

CELL PHONE BASED DTMF CONTROLLED GARAGE DOOR OPENING SYSTEM



An-Najah National University
Faculty of Engineering
Electrical Engineering Department

Graduation Project 2

SUPERVISOR :

Dr.Kamel Subhi.

Prepared By:

Fatima Dmaidi.

Majd Darweesh.

Noor Salman.

Presented in partial fulfillment of the
requirements for Bachelor degree in
Electrical Engineering.



Acknowledgment:

*Our first and greatest thanks are to **Allah**. He above all was, is, and will be the source of help and guidance that counts. His help kept us going through many frustrations and His guidance brought us back on track when our frustrations tended to drive us astray.*

*Next, we would like to express our deep gratitude to our supervisor **Dr. Kamel Saleh** for his effort and encouragement throughout the preparation of this project. His Knowledge and endless support was a great asset from which we learned plenty. He was very enthusiastic about the project, which gave us a motive to work harder and harder.*

We would like to thank friends who did their best whenever we needed or asked for anything. No matter what the task was or how busy they were, they always found the time.

Also we would like to thank our family that helps us very much and they were very generous with us.

It is difficult to acknowledge everyone who was involved in preparation of this project by name. Nevertheless we appreciate their help no matter how simple it might have been.

Finally, we owe our colleagues great thanks for supporting us through our college years and for making those years the best years of our life.

DISCLAIMER

This report was written by students at the Electrical and Telecommunication Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of contents

- List of figures:..... 5
- List of tables: 6
- Nomenclature or list of symbols..... 7
- Abstract:** 8
- Chapter One: Introduction.** 9
 - 1.1 Executive Summary..... 9
 - 1.2 Statement of the problem: 9
 - 1.2.1 General problem statement 9
 - 1.2.2 General Solution Approach..... 9
 - 1.3 Block diagram of the project..... 10
 - 1.4 Organization of the report. 11
- Chapter 2: Constraints, Standards/ Codes and Earlier course work** 11
 - 2.1 Constrains: 11
 - 2.2 Standards / codes..... 12
 - 2.3 Earlier course work: 12
- Chapter 3: Literature Review.**..... 13
- Chapter 4: Methodology**..... 14
 - 4.1. DTMF:..... 14
 - 4.1.1. What is DTMF (Dual Tone Multi Frequency)?..... 14
 - 4.1.2. DTMF Decoder:..... 17
 - 4.2. Arduino: 20
 - 4.2.1 .What is Arduino? 20
 - 4.2.2. Why Arduino?..... 20
 - 4.2.3. Arduino Uno: 21
 - 4.3. H-Bridge: 22
 - 4.3.1 Speed Control - PWM Technique 23
 - 4.4 MOSFET 25
 - 4.4.1 What is MOSFET? 25

4.4.2 How to use MOSFET as a switch?	26
4.4.3 IRFP450A MOSFET.....	26
4.4. Full bridge drive:	27
4.4.1. IR2110	27
4.5. Permanent Magnet DC Motor.	29
4.6. Flow Chart shows the details of operation of the project:	32
Chapter 5: Results and Analysis and discussion.	33
5.1 Simulation:	33
5.2 Results and Analysis of real implementation of the system.....	41
Chapter 6: conclusions and recommendations.	54
Chapter7: Feasibility Study	55
Appendices:	59
Appendix A.....	59

List of figures:

Figure 1: keypad with 16 buttons.	15
Figure 2: DTMF decoder circuit.....	17
Figure 3: DTMF Decoder.	19
Figure 4 :H-Bridge Topology.	22
Figure 5 :H-Bridge Topology – Forward Direction.	23
Figure 6 :H-Bridge Topology – Reverse Direction.	23
Figure 7 : Pulse Width Modulation Used For Motor Control.....	24
Figure 8: IR2110 circuit for high-voltage full-bridge drive with independent switch control.....	27
Figure 9: IR2110 Pin/Lead Assignments.....	28
Figure 10: Functional Diagram of IR2110.....	28
Figure 11:IR2110 Pin/Lead Definitions.	29
Figure 12:Structure of Magnet DC Motor.	29
Figure 13: Simulation model of the project.	33
Figure 14: H-Bridge inverter and DC Motor.	34
Figure 15: Speed variation on the speed scope.....	35
Figure 16: Pulses that enters the Gates of IGBT (gate 1 ,2,3,4) in order.	36
Figure 17: Distance variation on distance scope.....	37
Figure 18:Speed variation when pressing number 4.	38
Figure 19: output pulses when pressing number 4.....	38
Figure 20: Distance variation on distance scope.....	39
Figure 21: Automatic answering setting.	41
Figure 22: Automatic answering timer.	42
Figure 23: Connecting cell phone using headset to the decoder.	43
Figure 24: DTMF output when pressing number 2.....	45
Figure 25: 1 st pulses of 75% duty cycle.	46
Figure 26: pulses of 25% duty cycle.	47
Figure 27: pulses of 50% duty cycle.	47
Figure 28: pulses of 25% duty cycle.	48
Figure 29: pulses of 75% duty cycle.	48
Figure 30: pulses of 50% duty cycle.....	48
Figure 31: IR2110 circuit for high-voltage full-bridge drive with independent switch control.....	49
Figure 32: output of IR2110 between 7 and 8.....	50
Figure 33: output of IR2110 between 11 and 12.....	51
Figure 34: output of IR2110.....	51
Figure 35: results at the output of the MOSFET.	52
Figure 36: picture of motor connected to the door.	53

List of tables:

Table 1: The frequencies generated on pressing different phone keys.	16
Table 2: Table showing DTMF Low and High frequency tones and decoded output.	19
Table 3: Absolute maximum ratings of IRFP450A MOSFET	26
Table 4: Output of the DTMF decoder	43
Table 5: The items used at the product and suppliers with the regions.	55
Table 6: Description and cost of each item in the project.	57

Nomenclature or list of symbols

DTMF: dual tone multi frequency

IC: Integrated Circuit

DC: direct current

PM motor: permanent magnet motor

PMW: pulse width modulation

USB: universal serial bus

AC: Alternating current

ISCP: In Circuit Serial Programming

I/O: Input/ Output

MOSFET: metal–oxide–semiconductor field-effect transistor

IGBT: Insulated Gate Bipolar Transistor

UPS: Uninterruptible power supply

CMOS: complementary metal oxide semiconductor

LSTTL: Low-Power Schottky Transistor-Transistor Logic

HVIC: High-Voltage Integrated Circuit

SOIC: Small Outline Integrated Circuit

DIP: Integrated Project

EMF: Electromotive force

DCGD: Distance Controlling Garage Door

LED: light emitting diode.

Abstract:

The main objective of this project is to unlock a garage door by a mobile phone using a password and a unique number entered through the keypad of the phone. Opening and closing of garage doors involves human labor. In this proposed system, the opening and closing of a garage door is achieved by using a mobile phone. The owner can call to a mobile phone interfaced to the system which in turn is connected to the garage door that can open/close the door by entering the password. This method is very convenient as one doesn't have to get down of his car to open/close the door physically.

This project is based on the concept of DTMF (dual tone multi - frequency). Every numeric button on the keypad of a mobile phone generates a unique frequency when pressed. These frequencies are decoded by the DTMF decoder IC at the receiving end which is fed to the Arduino. If this decoded values (number entered by the user) matches with the number stored in the microcontroller, then the Arduino initiates a mechanism to open the door through a motor driver interface.

Chapter One: Introduction.

1.1 Executive Summary

The system will allow a user to control a garage door from a cellular phone and will be broken down in four main parts: The cellular phone (serving as a platform for instructions), the DTMF decoder (which decodes the entered number and password via user's cellular phone), the control unit (receiving, interpreting and issuing commands), and the controlled device (basically a motor connected to the door). The control unit will comprise a cellular module and a DTMF decoder and a microcontroller.

The basic door control scenario will start at the cellular phone, where the user will input the password and the command in the form of unique numbers. At the control unit, the cellular module will receive the command and transmit it to the DTMF decoder which will retrieve the entered number and transmit it to the microcontroller. The microcontroller will then interpret the command and issue the appropriate control signal to the motor (connected to the door to achieve the desired command, either opening or closing).

The main component of the control unit is ATmega328 Arduino Kit microcontroller. The total project cost is almost 120\$ excluding labor.

1.2 Statement of the problem:

1.2.1 General problem statement

The objective of this project is to develop a device that allows for a user to remotely control and monitor a garage door using a cellular phone. This system will be a powerful and flexible tool that will offer this service at any time, and from anywhere with the constraints of the technologies being applied.

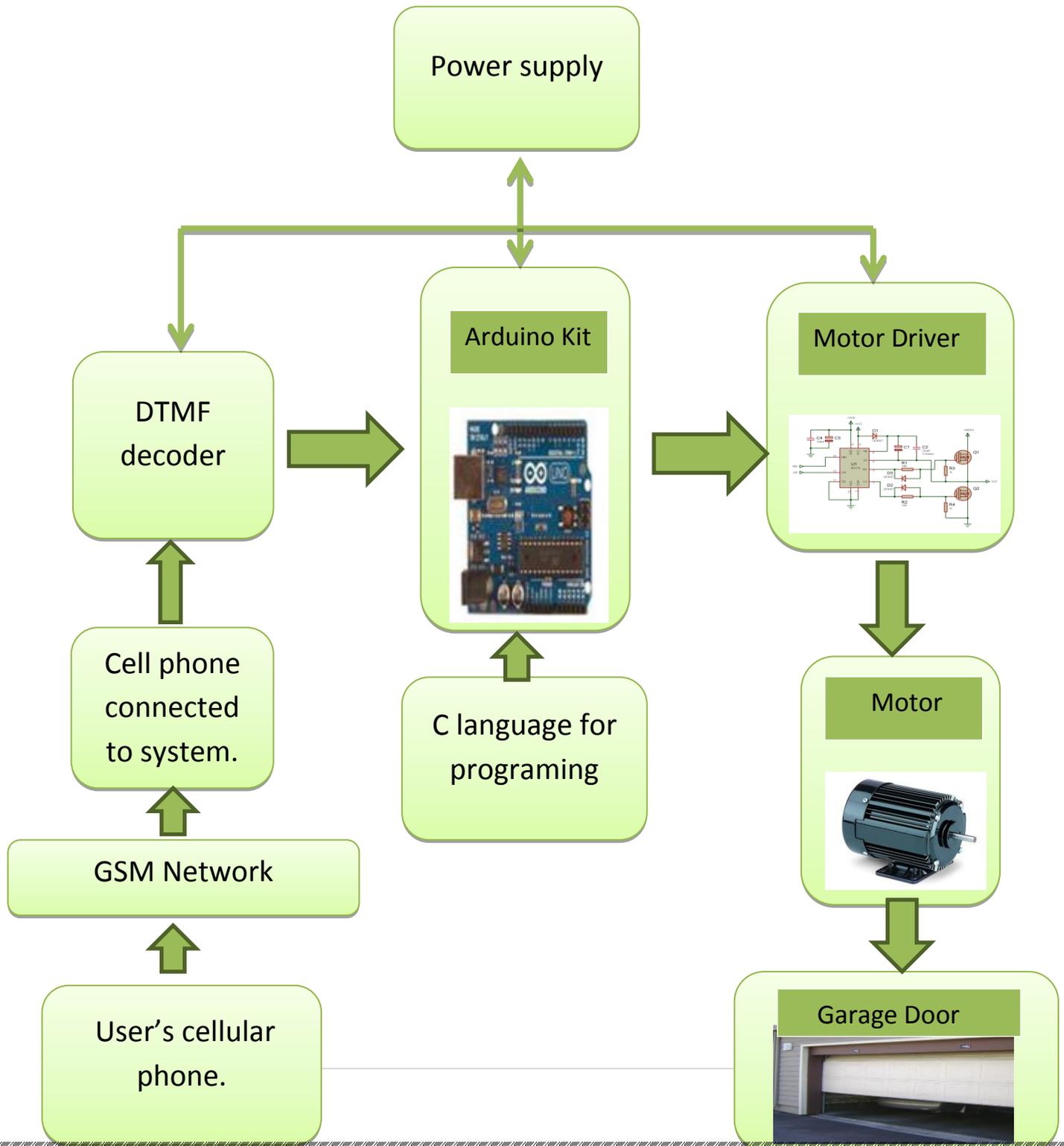
1.2.2 General Solution Approach

The proposed approach for designing this system is to implement a microcontroller-based control module that receives its instructions and commands from a cellular phone over the GSM network. The microcontroller then carries out the issued commands and sends the status of a given appliance or device back to

the cellular phone. For security purposes, a means of identification and user authentication will be implemented, and will combine caller identification with a password authorization.

1.3 Block diagram of the project.

The following block diagram illustrates the design of the system which we have built.



1.4 Organization of the report.

The report will begin by describing the constraints we faced, standards and codes we followed and earlier course work we learned from. Then literature review about previous projects we took advantage of. Then methodology, after that results and analysis, conclusions and recommendations are considered. Finally Feasibility Study is introduced.

Chapter 2: Constraints, Standards/ Codes and Earlier course work

2.1 Constrains:

In Our project we faced several constraints:

1. Unavailability of some electronic parts that we needed, so we had to bring them from the United States of America, i.e. (The IR2110 IC)
2. The sensitivity of some electronic parts (i.e. IR2110) which were damaged more than one time so we had to bring them from Jordan again, and we faced problems to deal with them because they differ from that we used before (in size) by making the wire rapping then welding them.
3. The difficulty of dealing with DTMF circuit in terms of connecting it with ARDUINO and show the pulses on the oscilloscope.
4. Since DTMF decoder deals with sum of sinusoidal signals (which must be decoded to give a binary output) the process is so sensitive and produce a noise that cause a bad effect, at first we connected it using bread board and the results did not show, then we weld it and the result did not appear clearly, so we had to implement it on a printed circuit to view the results more clearly and better.
5. The problem of changing some of the wires after welding them because they do not hold out the high current and maybe damage the circuit.

6. The difficulty of finding a suitable motor depending on the required voltage and current.
7. The difficulty of designing and making a wooden door so that the motor is able to move it right and left, so we needed several pieces like diodes gear .
8. The lack of equipment such as voltage sensor, current sensor, pico-scope, since we had to use pico-scope instead of oscilloscope to display results (floating point) and oscilloscope is grounded it is not able to achieve this goal.

But in spite of these constrains, we were able to complete the project successfully.

2.2 Standards / codes

We have implemented our project according to IEC (International Electro technical commission) standards.

2.3 Earlier course work:

Our group consist of two specialization (electrical and communication), so we will talk about each specialization alone.

In electrical specialization we took several course works that has been utilized in doing the project effectively:

1. Power electronics: In this course, we have learned about IGBT component and its working principle, and this has helped us in choosing the suitable one.
2. Control of electrical machine: In this course, we have learned about the arrangement of IGBT for making the H-Bridge for controlling the direction of the motor.
3. Electrical machines: In this course, we have learned about the Dc motor in general and the permanent magnetic motor in particular and we learned the principle of its work and that has helped us in choosing it to be in our project

In communication specialization we took one course work which is Digital Signal Processing , this course has helped us to understand how DTMF operates , its principle of work and the implementation of this using Mat-lab.

Chapter 3: Literature Review.

