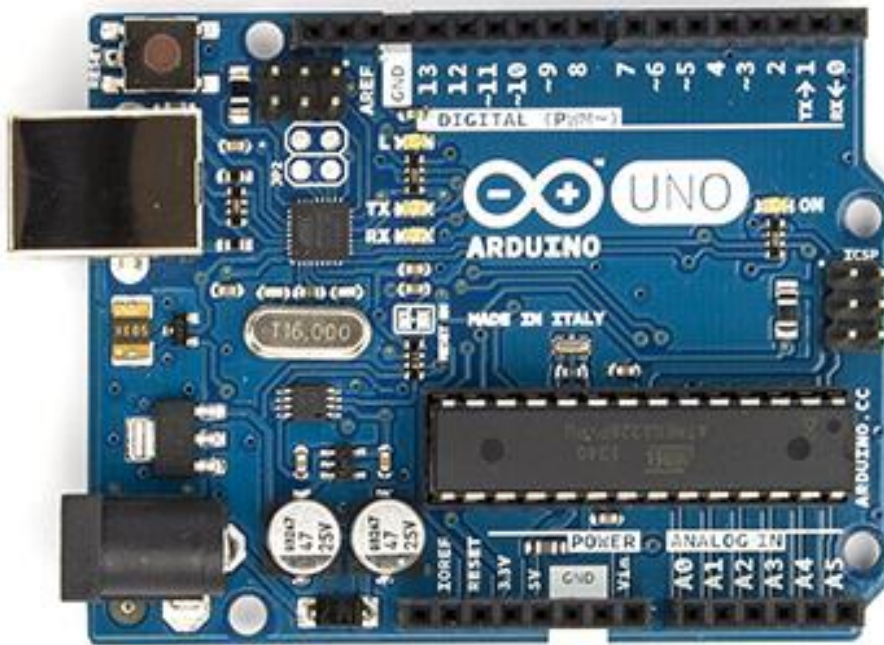


Fig 2.15: Arduino Uno R3

"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.

2.8.2 Technical specs

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6

Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

2.8.3 Documentation

The Uno is open-source hardware! You can build your own board using the following files:

2.8.4 Programming

The Uno can be programmed with the Arduino Software (IDE). Select "Arduino/Genuino Uno" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Uno comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then rese ing the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

2.8.5 Warnings

The Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

2.8.6 Differences with other boards

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

2.8.7 Power

The Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin.The input voltage to the Uno board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- GND. Ground pins.

- IOREF. This pin on the Uno board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

2.8.7 Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

2.8.8 Input and Output

See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with `analogReference()`.

- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.8.9 Communication

The Uno has a number of facilities for communicating with a computer, another Uno board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

2.8.10 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of

the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

Chapter 3

SYSTEM DESIGN AND ANALYSIS

3.1 Introduction

In this chapter helps those people who interested to build this project. To Design this project include into two parts, one is hardware design and another part is software design .In this chapter we discuss about those.

3.2 Block diagram

3.2.1 Block diagram of PIR Sensor Connection

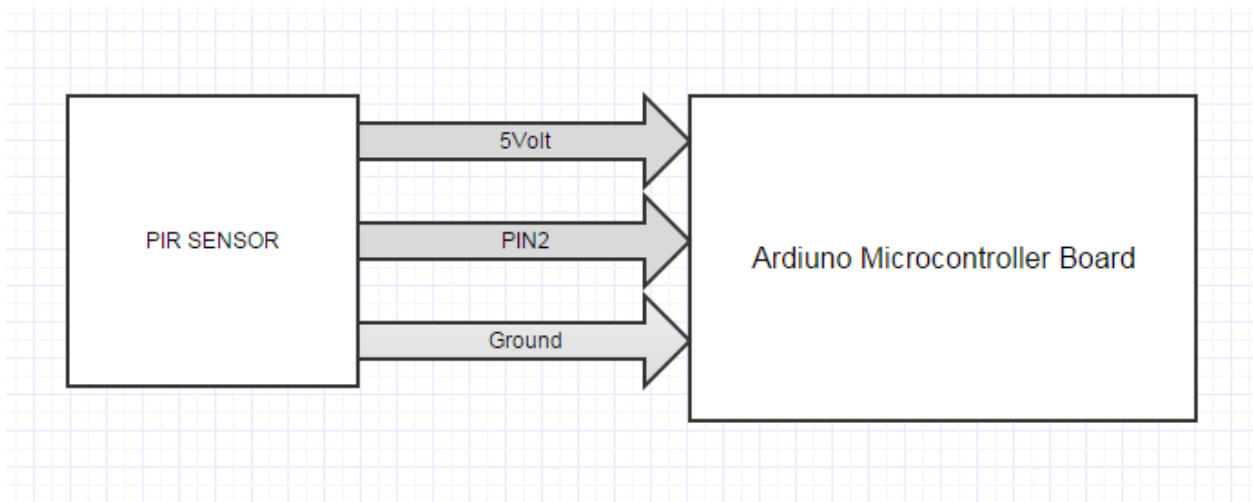


Figure 3.1: Block Diagram of PIR Sensor

3.2.2 Block diagram of Speaker Connection

In here 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.

