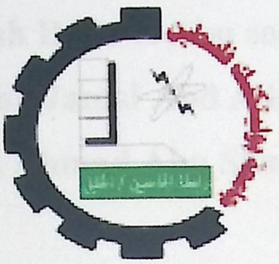
Palestine Polytechnic University



College of Engineering and Technology

Computer and Electrical Engineering Department

Graduation Project

Smart House

Project Team

Insherah Badawi Abu Soloom
Rana Jamal Abd Alla
Reem Ahmad Abu Sharar

Supervisor

Elayan Abu Garbyah

Hebron – Palestine
June_2009

Smart House

Project Team:

Insherah Badawi Abu soloom

Rana Jamal Abd Alla

Reem Ahmad Abu Sharar

Supervisors:

Eng: Elayan Abu Garbyah

Graduation Project Report

Submitted to the Department of Electrical and Computer Engineering in the College of Engineering and Technology

Palestine Polytechnic University

College of Engineering and Technology

Electrical and Computer Engineering Department

Hebron – Palestine

2009

جامعة بوليتكنك فلسطين

الخليل-فلسطين

كلية الهندسة و التكنولوجيا

دائرة الهندسة الكهربائية والحاسوب

اسم المشروع

Smart House

أسماء الطلبة

انشراح بدوي ابو سلعوم **رنا جمال عبدالله** **ريم احمد ابوشرار** •

بناءً على نظام كلية الهندسة والتكنولوجيا وإشراف ومتابعة المشرف المباشر على المشروع وموافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية والحاسوب، وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة أنظمة الحاسوب.

توقيع المشرف

.....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

.....

Abstract

The objective of this project is to design and implement warning and protection system against disasters that happen in urban houses such as fire and theft This system will react to fire occurrence it will light a red color in the place to indicate dangerous incident, Launch sound alarm, cut off electricity to the house, then power on a water pump working to extinguish the fire ,and finally the system will automatically dial a previously saved phone number giving

This system will react to theft crimes as follows. The System will light the place, launch the sound alarm, then the system will automatically dial a previously saved phone number giving a voice message telling that there is a theft crime in the house.

To increase the efficiency of this system, the sound alarm is designed to be password protected, so that no one is allowed to stop the sound alarm unless he enters the correct password. The user can monitor the status of all sensors and outputs via a password protected web interface which is reachable anywhere.

الملخص

هذا المشروع يعني بتصميم نظام حمايه وانذار ضد الكوارث, مثل مكافحة الحريق والسرقات حيث انه في حالة الحريق يعمل هذا النظام على اضاءة المكان باللون الاحمر للدلاله على وجود خطر في هذا المكان وكذلك يشتمل على انذار صوتي ويعمل ايضا على قطع الكهرباء من المنزل وتشغيل مضخة مياه التي تعمل على اخماد الحريق , وايضا يعمل على الاتصال اوتوماتيكيا بالرقم المخزن معطيا بذلك رساله صوتيه توضح انه يوجد هنالك حريق بالمنزل.

اما في حالة السرقات فان هذا النظام اضافه على انه يعطي انذار صوتيا ويضيء المكان فانه يعمل ايضا على الاتصال بالرقم المخزن معطيا رساله صوتيه تبين ان هنالك احد بالمنزل.

لزيادة فاعلية هذا النظام فانه عند حدوث حريق او سرقة فانه لا يسمح لغير مستخدم هذا النظام بايقاف الانذار الصوتي وذلك عن طريق ادخال كلمة المرور للنظام وكذلك يستطيع المستخدم مراقبة حالة كل من المجسات وايضا المخرجات اينما تواجد وذلك عن طريق صفحة ويب يتم الدخول اليها ايضا عن طريق كلمة مرور اخرى.

Dedication

To our beloved Parents.

Acknowledgment

In the name of Allah, Most Gracious, Most Merciful

All praise and glory to Almighty Allah who gave us courage and patience to carry out this work. Peace and blessing of Allah be upon last Prophet Muhammad (Peace Be upon Him).

To our supervisors, Mr. Elayan Abu Grbya for their insightful inputs, valuable discussions, & the assistance they have provided throughout the lifetime of this project.

To Mr.Majdi Zalloum & wadah sultan for generous support.

To Eng.Sami salameen & Mr Hassan abu rajab for the efforts they have paid .

To Mohammad Abu Sharar, Aya , Fidaa, Ameera, & for their continuous encouragement, and priceless support.

Table of Contents

Chapter Three: Software Requirements Specification

Chapter Four: System Design

Chapter Five : Software System Design

Chapter Six : System Testing

Chapter Seven : Conclusion and Future Work

List of Tables

Table	Page

list of Figures

Chapter One

Introduction

1.1 Overview

1.2 Project Importance

1.3 Project Objective

1.4 Literature Review

1.5 Time Plane

1.6 Cost Estimated and Budget Breakdown

1.7 Technology Risk

1.8 The Structure of the Report

1.9 Summary

Chapter One
Introduction

This chapter gives a general overview of the Smart House project and the project importance, also this chapter presents some previous projects that work on similar systems, and finally the estimated cost and the time plan for the project are illustrated briefly.

1.1 Overview

For many years ago, the engineers and the security systems developer's aims to develop the most secure and safety system that couldn't be hacked, and that's lately in twentieth century a lot of security systems and security idea appears in the application area.

Smart Home is a name we use for a Home Security System with digital features and sensors for senses the room smoke and motion .

This project aims to secure the hall house either the house owner was at home or out, and prevent or decrease the fire damages, humanity losses and thefts accidents, that we looking for a safe society.