A DTMF technology is usually a wide application that allows the people freedom and flexibility to live a safe and comfortable life. Such save and time and use energy efficiently. DTMF technology is still at its infancy stage with a lot of research going on for its advancement. However, success has been recorded in this field as some working DTMF projects are available. The features in this field include:

- **Cell Phone Based Remote Home Control System:**

The objective of this project is to enable users to remotely control their home appliances and systems using a cell phone-based interface. To access the control unit, the user should send an authentication code along with the required function to his/her home control system via GSM. Upon being properly authenticated, the cell phone-based interface at home would relay the commands to a microcontroller that would perform the required function, and return a function completion code that would be sent to the source of the original command (user's cell phone).

As a result, the extensive capabilities of this system are what make it so interesting. From the convenience of a simple cell phone, a user is able to control and monitor virtually any electrical device in a household. This makes it possible for users to rest assured that their belongings are secure, that the garage door is shut, and that the television was not left running when they left the house to just list a few of the many uses of this system.[1]

- **DTMF Signal Controlled House Monitoring System (HMS):**

Purpose of our course project is to develop DTMF signal controlled house monitoring system based on PIC Microcontroller. The device is connected to phone line and is controlled by DTMF signal emitted by DTMF encoder in stationary or mobile phones. In presence of incoming call the device pick-up the phone and answer the call connecting to the line. User can press buttons and control the system remotely. Function of keys should be known initially.

As a result the project contained some function that was able to achieve successfully , this functions are:

- Speakerphone monitoring (you can listen the sound in remote premise),
- Room temperature monitor (sound indication),
- Output for heating switch (we can turn on/off the external heating),
- input for security check (if it was alarmed, the input must be high),
- Alarm controller (we can activate alarm system manually),
- Automatic pick-up (raises after 4 seconds or could be changed).[2]

Chapter 4: Methodology

This chapter includes the principle of work of all the fundamental devices in our project, and after each explanation, a description of the device, circuit or IC that we chose to use will be included.

4.1. DTMF:

4.1.1. What is DTMF (Dual Tone Multi Frequency)?

DTMF is a common communication term for touch tone phones. The tones formed when dialing on the keypad (DTMF digits or DTMF number) on the phone can be used to characterize the digits, and some different tone is used for each digit. DTMF signaling is mainly used in DTMF phones, telephone switching system.

Yet, there is always a possibility that a random sound will be on the similar frequency which will trip up the DTMF sounds system. It was recommended that if two tones were used to represent a digit, the probability of a false signal happening

is ruled out, thus the name ‘Dual Tone’. This is the basis of using dual tone in DTMF communication. DTMF dialing uses a keypad with 12 or 16 buttons. Each key pressed on the keypad generates two tones of particular frequencies, so a voice or a random signal cannot mimic DTMF signaling tones. One tone is generated from a High DTMF frequency group of tones and the other from Low DTMF frequency group. [3]



1	2 ABC	3 DEF	A
4 GHI	5 JKL	6 MNO	B
7 PQRS	8 TUV	9 WXYZ	C
*	0 +	#	D

Figure 1: keypad with 16 buttons.

When a button is pressed, both the row and the column tones are generated by the telephone or touch tone instrument.

These two tones will be distinctive and different from tones of other keys. So there is a low and high frequency associated with a button, it is essentially the sum of two waves is transmitted. This elementary principle can be extended to a range of applications. [3]

➤ Row and Column Frequency Correspondence:

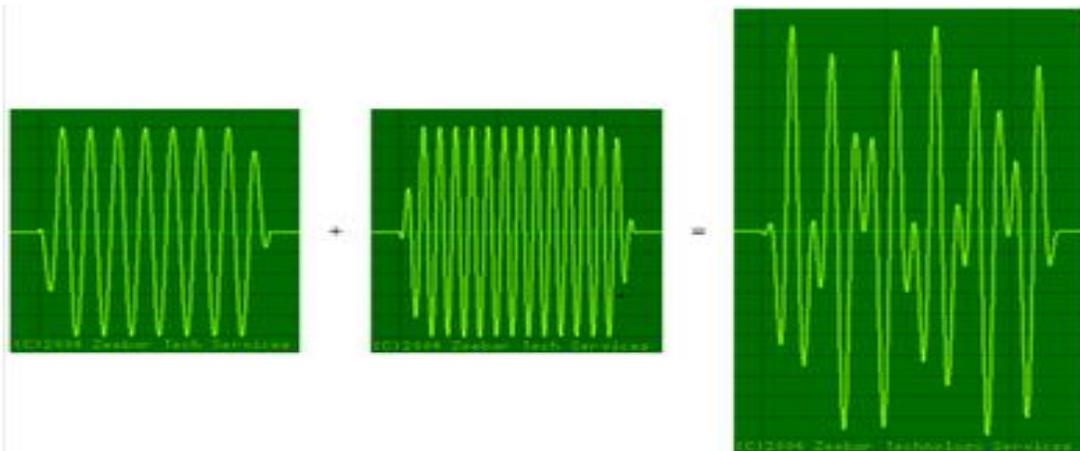
The frequencies generated on pressing different phone keys are shown in table1:

DTMF Frequency(Hz)	1209	1336	1447
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

Table 1: The frequencies generated on pressing different phone keys.

Each row and column of the keypad corresponds to a certain tone and creates a specific frequency. Each button lies at the intersection of the two tones.

For example, in order to generate the DTMF tone for “1”, we should mix a pure 697Hz signal with a pure 1209Hz signal.



$$697\text{Hz sine wave} + 1209\text{Hz sine wave} = \text{DTMF tone "1"}$$

DTMF signals can be transmitted over a radio to switch ON or switch OFF home appliances, flash lights, motors, cameras, warning systems, irrigation systems and so on. These encoded data can be stored and processed in a microcontroller to perform different tasks.

4.1.2. DTMF Decoder:

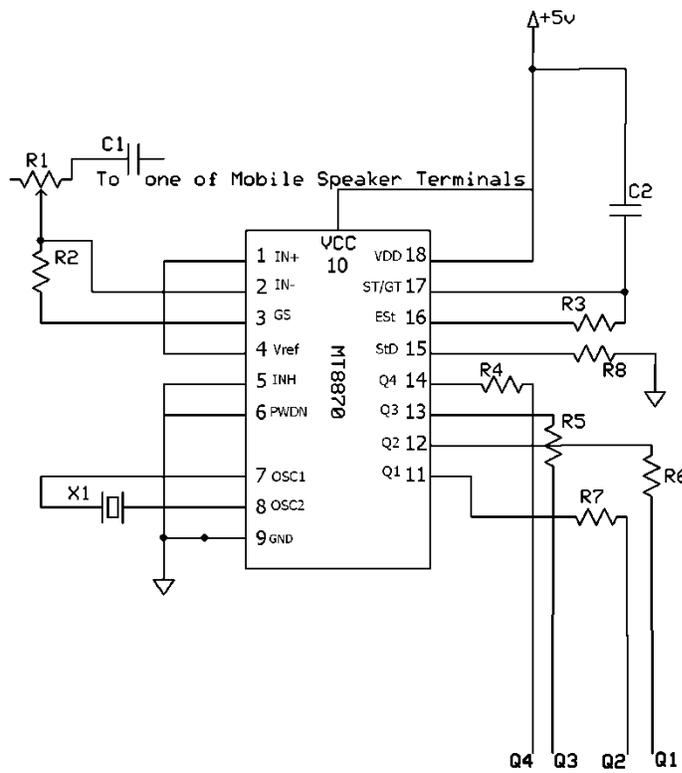


Figure 2: DTMF decoder circuit.

This DTMF decoder circuit identifies the dial tone from the telephone line and decodes the key pressed on the remote telephone. Here for the detection of DTMF signaling, we are using the IC MT8870DE which is a touch tone decoder IC. It decodes the input DTMF to 5 digital outputs. The M-8870 DTMF decoder IC uses a digital counting technique to determine the frequencies of the limited

tones and to verify that they correspond to standard DTMF frequencies. The DTMF tone is a form of one way communication between the dialer and the telephone exchange. The whole communication consists of the touch tone initiator and the tone decoder or detector [3].The decoded bits can be interfaced to a computer or microcontroller or (Arduino in our case) for further application.

➤ Working of DTMF decoder circuit:

DTMF keypads are employed in almost all landline and mobile handsets. Thus this technology is used in the telephone switching centers to identify the number dialed by the caller.

The decoder distinguishes the DTMF tones and produces the binary sequence equivalent to key pressed in a DTMF keypad.

The circuit uses MT8870 DTMF decoder IC which decodes tone generated by the keypad of cell phone. DTMF signals can be tapped directly from the microphone pin of cell phone device. Cut the microphone wire and you will get two wires red and green. The red wire is the DTMF input to the circuit.

The signals from the microphone wire are processed by the DTMF decoder IC which generates an equivalent binary sequence as a parallel output like Q1, Q2, Q3, and Q4.

Table2 showing DTMF Low and High frequency tones and decoded output.

Button	Low DTMF frequency (Hz)	High DTMF frequency (Hz)	Binary coded output			
			Q1	Q2	Q3	Q4
1	697	1209	0	0	0	1
2	697	1336	0	0	1	0
3	697	1477	0	0	1	1
4	770	1209	0	1	0	0
5	770	1336	0	1	0	1
6	770	1477	0	1	1	0
7	852	1209	0	1	1	1
8	852	1336	1	0	0	0
9	852	1477	1	0	0	1
0	941	1336	1	0	1	0
*	941	1209	1	0	1	1
#	941	1477	1	1	0	0

Table 2: Table showing DTMF Low and High frequency tones and decoded output.

We built our DTMF decoder to decode tones coming from the user's mobile phone and it is shown in figure 3.



Figure 3: DTMF Decoder.

4.2. Arduino:

4.2.1 .What is Arduino?

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicating with software running on the computer.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment. [4]

4.2.2. Why Arduino?

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

clear programming environment, Open source and extensible software, Open source and extensible hardware. [4]

In our project we used the Arduino Uno.

4.2.3. Arduino Uno:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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 an AC-to-DC adapter or battery to get started. [4]



4.2.4. Specifications:

- ATmega328 Microcontroller
- 7-12V Input Voltage.
- 14 Digital I/O Pins (6 can be used as PWM outputs).
- 6 Analog Input Pins.
- 40mA DC Current per I/O Pin.
- 16 MHz Clock Speed.

On arduino we wrote a code that takes orders from user and directs them to the system, this code is found in appendix A.

4.3. H-Bridge:

An H-Bridge is an electronic power circuit that allows motor speed and direction to be controlled. Often motors are controlled from some kind of "brain" or micro controller to accomplish a mechanical goal. The micro controller provides the instructions to the motors, but it cannot provide the power required to drive the motors. An H-bridge circuit inputs the micro controller instructions and amplifies them to drive a mechanical motor. The H-bridge takes in the small electrical signal and translates it into high power output for the mechanical motor.

The H-Bridge electronic structure is explicit in the name of the circuit - **H -- Bridge**. The power electronics actually form a letter H configuration, as shown in Figure 4. The switches are symbolic of the electronic Power MOSFETs which are used for switching.

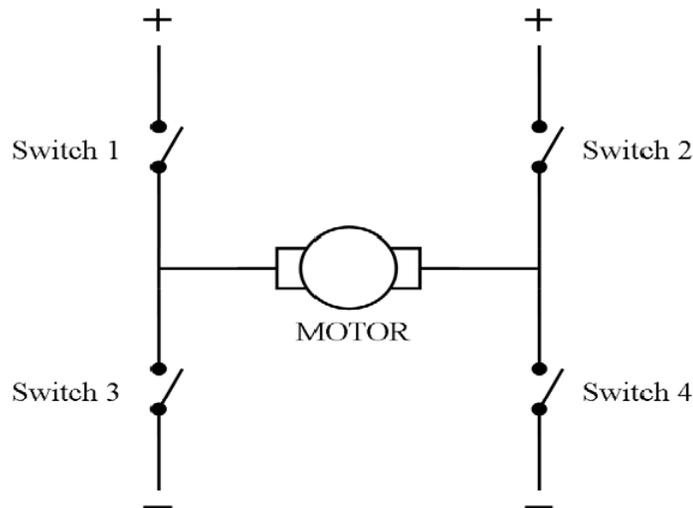


Figure 4 :H-Bridge Topology.

If it is desired to turn the motor on in the *forward* direction, switches 1 and 4 must be closed to power the motor. Figure 5 below is the H-Bridge driving the motor in the forward direction. And if it is desired to turn the motor on in the *reverse* direction, switches 2 and 3 must be closed to power the motor. Figure 6 below is the H-Bridge driving the motor in the reverse direction.

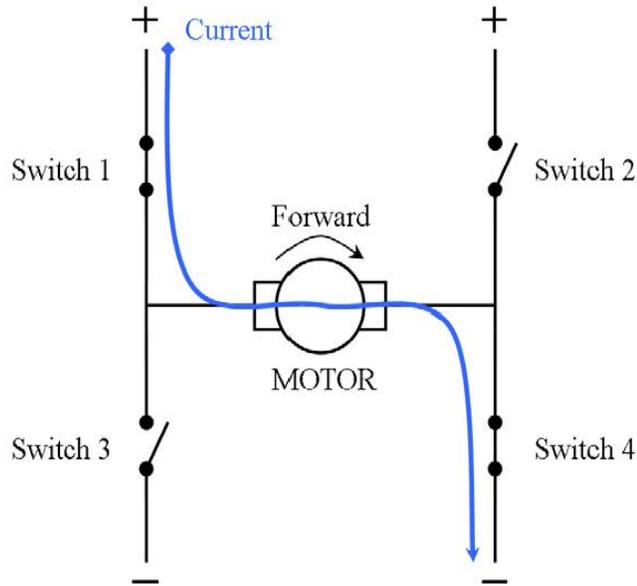


Figure 5 :H-Bridge Topology – Forward Direction.

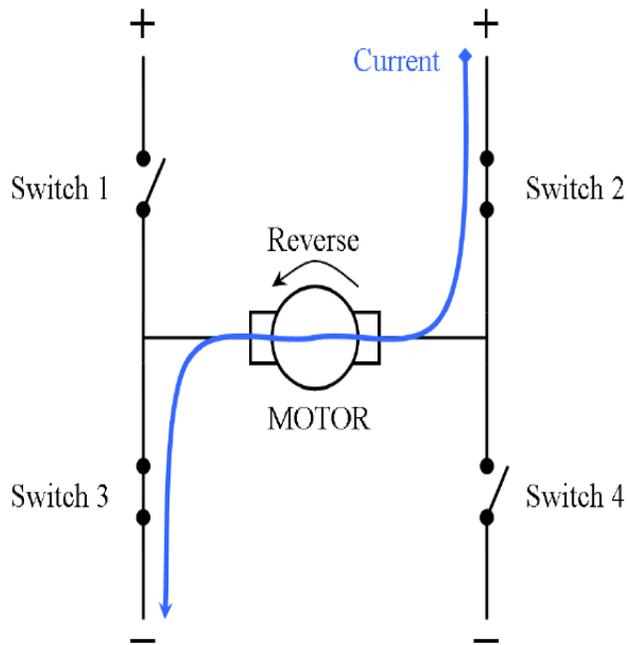


Figure 6 :H-Bridge Topology – Reverse Direction.