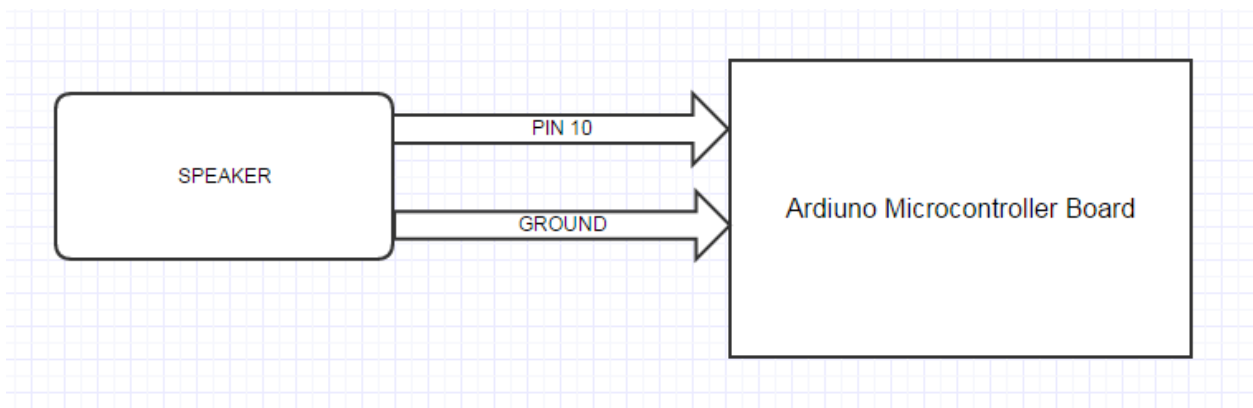


Figure 3.2: Block Diagram of Speaker

we used a speaker and it is also directly connected to the Arduino. In the Speaker 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.