Any way this security system will use sound alert (buzz alarm) when detect any of senses by sensors we used, also the system will send alarm using Phone dealing the owner mobile or other numbers decided by owner stored at phone memory acting with system, we also give the chance for owner to see the status of the input and outputs any where he /she exists by designated web page.

Smart home which aim for full security system is not completely a new idea, in fact it was initiated lately in twentieth century, while every project having same description trying to add some other function or features to improve the system, and that we aims to add internet helping the user to be sure that be enabled all the inputs sensors and output before leaving the house.

Finally, Smart Home is a good idea for interact between technology and people, for there safety and comfort ability, to be sure that there children, there money and there entire home wouldn't be at risk, not any more.

1.2 Project Importance

The smart house project aims to built a safety house without any dangerous to threaten this house, either the owners inside or outside the house. For example calling the owner when they are outside the house if any dangerous occurs.

1.3 Smart Home Objectives

The project has the following objectives:

- Encourage people to feed the value of technology by letting those uses such system.
- Try to eliminate the fire accidents, injuries and casualties.

- Create reliable and sensitive system in order to get the right action at the right moment.

- Helping the people to feel safe while they are outside their home.

- Building Safety Interfaces for the system by using LCD (A liquid crystal display) screen to allow specific user to use this system.

1.4 Literature Review

We can categorize the Literature Survey into two main groups:

1.4.1 PPU Projects

[A] Project Title:-"Micro web Server for Controlling and Monitoring Applications"

This project was introduced and approved in the 2002/2003 academic year in our university. It used the IPu 8930 micro_web server and the java applets software to control some simple home applications.

Project Description:

The project is considered as a home automation system, in which micro web server controls and manages many 'dumb' non_computarized devices. It is a system which was designed to control various devices.

These are:

- Lightening systems.
- Temperature control.
- Alarm system.
- Fan controlling.

[B] Project Title: "نظام إنذار وإرشاد ذكي داخل المؤسسات الكبرى"

This project was introduced and approved in 2006/2007 academic year in our university. It used the PIC microcontroller 18F4520 and C Programming language software to control this intelligent security system.

Project Description:

The aim of the project was to designing an intelligent security guidance system that can be used in large building and constructions. This system behaves automatically to worn and guide workers in such large constructions through RED/GREEN lighting lanes that are activated by a controlling system. The controlling system is triggered by different types of fire, smoke and other disastrous detector.

[C] Project Title:" Remote Control Monitoring System"

This project was introduced and approved in the 2001/2002 academic year in our university. It is used the smart modem 2400 as external modem and the AT commands software to tell the modem what to do.

Project Description:

This project is about monitoring and controlling system using telephone line. The project is stand alone intermediate system which contains modem interface with microcontroller system connected through telephone line with remote modem on PC.

1.4.2 Other Projects:-

Through searching the internet, we had seen various projects and suggestion tacking the idea of our project, some of them are tightly related to it, such as:

[A] Project Title: "Smart Home"

This project was introduced and approved in the 2003/2004 academic year in senior university; it is used PIC 016f874 and assembly language to control some simple home application.

Project Description

This project have two goals, the first goal is alert the home owner to any problems that are encountered, even when the house is unoccupied, the second goal is the system continuously monitors of fire, flooding, carbon, monoxide, power outgas, and temperature.

[B] Project Title:" Home Security System"

This project was introduced and approved in the 1998/1999 academic year in the Cornell University. The project the project idea to design home security system with three zones.

Project Description:

The aim of the project was to design the alarm security system include the following features

- Sounding of alarm upon violation of a zone.
- Display to LCD which zones violated .
- Steady green LED = "Ready..." state.
- Steady red LED = "Armed!!!" state .
- Flashing green LED = warning—some zones currently in fault .

1.5 Time Plane

The time plan views the stages in designing and building the system components. The section includes two time schedules; the first one shows what is done in the first semester, while the second shows the task scheduling for the second semester.

Table 1-1 Time Planning(first semester)

Weeks / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Project determination																
Data collection																
Literature review																
Design and analysis																
Documentation																

Table 1-2 Time Planning(second semester)

Weeks / Task	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Software design																
Hardware implementation																
Software testing																
Hardware testing																
Integrating system testing																
Documentation																

1.6 Cost Estimated and Budget Breakdown

This section lists the overall cost of the project, the cost includes the hardware cost, software cost, and the human budget.

1.6.1 Hardware Cost:

Includes the cost of the component that is used to implement the project. Table 1-3 shows these cost.

Table 1-3 hardware cost.

Component	Quantity	Price
Atmel Mega 32	1	14$
Visonic Speech Dialer	1	100$
XPort server	1	55$
Water Valve	1	10$
Resister	30	10$
Capacitor	10	10$
LEDS	5	2$
Smoke sensor	1	15$
Motion sensor	1	21$
Breadboard	1	6$
Function Generator(rent)	1	Free
LCD	1	8$

Keypad	1	12$
OpAmp Amplifier	6	10$
Transistor	3	8$
Lump	1	2$
wires	200	10$

1.6.2 Software Cost :

Includes the cost of software that is used to implement the project. Table 1-4 shows these costs.

Table 1-3 Software Cost.

Component	Price
Windows XP	free
Microsoft office 2003	2$
Code vision avr studio	free
Orcad Family Release 9.2	free
Visio	free
Total cost	2$

1.6.3 Human Resource Budget:

Include the budget of human operation. The cost is distributed on three engineers, each engineer costs $200 weekly.

1.7 Project risk management

1.7.1 technology Risks:

Some risks may happen due to software or hardware used in the system.

Hardware risk

- Sensors malfunctions.
- Telephone malfunctions.

Software risks

- Problems that might occur using the software itself.

1.7.1 People risks

- Member of team gets sick
- Member of the team got unavailable for any reason.