4.3.1 Speed Control - PWM Technique

The motor is controlled by the 4 switches above. For the speed control explanation that follows only switches 1 and 4 will be considered because speed

control is identical in the forward and reverse direction. Say the switches 1 and 4 are turned on, the motor will eventually run at full speed. Similarly if only switch 4 is turned on while switch 1 is off the motor stops. Using this system, how could the motor be running at 1/2 of the full speed? The answer is actually quite simple; turn switch 1 on for half the time and turn it off for the other half. In order to implement this system in reality, one must consider two main factors, namely frequency and duty cycle.

Frequency: Using the switch example, the frequency would be how fast the switch was turned on and off. If the frequency is too low (switch is changed slowly), then the motor will run at full speed when the switch is on, and completely stop when the switch is off. But if the frequency is too high, the switch may mechanically fail. In reality there is no switch, but rather an electronic board named an H-Bridge that switches the motor on and off. So in electrical terms; if the frequency is too low, the time constant of the motor has enough time to fully switch between on and off. Similarly the upper limit on the frequency is the limit that the H-Bridge board will support, analogous to the mechanical switch.

Duty Cycle: The duty cycle is analogous to how long the upper switch (switch 1) remains on as a percentage of the total switching time. In essence it is an average of how much power is being delivered to the motor. Duty cycle gives the proportional speed control of the motor. Figure 6 is an example of 1/4, 1/2, and 3/4 duty cycles. Effectively, these duty cycles would run the motor at 1/4, 1/2, and 3/4 of full speed respectively. [5]

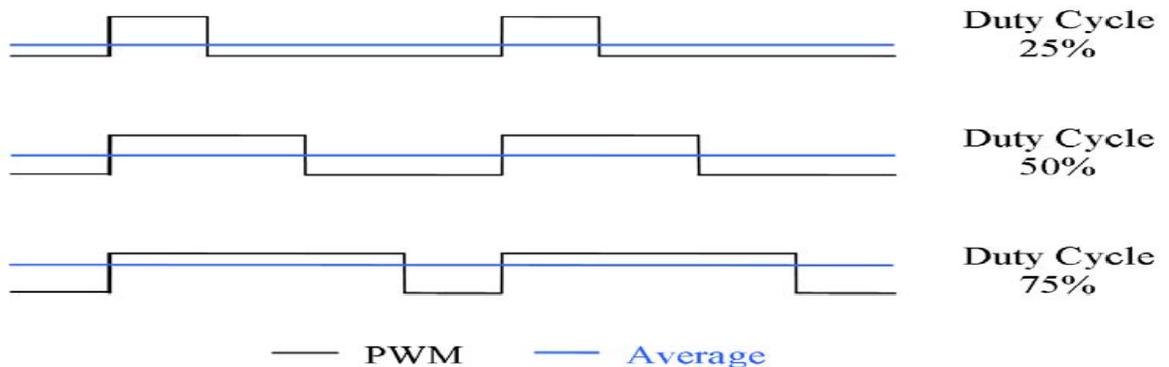


Figure 7 : Pulse Width Modulation Used For Motor Control.

So to convert the fixed voltage into a variable voltage \ variable current source for the speed control of the dc motor drive, the variable dc voltage is controlled by chopping the input voltage by varying the on and off times of a converter, and as shown above the type of convertor capable of such function is known as H-bridge.

4.4 MOSFET

4.4.1 What is MOSFET?

MOSFET stands for metal-oxide semiconductor field-effect transistor. It is a special type of field-effect transistor (FET). Unlike BJT which is ‘current controlled’, the MOSFET is a voltage controlled device. The MOSFET has “gate“, “Drain” and “Source” terminals instead of a “base”, “collector”, and “emitter” terminals in a bipolar transistor. By applying voltage at the gate, it generates an electrical field to control the current flow through the channel between drain and source, and there is no current flow from the gate into the MOSFET.

A MOSFET may be thought of as a variable resistor, where the Gate-Source voltage difference can control the Drain-Source Resistance. When there is no applying voltage between the Gate-Source, the Drain-Source resistance is very high, which is almost like an open circuit, so no current may flow through the Drain-Source. When Gate-Source potential difference is applied, the Drain-Source resistance is reduced, and there will be current flowing through Drain-Source, which is now an open circuit.

In a nutshell, a FET is controlled by the Gate-Source voltage applied (which regulates the electrical field across a channel), like pinching or opening a straw and stopping or allowing current flowing. Because of this property, FETs are great for large current flow, and the MOSFET is commonly used as a switch.

4.4.2 How to use MOSFET as a switch?

To use a MOSFET as a switch, you have to have its gate voltage (V_{gs}) higher than the source. If you connect the gate to the source ($V_{gs}=0$) it is turned off. For example we have a IRFZ44N which is a “standard” MOSFET and only turns on when $V_{gs}=10V - 20V$. But usually we try not to push it too hard so 10V-15V is common for V_{gs} for this type of MOSFET. However if you want to drive this from an Arduino which is running at 5V, you will need a “logic-level” MOSFET that can be turned on at 5V ($V_{gs} = 5V$). For example, the ST STP55NF06L. You should also have a resistor in series with the Arduino output to limit the current, since the gate is highly capacitive and can draw a big instantaneous current when you try to turn it on. Around 220 ohms is a good value.[6]

To build our H-bridge we chose to use IRFP450A MOSFET, and we will state some of its features and specifications:

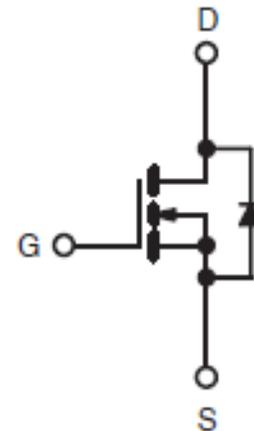
4.4.3 IRFP450A MOSFET

APPLICATIONS

- Switch Mode Power Supply (SMPS).
- Uninterruptable Power Supply.
- High Speed Power Switching.

TYPICAL SMPS TOPOLOGIES

- Two Transistor Forward.
- Half Bridge, Full Bridge.
- PFC Boost. [7]



ABSOLUTE MAXIMUM RATINGS $T_C = 25\text{ }^\circ\text{C}$, unless otherwise noted			
PARAMETER	SYMBOL	LIMIT	UNIT
Drain-Source Voltage	V_{DS}	500	V
Gate-Source Voltage	V_{GS}	± 30	
Continuous Drain Current	V_{GS} at 10 V	$T_C = 25\text{ }^\circ\text{C}$	14
		$T_C = 100\text{ }^\circ\text{C}$	8.7
Pulsed Drain Current ^a		I_{DM}	56
Linear Derating Factor			1.5
Single Pulse Avalanche Energy ^b		E_{AS}	760
Repetitive Avalanche Current ^a		I_{AR}	14
Repetitive Avalanche Energy ^a		E_{AR}	19
Maximum Power Dissipation	$T_C = 25\text{ }^\circ\text{C}$	P_D	190
Peak Diode Recovery dV/dt^c		dV/dt	4.1
Operating Junction and Storage Temperature Range		T_J, T_{stg}	- 55 to + 150
Soldering Recommendations (Peak Temperature)	for 10 s		300 ^d
Mounting Torque	6-32 or M3 screw		10
			1.1

In our system we need to use MOSFETs configured as high-side and low-side switches. Since we have a bridge circuits, in full-bridge circuits we have 2 high-side MOSFETs and 2 low-side MOSFETs. In such situations, there is a need to use high-side drive circuitry alongside low-side drive circuitry. The most common way of driving MOSFETs in such cases is to use high-low side MOSFET drivers. The most popular such driver chip is the IR2110. The circuit required for the system is shown in figure 8.

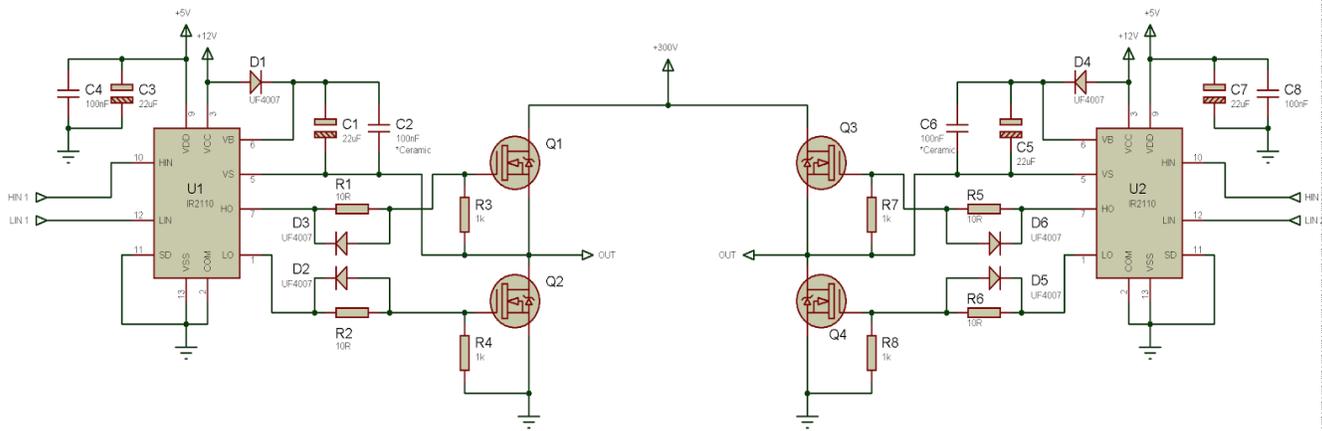


Figure 8: IR2110 circuit for high-voltage full-bridge drive with independent switch control.

4.4. Full bridge drive:

4.4.1. IR2110

The IR2110 are high voltage, high speed power MOSFET and IGBT drives with independent high and low side referenced output channel.

The IR2110 comes in two packages-14 pin through-hole p DIP package and the 16-pin surface mount SOIC package.

Proprietary HVIC and latch immune CMOS technologies enable ruggedized monolithic construction. Logic inputs are compatible with standard CMOS or LSTTL output, down to 3.3V logic.

The output drivers feature a high pulse current buffer stage designed for minimum driver cross-conduction. Propagation delays are matched to simplify use in high frequency applications. The floating channel can be used to drive an N-channel power MOSFET or IGBT in the high side configuration which operates up to 500 or 600 volts. [8]

In our project we used both 14 pin through-hole p DIP package and the 16-pin surface mount SOIC package.

Lead Assignments

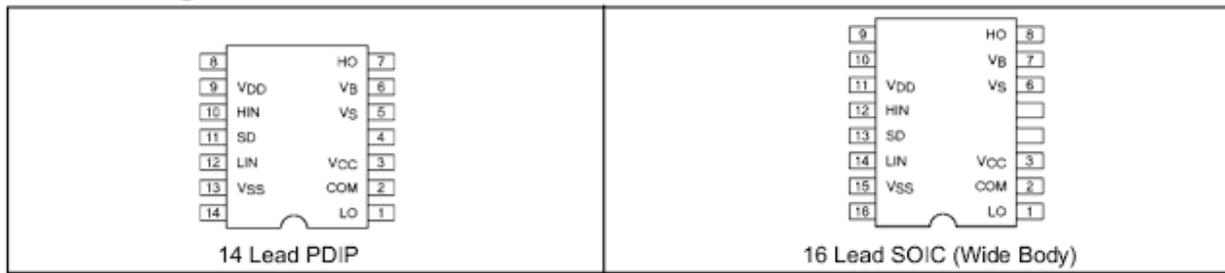


Figure 9: IR2110 Pin/Lead Assignments.

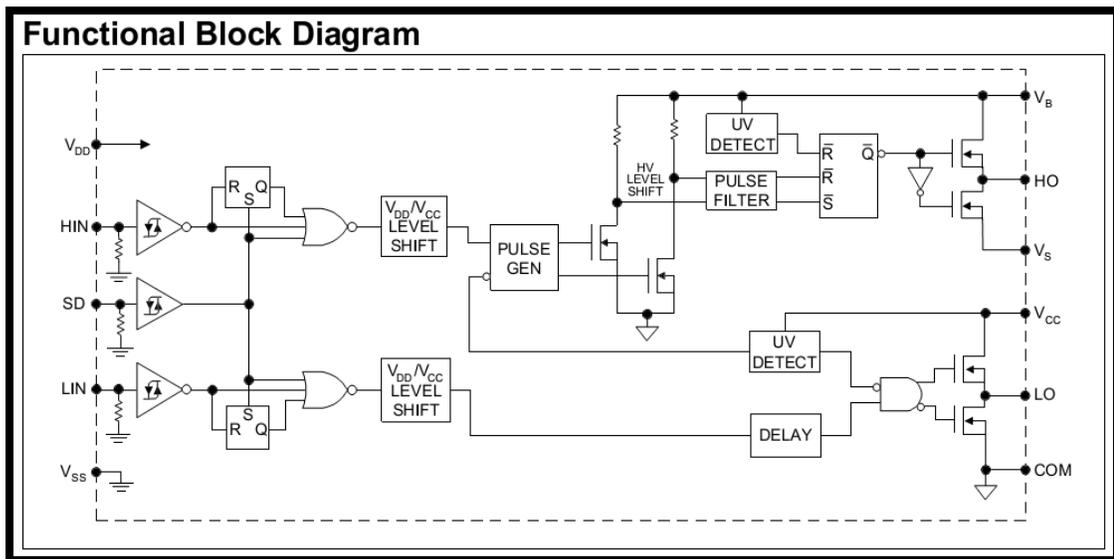


Figure 10: Functional Diagram of IR2110.

Lead Definitions

Symbol	Description
V _{DD}	Logic supply
HIN	Logic input for high side gate driver output (HO), in phase
SD	Logic input for shutdown
LIN	Logic input for low side gate driver output (LO), in phase
V _{SS}	Logic ground
V _B	High side floating supply
HO	High side gate drive output
V _S	High side floating supply return
V _{CC}	Low side supply
LO	Low side gate drive output
COM	Low side return

Figure 11:IR2110 Pin/Lead Definitions.

4.5. Permanent Magnet DC Motor.

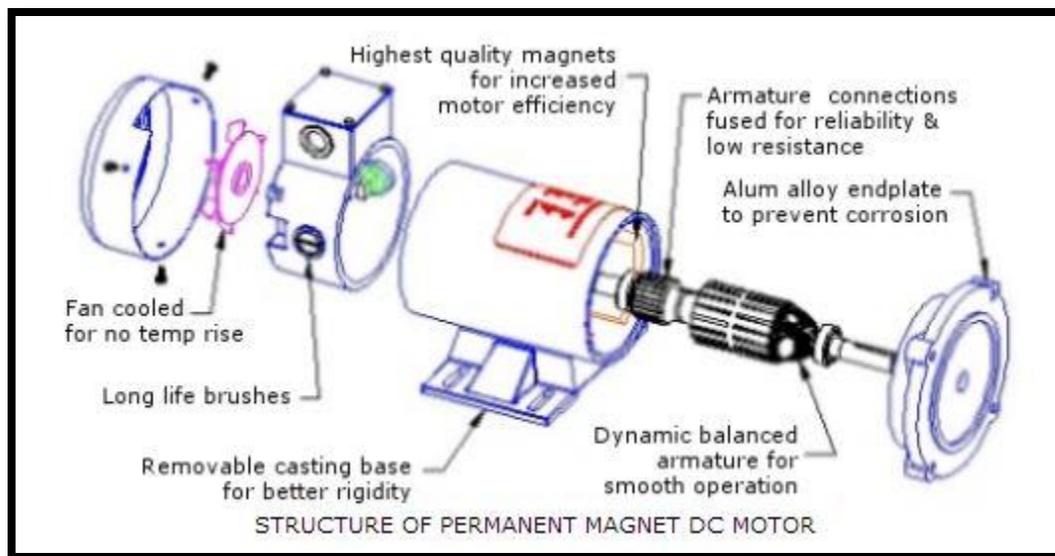


Figure 12:Structure of Magnet DC Motor.