3.2.3 Block diagram of LED Connection

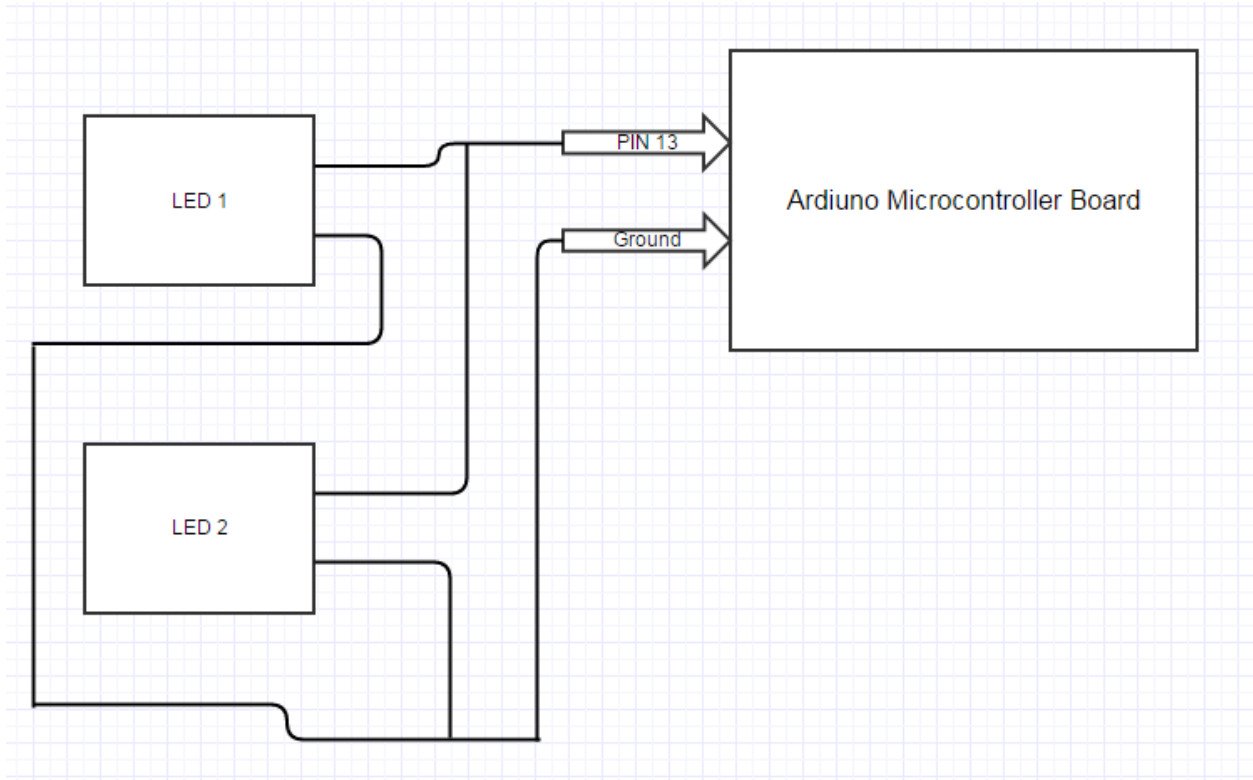


Figure 3.3: Block Diagram of 2 LED connected with Arduino

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.

3.2.4 Block diagram of one channel 5V relay

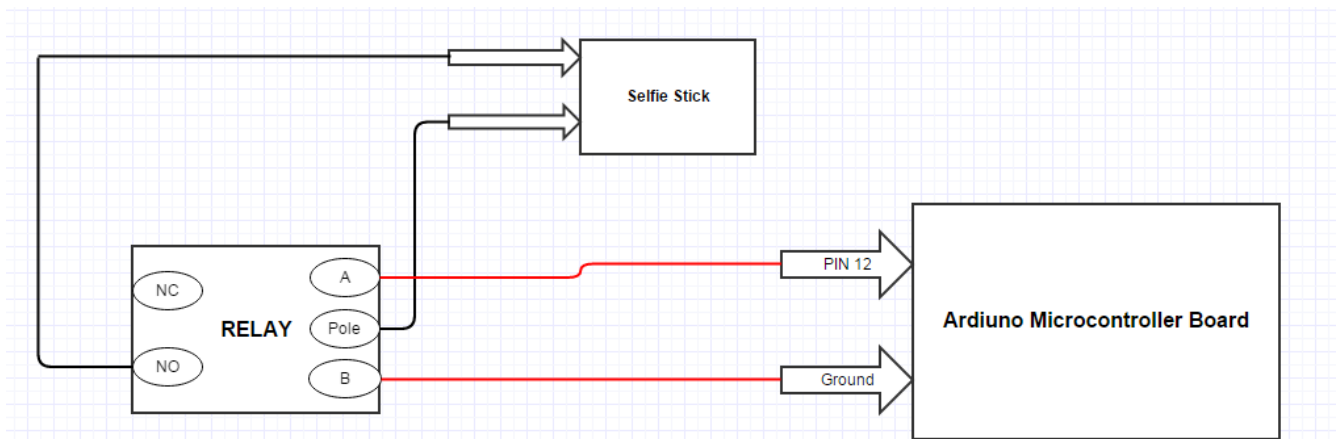


Figure 3.4: Block Diagram of Relay connected with selfie stick and Arduino

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 Arduino Microcontroller

3.2.5 Combination of all component

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 speaker and it is also directly connected to the Arduino. In the Speaker 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

