ARDUINO Microcontroller and Associated Peripherals

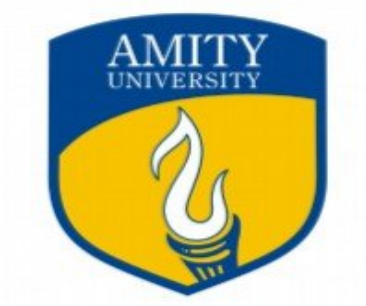
A TRAINING REPORT SUBMITTED TO
AMITY UNIVERSITY, GURGAON
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
Electronics and Communication

By

Ankita Sakhuja
(ROLL NO. **A50105111078**)

Under the esteemed guidance of
Mr. Meharban Singh
Director
Optimus Research Labs Pvt Ltd , New Delhi



Department of Electronics and Communication Engineering
Amity School of Engineering and Technology
AMITY UNIVERSITY, HARYANA, INDIA
SEPTEMBER-2014

DECLARATION

I **Ankita Sakhuja ,** student of **Bachelor of Technology (Electronics and Communication), 7th** Semester **Enrolment No. A50105111078** hereby declare that the training work entitled **" Arduino microcontroller and associated peripherals "** which is being submitted to Department of Electronics and Communication Engineering, AMITY UNIVERSITY, HARYANA is my authentic work record carried out at Lab during my project session**.**

Date: 22 Sep 2014 Ankita Sakhuja

Roll. No.A50105111078

OPTIMUS
TECHNOLOGY LABS
Training Students in Embedded, IT & Electronics

CERTIFICATE

This is to certify that report entitled

"**Arduino microcontroller and associated peripherals**" which is submitted by **ANKITA SAKHUJA** in partial fulfilment of the requirement for the award of degree B.Tech in **Electronics & Communication** to **Amity University Haryana** is a record of the candidate own work carried out by her under my supervision. With the knowledge she gained during this training, she developed a DTMF controlled Home Automation System .The matter embodied in this is original and has not been submitted for the award of any other degree.

Date: 20-07-14

Mr . Meharban Singh

Director

ACKNOWLEDGEMENT

I would like to extend my gratitude and my sincere thanks to my honourable, esteemed supervisors **Dr. Priti Singh** . I sincerely thank for his exemplary guidance and encouragement. I would like to thank **Mr. Mahendra Singh Meena** for his guidance and immense support .Their trust and support inspired me in the most important moments of my making right decisions and I am glad to work with them.

My special thank goes to **Mr. Meharban Singh,** Director of **Optimus Research Labs Pvt. Ltd. ,** New Delhi for his encouragement, support and providing the facilities for the completion of the training. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

Last but not least I would like to thank my parents, who taught me the value of the hard work by their own example.

Ankita Sakhuja

ABSTRACT

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. The boards can be built by hand or purchased preassembled; the software can be downloaded for free. The hardware reference designs (CAD files) are available under an open-source license, you are free to adapt them to your needs.

Generally, appliances used in our home are controlled with the help of switches. These days, you can see automation of these appliances using many technologies. This report presents the controlling of home appliances using DTMF technology.

For the whole system, it will setup based with the Arduino Uno and DTMF module. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing).

List of Contents

List of Figures

Figure Name

CHAPTER 1 ARDUINO

ARDUINO interface boards provide the engineers ,artists ,designers , hobbyists with technology with a low-cost ,easy-to-use technology to create their creative, interactive objects ,useful projects etc . A whole new breed of projects can now be built that can be controlled from a computer.

1.1 WHAT IS ARDUINO ?

ARDUINO is a open source electronics prototyping platform based on flexible ,easy-to-use Hardware and Software .

It's an open-source physical computing platform based on a microcontroller board ,and a development environment for writing software for the board . In simple words , Arduino is a small microcontroller board with a USB plug to connect to your computer and a number of connection sockets that can be wired up to external electronics ,such as motors ,relays ,light sensors ,laser diodes , loudspeakers , microphones , etc .They can either be powered through the USB connection from the computer or from a 9V battery .They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently.

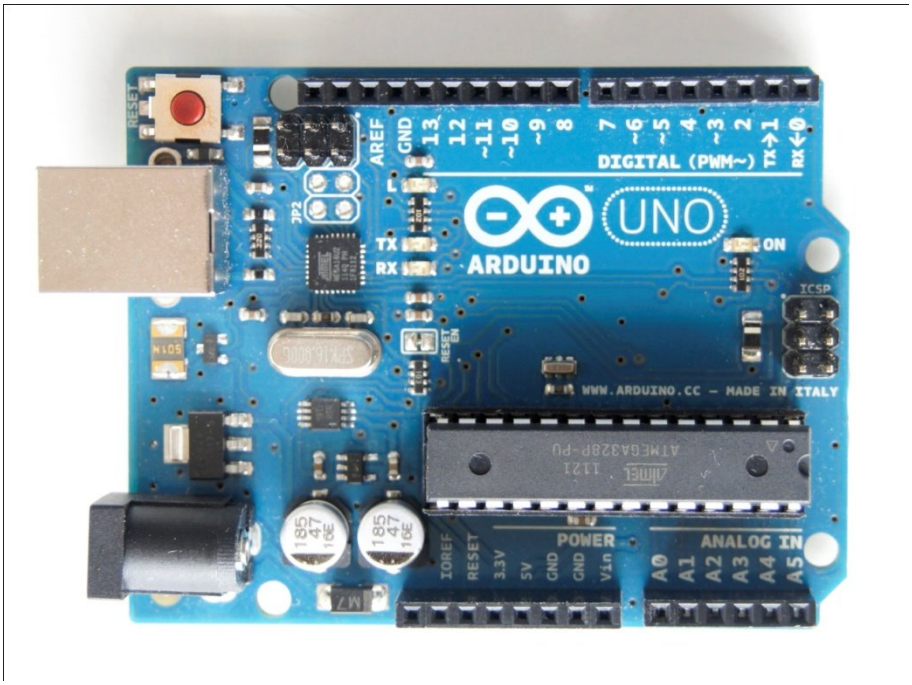


Fig 1.1 Arduino UNO

Anyone can buy this device through online auction site or search engine .Since the Arduino is an open-source hardware designs , hence people create their own clones of the Arduino and sell them ,so the market for the boards is competitive .An official Arduino costs about $30,and a clone often less than $20.
The name ―Arduino is reserved by the original makers. However, clone Arduino designs often have the letters ―duino on the end of their name, for example, Freeduino or DFRduino.

The software for programming your Arduino is easy to use and also freely available for Windows , Mac, and LINUX computers at no cost.

1.2 EMBEDDED SYSTEMS

The main objective of undertaking this topic in our report is very simple because Arduino is a kind of a microcontroller and microcontroller cannot be explained without introducing to Embedded systems.

An embedded system is a system that has software embedded into hardware, which makes system dedicated for an application (s) or specific part of an application or product or part of a larger system. It processes a fixed set of pre-programmed instructions to control electromechanical equipment which may be part of an even larger system (not a computer with keyboard, display, etc).

Furthermore any machinery which has **mechanical parts, electronic circuits and software** can be termed as an **embedded system**.