1.7.2 Risk Avoidance

- Regular maintenance and updating components of hardware and software.
- Taking good care of the team members weel being during the project time.

Risk Management

- Preparing a backup version of the software developed.
- Having extra hardware components so when a problem occurs a new alternative is immediately replaced.
- People risks could be handled by the distribution of the work load.

1.8 The Structure of the Report

The documentation for this project is categorized into seven chapters, the report structure view the outline for the discussed subjects in each chapter. The outline of all chapters is summarized briefly as follows :

- Chapter One: Introduction

Demonstrate an overview about the systems, a literature review, estimated cost and time planning , risk management and risk avoidance, finally this chapter shows the structure of the project report.

- Chapter Two: Theoretical Background

Focuses of theories and materials that are related to the project. It mentions the laws behind the used hardware or software project components.

- Chapter Three: Design Concepts

Describes the detailed system objectives, design options, a general block diagrams and the system interaction with the surrounding environment.

- Chapter Four: Hardware System Design

Describes the detailed subsystems design, overall system design (schematic diagrams) and the user system interfacing circuits.

- Chapter Five: Software System Design

Describes the detailed system algorithms, flowcharts pseudo code for the overall procedures in the project

- Chapter Six: System Implementation and Testing

Describes the actual project implementation, prototypes. It shows the individual component testing, subsystem testing, software testing and the integrated system testing.

- Chapter Seven: Conclusions and Future Work

Describes some suggestions for future enhancement, explains the conclusions and the problems that faced the team.

1.9 Summary

This chapter introduced an overview about the project, and presented the importance of the project, and discussed some previous projects which deal with similar projects ,then presented the estimated cost, time plane, finally the structure of the project report.

Chapter Two

Theoretical Background

2.1 Microcontrollers

2.2 Serial Interface

2.3 XPort Embedded Device Server

2.4 Visonic 2-Channell speech dialer DL-125CA.

2.5 sensors

2.6 16×2 LCD

2.7 4×4 Keypad

2.8 Piezoelectric Buzzer

2.9 Pumps

2.10 summary

Chapter two

Theoretical Background

We need intermediate system that is going to be connected to the application through a certain interface circuit. It must work as a computer containing, R/W (Read/Write) memory unit in order to store data, I/O ports (Input/Output) ports to transmit and receive data or control signals, all of these needful features which are found in the microcontroller chip. It is some times called single-chip microcontroller can work in the control system. Also contains XPort embedded device severe and Visonic 2-chaneel speech dialer DL-125C.

This chapter will discuss theoretical information about:

1. Microcontroller.
2. Serial interface.
3. XPort Embedded Device Server.
4. Visonic 2-chaneel speech dialer DL-125C.
5. Sensors.
6. 16 ×2 LCD.
7. 4 ×4 keypad.
8. Piezoelectric Buzzer.
9. Pumps.

2.1 Microcontrollers

Microcontroller is a functional computer system on single chip, which contains a microprocessor core, some memory (RAM/ROM) to store data, parallel and serial input/output ports to transmit and receive data, timer, and analog to digital converter. Microcontroller is used to control product and device automatically, such as remote control and automobile engine control system. Because of the low cost and easy of integration within an application they are used whenever possible to reduce the chip count of piece [1].

2.1.1 ATMEL AVR microcontroller:

The ATMEL AVR is a single chip and modified Harvard architecture. The Harvard architecture is means that the data and program are stored in separate memory space. The AVR was the one of the microcontroller family which is used on chip flash memory for program storage as EEPROM, and low power and high performance device. AVR microcontroller can handle demanding 8 and 16 bit applications. The AVR architecture ensures easy application development and fast code execution combine with the lowest possible power consumption, and I/O structure in the AVR is limit the need for external component and reduces developmental cost, variety of timers, UARTS, analog comparators, and watch dog timer .AVR microcontroller may be programmed using assembly or a higher level languages such as c and java language [2].

2.1.2 Atmelmega32

ATmelmega32 is low power CMOS 8bit microcontroller based on RISC architecture, executing powerful instructions in a single clock cycle. It is also contains 32 general purpose working registers and all these registers are directly connected to the ALU(Arithmetic Logic Unit),and allowing two independent registers to be accessed in one single instruction executed in one clock. ATmega32 AVR is supported with full suite of program and system development, such as C compiler and macro assemblers[3].

Figure 2.1 Atmel Mega32

Features:

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  – 131 Powerful Instructions – Most Single-clock Cycle Execution
  – 32 x 8 General Purpose Working Registers.

- Nonvolatile Program and Data Memories
  – 32K Bytes of In-System Self-Programmable Flash

– 1024 Bytes EEPROM Endurance: 100,000 Write/Erase Cycles
– 2K Byte Internal SRAM

- JTAG (IEEE std. 1149.1 Compliant) Interface

- Peripheral Features
  – Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  – One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  – Real Time Counter with Separate Oscillator
  – 8-channel, 10-bit ADC
  – Byte-oriented Two-wire Serial Interface
  – Programmable Serial USART
  – Master/Slave SPI Serial Interface

- Special Microcontroller Features
  – Power-on Reset and Programmable Brown-out Detection
  – Internal Calibrated RC Oscillator
  – External and Internal Interrupt Sources
  – Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

- I/O and Packages
  – 32 Programmable I/O Lines
  – 40-pin PDIP, 44-lead TQFP, and 44-pad MLF

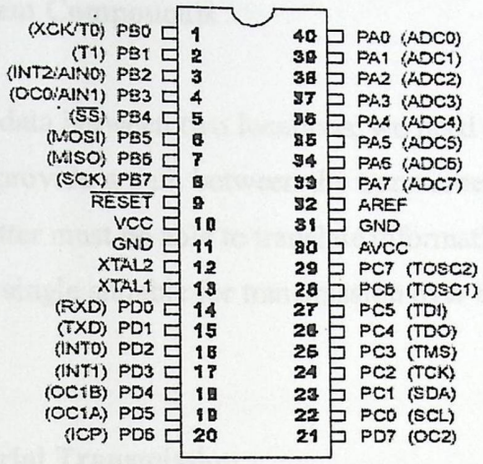


Figure 2.2 Pins of Atmel Mega32

All of these feature make Atmel mega 32 ideal in serving our needs for the project in single chip. Atmel mega32 in our project work as computer and intermediate system, the Atmel mega32 receives data such as accept the signal from the smoke sensor when there is afire to activate the buzzer and phone dialer as example.