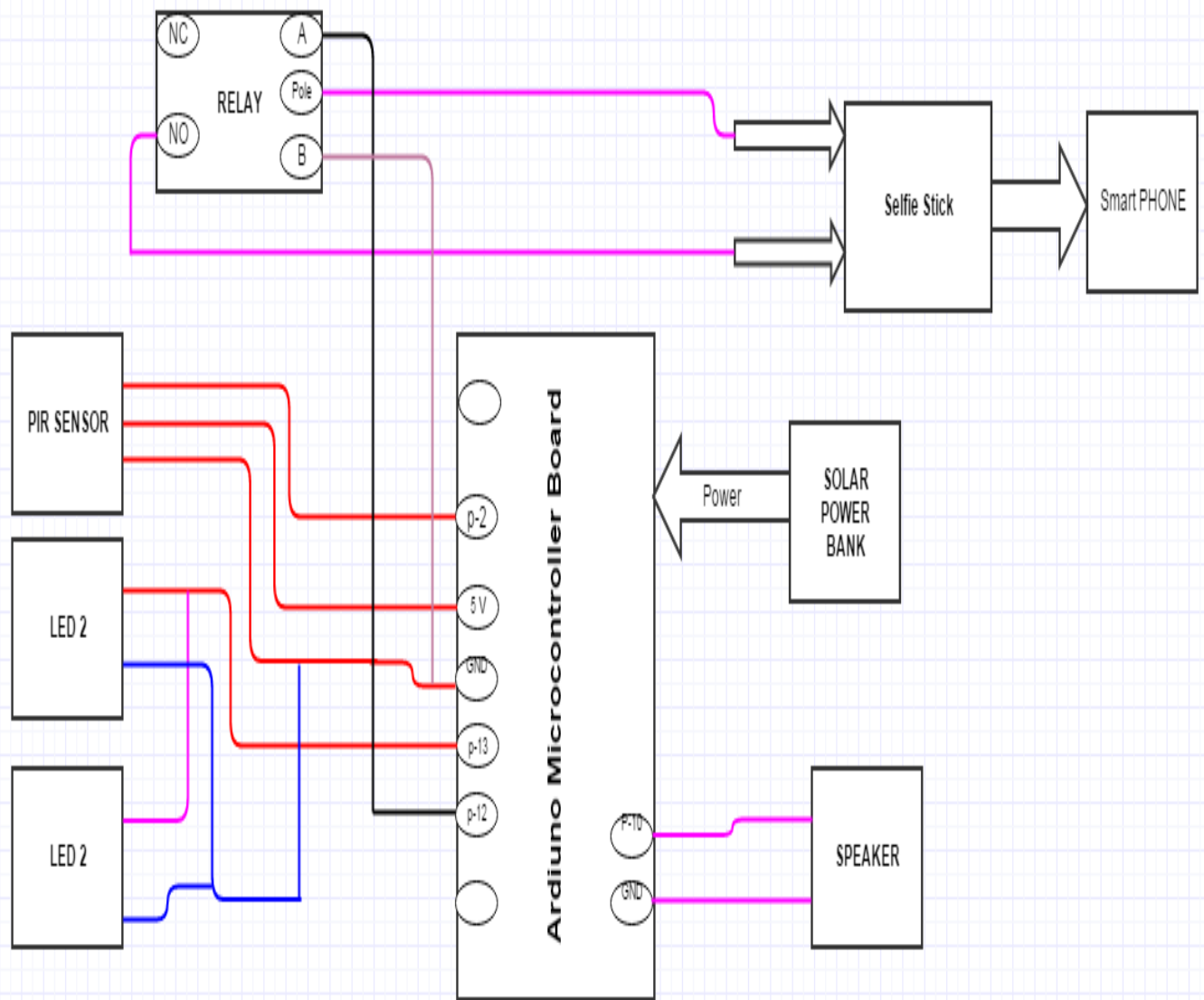


Fig 3.5: Block Diagram of full component connection

and pole is connected the selfie stick switch and port A and B is connected to the pin 12 and ground respectively of the Arduino Microcontroller. A selfie stick supported smart phone is use for captured the picture of moving object. It is connected to the Relay.

3.3 Software Design

Software design is divided into two parts. First we write the Arduino program in Arduino software. Then we compile it to the Arduino hardware. This Arduino command is control the Arduino hardware and other circuit connection. For making connection between Arduino and android mobile we need to install Arduino Bluetooth controller software on our mobile phone.

3.3.1 Installing Arduino

Arduino runs on Windows. Go to the Arduino software web site at <http://arduino.cc/en/Main/Software> and download the version of the software compatible with our system. We use Arduino 1.0.5 version.

3.3.2 Verifying the Hardware

Now that we have the Arduino IDE software installed, let's connect the computer to the mc board, load a small program, and verify that all components are working together. First, need to connect the USB cable to our mc board and then plug the other end of the USB cable into our computer.

3.3.3 Arduino Language

The Arduino language is implemented in C/C++ and based in Wiring. When we write an Arduino sketch, we are implicitly making use of the Wiring library, which is included with the Arduino IDE. This allows us to make run able programs by using only two functions: setup () and loop (). As mentioned, the Wiring language is inspired by Processing, and the Arduino language structure is inherited from the Processing language, where the equivalent functions are called setup (). We need to include both functions in every Arduino program, even if we don't need one of them. Let's analyze the structure of a simple Arduino sketch using again the Blink example.

3.3.4 Logical Diagram

We have a logical diagram for this project. Here android mobile phone works as a input signal which start from beginning part. We have got output from electronic devi