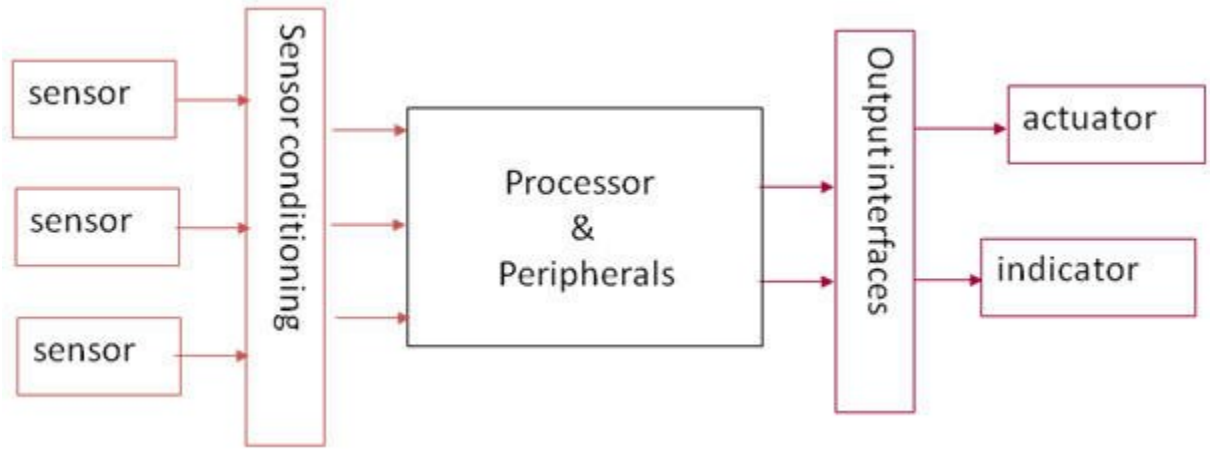


Fig 1.2 Block diagram of Embedded system

1.2.1 ELEMENTS OF AN EMBEDDED SYSTEM

- **Microcontroller**

Microcontrollers are single chip semi conductor device which is a computer on chip. In other words it is a smaller version of a computer but not a complete computer. Its CPU contains an Arithmetic & Logic Unit (ALU), a Program Counter (PC), a Stack Pointer (SP), registers, a clock and interrupts circuit on a single chip. To make complete micro computer, one must add memory usually ROM and RAM, memory decoder, an oscillator, a number of serial and parallel ports

However, the uC by itself, cannot accomplish much; it needs several external inputs :power, for one; a steady clock signal, for another. Also, the job of programming it has to be accomplished by an external circuit. So typically, a microcontroller is used along with a circuit which provides these things to it; this combination is called a microcontroller board. The Arduino Uno that you have received, is one such microcontroller board. The actual microcontroller at its heart is the chip called **Atmega328.**
The advantages that Arduino offers over other microcontroller boards are largely in terms of reliability of the circuit hardware as well as the ease of programming and using it.

A Microcontroller is designed to cater for large amount of applications and hence is produced in bulk. Using it in an embedded system offers various benefits. Design time is low as only software is to be developed, no digital design is involved. **Typical characteristics of a Microcontroller are relatively high cost, high speeds, higher Power consumption, large architecture, large memory size, onboard flash and cache, an external bus interface for greater memory usage.**

- **Software**

If hardware forms the body of the embedded systems, software is a soul of embedded systems. The software lends the functionality to an embedded system. Variety of languages has emerged as the tasks to be performed by embedded systems are of quite diverse nature.

- **Interface to physical world**

Embedded systems interface with the physical world through Sensors and Actuators. Sensors act as "senses" for an embedded system, whereas actuators act as "limbs". Varieties of sensors working on variety of sensing principles are used in embedded systems. Light, temperature, acceleration, speed, mass, distance, etc. are common physical parameters which are sensed. Motors are one of the actuators used in many systems.

1.2.2 APPLICATION AREAS OF EMBEDDED SYSTEMS

- Medical Systems
  - pace maker, patient monitoring systems, injection systems, intensive care units
- Office Equipment
  - printer, copier, fax
- Tools
  - Multimeter, oscilloscope, line tester, GPS
- Banking
  - ATMs, statement printers
- Transportation
  - (Planes/Trains/[Automobiles] and Boats)
  - radar, traffic lights, signaling systems

1.3 HISTORY OF ARDUINO

While teaching a physical computing class at the Interaction Design Institute Ivrea in 2005, Massimo Banzi's students were unwilling to spend the 76 Euros for the BASIC Stamp microcontroller commonly used in such applications. Banzi and his collegues looked for alternatives, finally settling on the
wiring platform developed by one of Banzi's students. In his own words:—…**we started to figure out how could we make the whole platform even simpler, even cheaper,**
even easier to use. And then we started to essentially reimplement the whole thing as an open source project.
Once they had a prototype, a student wrote the software that would allow wiring programs to run on the new platform.

- Upon seeing the project, visiting professor Casey Reas suggested that there might be wider applications than just design schools for the new product. The prototype was redesigned for mass production and a test run of 200 boards was made. Orders began coming in from other design schools and the students looking for Arduino's, and the **Arduino project was born.**

- **Massimo Banzi** and **David Cuartielles** became it's founders.

- ARDUINO is an Italian word, meaning STRONG FRIEND .

The English version of the name is ―Hardwin. As of May2011,more than 300,000 Arduino units are in the world

1.4 VERSIONS OF ARDUINO

There have been many revisions of the USB Arduino .Some of them are :

❖ **Arduino UNO**

This is the latest revision of the basic Arduino USB board. It connects to the computer with a standard USB cable and contains everything else you need to program and use the board. It can be extended with a variety of shields: custom daughter-boards with specific features. It is similar to the Duemilanove, a different USB-to-serial chip the ATMega8U2,and newly designed labeling to make inputs and outputs easier to identify.

Figure 1.3 Arduino Uno Board

❖ **Arduino Mega 2560**

A larger, more powerful Arduino board : digital pins, PWM pins, analog inputs, serial ports, etc. The version of the Mega released with the Uno, this version features the Atmega2560, which has twice the memory, and uses the ATMega 8U2 for USB-to-serial communication.

Figure 1.4 Arduino Mega Board

❖ **Arduino Duemilanove**

The Duemilanove automatically selects the appropriate power supply (USB or external power), eliminating the need for the power selection jumper found on previous boards .It also adds an easiest to cut trace for disabling the auto-reset, along with a solder jumper for re-enabling it.

Note: Around March 1st, 2009, the Duemilanove started to ship with the ATmega328p instead of the ATmega168

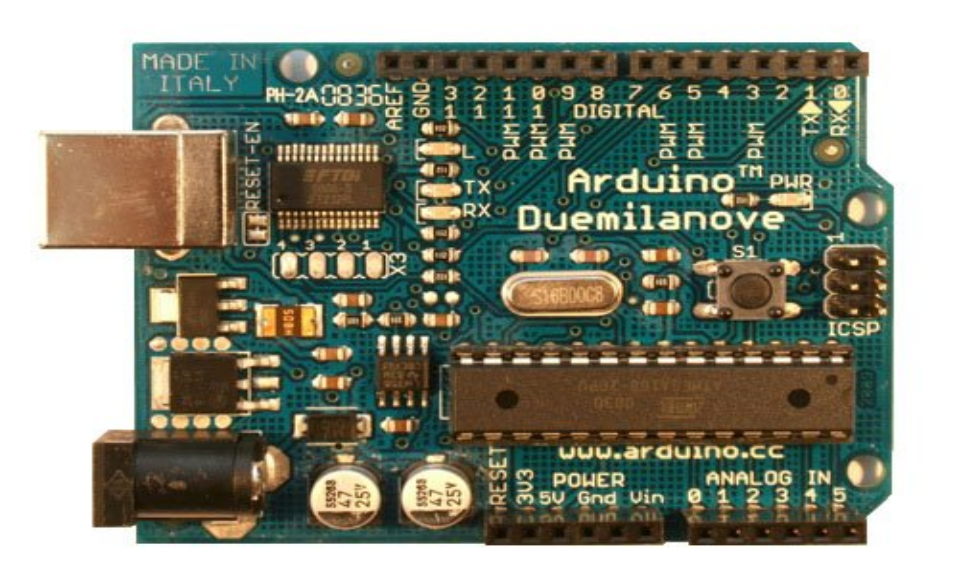


Figure 1.5 Arduino Duemilanove Board

❖ **Arduino Fio**

An Arduino intended for use as a wireless node. It has a header for an XBee radio, a connector for a LiPo battery, and a battery charging circuit.