The principle of working for the PM motor based on that a force is generated when current flows in an inductor that is placed in a magnetic field, these motors are known for their simplicity and low maintenance. [9]

A PM motor does not have a field winding on the stator frame, instead relying on PMs to provide the magnetic field against which the rotor field interacts to produce torque. Because this field is fixed, it cannot be adjusted for speed control. PM fields (stators) are convenient in miniature motors to eliminate the power consumption of the field winding; field windings were more practical to obtain the needed amount of flux [10]

The rotational speed of a permanent magnetic DC motor is proportional to the EMF in its coil (= the voltage applied to it minus voltage lost on its resistance), and the torque is proportional to the current. Speed control can be achieved by variable battery tapings, variable supply voltage, resistors or electronic controls. The direction of a wound field DC motor can be changed by reversing either the field or armature connections but not both.

A permanent magnet DC motor is characterized by a linear relationship between stall torque when the torque is maximum with the shaft at standstill and no load speed, with no applied shaft torque and maximum output speed. There is a quadratic power relationship between these two speed-axis points.

❖ The torque equal = $k\Phi\omega$;

Where K is constant, Φ is the magnetic flux and ω is the speed of the motor

❖ The back-EMF (voltage) = $k\Phi I_a$.

Where the I_a is the Armature current.

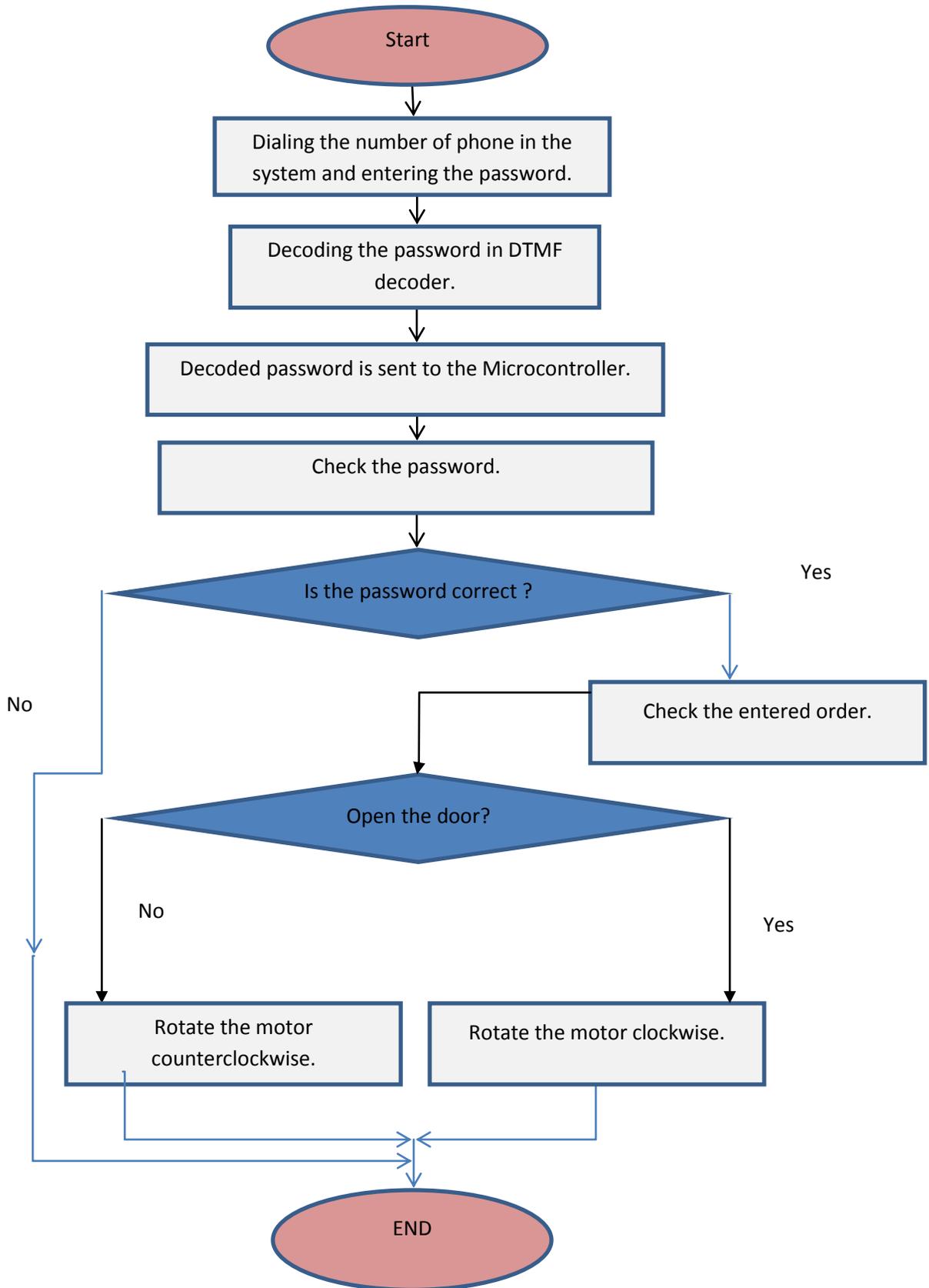
We chose to work with PM DC motor with the following specifications:

- Type of the motor: Permanent Magnet DC motor taken from Electric screwdriver.
- Rated voltage: 12 volts, but 6 volts were enough to be used because 12 volts makes the rotation of the motor very quickly (since we built a small

model of garage door) which leads the garage door to move also in a very quick mode which is unacceptable .

- Rated current: depends on the load but in our project we need just one or two amperes.

4.6. Flow Chart shows the details of operation of the project:



Chapter 5: Results and Analysis and discussion.

5.1 Simulation:

Here we introduce our simulation results and discussion:

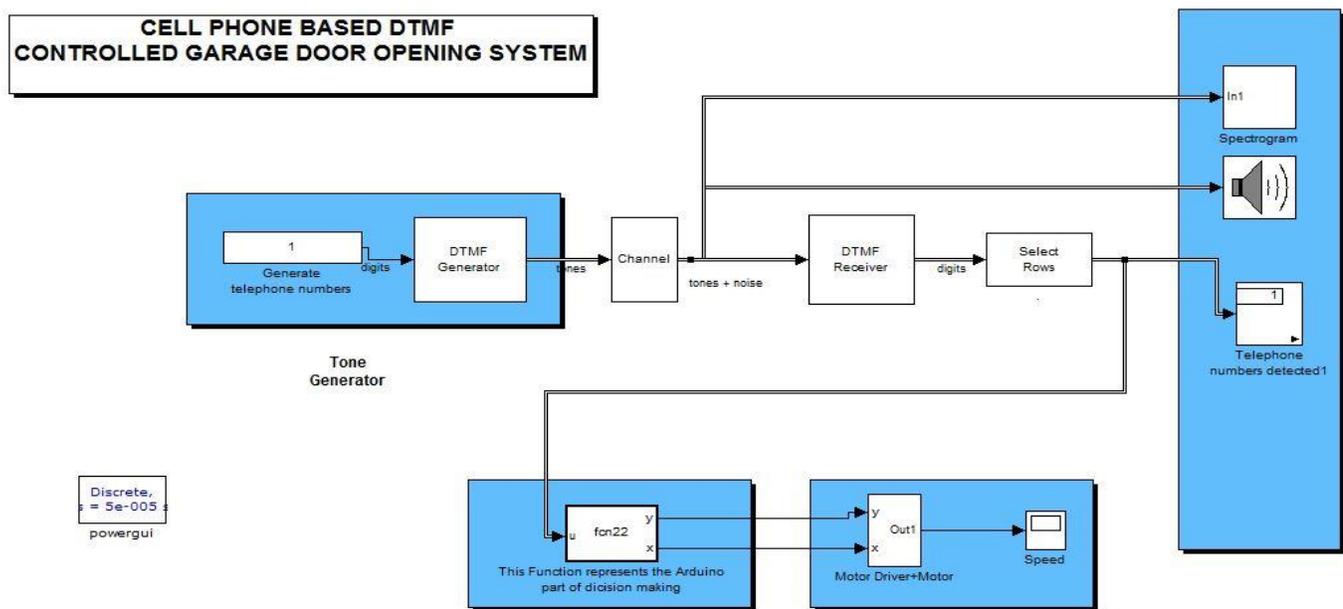


Figure 13: Simulation model of the project.

Figure 13 shows the the Matlab\Simulink model of H-Bridge inverter .It consists of four IGBT's/diodes .The H-bidge inverter is connected to the DC motor.

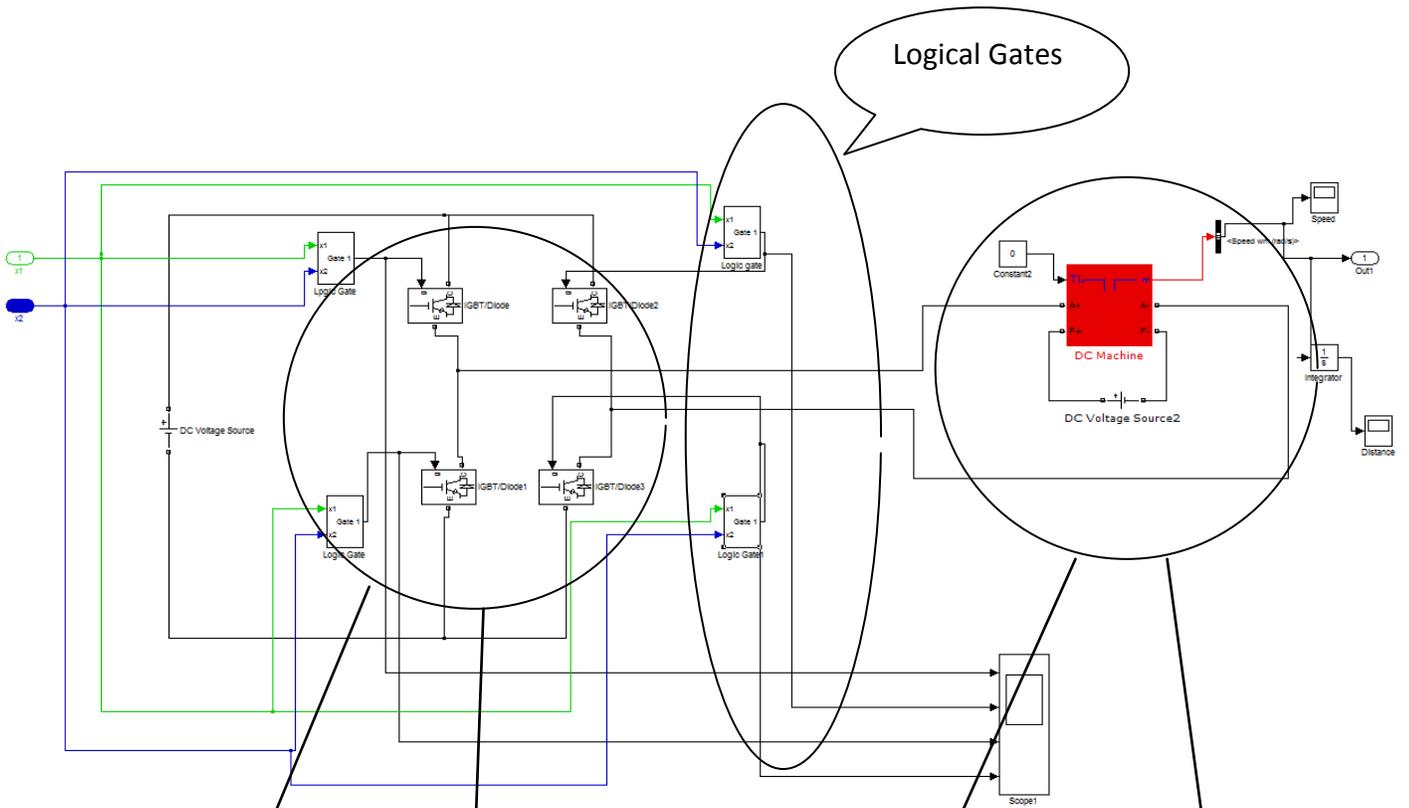
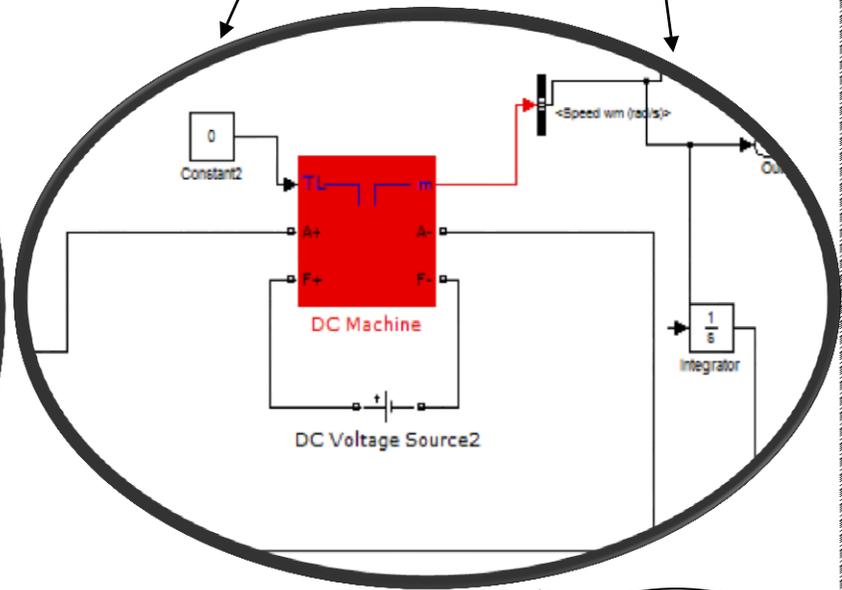
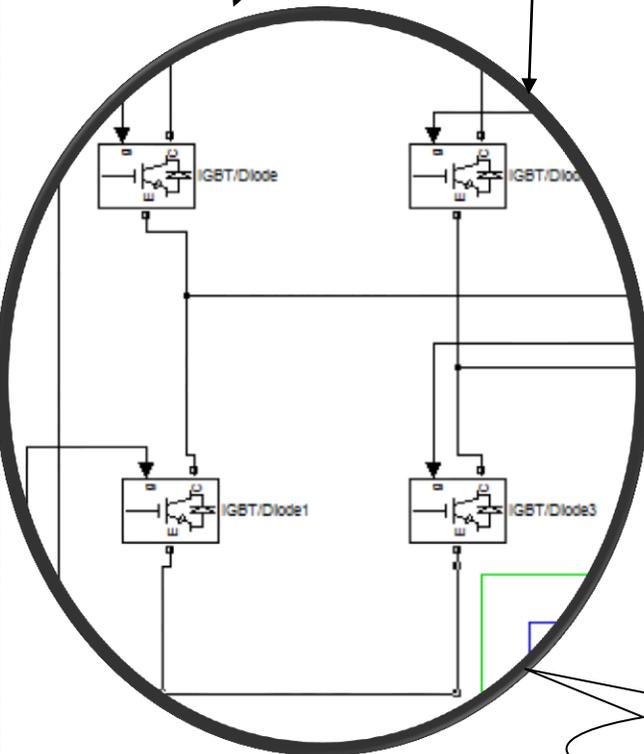


Figure 14: H-Bridge inverter and DC Motor.