3.3.5 Flow Chart

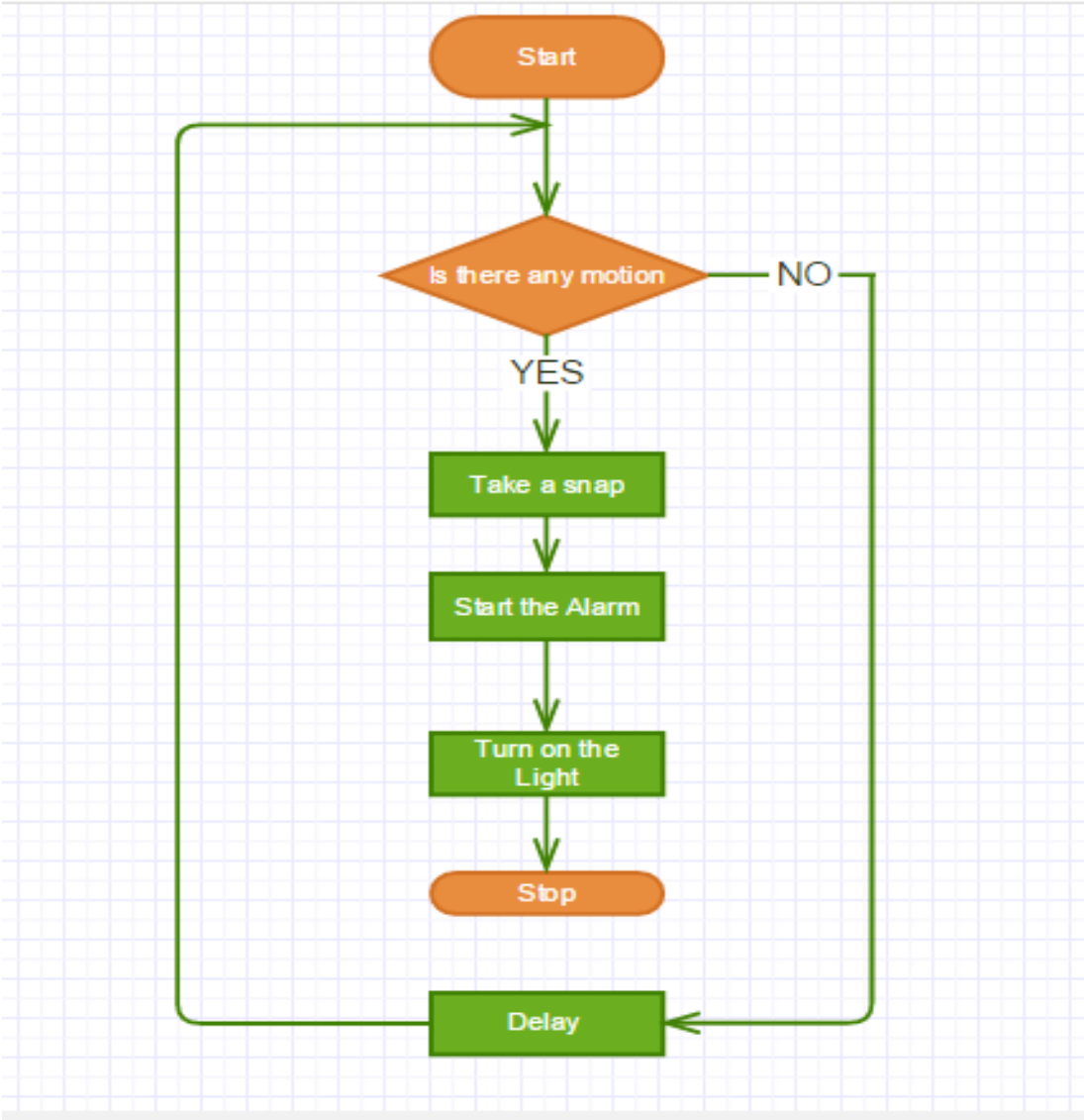


Fig3.6: Flow Chart

Chapter 4

Implementation and Result

4.1 Implementation and Result

All the parts are connected as circuit design. Then we upload the programming code in the Arduino and we get positive result. It works properly according to our design.