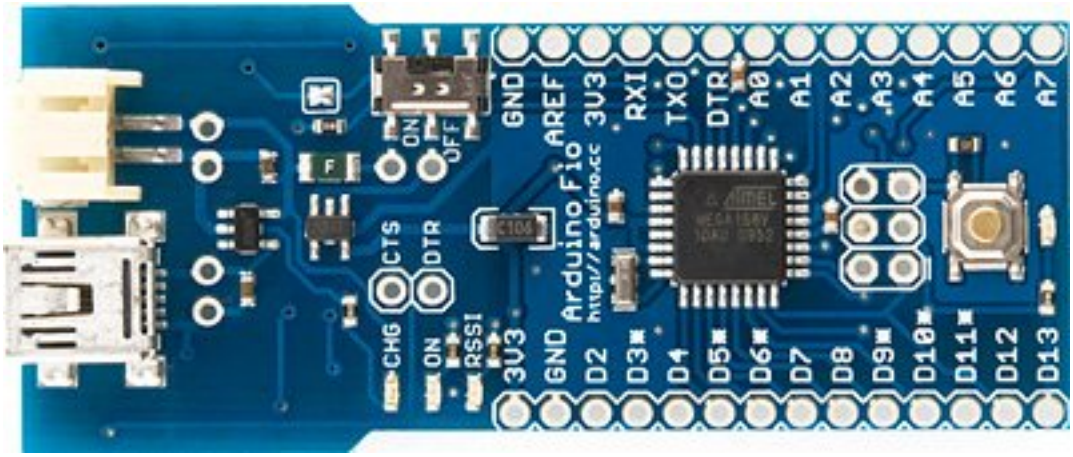


Figure 1.6 Arduino Fio Board

❖ **LilyPad Arduino**

A stripped-down , circular Arduino board designed for stitching into clothing and other fabric/flexible applications. It needs an additional adapter to communicate with a computer.

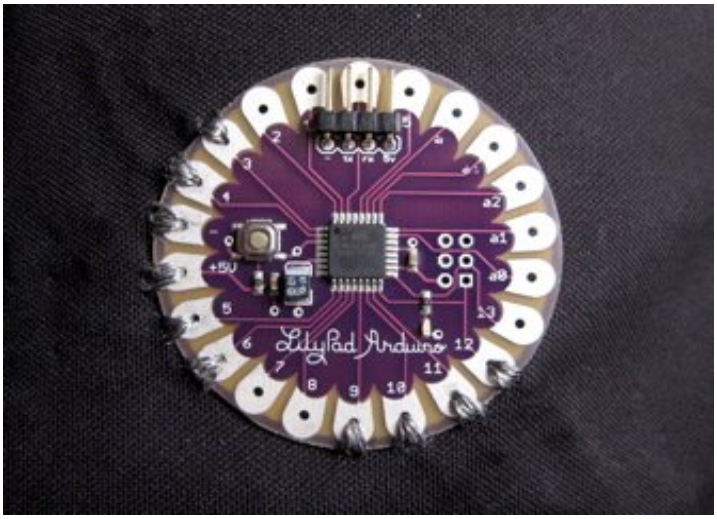


Figure 1.7 Arduino LilyPad Board

❖ **Arduino Diecimila**

The main change in the Arduino Diecimila is that it can be reset from the computer, without the need to physically press the reset button on the board. The Diecimila uses a low dropout voltage regulator which lowers the board's power consumption when powered by an external supply (AC/DC adapter or battery). It also provides pin headers for the reset line and for 3.3V. There is a built-in LED on pin 13. Some blue Diecimila boards say "Prototype – Limited Edition" but are in fact fully-tested production boards (the actual prototypes are red).

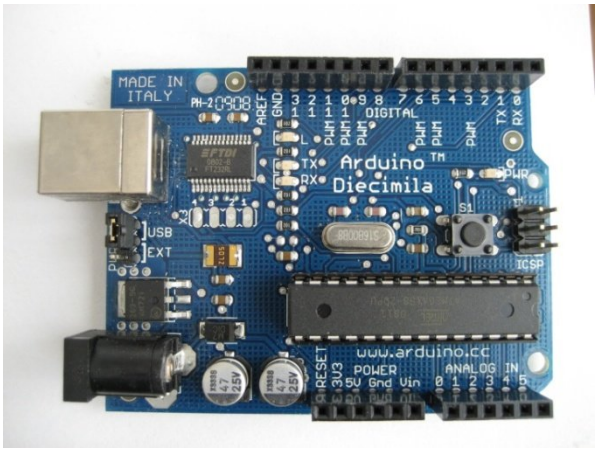


Figure 1.8 Arduino Diecimila Board

❖ **Arduino NG Rev.C**

Revision C of the Arduino NG does not have a built-in LED on pin 13 - instead you'll see two small unused solder pads near the labels "GND" and "13".There is, however, about 1000 ohms of resistance on pin 13, so you can connect an LED without external resistor.

Figure 1.9 Arduino NG Rev.C Board

❖ **Arduino Extreme**

The Arduino Extreme uses many more surface mount components than previous USB Arduino boards and comes with female pin headers. It also has RX and TX LEDs that indicate when data is being sent to or from the board.

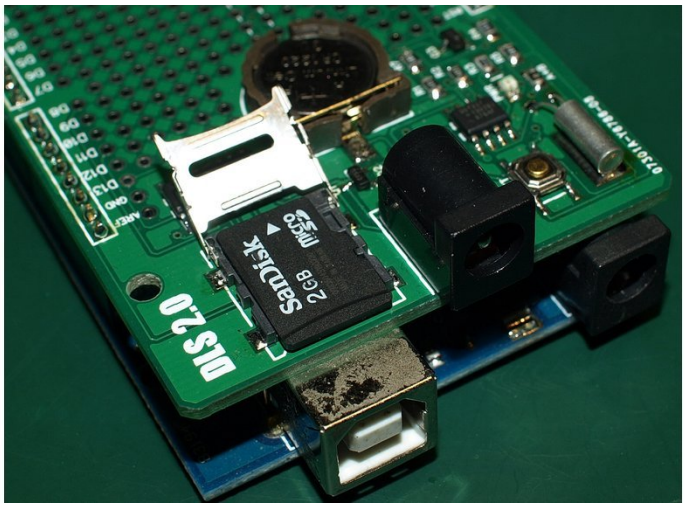


Figure 1.10 Arduino Extreme Board

❖ **Arduino Mini 04**

On this version of the Arduino Mini, two of the pins are changed. The third pin became reset (instead of ground) and fourth pin became ground (instead of being unconnected). These boards are labeled "Mini 04"

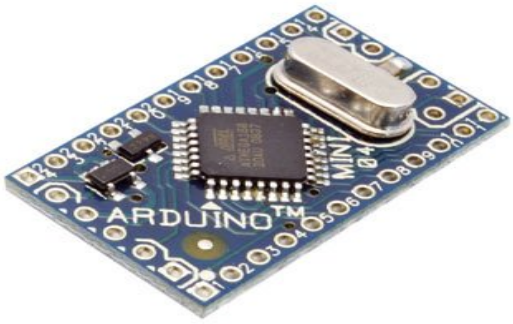


Figure 1.11 Arduino Mini 04 Board

CHAPTER 2 PLATFORM

In this section , we will discuss about platform of Arduino both at hardware level as well as at software level .