H-Bridge

DC Motor

The principle of work of our model is Distance Controlling Garage Door .We want to implement this by using DTMF decoder circuit connected to a cell phone mobile.(here in the simulation we generated a tone which was decoded in the DTMF receiver to give the number that was dialed).

The numbers we used are 3, 4 (3 for opening the door and 4 to close it), of course a more complicated password can be used in the same way.

So for example when you enter number 3 using mobile phone, the tones will be decoded in the DTMF decoder.

Then the decoded number will be sent to the Arduino Kit which will decide whether the number is for opening or closing. Based on this decision pulses will be generated.

And since our case here is opening (number 3 was pressed) the door will be opened and the motor will be rotating clockwise.

✚ The whole model results after running the simulation when you press number 3 is shown in the figure below :

✓ The results on the scope that shows the speed of the motor:



Figure 15: Speed variation on the speed scope

For the opening case (pressing number three) , the DC machine will act as a motor and the speed will increase gradually , in other words there will be acceleration and the induced torque will be larger than the load torque which will force the motor to rotate clockwise in order to open the door . In fact it will open a defined distance which is 3 meters in our project. When the motor reaches its maximum speed the machine will start acting as a generator and the system will decelerate and the speed will continue decreasing until it reaches its minimum value which is equal to zero as shown in figure 15.

✓ Output pulses when pressing number 3 :

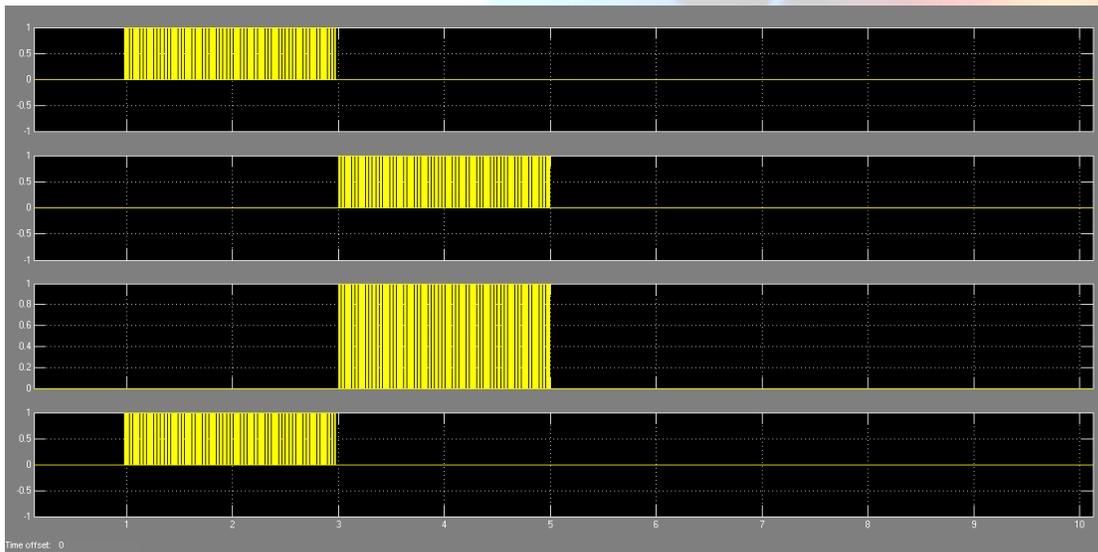


Figure 16: Pulses that enters the Gates of IGBT (gate 1 ,2,3,4) in order.

Two IGBT's can work at once , for example IGBT's with numbers 1 and 3 can't work together , only IGBT's on the diagonal line can work together so IGBT's 1 and 4 are working in this case to execute opening the door and IGBT's 2 and 3 are related to the back emf which stops the opening order .

✓ The distance variation on the distance scope (when 3 is pressed) :

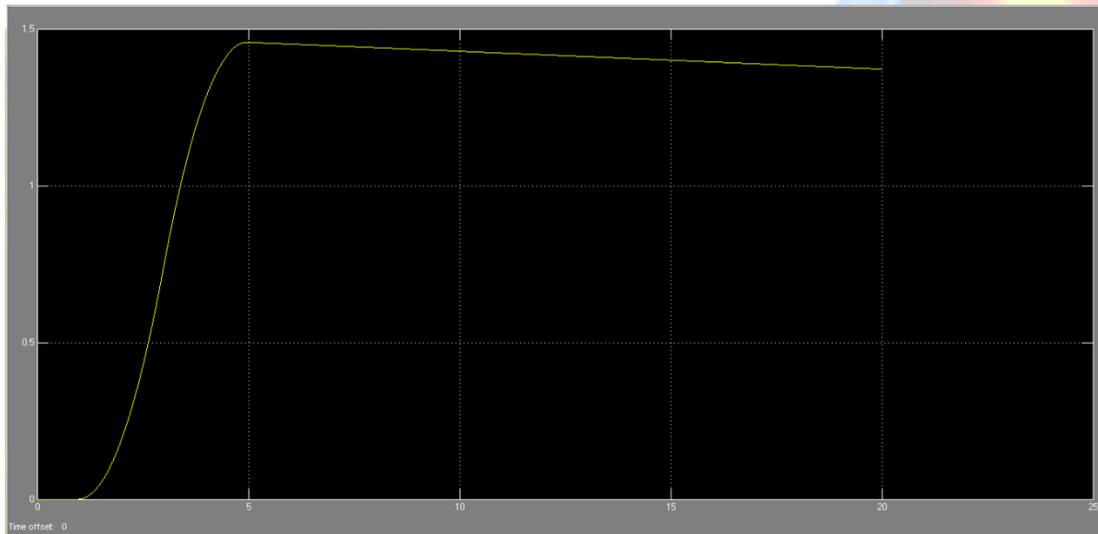


Figure 17: Distance variation on distance scope.

Since the distance is the integral of the speed then we must calculate the total area under the curve of the speed. We found it 3, then the maximum distance which the door can open will be 3 meters. We controlled this distance by changing the duty cycle, which we have controlled by changing the voltage as shown in figure 17.

✚ The whole model after running the simulation when you press number 4 is shown in the figure below :

✓ Output speed when pressing number 4 :

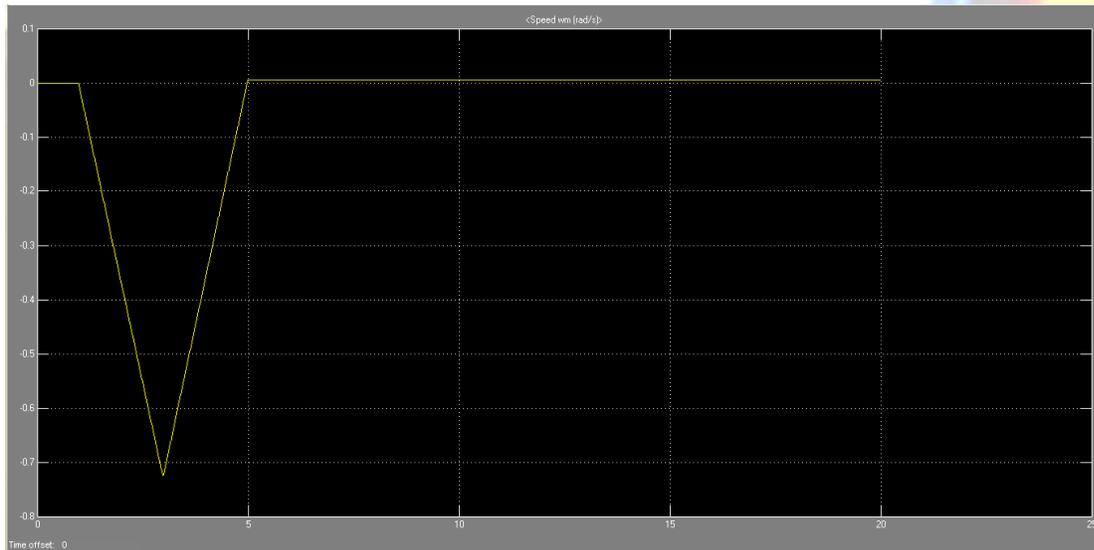


Figure 18: Speed variation when pressing number 4.

In the second case which is closing the door (pressing number 4), the opposite of what happened in the above case will happen and the motor will rotate counter clockwise. The torque induced will increase but in the negative direction which makes the speed decrease until it reaches its minimum value and then it will increase gradually until it reaches zero which means that the order has been executed and the door was closed as shown in figure 18 .

✓ Output pulses when pressing number 4 :

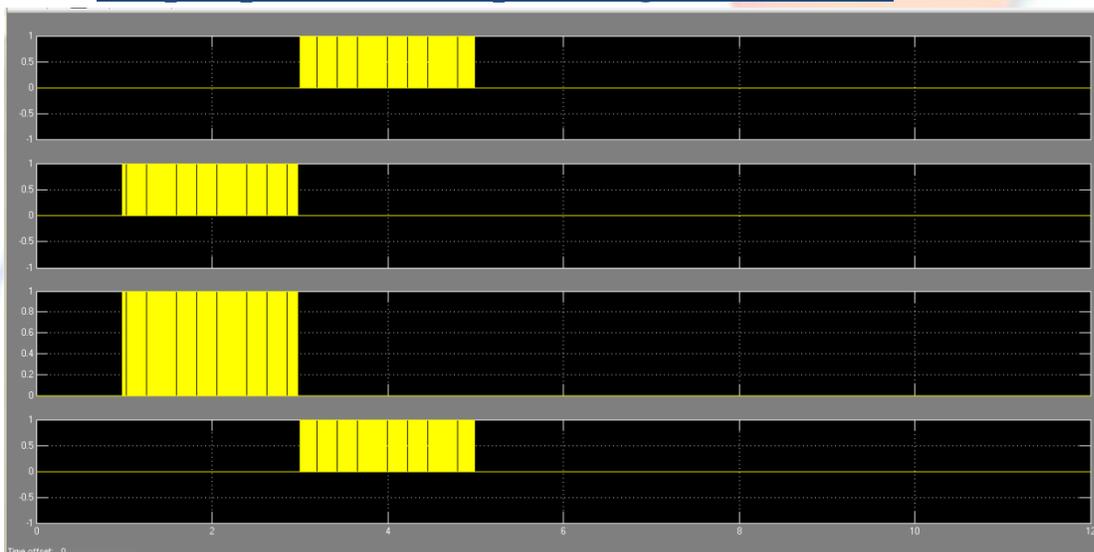


Figure 19: output pulses when pressing number 4.

In this case IGBT's 2 and 3 are working in order to close the door, and the rest IGBT's (1 and 4) are related to the back emf in order to stop the closing proses.

✓ **Output distance when pressing number 4:**

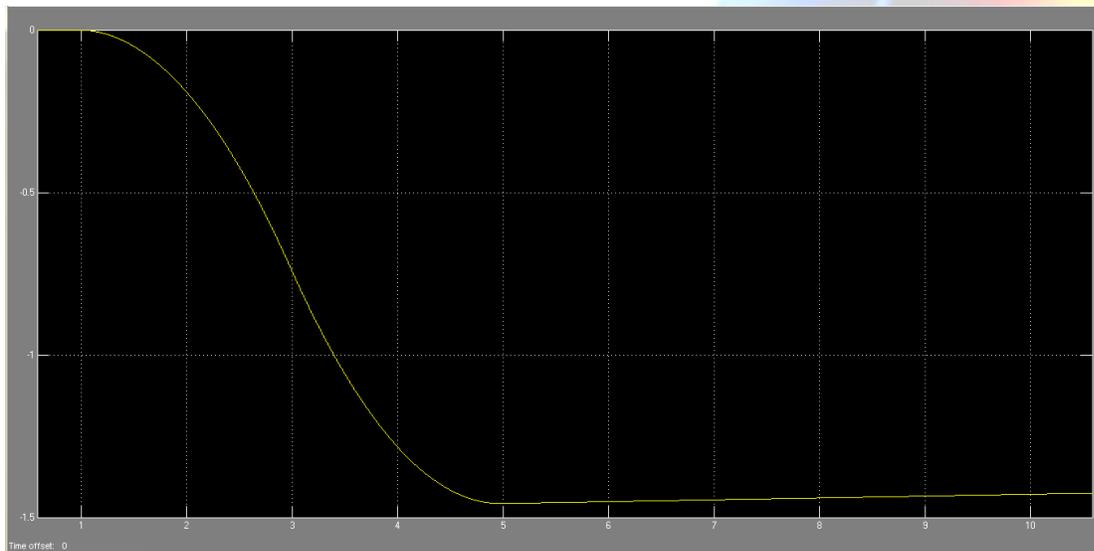


Figure 20: Distance variation on distance scope

As for the distance in the closing case , it the same as the first case except that the door will close gradually after it was opened for a distance equals 3 meters and the curve of the distance shows obviously the decreasing of the curve from the maximum value (3 meters) to the zero point as shown in figure 20 .

Final conclusions and notes about the model:

Once we had run the simulation we noticed that the results were different from each other when pressing number three and number four.

When we pressed number one which means opening the door the speed will start increasing from zero at first until it reaches its maximum value and then will decrease gradually until it returns to zero again. And the opposite thing happened when we had pressed number four.

The speed here started decreasing from zero to its maximum value and then increasing until it reaches zero again. As for the pulses, we had seen that only two IGBT's/Diodes work when pressing a number.

When we pressed number one only the first and the fourth IGBT's worked. But when we pressed number two only the second and the third IGBT's worked. Actually this is also related to the distance, and also as we studied before that the electromotive force (emf) equals to $(k * \Phi * w)$.

So it is obvious that the (emf) depends on the speed (w). Which will also affect the distance. So when press number one the (emf) which will be found between the terms of the two working IGBT's (1, 4) will be greater than the main voltage from the DC voltage source which force the motor to rotate clockwise.

Otherwise when we press number two the (emf) which will be found between the terms of the two working diodes (2, 3) will be less than the main voltage from the DC voltage source which force the motor to rotate counter clockwise, and between these two stages of rotating, there will be a stop station for the motor.

Notes:

Note 1: we used a pulse generator to generate the pulses shown in figures 16, 19 above which illustrate the IGBT's that work at the same time when pressing a unique number.

Note 2: If you want to know which IGBT is working at any instant of time you can put a voltage measurement between the collector and the emitter and if you found a low voltage difference between them then you can judge that this IGBT is

working, otherwise if you found the voltage difference is high then you can judge that it is not working.

5.2 Results and Analysis of real implementation of the system.

In this section we will walk through our project step by step to introduce obtained results.