4.1.1 System in Idle Mode (Circuit design)

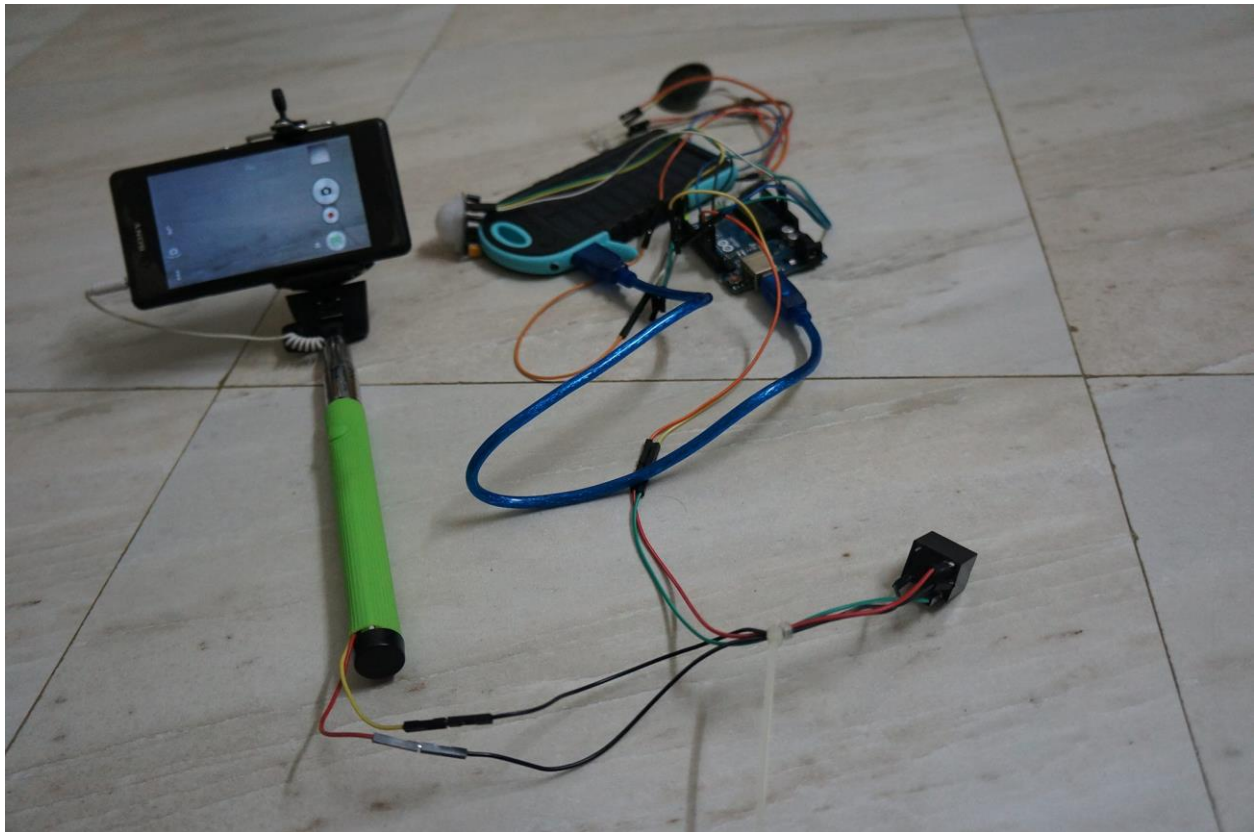


Fig 4.1: In the idle mode connection

4.1.2 System in Active Mode (Circuit design)

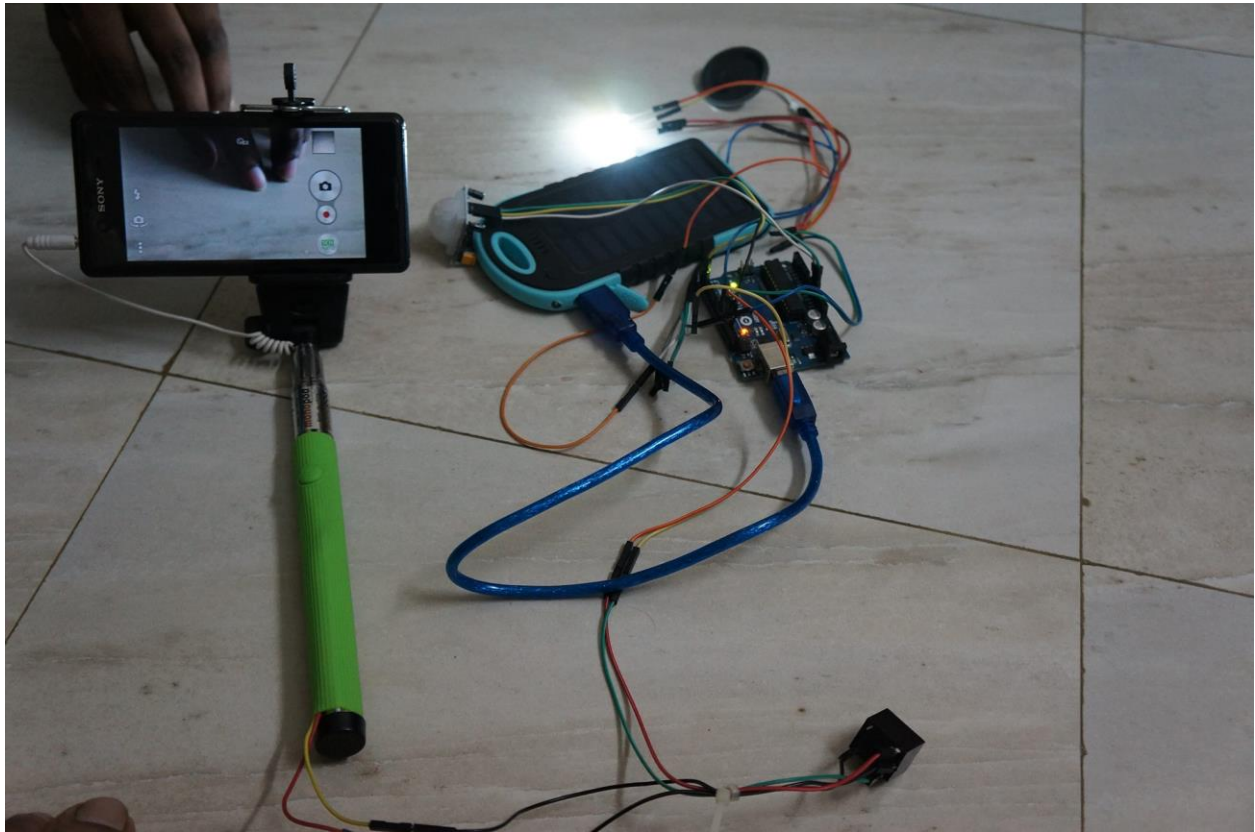


Fig 4.2: In the Active mode connection

When we uploaded the program in arduino and connected the circuit for observing the system in fig 4.1, there was no alarm and no snapshot was taken by camera because no motion was detected by the sensor. On the other hand, the sensor in figure 4.2 detected motion and we got alarm light and snapshot.

4.1.3 Practical set-up and Result



Fig 4.3: system on idle mode

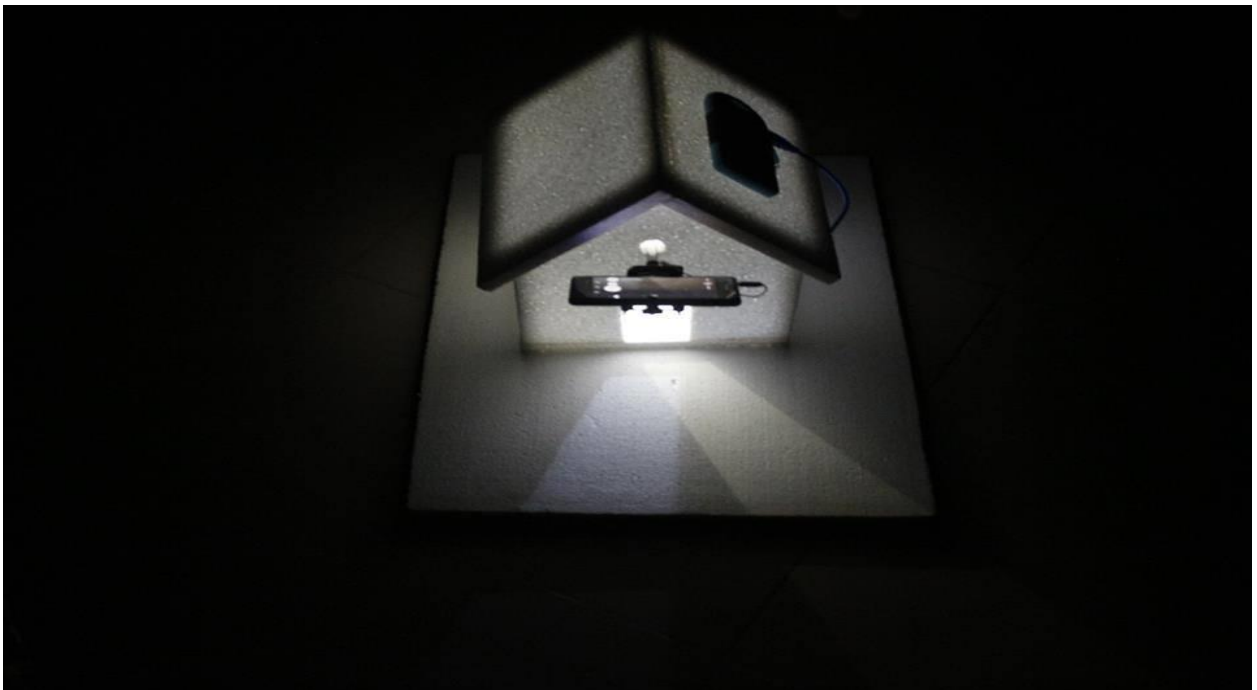


Fig 4.4: system on Active mode

After designing the circuit we setup the circuit in a demo home which is shown in the figure. In the above figure, it is shown that a system is setup for security purpose in a home. When an object moves in the sensor range, then an input signal is generated in the system. When the input signal is generated, the LED lights are lilt, an alarm is generated and the phone camera captures that moving object automatically. In figure 4.3, there was no moving object and system is in idle mode, but in figure 4.4, a moving object is passes through the system and the LEDs are lits. As a result, the alarm is active. In this way, a user may easily find out what happens on his home. So, it has a great impact in the security of a house.