2.2 serial interface

It will cover the following topics:

1- Transmission system component.
2- Parallel versus serial transmission.
3- USART.
4- RS232 for serial interface.

2.2.1 Transmission System Components

When we transmit data between two locations, we need a transmitter, receiver and transmission medium that provide a path between the transmitter and receiver. In addition to transmit signals, transmitter must be able to translate information from a form created by humans or machines into a single suitable for transmission over the transitive medium.

2.2.2 Parallel Versus Serial Transmission

when the data is represented as a group of bits, and transmitted all bits in the same time, this type of transmission is called parallel transmission, because parallel transmission deals with 1 byte (8 bits) of data need 8 line to transmit data and other line to transmit control signal. So this transmission method becomes expensive, when need to transmit data for long distance.

So another method was developed, this method is called serial transmission.
In serial transmission one bit transmit at a time, in this method the number of lines is reduced to single line, another advantage of this method is to transmit data through single line for long distance which is more flexible than parallel transmission.

2.2.3 USART

The USART devices are used to communicate with devices outside the microcontroller, with the USART in asynchronous mode.

2.2.4 RS 232 Standard For Serial Interface

RS232 standard is one of the oldest physical communication standard in computer world the standard defined as low cost serial communication in robust way where bits are sent sequentially on a line, it was originally define for connecting devices such as computer, terminal and printer, this equipment is connected through their serial port[4].

The original serial port limits the maximum transfer speed to 20 kbps, to overcome these limitations, the RS232-E standard allows much higher communication speeds than its predecessor. We can define the communication method in RS232 stand as asynchronous serial communication method [4].

In this project the sensor is connect to the Atmel chip, so the results which is received sent out to XPort .

2.3 XPort Embedded Device Server

XPort is a complete solution (hardware and software) for web enabling your edge device with serial interface, it can also be defined and it is embedded device server which removes the complexity of designing network connectivity into a product by incorporating all of the required hardware and software inside a single embedded Ethernet solution. Smaller than your thumb. The XPort offers the highest level of integration available in device server, within compact RJ45 package, highest speed serial port, and three programmable I/O pins[5].

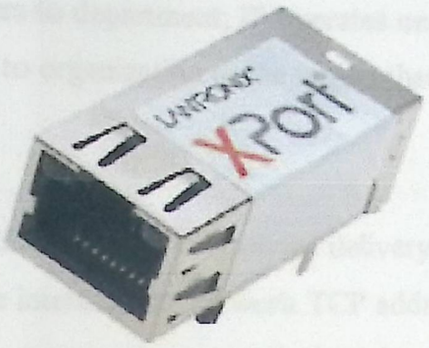


Figure 2.3 Pins of XPort Embedded Device Server

The XPort is a self contained TCP/IP server with the ability to both store dynamic web pages. The device is configurable through a web interface as well as through a serial connection compatible with our USART. Currently, the XPort is used as a transparent serial connection between the host microcontroller and a server connected to the XPort over the internet [5].

2.3.1 TCP/IP Protocol

TCP/IP (Transmission Control Protocol/Internal protocol) is the basic communication language or protocol of the internet TCP/IP protocol is composed of two layers:

- IP Protocol: - is responsible for moving packet of data from node to node. IP forward packets based on a four byte destination address (The IP number).the internet assign ranges of numbers to different organization, the organization assign

groups of their numbers to department; IP operates on gateway machines that move data from department to organization to the region then around the world[6].

- TCP: - is responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network TCP adds support detect errors or lost data and to trigger retransmission until that data is correctly and completely received[6].

2.4 Visonic 2-chaneel speech dialer DL- 125C

The DL-125C is a programmable speech dialer with two alarm inputs. It is designed for verbal reporting of two separate events, one event per input. Each event can be reported to 4 different remote telephones, or both events can be reported to the same 4 telephones. The telephone numbers of the called parties may be frequently reprogrammed by the user.

A communication session with the first / second group of telephones is initiated by triggering alarm inputs Z-1 / Z-2, or by pressing AL-1 / AL-2 on the front panel, respectively. The dialer performs certain functions in response to DTMF control commands received from remote telephones. The DL-125C is packaged 12 keys serve for entering data, and 4 are function keys. Programmed data is retained in an EEPROM Two models are available (12 V or 24 V versions upon request): DL-125C - allows the user to

stop the communication session by pressing the STOP pushbutton (provided that momentary alarm contacts are used).

Two different events can be reported to remote telephones. Stand-alone 2-input 24-hour alarm system, triggered directly by a smoke detectors or motion sensor. The overall length of the speech message that can be recorded is limited to 20 seconds. Within this limit, the message can be composed of two pre-recorded segments.