The first component in our system is the cellular phone that plays a dominant rule in delivering user's orders to the system, since it is serving as a platform for instructions. For this phone some settings are necessary to enable it to do its purpose effectively. From the phone settings > call settings, user should choose the Automatic answering mode (which only operates with a connected headset), this is shown in figure 21 below:

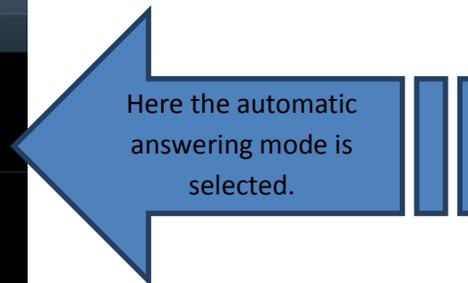
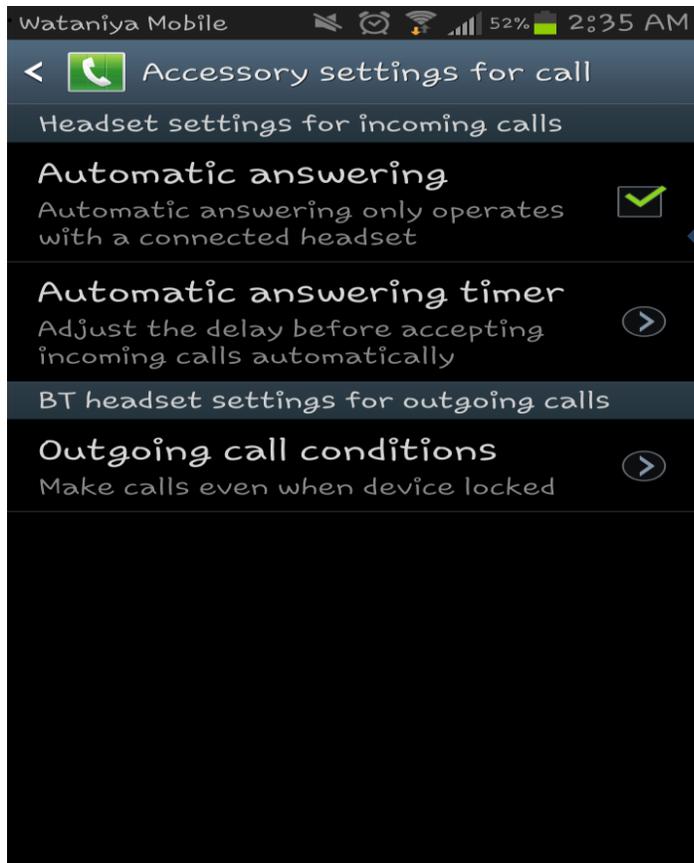


Figure 21: Automatic answering setting.

The previous step is an important step since it is intended to control the garage remotely, which means that the user calls the phone connected to the system and it answers automatically to receive orders from the user and forward them to next component of the system. It is possible also in some devices to set an automatic answering timer (this controls how many seconds the phone waits before answering the user's call, this shown in figure 22.

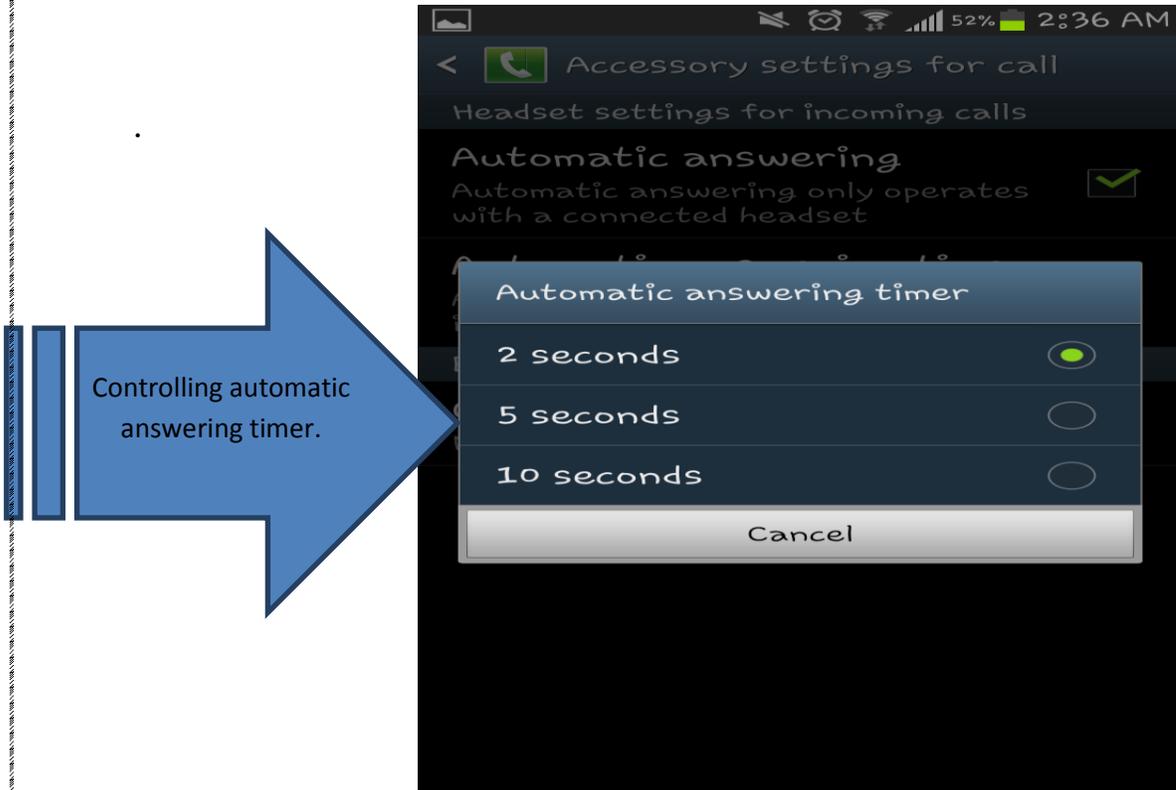


Figure 22: Automatic answering timer.

This phone is connected to the system using a headset to insure that the system is able to get the tones resulting from pressing numbers by user's device during the call that intends to control the door.

The second part of the system is the DTMF decoder which is extremely important in translating the user's orders into an output (binary encoded) that is understandable by the controller, DTMF decoder takes the tones from the cellular phone via the headset and decode them to give to corresponding number, which is then forwarded to the microcontroller. This is shown in figure 23.

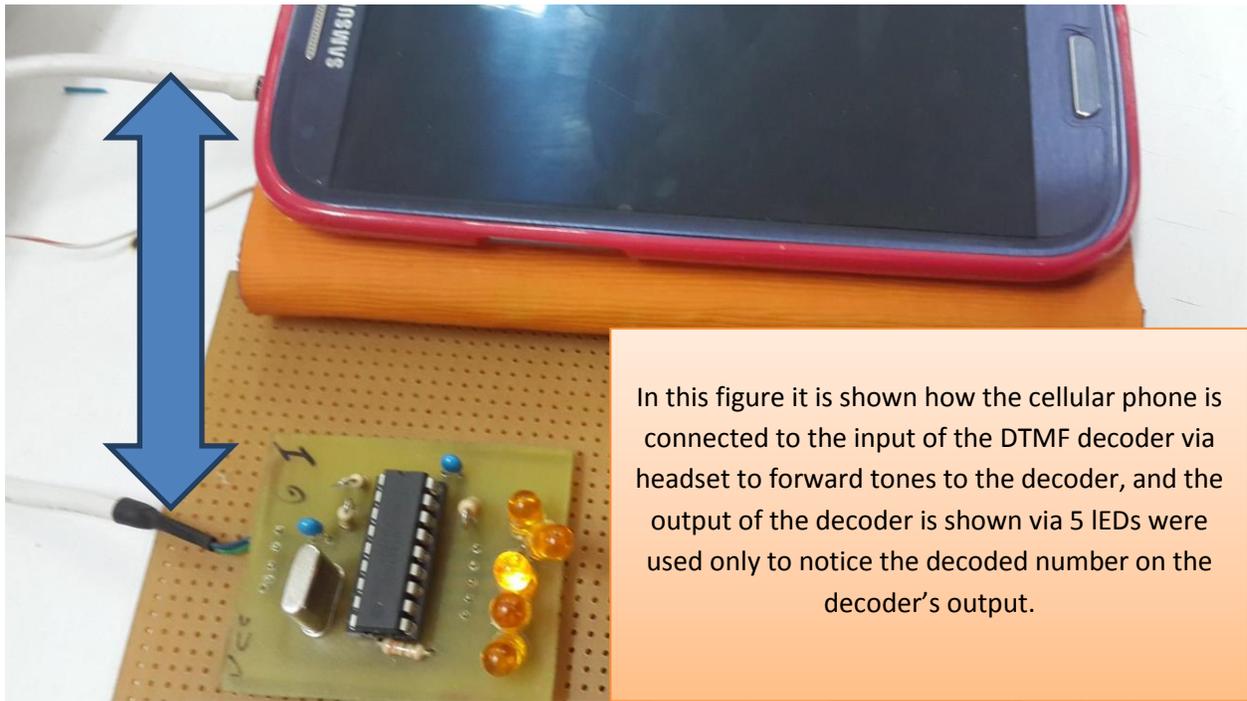


Figure 23: Connecting cell phone using headset to the decoder.

The expected output of the decoder is one of the following options shown in table4.

Button	Low DTMF frequency (Hz)	High DTMF frequency (Hz)	Binary coded output			
			Q1	Q2	Q3	Q4
1	697	1209	0	0	0	1
2	697	1336	0	0	1	0
3	697	1477	0	0	1	1
4	770	1209	0	1	0	0
5	770	1336	0	1	0	1
6	770	1477	0	1	1	0
7	852	1209	0	1	1	1
8	852	1336	1	0	0	0
9	852	1477	1	0	0	1
0	941	1336	1	0	1	0
*	941	1209	1	0	1	1
#	941	1477	1	1	0	0

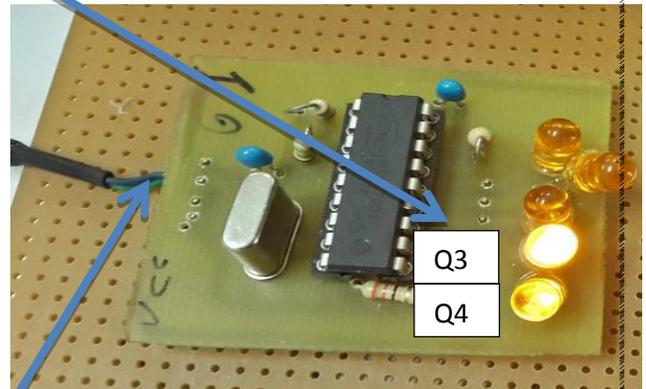
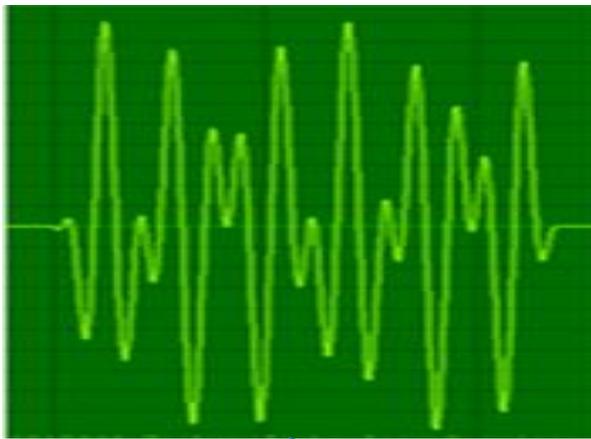
Table 4: Output of the DTMF decoder.

- And to make the process clear an example is given below:

4.The output of the decoder (as shown in table...) should be >>

Q1	Q2	Q3	Q4
0	0	1	1

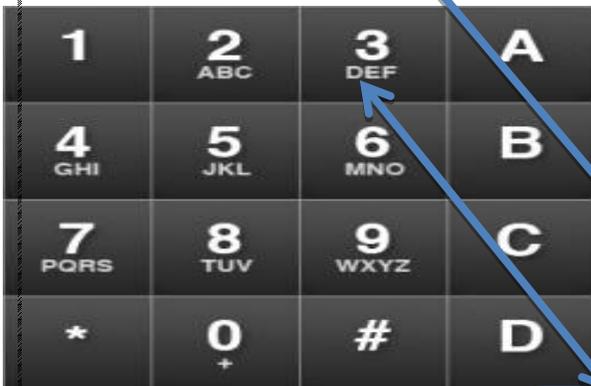
5.The output of the decoder is shown by LEDs (Q3,Q4 became ON)



3.The resulting tone of pressing 3 will be the input of the decoder.

2.The resulting tone of pressing 3.

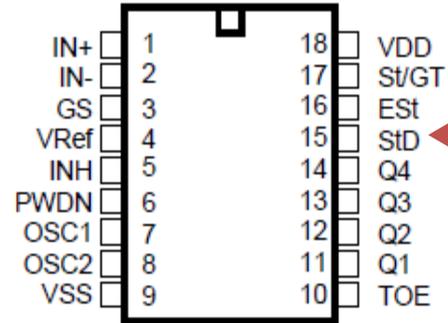
1. Press 3 on the phone's keypad.



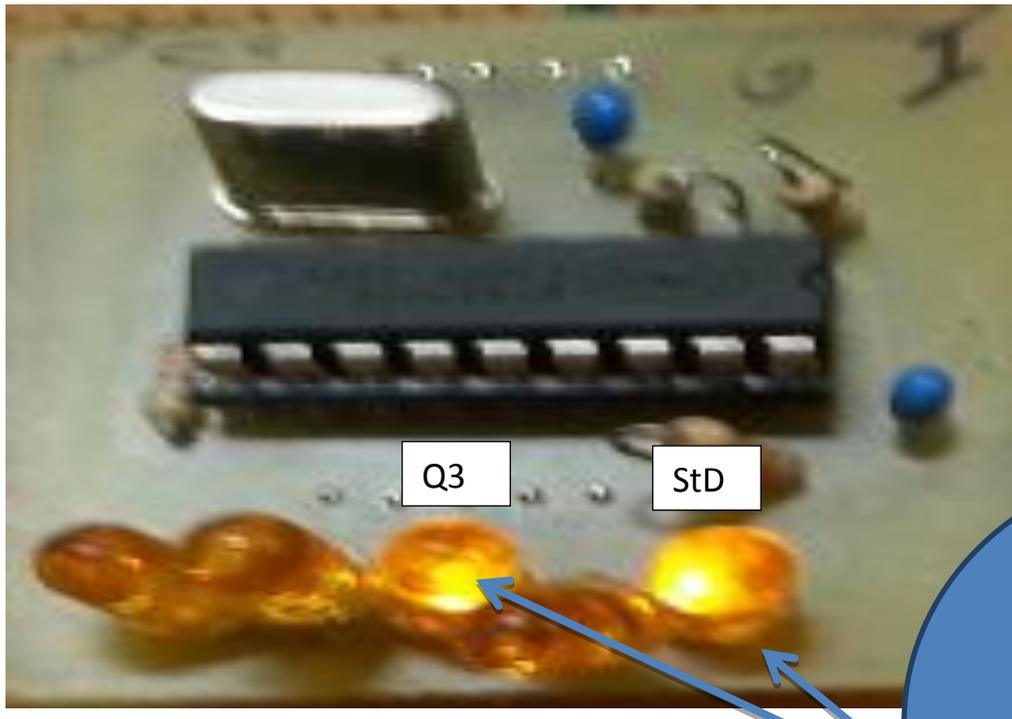
- Another important output from the DTMF decoder is put on the Std pin.

Std: Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated. Returns to logic low when the voltage on St/GT falls below VTSt.