2.1 ARDUINO BOARD LAYOUT

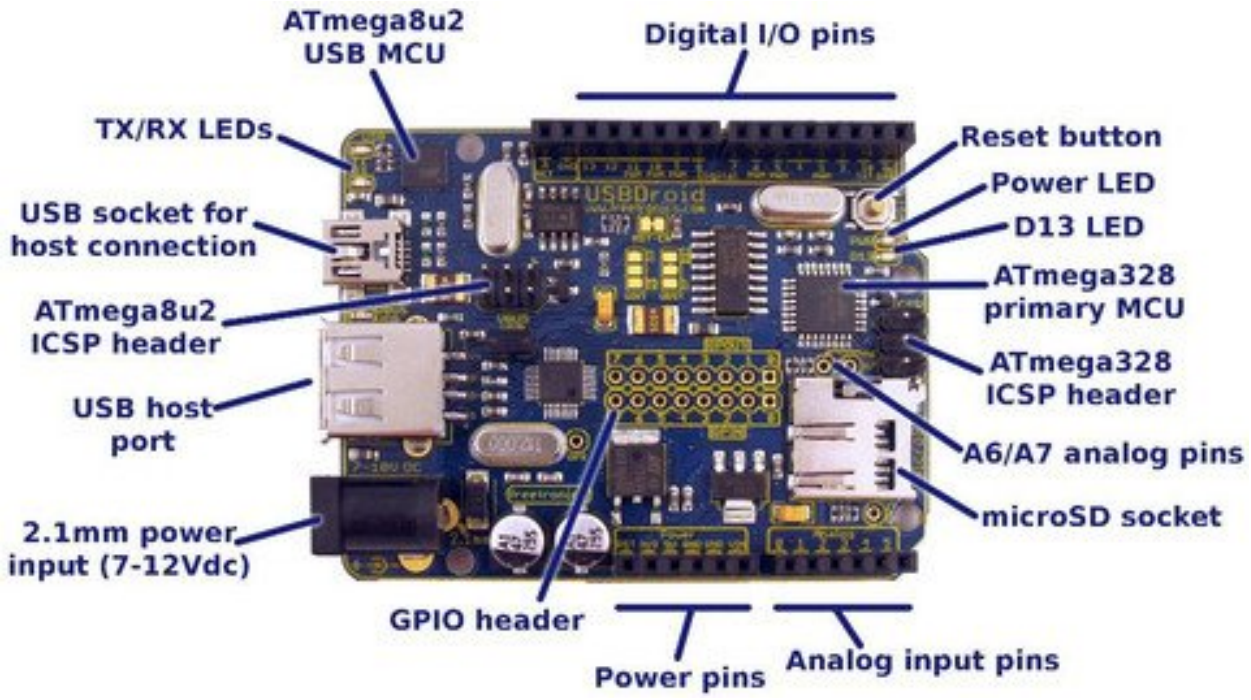


Figure 2.1 Explanation of pins(Arduino Uno)

2.1.1 ATmega8(Microcontroller)

- 16 MHz
- 8 Kbyte Flash RAM(1K taken by the boot loader)
- 1 Kbyte RAM(eg. for auto/local variables and stack)
- 14 digital Input/Output Ports

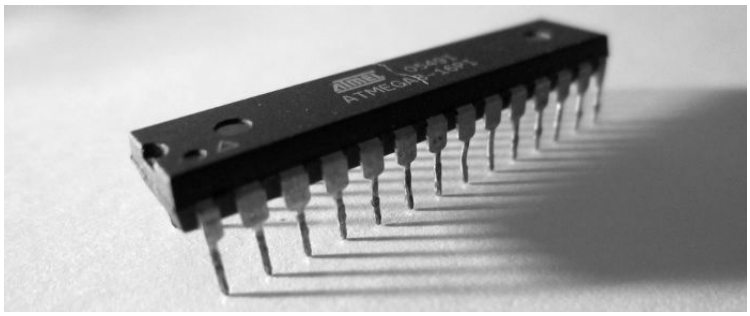


Figure 2.2 ATmega8(microcontroller)

2.1.2 Single chip USB to async. Serial data transfer interface

- USB 2.0 compatible
- Transmit and receive LED frive signals
- 256 Byte receive,128 Byte transmit buffer
- Data transfer rate from 300bits/sec to 2 Mb/sec

Figure 2.3 Async. Serial Data Transfer Interface

2.1.3 External Power

The power requirement for ARDUINO is 9 to 12V DC, 250mA or more , 2.1mm plug ,centre pin positive.

The OFF-the shelf adapter

- must be a DC adapter (i.e. it has to put out DC, not AC)
- should be between 9V and 12V DC
- must be rated for a minimum of 250mA current output, although you will likely want must have a 2.1mm power plug on the Arduino end, and
- the plug must be "centre positive", that is, the middle pin of the plug has to be the +connection.

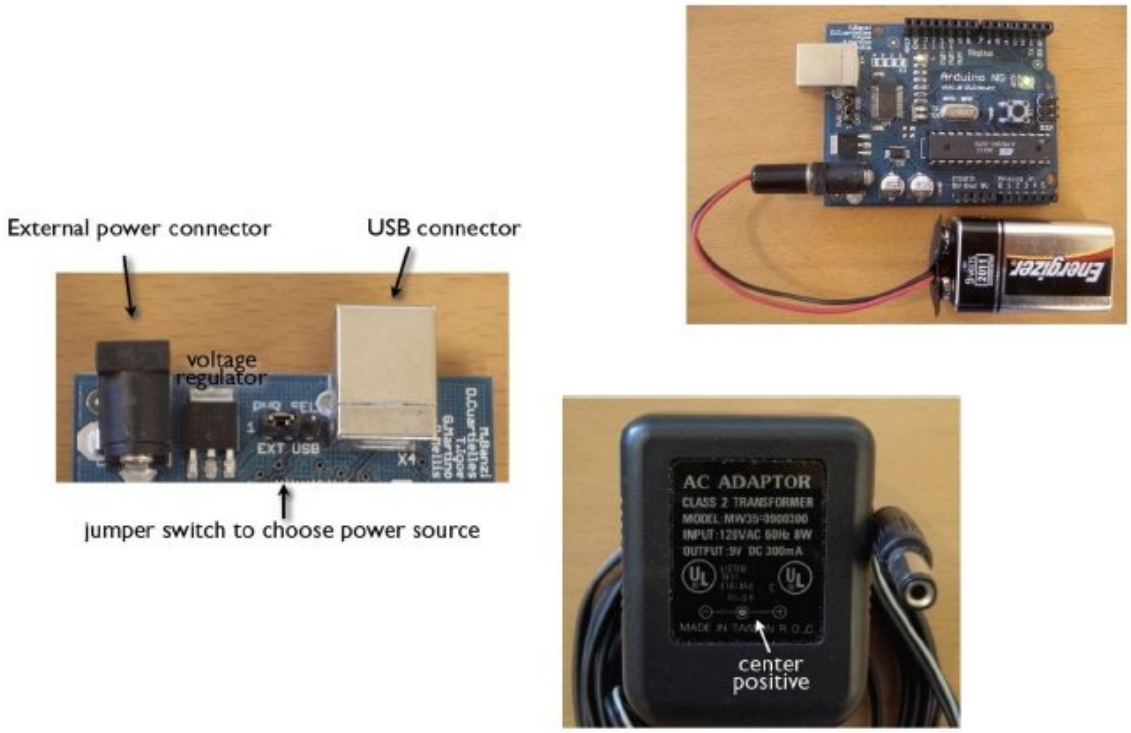


Figure 2.4 Representation of different power supplies

2.2 SOFTWARE

- The Arduino integrated development environment (IDE) is a **cross-platform** application written in **Java**, and is derived from the IDE for the Processing programming language and the Wiring projects.