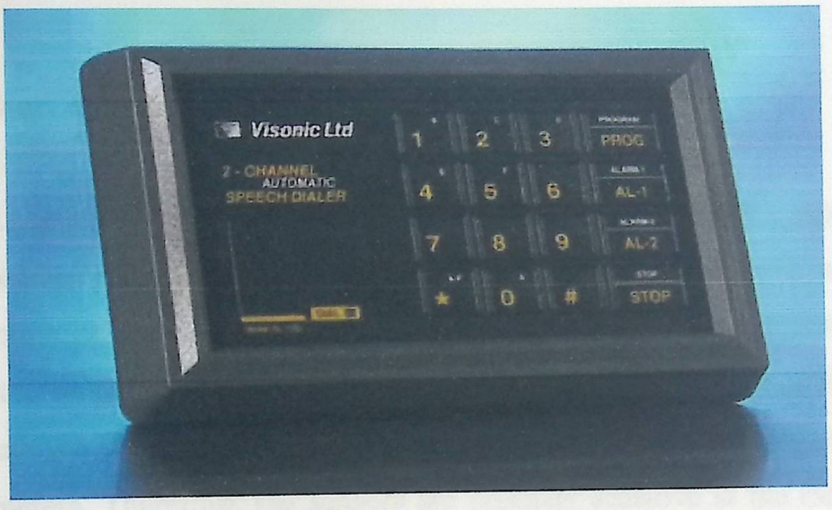


Figure 2.4 Visonic speech dialer DL- 125C

About this project, this system provides ability for the user to store four numbers and record a short massage about 20 second for both zone. When any of the sensors activated it call the desired numbers that the user specify, then play the prerecorded message which about the status of the house, for example in case of smoke sensor the massage will tell the user there is a fire in the house, then calling fire Brigade, otherwise in motion detector it will tell him there some one moving in the house.

2.5 Sensors:

Sensors can be defined as a device takes information about physical stimulus (such as temperature, smoke and motion) and turns it into a signal which can be measured.

The main sensor used in this project are :

1. Motion sensor.
2. Smoke Sensor.

2.5.1 IR motion detector

The IR motion sensor is used to sense feature on most security system is a passive system that detects infrared energy. These sensors are therefore known as PIR passive infrared) detectors or pyroelectric sensors. In order to make a sensor that can detect a human being, the sensor should be sensitive to the temperature of a human body. Humans, skin temperature are about 20 degree Celsius, radiate energy with a wavelength between 9 and 10 micrometers. Therefore, the sensors are typically sensitive in the range of 8 to 12 micrometer [10].

Figure 2.5 Motion Sensors

The sensors themselves are simple electronic components not unlike photo sensors. The motion sensors is sensitive to motion, but not to a person who is standing still. That is because the sensor is designed to detect a fairly rapid change in the mount of infrared energy in front of it. When a person walks by, the amount of infrared energy in the field of view is changes rapidly and is easily defected [9].

2.5.2 Smoke Detector

Smoke detector is a dvice that detects smoke and issues an alarm to alert nearly people that there is a potential fire. A household smoke detector will typically be mounted in a disk shaped plastic enclosure about 150mm in diameter and 25mm thick, but the shape can vary by manufacturer [11].

Figure 2.6 Smoke Sensors

Because smoke rises, most detectors are mounted on the ceiling or on a wall near the ceiling to avoid the nuisance of false alarms; most smoke detectors are amounted away

from kitchens. To increase the chance of waking seeping occupants, most homes have at least one smoke detector near any bedroom.

Smoke detectors are usually powered by one or more batteries but some can be connected directly to household wiring. Smoke detectors may operate alone, be interconnected to cause all detectors in an area to sound an alarm if one is triggered, smoke detector with flashing light are available for deaf or hearing impaired [11].

2.6 16×2 LCD

The LCD is used to operate as interface for the system to enable or disable the sensors by using a simple password to enter the system. And if any sensor goes wrong, the LCD will output error and with proper initial for sensor (T for temperature, M for motion, S for smoke).

The LCD display contains 2 lines by 16 characters and provides basic text wrapping so that your text look right on the display. The serial LCD allow you to move cursor to any location on the display screen with a single instruction, when you first turn on the LCD, the cursor is in the left most position on the top line, and when the LCD receive the first character, display that character at the current cursor position and then move cursor one position to the right [12].

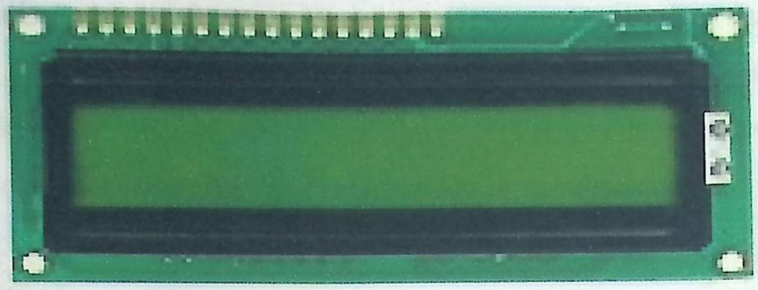


Figure 2.7 16×2 LCD

The LCD contains the following features:

1. Diplay all ASCII characters directly to the display.
2. Warps to the next line automatically for easy display of text string.
3. Work at 2400, 9600, and 19,200 band.
4. Move the cursor any where on the display with single command.
5. Clear the whole display with single command.
6. Module Dimensions: 66(W) x 26(H) x 12(T) mm.

2.7 4×4 Keypad

Keypad is the input device and is a standard matrix keypad. The keypad contains 16 key symmetrically arranged in four rows with four keys, each column and row of the keypad is connected to an I/O (INPUT/OUTPUT) pin[13].

The microcontroller continuously scans the keypad setting all row pin except one to high and reading all column pins. The keypad driver uses timer/counter over flow the

active keypad row is increased and wrapped around, when necessary. Key push is register and decodes via the pin change interrupts. The smaller keypad consider replacing the 4×4 keypad with a 4x3 keypad typically only numerical keys and one or two special keys are required. This approach saves one I/O pins [13].