- In simple words : when a tone is introduced to the decoder StD pin presents a logic high an immediately logic low again.(this property of StD helps to detect the output of the decoder by the controller).



- This can be shown in the figure 24 where pressing number 2 results in logic high output on the StD, from table 4 Q3 is also going to have a logic high output.



The moment number 2 is pressed StD goes high, then low. As for Q3 it keeps a high output until another number is pressed.

Figure 24: DTMF output when pressing number 2.



The third part of the system is microcontroller, which will take the DTMF output (binary encoded) and using a suitable code configured on arduino a process of checking the controller’s input coming from decoder (to decide the resulting action) will take place.

An important step will be done by the microcontroller before executing the user’s order which is “authentication”. Authentication is very necessary in our system to prevent unauthorized person from accessing the system and giving orders. This was done by giving the user a unique password to give him the right to access the system, when the user wants to open the door, he has to enter the correct password, the arduino will check it and then orders will be taken from the user.

After the authentication part is done, the next step for the microcontroller is to check the order (opening or closing), to open the door number 3 for example has to be entered by user, to close it number 4 is the number to be entered. In each case a certain sequence of pulses will be sent from arduino to the motor drive to determine the direction of rotation of motor.

In the following pulses in each case are shown. In figures 25,26,27 it is shown what sequence of pulses is produced to open the door, in figure28,29,30 the sequence required to close the door is shown.

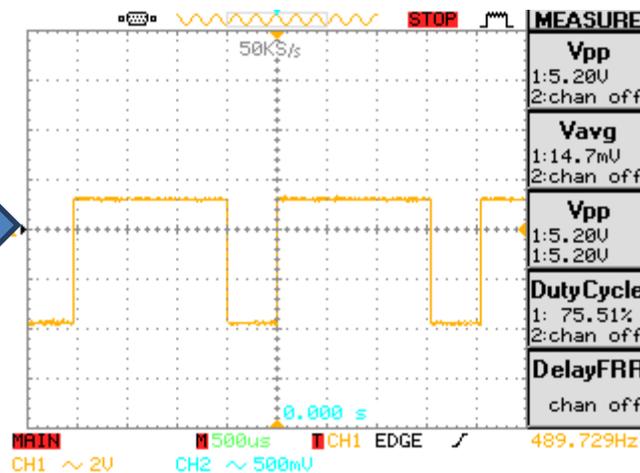
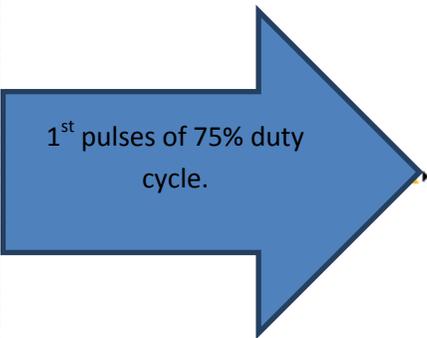
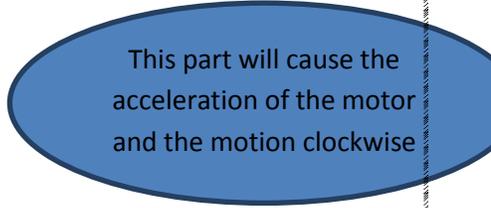
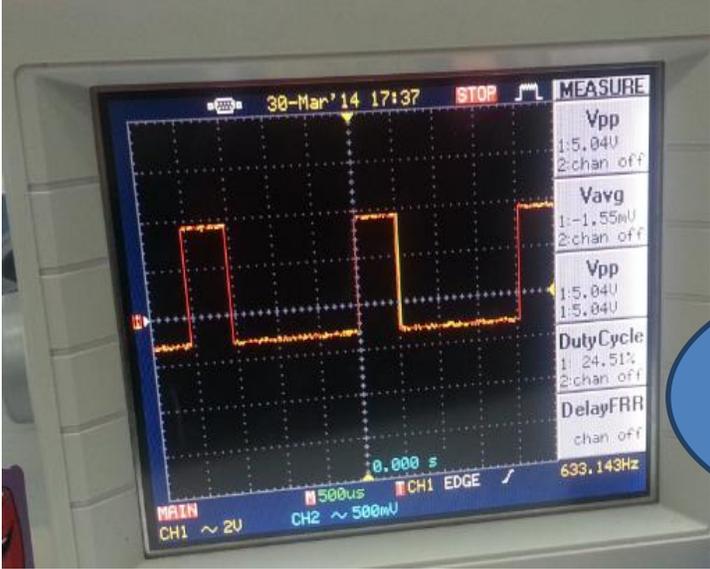


Figure 25: 1st pulses of 75% duty cycle.



Then pulses of 25% duty cycle.



This part will cause the deceleration of the motor in the last few seconds of opening process.

Figure 26: pulses of 25% duty cycle.

Then pulses of 50% duty cycle.

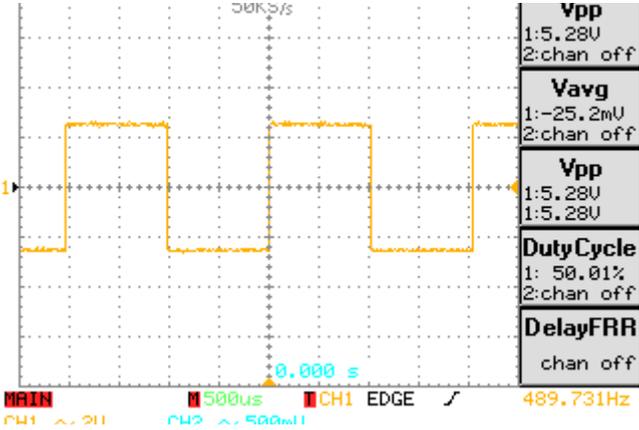
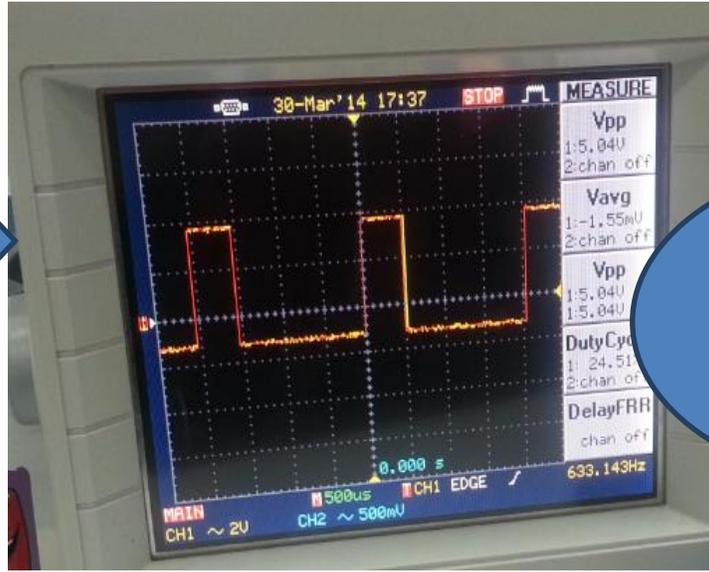


Figure 27: pulses of 50% duty cycle.

As for the closing option the sequence that will be produced is shown in figures 28,29,30.

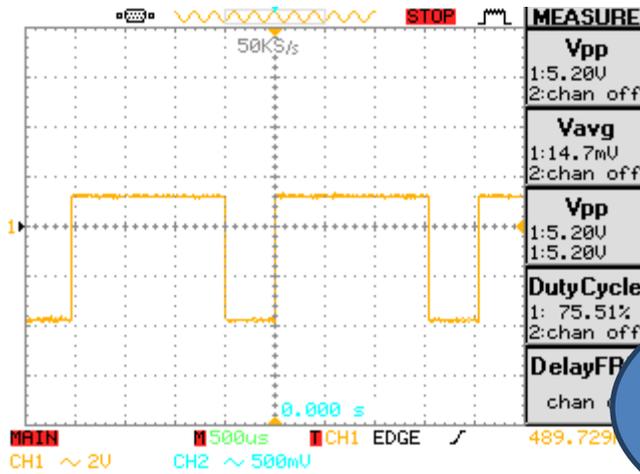
1st pulses of 25% duty cycle.



This part will cause the acceleration of the motor and the motion counterclockwise.

Figure 28: pulses of 25% duty cycle.

Then pulses of 75% duty cycle.



This part will cause the deceleration of the motor in the last few seconds of closing process.

Figure 29: pulses of 75% duty cycle.

Then pulses of 50% duty cycle.

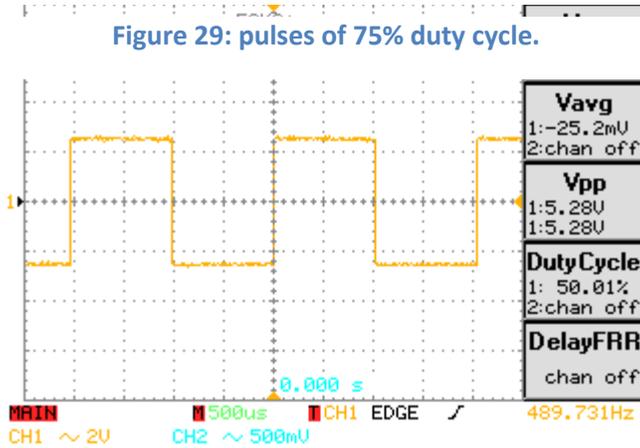


Figure 30: pulses of 50% duty cycle

After producing one of those sequences, the sequence will be directed to the motor drive.

The fourth part of the system is a motor drive:

In our system we need to use MOSFETs configured as high-side and low-side switches. Since we have a bridge circuits, in full-bridge circuits we have 2 high-side MOSFETs and 2 low-side MOSFETs. In such situations, there is a need to use high-side drive circuitry alongside low-side drive circuitry. The most common way of driving MOSFETs in such cases is to use high-low side MOSFET drivers. The most popular such driver chip is the IR2110.

The circuit required for the system is shown in figure 31.

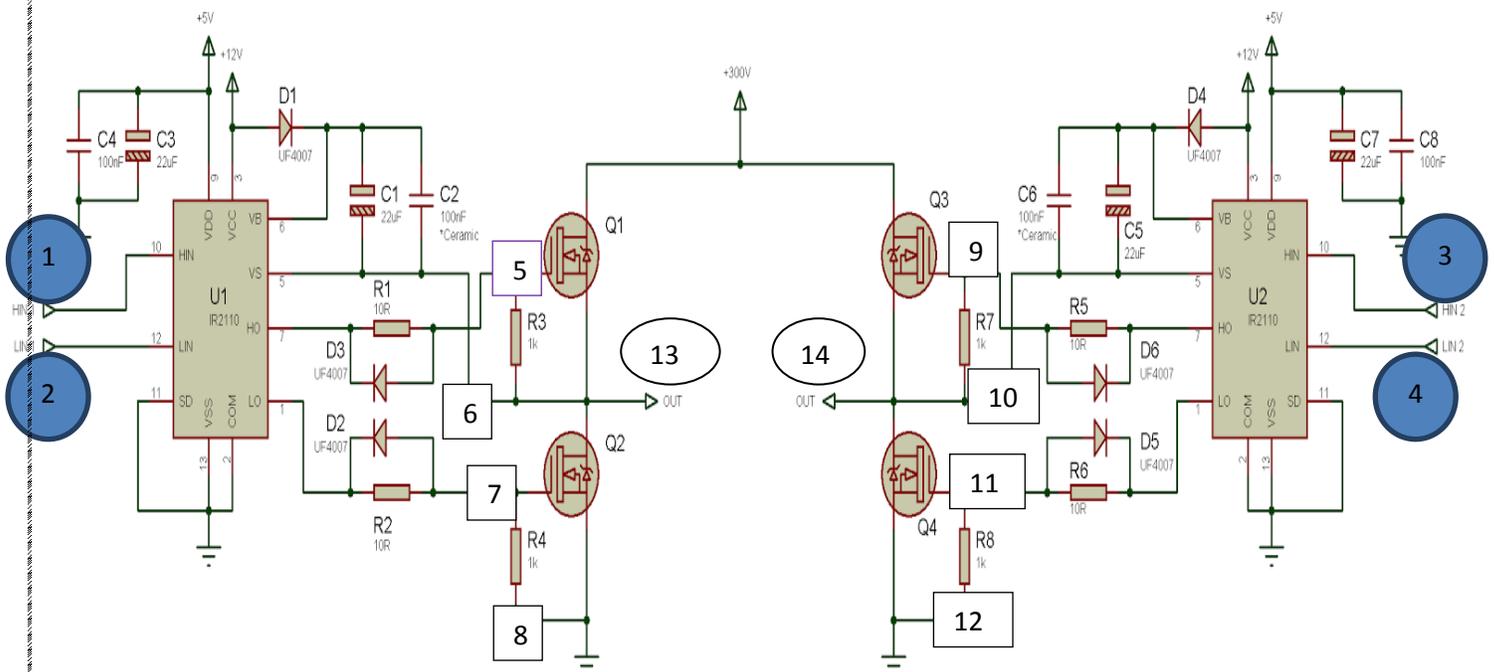


Figure 31: IR2110 circuit for high-voltage full-bridge drive with independent switch control.

In this circuit the sequence of pulses that was mentioned earlier (produced by arduino) will enter the circuit from pins labeled as 1,2,3,4 of IR2110 in figure 31. At points 1 and 2 the pulses will be entered to point 1 directly as for point 2 pulses will be inverted first then will enter the IR2110. The same for points 3,4, pulses will be entered to point 4 directly as for point 3 pulses will be inverted first then will enter the IR2110. Notice that for the bridge to work input in point 1,4 must be the same, input in points 2,3 must be the same .

The purpose of using IR2110 is to produce pulses with more than 10 volt magnitude to be able to run the IGBTs (making up the H Bridge) as switches, and also to isolate low side and high side. The output of IR2110 will be presented between points 5 and 6, 7 and 8, 9 and 10, 11 and 12.

In figure 32 we see the output of IR2110 between 7 and 8. It is shown how IR2110 gave pulses with more than 10 Vp-p.

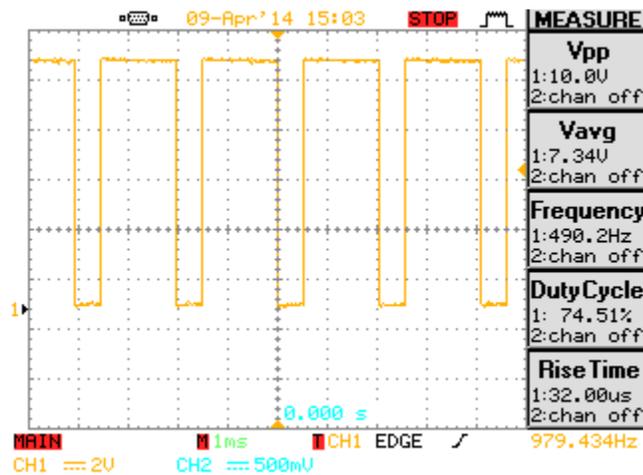


Figure 32: output of IR2110 between 7 and 8

In figure 33 we see the output of IR2110 between 11 and 12. It is shown how IR2110 gave pulses with more than 10 Vp-p and 25% duty cycle.