- It is designed to introduce programming to artists and other newcomers unfamiliar with software development.

- It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click.

- A program or code written for Arduino is called a "sketch".

- Arduino programs are written in C or C++.

- The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier.

- Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

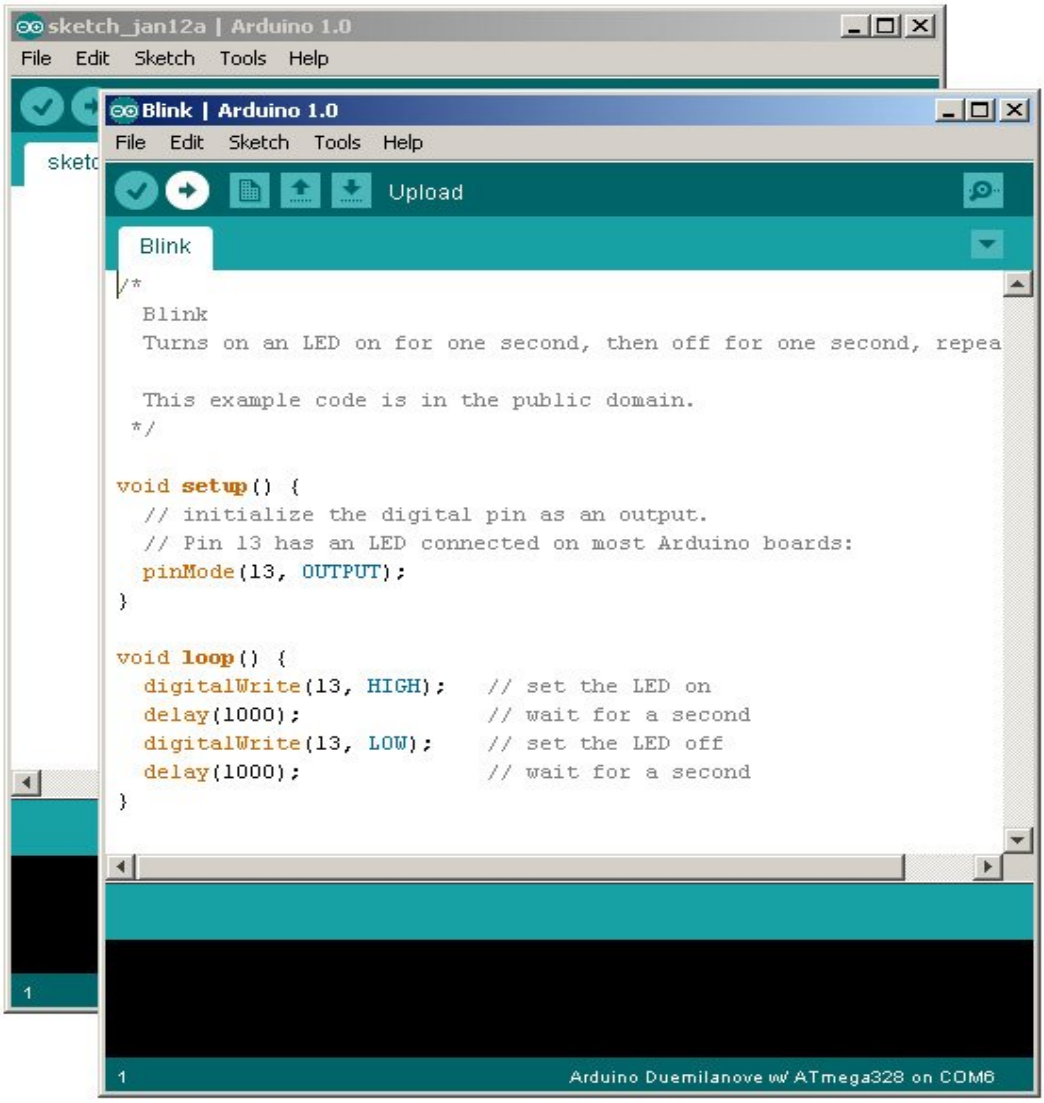


Figure 2.5 Demonstrating a program on IDE

- The **Arduino website** describes this software as follows:

The open-source Arduino environment makes it easy to write code and upload it to the i/o board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing, avr-gcc, and other open source software.

- It's easy to set up, with installers for Windows, Mac OS X, and Linux.

We use the latest version (1.5.5 at the time of this writing), from this location:

http://arduino.cc/en/Main/Software#toc3

Once installed and started up, you'll see something like this:

Figure 2.6 Demonstration of loop() and setup() statements

2.2.1 BASICS OF ARDUINO PROGRAMMING

- Arduino programs can be divided in three main parts:
  - structure
  - values
    (variables and constants), and
  - functions

- Available datatypes in ARDUINO IDE are

  - Void
  - Boolean

- ❖ char (0-255)

- ⊗ byte- 8 bit data (0 – 255)
- ⊗ int- 16-bit data (32,767 - -32,768)
- ⊗ long – 32 bit data (2,147,483,647 to -2,147,483,648)
- ⊗ float
- ⊗ double
- ⊗ string
- ⊗ char array
- ⊗ String- object
- ⊗ array

- **Arithmetic operators**

Arithmetic operators include addition , subtraction ,multiplication and division . For math that requires fractions you can use float variables ,if you can bear large size and slow computation speeds in your microcontroller.

e.g. ,

```
y=y+3 ;
x = x – 7;
i = j * 6;
r = r / 5;
```

- **Comparision operators**

Comparisons of one variable or constant against another are often used in if statements to test if a specified condition is true.

e.g.

```
x == y // x is equal to y
x != y // x is not equal to y
x < y // x is less than y
x > y // x is greater than y
x <= y // x is less than or equal to y
x >= y // x is greater than or equal to y
```

- **Logical operators**

Logical operators are usually a way to logically combine two expressions and return a TRUE or FALSE depending on the operator .There are three logical operators, AND, OR, and NOT.
e.g. Logical AND: if (x > 0 && x < 5) // true only if both expressions are true

Logical OR :if (x > 0 || y > 0) // true if either expression is true

Logical NOT: if (!x > 0) // true only if expression

- **TRUE/FALSE**

These are Boolean constants that define logic levels of the Arduino.
FALSE is easily defined as 0 (zero)
TRUE is often defined as 1, but can also be anything else except zero.

- **HIGH/LOW**

These constants define pin levels as HIGH or LOW and are used when reading or writing to digital pins.