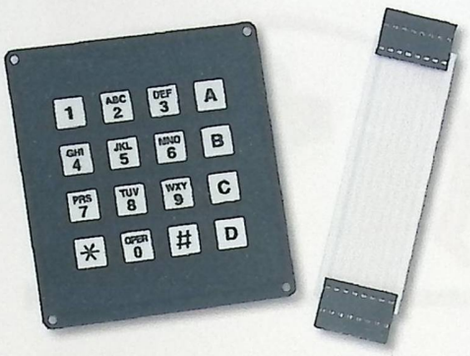


Figure 2.8 4×4 Keypad

Using keypad in this project is helping to create secure system by entering a simple password via this key. It is used to enable or disable the sensors. And if the user enter the correct password the buzzer and voice playback will stop. the user can now navigate throw our menu and make any changes in the setting.

2.8 Piezoelectric Buzzer

Piezoelectric used to get alarm sound when the smoke sensor is activate. Piezoelectric sound components produce clear penetrating tones, free of harmonics, the devices have high acoustics output and low power, requirements making them ideal for

micro controller application. The sound source of the piezoelectric buzzer is the diaphragm. Applying a DC voltage to the device causes mechanical distortion to the diaphragm and applying an AC voltage will move the diaphragm in a repeated bending motion, creating sound waves. The tone of the piezoelectric buzzer is directly proportional to the frequency of the AC signal applied [14].

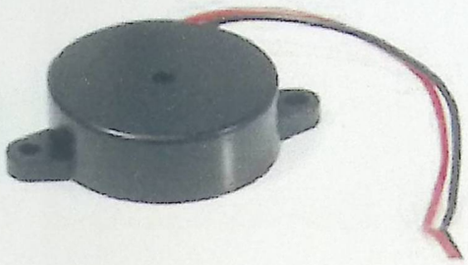


Figure 2.9 Piezoelectric Buzzer

2.9 Water Pumps

The fire pumps are the organs of large craters that water flows which are used in fire to putting out the fire. When there is a fire the smoke detector will activate which is cause to activating the pump by pumping water in place where the fire.

2.10 Summary

This chapter the theoretical background of the project main idea was demonstrated, and then this chapter described the hardware related to the project. The chapter talks about

the AVR Atmel mega32, sensors and it is type, the XPort server, explaining the serial interface and other component.

Chapter Three

Project Conceptual Design

3.1 Project Objective.

3.2 Design Options .

3.3 Project Design Block Diagrams.

3.4 General Block Diagram.

3.5 How Does System Work.

3.6 Summary.

Chapter Three
Project Conceptual Design

In this chapter, we are going to describe the project objectives, design concept, and the general block diagrams that explain this security system. It represents main components of the system, such as microcontroller units and sensors. Finally this chapter explains how the system works.

3.1 Project Objective

The project objective as follows:

- Design Smart House system that consist of multiple units (as show in figure 3-1) detection, transmitter, and receiver units.

- Connected the first unit which consists of sensors with the Atmel mega32 Microcontroller and building the required circuit

- Connect the Atmel mega32 microcontroller to the visonic phone dialer to make the dialing to the owner through it. Where they are both in transmitter unit.

3.2 Design Options

There are many issues that we must consider in order to build a functional, successful and reliable security system. These issues involve proper design, and choice of major components, there are may alternative options for choosing each component and this section talks about each available option.

3.2.1 Microcontroller

Microcontroller controls different needed functions for the entire project. We have two available options:

1. Atmel 89C51.
2. Atmel mega32.

- **Atmel 89C51**

Atmel 89C51 Microcontroller, 8-bit Microcontroller with 4K bytes flashes. And contains the following features [15]:

Features:

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles

- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

- **Atmel Mega32**

This project uses Atmel mega32. 8-bit Microcontroller with 32K Bytes in-System Programmable Flash, it contains the following features [3]:

Features

- High-performance, and Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
- Nonvolatile Program and Data Memories
- JTAG (IEEE std. 1149.1 Compliant) Interface
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down,
  - Standby and Extended Standby.

3.2.2 External Modem

The external modem used to store phone number, and automatically call that numbers. There are two suitable option are available

1. Hayes smart modem.
2. Visonic phone dialer.

- **Hayes Smart Modem**

Modem can perform many different functions, such as automatic dialing, negotiation of the method of modulation used to communicate with distance modem, error detection and correction, and support transmission of both asynchronous and synchronous data, and other functions. Every modem follows certain sets of code which tell the modem what to do. And when we used this modem we need use the voice recognition and the microphone to record the voice [17].

- **Visonic phone dialer**

This project uses that uses visonic phone dialer. It's key feature is in chapter two, and the option of choosing visonic phone dialer refers to that it has the facility to record a short 20 second voice massage without needed external voice recognition and microphone. It also can store four telephone numbers from the user choice [7].

3.2.3 Embedded Web Server.

The microcontroller can connect to the internet through embedded web server. There are two suitable options are available.

1. IPμ8930.
2. XPort embedded device server.

- **IPμ8930**

The IPμ8930 is a general purpose network controller and web server which make it easy to monitor control and communication with remote sensor. IPμ8930 can directly used for interfacing with analog sensor, which makes information can be used directly available through network, and directly is connected to 3-20 ma, 0-5 v. IPμ8930 can connect to the internet via rj-45 Ethernet connectors. In this project IPμ8930 not used, because number of available I/O (Input/Output) pin is not enough to connect all input and out put component [17].

- **XPort Embedded Web Server**

This project uses XPort embedded web server, the key features of XPort explains in chapter two, and the choice of choosing XPort embedded web server refers to [5] :

- XPort is a complete solution (hardware and software) for web that enable your edge device with serial interface.