Chapter –5

Discussion

5.1 Discuss

The smart home security system using mobile phone camera for snapshot has been designed for a demo version. We use 5V solar power supply for relay correction, we use PIR sensor for detection moving object and also use LED Light and speaker. Finally, we have designed and developed the whole control system and tested using smart mobile camera. We fixed all the problem's encountered during the design and testing of the system. Finally, we successfully achieved our goals.

5.2 For research purpose

We can add IP Camera which transfer videos through internet system for users who may need it for research. We can also add some sensors like smoke, fire, heat light etc. Users may choose which ones they need, they may be a limitation on number of features depending on the

headwear will get uploaded too with better security and more expanded ability. But anyway, this will make up for a decent security system for research purpose.

5.3 For security purpose

With so many cool features and applications, security system serves many different purposes and provides some great benefits. Smart home systems provide exceptional security solutions which will make you, and your family, feel more in control of your home.

5.4 For save energy

Automatic security system manage our home energy consumption to help lower the energy bills.

Chapter 6 Conclusion and future work scope

6.1 Suggestion for future work

We tried to make a smart home security system but because of insufficient time, we could not implement of some parts which make a home more secured. If anyone is interested to make to security smarter, they can added smoke, fire, heat detection sensor etc. They can also use automatic water tank system. If an internet connected camera is connected to the system, then the whole house can be observed even when nobody's at home.

6.2 Difficulties that we faced

We started from ground zero level with no knowledge on anything. Then, we tried to implement an idea even though there were some difficulties. As all elements of this project were not available in local market, we could not manage TTL serial camera. Finally, we took help from our advisor and internet.

6.3 Conclusion

However, we are happy to complete the project successfully against all odds. We sorted out all the problems and were fortunately able to fix them all. As we were successful to achieve our primary goals, we are opting for the future advancement of the project. We hope that, we will be able to complete all the features needed for more smart home security system.

APPENDIX

Programming Code for Whole System

```
int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;         // choose the input pin (for PIR sensor)
int snapPin = 12;
int pirState = LOW;      // we start, assuming no motion detected
int val = 0;             // variable for reading the pin status
int pinSpeaker = 10;     //Set up a speaker on a PWM pin (digital 9, 10, or 11)

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare sensor as input
  pinMode(pinSpeaker, OUTPUT);
  pinMode(snapPin,OUTPUT);
  Serial.begin(9600);
}

void loop(){
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {          // check if the input is HIGH
    digitalWrite(ledPin, HIGH); // turn LED ON
    digitalWrite(snapPin,HIGH);
    playTone(300, 160);
    delay(150);
    digitalWrite(snapPin,LOW);
```

```

if (pirState == LOW) {
    // we have just turned on
    Serial.println("Motion detected!");
    // We only want to print on the output change, not state
    pirState = HIGH;
}
} else {
    digitalWrite(ledPin, LOW); // turn LED OFF
    playTone(0, 0);
    delay(300);
    if (pirState == HIGH){
        // we have just turned off
        Serial.println("Motion ended!");
        // We only want to print on the output change, not state
        pirState = LOW;
    }
}
}

// duration in mSecs, frequency in hertz
void playTone(long duration, int freq) {
    duration *= 1000;
    int period = (1.0 / freq) * 1000000;
    long elapsed_time = 0;
    while (elapsed_time < duration) {
        digitalWrite(pinSpeaker,HIGH);
        delayMicroseconds(period / 2);
        digitalWrite(pinSpeaker, LOW);
    }
}

```



```
    delayMicroseconds(period / 2);  
    elapsed_time += (period);  
  }  
}
```

Reference:

1. <http://internetofthingsagenda.techtarget.com/definition/microcontroller>
2. <https://en.wikipedia.org/wiki/Arduino>
3. <http://www.kenleung.ca/portfolio/arduino-a-brief-history-3/>
4. <https://www.arduino.cc/en/Main/ArduinoBoardUno>
5. <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>
6. <http://howtomechatronics.com/tutorials/arduino/how-pir-sensor-works-and-how-to-use-it-with-arduino/>
7. <http://www.robotoid.com/appnotes/sensors-passive-infrared.html>
8. https://en.wikipedia.org/wiki/Light-emitting_diode
9. http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_2151486_-1
10. <http://makezine.com/projects/pir-sensor-arduino-alarm/>