HIGH is defined as logic level 1, ON, or 5 volts

LOW is logic level 0, OFF, or 0 volts.

e.g. digitalWrite(13, HIGH);

- **INPUT/OUTPUT**

  These are used to define whether the port would be an input port or an output port used in the program.

e.g.
pinmode(13, OUTPUT);

2.2.2 Basic Terminologies in ARDUINO

- **Analog to digital converter(ADC)**

The process of Analog to digital conversion is shown in figure . This term is usually used in Arduino because it has six analog input pins whose output is easily converted into digital for carrying out the process .

Arduino	Processor	Flash KiB	EEPROM KiB	SRAM KiB	Digital I/O pins	...with PWM	Analog input pins	USB Interface type	Dimensions inches	Dimensions mm
Diecimila	ATmega168	16	0.5	1	14	6	6	FTDI	2.7" x 2.1"	68.6mm x 53.3mm
Duemilanove	ATmega168/328P	16/32	0.5/1	1/2	14	6	6	FTDI	2.7" x 2.1"	68.6mm x 53.3mm
Uno	ATmega328P	32	1	2	14	6	6	ATmega8U2	2.7" x 2.1"	68.6mm x 53.3mm
Mega	ATmega1280	128	4	8	54	14	16	FTDI	4" x 2.1"	101.6mm x 53.3mm
Mega2560	ATmega2560	256	4	8	54	14	16	ATmega8U2	4" x 2.1"	101.6mm x 53.3mm
Fio	ATmega328P	32	1	2	14	6	8	None	1.6" x 1.1"	40.6mm x 27.9mm
Nano	ATmega168 or ATmega328	16/32	0.5/1	1/2	14	6	8	FTDI	1.70" x 0.73"	43mm x 18mm
LilyPad	ATmega168V or ATmega328V	16	0.5	1	14	6	6	None	2" ø	50mm ø

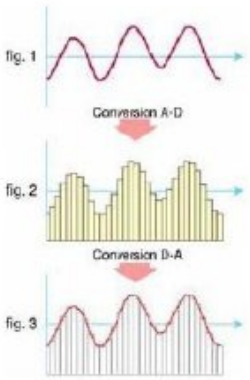


Figure 2.7 A-D and D-A waveforms

- **Pulse width modulation (PWM)**

– An Arduino Uno has 14 digital input/output (I/O) pins. The pinMode function is used to configure a pin as an input or output. When a digital I/O pin is
configured as an input, digitalRead reads the state of the pin, which will be either HIGH or LOW.
– On an Arduino Uno, PWM output is possible on digital I/O pins 3, 5, 6, 9, 10 and 11.
– "analogWrite()" is used for PWM functionality. It writes an analog value (PWM wave) to a pin.
– It Can be used to light a LED at varying brightness or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady square wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite() on the same pin). The frequency of the PWM signal on most pins is approximately 490 Hz.
– On the Uno and similar boards, pins 5 and 6 have a frequency of approximately 980 Hz. Pins 3 and 11 on the Leonardo also run at 980 Hz.
– Unlike the PWM pins, DAC0 andDAC1 are Digital to Analog converters, and act as true analog outputs.
– The *analogWrite* function has nothing to do with the analog pins or the *analogRead* function.

- Syntax

analogWrite(pin, value)

- Parameters

pin: the pin to write to.
value: the duty cycle: between 0 (always off) and 255 (always on).

- Returns

nothing

- Notes and Known Issues

The PWM outputs generated on pins 5 and 6 will have higher-than-expected duty cycles. This is because of interactions with the millis() and delay() functions, which share the same internal timer used to generate those PWM outputs. This will be noticed mostly on low duty-cycle settings (e.g 0 - 10) and may result in a value of 0 not fully turning off the output on pins 5 and 6.

CHAPTER 3

PROJECT: DTMF BASED HOME AUTOMATION SYSTEM

This project works on the principle of DTMF (Dual tone multi frequency). This circuit can be used to control or on/off the load i.e. electric equipments like fan, bulb, water pump, etc.

We know that, in industries loads are distributed over large distances, and it is not convenient to operate all those equipments time to time and it is also time consuming.

Also in agricultural fields water pumps and other electric systems are spread over large area and it is hard to operate these systems to farmers. We can use this circuit in home too, therefore this can also be called as DTMF based home automation project.

3.1 INTRODUCTION

Generally, appliances used in our home are controlled with the help of switches. These days, you can see automation of these appliances using many technologies. This portion presents the controlling of home appliances using DTMF technology.

DTMF is acronym for Dual Tone Multi Frequency. So, just think when you make call for customer care, they will ask you to press 1, 2 or any other number. When you press a number from your mobile, one particular action is happening. All this is because of DTMF. When a button is pressed in your mobile keypad, it will generate a tone of two frequencies. These tones are called row and column frequencies.

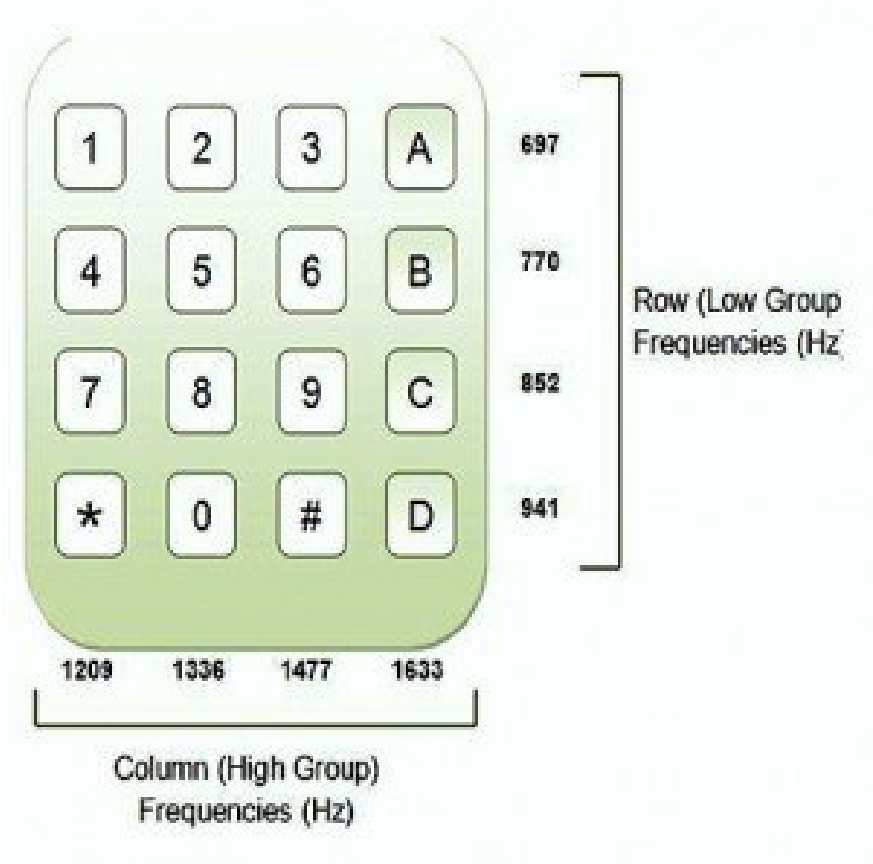


Figure 3.1 Working with DTMF frequencies

3.2 CIRCUIT COMPONENTS

We need following components in our circuit in order to demonstrate the working of our program:

- **DTMF Decoder 8870**
- **Arduino UNO**
- **I/0 card**
- **Female connectors**
- **BCD Display**
- **USB Cable**

- **DTMF (Dual Tone Multiple Frequency) Decoder 8870**

This circuit detects the dial tone from a telephone line and decodes the keypad pressed on the remote telephone. The dial tone we heard when we pick up the phone set is call Dual Tone Multi-Frequency, DTMF in short. The name was given because the tone that we heard over the phone is actually making up of two distinct frequency tones, hence the name dual tone. The DTMF tone is a form of one way communication between the dialer and the telephone exchange. A complete communication consists of the tone generator and the tone decoder. When a key is being pressed on the matrix keypad, it generates a unique tone consisting of two audible tone frequencies.