- The XPort is a self contained TCP/IP server with the ability to both store dynamic web pages and act as a serial to Ethernet converter.
- The cost is not high.

3.2.4 Alarm System

Alarm system is needed to give special sound when one of the sensors goes wrong. We have two available suitable options

1. Alarm using 555.
2. Buzzer.

- **Using D555 Timer**

It is circuit built from: 555 timer and RC circuit edge by a simple speaker. This circuit is activated when Atmel mega 32 microcontroller command it to work according to certain condition. The circuit gives a good sound, but in order to achieve a relatively reliable high sound, we can connect it to the amplifier, there by give a much stronger sound.

- **Buzzer**

This project uses buzzer, the key features of buzzer is explained in chapter two. This buzzer when triggered by Atmel mega32 gives a special sound indicating the error occur in one of the sensor such as (motion sensor). The reasons for choosing this option are:

1. It is relatively very simple circuit with only two components.
2. Buzzer is much available in marketing.
3. it gives the needed sound .

3.3 Project Design Block Diagrams

This section shows the block diagrams of Smart Home system. It consists of several units as follows:

1. Detection unit, this unit consists of sensors (smoke, and motion sensor).
2. Transmitter unit, this unit consists of Atmel mega32 microcontroller; and visonic phone dialer.
3. Receiver unit, this unit consists of LCD, LED'S, buzz, water pump, and user account and owner phone.

3.4 General Block Diagram

This diagram illustrates all general components of the system.

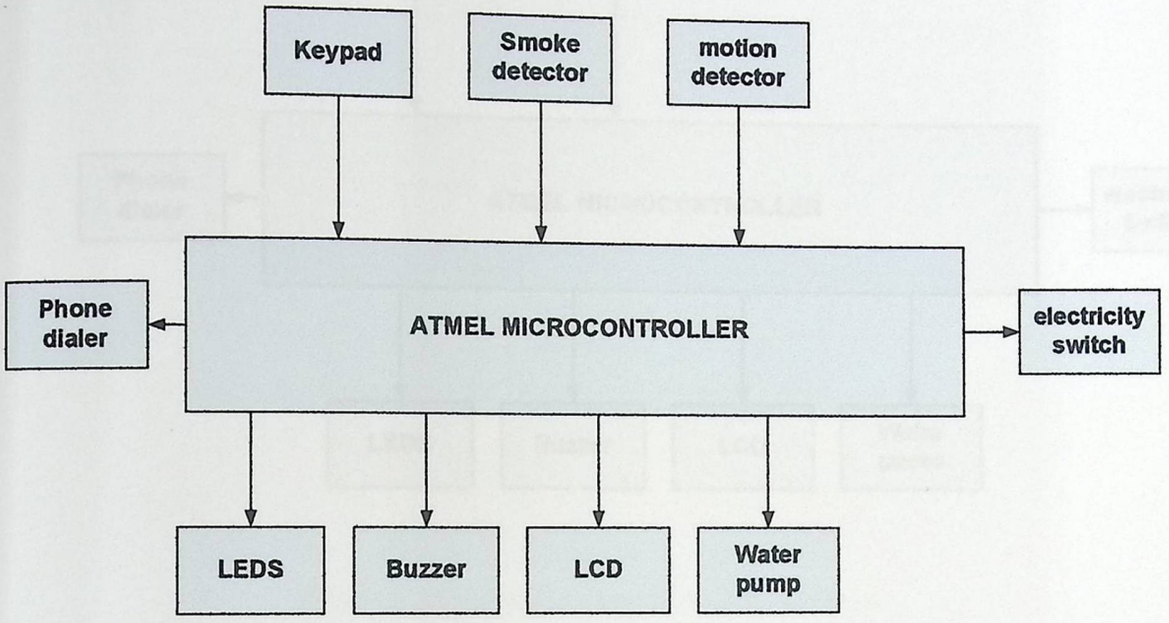


Figure 3.1 General Block Diagram.

3.4.1 Smoke Detector Block

This block represents smoke sensor. Which is connected to the Atmel mega32 microcontroller, when the smoke sensor is activated it sends signal to microcontroller, the microcontroller sends a signal to turn off the electricity then activate water pump, LED, buzzer alarm, and visonic phone dialer and the status of this detector is displayed on the LCD, and the enable or disable for the sensor done via keypad. And small application of using internet is trying to done through this project to check the status of the smoke

detector by xport server web page.