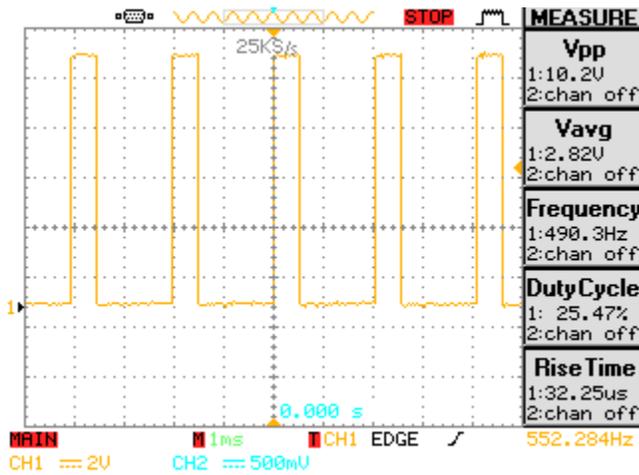


Figure 33: output of IR2110 between 11 and 12

Both signals are shown in figure 34.

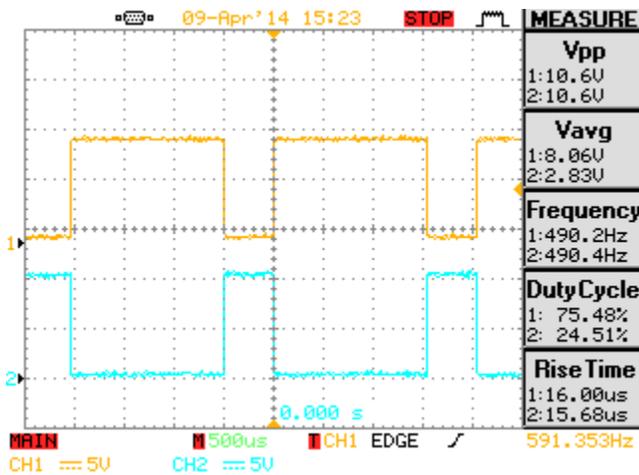


Figure 34: output of IR2110.

The output of the IGBT (i.e. at point 13 on figure 31) is shown in figure 35.

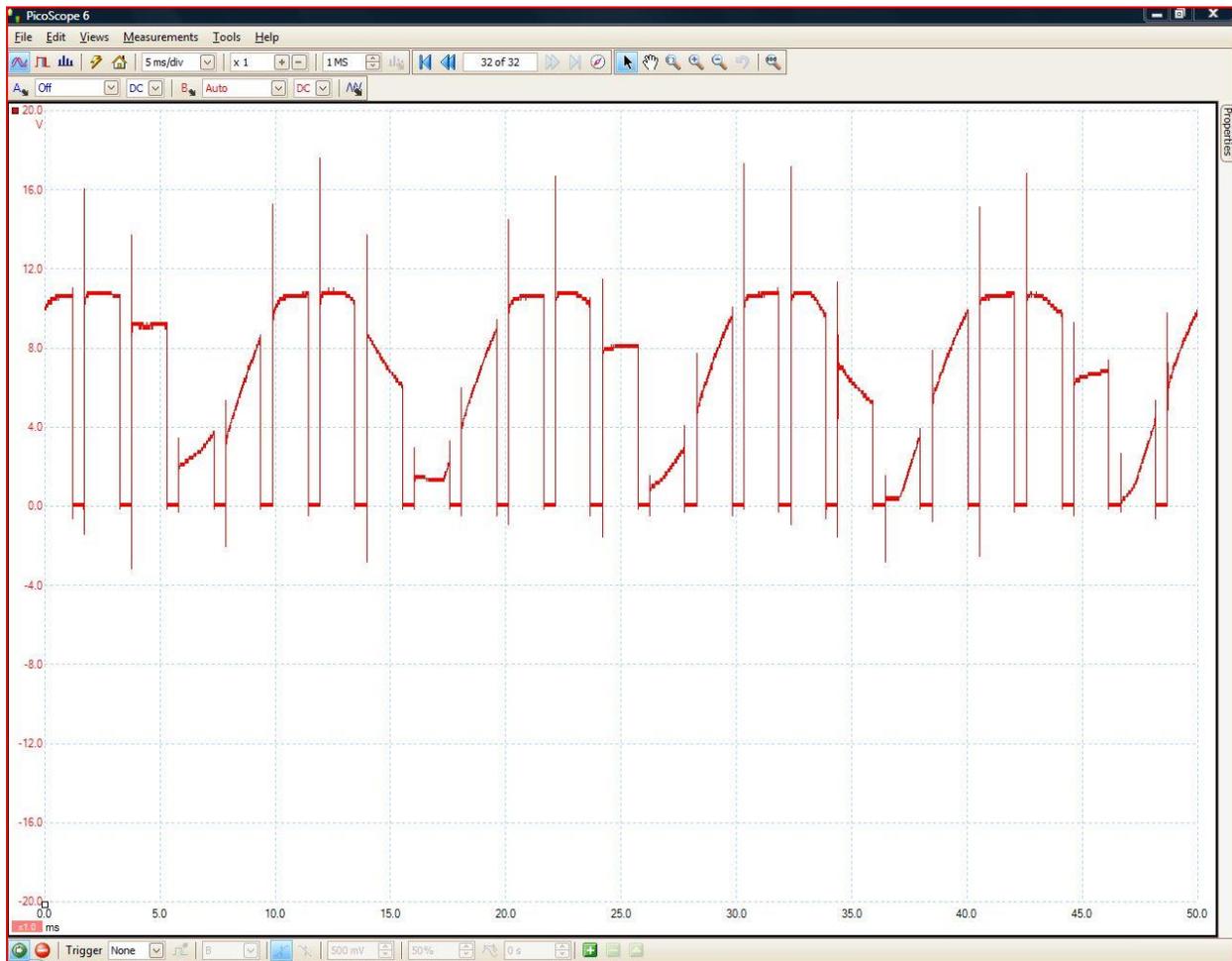


Figure 35: results at the output of the MOSFET.

The fifth part of our system is the motor that is connected between points 13, 14 on figure 30. This motor is connected to the door (the model of the door), this shown in figure 36.



Figure 36: picture of motor connected to the door.

So this door which we designed to experience our system is going to be opened and closed by entering number 3 or 4 on the user's mobile phone wherever he is if he has the correct password.

Chapter 6: conclusions and recommendations.

In conclusion we were able to build a system which is reliable and capable of controlling a garage door remotely. A DTMF decoding circuit was built to convert tones into binary output, and a reliable arduino code that takes instructions from user phone and turn them into sequence of pulses that is able to run a motor in the desired direction was built. The authentication part was taken in consideration to prevent anyone other than the owner to control the door, this is done by giving the user a password that gives him the right to access the system, then a motor drive circuit represents a dominant part to enable H-Bridge to work as desired to run a DC motor that is connected with the door that is required to be controlled.

What we learned in this project had a great value. First we were able to write a complete arduino code works as desired; this was a new thing we learned. Another thing is controlling a motor (speed and direction) and building the required circuit to achieve this purpose. Also we were able to build a model of a gate to examine our system and to make sure that it is working as desired.

On the other hand we have a future work plan to improve our system, first by using a sensor to check the status of the door before applying the user's order, and then by making the system send a message to the user to confirm the success of the process, this will require the use of a GSM module that will take instructions from the microcontroller to send messages to the user mobile, another thing we plan to do is to give the user the option to control the system either by entering the password then the controlling number or by sending a message to the system to give the required order.

Chapter 7: Feasibility Study

Assessment of the current market:

Regarding to the Palestinian market we can note that there is no similar product can be used to work as the product which we are designing.

Availability of the items and the parts at the Palestinian market:

The simplicity of the product (Distance Controlling Garage Door (DCGD)) provides significant advantages of the availability of the items and parts that used at the product and the following table shows the items and the suppliers.

Item No.	Description	Suppliers/ Region
1	DTMF cct	Electro tech/ Nablus
2	DC motor	Electro tech/ Nablus
3	Transformer	Electro tech/ Nablus
4	Arduino	Electro tech/ Nablus
5	IGBT driver (IR2110)	Ordered online from USA
6	IGBT Diode	Electro tech/ Nablus

Table 5: The items used at the product and suppliers with the regions.

The previous table obviously shows the availability of most of the items at Palestinian market in different regions which make it very easy to get these

items from the suppliers in ideal time to facilitate the processes of designing the product.

✓ Strengths and Weaknesses of the product (DCGD):

Strengths:

- Availability of most of the items at the Palestinian market.
- Simplicity of the processes in designing the product.
- New product idea at Palestinian market which provide a large market share.
- Ease of programming the Arduino.
- Ease of dealing with Matlab/Simulink.

Weaknesses:

- ✓ The cost of the product must be very low to reduce the possibility of exporting the same product from other countries especially from China.
- ✓ It might be easier if the entire item found in Palestine.

Cost Analysis:

Item No.	Description	Quantity	Price/Item	Cost	Availability
1	DTMF cct	1	23	23 NIS	Yes
2	DC PM motor	1	1	50 NIS	Yes
3	Motor drive circuit	1	-	38 NIS	Yes
4	Arduino kit	1	170	170 NIS	Yes
5	IGBT driver (IR2110)	2	16	32 NIS	Yes
6	IGBT\Diode	4	20	80 NIS	Yes

Table 6: Description and cost of each item in the project.

According to the previous table the estimated total cost is which is almost 120\$ reasonable and make the product highly competitive with any possibility of similar products.

References:

- [1] "SENIORD,"[Online].Available: <http://seniord.ece.iastate.edu/may0613/project%20plan.doc>.
- [2] "DTMF SIGNAL ,"[Online]. Available: <http://edutwin.com/t-DTMF-Signal-Controlled-House-Monotoring-System-HMS-full-report>. [Accessed 2 4 2014] .
- [3] "CircuitsGallery," [Online].
- [4] "ARDUINO," [Online].
- [5] V.Sieben, "A High Power H-Bridge,"2003.
- [6] "BLOG," [Online]. Available: www.blog.oscarlang.net. [Accessed 20 9 2013].
- [7] "datasheet," [Online]. Available:www.alldatasheet.com. [Accessed 20 2 2014].
- [8] I.Rectifier , IR2110(-1-2)(s)pbf/IR2112(-1-2)(s)pbf.
- [9] "ELECTRIC MOTORS," [Online].Available : <http://www.ohioelectricmotors.com/> . [Accessed 10 10 2013]
- [10] "[Online]. Available : <http://www.eecs.ucf.edu/>[Accessed 10 10 2013].

Appendices:

Appendix A

Arduino code:

```
/* Defining DTMF pins */  
  
int q1_pin = 2;  
  
int q2_pin = 3;  
  
int q3_pin = 4;  
  
int q4_pin = 5;  
  
int std_pin = 7;  
  
int f1=1;  
  
int f2=1;  
  
int f3=1;  
  
int f4=1;  
  
int start=0;  
  
int index=0;  
  
char pass[2];  
  
unsigned long time1;  
  
unsigned long time2;  
  
  
/* Defining IR pins */  
  
int ir11 = 10;  
  
int ir22 = 11;
```

```
/* Global variables*/  
char DTMF_code;  
  
void setup()  
{  
  /* DTMF pins as an input */  
  pinMode(std_pin, INPUT);  
  pinMode(q1_pin, INPUT);  
  pinMode(q2_pin, INPUT);  
  pinMode(q3_pin, INPUT);  
  pinMode(q4_pin, INPUT);  
  
  /* IR pins as an output */  
  pinMode(ir11,OUTPUT);  
  pinMode(ir22,OUTPUT);  
  
  pinMode(13,OUTPUT);  
  time1=millis();  
}  
void loop()  
{  
  int std = digitalRead(std_pin);  
  time2=millis();
```

```
if((time2-time1)>3000 && start==0){
```

```
    pass[0]='9999';
```

```
    pass[1]='9999';
```

```
    index=0;
```

```
    start=1;
```

```
    time1=millis();
```

```
}
```

```
if((time2-time1)>5000){
```

```
    f1=1;
```

```
    pass[0]='9999';
```

```
    pass[1]='9999';
```

```
    index=0;
```

```
    time1=millis();
```

```
}
```

```
if (std==LOW)
```

```
{
```

```
    DTMF_code=checkForDTMFpins();
```

```
    if(index<2){
```

```
        pass[index]= DTMF_code;
```

```
        ++index;
```

```
    }
```

```
if(DTMF_code!='44'){
digitalWrite(13,HIGH);
delay(500);
digitalWrite(13,LOW);
}
if(f1==0){
if (DTMF_code=='3')
{
analogWrite(ir11,191.25);
delay(3000);
analogWrite(ir11,63.75);
delay(3000);
analogWrite(ir11,127.5);

analogWrite(ir22,191.25);

delay(3000);
analogWrite(ir22,63.75);
delay(3000);
analogWrite(ir22,127.5);
}
else if (DTMF_code=='4')
{
analogWrite(ir11,63.75);
```

```
    delay(3000);

    analogWrite(ir11,191.25);

    delay(3000);

    analogWrite(ir11,127.5);

    analogWrite(ir22,63.75);

    delay(3000);

    analogWrite(ir22,191.25);

    delay(3000);

    analogWrite(ir22,127.5);
  }
}

if(f1==1){
  if(pass[0]=='3'){
    if(pass[1]=='3'){

      f1=0;

      time1=millis();

      digitalWrite(13,LOW);

      delay(100);

      digitalWrite(13,HIGH);

      delay(500);
```

```
digitalWrite(13,LOW);  
  
}  
  
}  
  
}  
  
}  
  
}
```

```
char checkForDTMFpins (){
```

```
//delay(500);
```

```
if (digitalRead(q4_pin)==LOW && digitalRead(q3_pin)==HIGH && digitalRead(q2_pin)==LOW &&  
digitalRead(q1_pin)==HIGH)
```

```
return '0';
```

```
else if (digitalRead(q4_pin)==HIGH && digitalRead(q3_pin)==HIGH && digitalRead(q2_pin)==HIGH &&  
digitalRead(q1_pin)==LOW)
```

```
return '1';
```

```
else if (digitalRead(q4_pin)==HIGH && digitalRead(q3_pin)==HIGH && digitalRead(q2_pin)==LOW &&  
digitalRead(q1_pin)==HIGH)
```

```
return '2';

else if (digitalRead(q4_pin)==HIGH && digitalRead(q3_pin)==HIGH && digitalRead(q2_pin)==LOW &&
digitalRead(q1_pin)==LOW)

return '3';

else if (digitalRead(q4_pin)==HIGH && digitalRead(q3_pin)==LOW && digitalRead(q2_pin)==HIGH &&
digitalRead(q1_pin)==HIGH)

return '4';

else if (digitalRead(q4_pin)==HIGH && digitalRead(q3_pin)==LOW && digitalRead(q2_pin)==HIGH &&
digitalRead(q1_pin)==LOW)

return '5';

else if (digitalRead(q4_pin)==HIGH && digitalRead(q3_pin)==LOW && digitalRead(q2_pin)==LOW &&
digitalRead(q1_pin)==HIGH)

return '6';

else if (digitalRead(q4_pin)==LOW && digitalRead(q3_pin)==HIGH && digitalRead(q2_pin)==LOW &&
digitalRead(q1_pin)==LOW)

return '*';

else if (digitalRead(q4_pin)==LOW && digitalRead(q3_pin)==LOW && digitalRead(q2_pin)==HIGH &&
digitalRead(q1_pin)==HIGH)

return '#';

else

return '44';

}
```