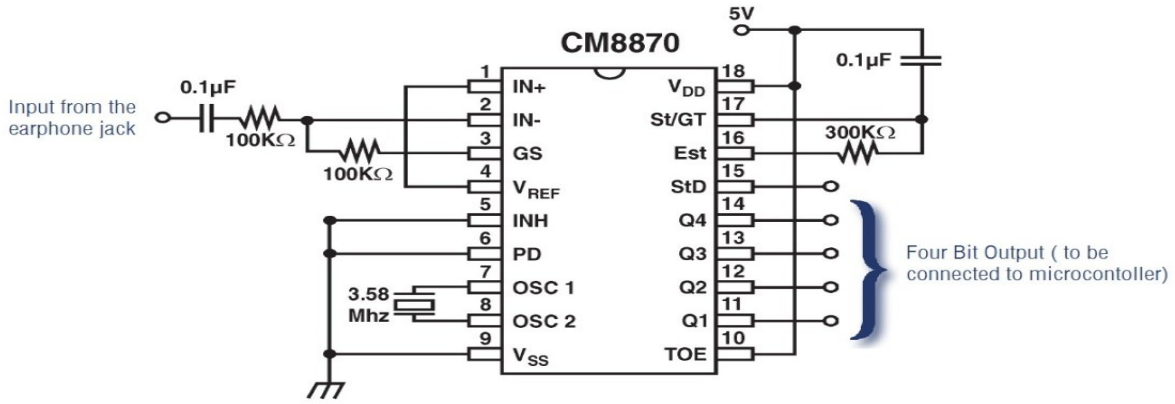


Figure 3.2 Pin Diagram of DTMF decoder 8870

For example, if the key '1' is being press on the phone, the tone you hear is actually consisting of a 697 Hz & 1209 Hz sine signal. Pressing key '9' will generate the tone form by 852 Hz & 1477 Hz. The frequency use in the dial tone system is of audible range suitable for transmission over the telephone cable.

- **Arduino UNO**

  Four input lines coming from the DTMF decoder is considered as input lines by our microcontroller. We decode the DTMF output and provide the output to our electrical components through microcontroller using control statements.

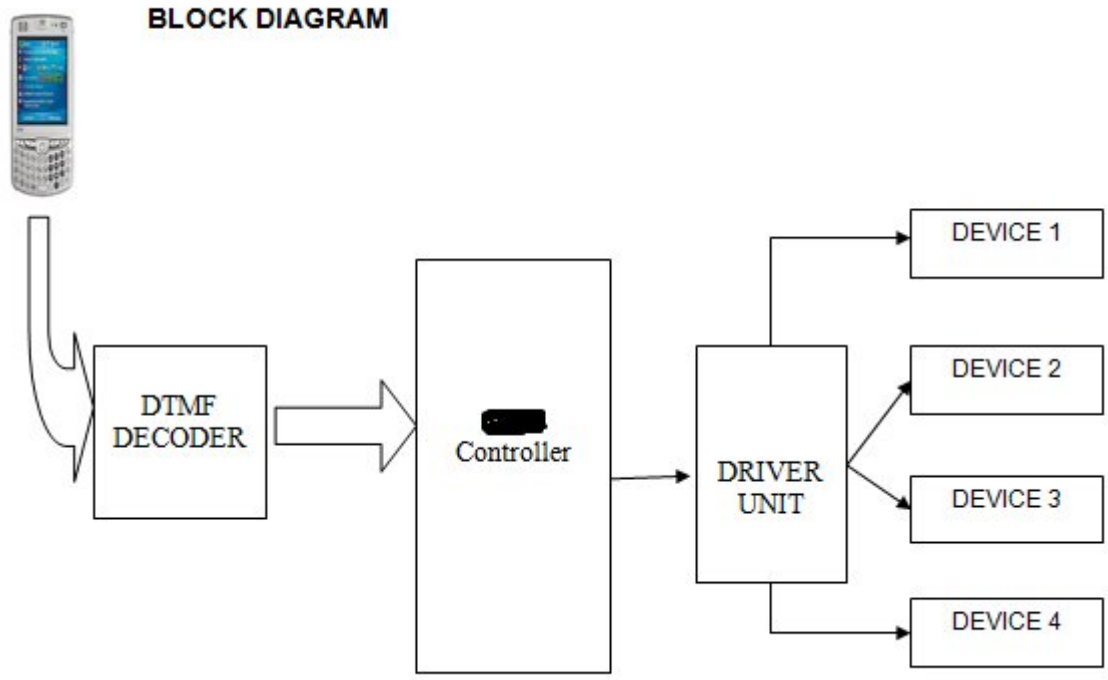


Figure 3.3 Block Diagram Representation of circuit

3.3 PROGRAM CODE

- **For interfacing seven segment display with UNO**

```
void setup()
{
 for(int i=2;i<6;i++)
 {
  pinMode(i,INPUT);
 }
 for(int i=6;i<14;i++)
 {
  pinMode(i,OUTPUT);
 }

 Serial.begin(9600);
}
```

```
int data[10][8]=
{
  {0,0,0,0,0,0,1,1},
 {1,0,0,1,1,1,1,1},
 {0,0,1,0,0,1,0,1},
 {0,0,0,0,1,1,0,1},
 {1,0,0,1,1,0,0,1},
 {0,1,0,0,1,0,0,1},
 {0,1,0,0,0,0,0,1},
 {0,0,0,1,1,1,0,1},
 {0,0,0,0,0,0,0,1},
 {0,0,0,0,1,0,0,1}
 };

 void loop()
 {
   int inputdata[4],sum;

   for(int i=2;i<6;i++)
   {
     inputdata[i-2]=digitalRead(i);
   }

   sum=inputdata[0]*1+inputdata[1]*2+inputdata[2]*4+inputdata[3]*8 ;
  Serial.println(sum);

//*************************************************************************************
******

  for(int i=0;i<8;i++)
  {
    digitalWrite(i+6, data[sum][i]);
  }
  delay(5);


 }
```

- **For interfacing I/O card with UNO**

```
int binary_data_from_DTMF[4]={0,0,0,0};
int sum=0;
int fan1=10;
int fan2=11;
int fan3=12;
int fan4=13;



void room1()
{
  digitalWrite(fan1,HIGH);

  }
void room2()
{
  digitalWrite(fan2,HIGH);

  }
void room3()
{
  digitalWrite(fan3,HIGH);

  }

void room4()
{
  digitalWrite(fan4,HIGH);


}

 void setup()
 {
   // INPUT FROM DTMF
   for(int i=6;i<10;i++)
   {
```

```
    pinMode(i,INPUT);
  }

  // OUTPUT FOR IO CARD
  for(int i=4;i<8;i++)
  {
    pinMode(i,OUTPUT);
    digitalWrite(i,LOW);
  }
}

 void loop()
{
  for(int i=6;i<10;i++)
  {
    binary_data_from_DTMF[i]=digitalRead(i);
  }

sum=((binary_data_from_DTMF[6]*1)+(binary_data_from_DTMF[7]*2)+(binary_data_fr
om_DTMF[8]*4)+(binary_data_from_DTMF[9]*8));

 if(sum==1)
 {
   room1();
 }
   else if(sum==2)
 {
   room2();
 }
   else if(sum==3)
 {
   room3();
 }

 else if(sum==4)
 {
   room4();
 }
}
```

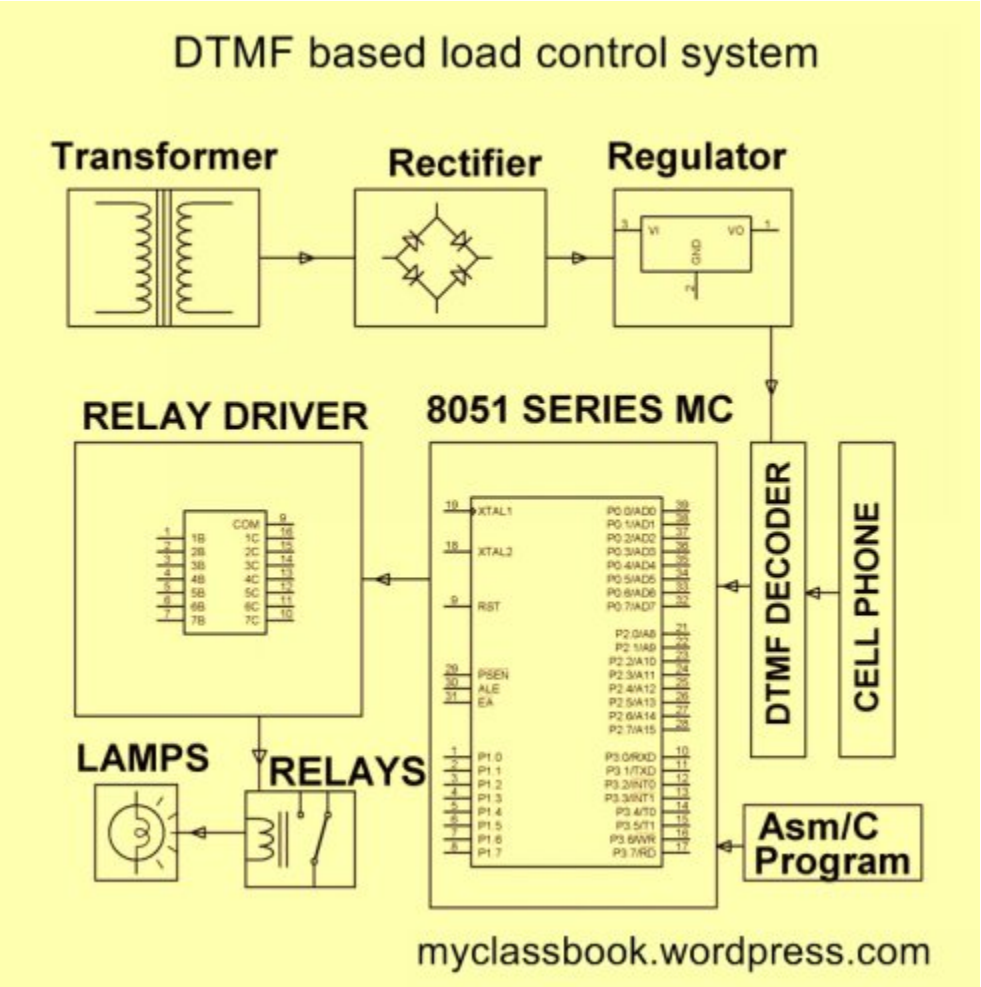


Figure 3.4 Controlling relays and load with DTMF circuit

3.4 ADVANTAGES AND LIMITATIONS

There are certain advantages and disadvantages which are encountered after making this circuit for home automation. Some of them are given as follows :

Advantages of DTMF Controlled Home Automation System Circuit:

1. One can control home appliances from anywhere.
2. It reduces wastage of electricity when we forgot to switch off the lights & fans and gone outside.
3. It is very low cost compared to other technologies like GSM.

Limitations of DTMF Controlled Home Appliances:

1. No security. Anyone can control the appliances by calling the mobile connected to module.
2. Number of appliances is limited as our mobile can generate only 16 tones.

CHAPTER 4 Why Arduino Has Won And Why It's Here To Stay?

- **The IDE Runs on Macs, Linux, and Win**

  The IDE works on a Mac, Win, and Linux, and it's completely open source. The IDE is how you program the Arduino — it's based on Processing (a graphics programming language and development system popular with artists and designers), which has been around for a long time. It runs on Macs and Linux, not just Windows, and that matters if you want to be inclusive.

- **The Driver Actually Work On Macs, Linux, and Win**

  Again, like the IDE, the drivers to use the board work on Mac, Win, Linux, and the FTDI drivers "just work." Sticking with serial, a well understood (but slow) interface, was a good call.

- **Libraries, Easy-to-Do Simple Things, Easy-to-Do Hard Things**

  There are tons of object-wrapped libraries to do complex things, like writing to SD cards, LCD screens, parsing GPS. And there's are also libraries to do simple things, like twiddle pins or debounce buttons.

- **Lightwight, Runs on the Metal**

  The code runs directly on bare metal, with a well-tested and understood compiler (we would even say that avr-gcc is the default/standard compiler for AVR.) It's not interpreted like .NET or BASIC. It's fast, it's small, it's lightweight, and you can use the HEX file to program fresh chips in bulk.

- **Sensors**
  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.

- **Simple, But Not Too Simple**

  Many dev 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.

- **Not Made By a Chip Maker**

  The board was not designed by a chip maker. Why is this important? Chip makers often want to show how their product is different so they add weird things to differentiate themselves. The Arduino highlights commonalities between microcontrollers, not the differences

- **Low Cost**

  You can get an Arduino for $30, and we'll probably see $20 Arduinos soon. Many dev boards start at $50 and could easily get to $100+, although now we're seeing chip companies start to realize that it's worthwhile to have a more pragmatic pricing strategy.

- **Open Source**

  While it's nice that Arduino is open source, and commercial use is allowed if you make a clone, it's not the biggest reason, which is why it's down near the end of the list. Specialized derivatives can be made without paying someone or asking anyone

Why Arduino is Here to Stay

The barrier to entry isn't a monetary one, it's a philosophical one. This requires boldness and getting out of committee-think. A chip company needs to show off chips — they don't care about Mac support, or writing tons of software, libraries, and IDEs. Chip companies are (historically) the ones who usually make the platforms. We'll see some of the big players flood the market with subsidized hardware to beat the $30 price point of the Arduino, but that doesn't matter if the Arduino support and quality stay high.

Why else is it here to stay? The community. How can you get 100,000+ people to jump ship? You can't. To get close, you'll need to develop something just like the Arduino, support its shields and accessories, and write a lot of code (something chip companies hate to do.) Great software for multiple systems, lots of libraries, drivers that work, simple, low cost, and open source.

REFERENCES

[1] Tero Karvinen , Kimmo Karvinen , *Make a Mind Controlled Arduino Robot (Creating With Microcontrollers Eeg, Sensors, and Motors).*

[2] Michael Margolis , *Make an Arduino Controlled Robot*

[3] Electronicshub.org

[4] Maik Schmidt, *Arduino: A Quick-Start Guide (Pragmatic Programmers)*