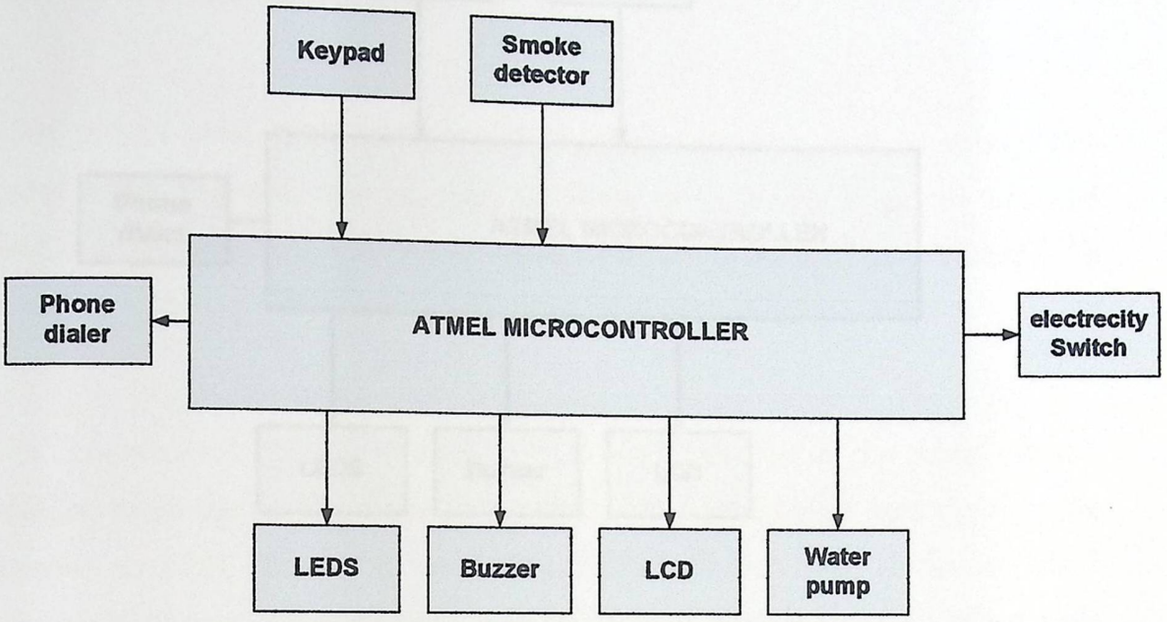


Figure 3-2 Block Diagram of Smoke Detector.

3.4.2 Motion Detector Block diagram

This block represents motion detector. Which Connected to the Atmel mega32 microcontroller, when the motion detector is activate and detect object send signal to the microcontroller. then the microcontroller send signals to activate buzzer alarm, LED and visonic phone dialer to dials the desired number and then telling them the prerecorded message. The LCD which is used to display which sensor is activated. And small application of using internet is trying to done through this project to check the status of the motion detector by xport server web page.

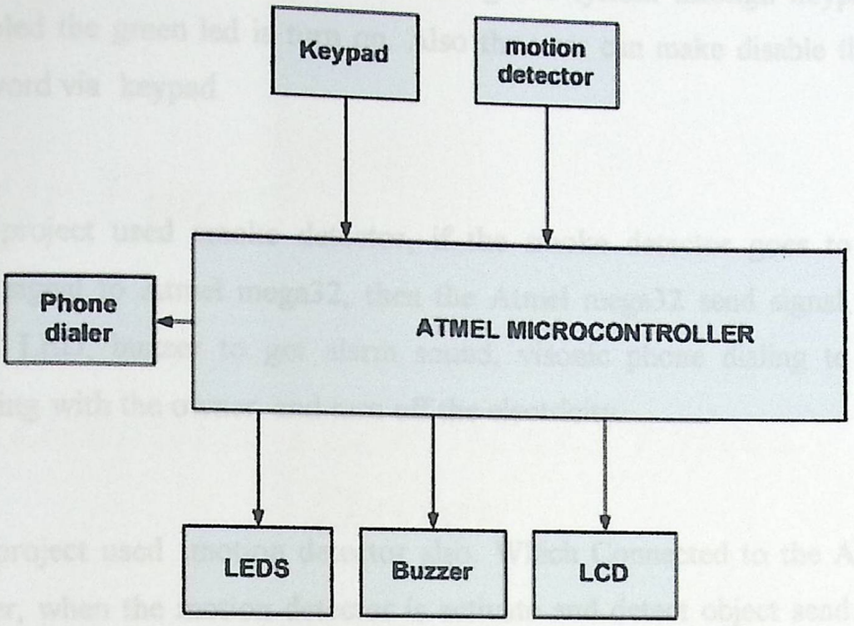


Figure 3-3: Block Diagram of motion detector

3.5 How Does System Work

Smart House security system as we explained before consist of 3 units (detection, transmitter, and receiver unit). detection unit consist of two different sensors (smoke, and motion sensor), Then the Atmel mega32 microcontroller from the transmitter unit collect information from the sensors , the receiver unit consist of 5 units (buzzer, LED, LCD, valve , visonic phone dialer).

ough keypad. After that the user can enabling the system through keypad, after the tem is enabled the green led is turn on. Also the user can make disable the system by tering password via keypad.

In this project used smoke detector, if the smoke detector goes to wrong, this detector send signal to Atmel mega32, then the Atmel mega32 send signal, to activated water pump, LED, buzzer to get alarm sound, visonic phone dialing to activate the automatic dialing with the owner, and turn off the electricity.

This project used motion detector also. Which Connected to the Atmel mega32 microcontroller, when the motion detector is activate and detect object send signal to the microcontroller, to LED, buzzer, phone dialing ,LCD which is used to display which sensor is activate, visonic phone dialer to dials the desired number and then telling them the prerecorded message.

And in the last time of this semester a small testing is done to improved the system. This testing about using an internet application to check the status of the inputs (motion sensor and smoke detector) and also the outputs by xport server web page.

3.6 Summary

This chapter contained the main objectives and requirements in the project, and showed the general block diagram of the project and explain each sub block diagram, and it represented the available design options. Finally it explained how the does system work.

through keypad. After that the user can enabling the system through keypad, after the system is enabled the green led is turn on. Also the user can make disable the system by entering password via keypad.

In this project used smoke detector, if the smoke detector goes to wrong, this detector send signal to Atmel mega32, then the Atmel mega32 send signal, to activated water pump, LED, buzzer to get alarm sound, visonic phone dialing to activate the automatic dialing with the owner, and turn off the electricity.

This project used motion detector also. Which Connected to the Atmel mega32 microcontroller, when the motion detector is activate and detect object send signal to the microcontroller, to LED, buzzer, phone dialing ,LCD which is used to display which sensor is activate, visonic phone dialer to dials the desired number and then telling them the prerecorded message.

And in the last time of this semester a small testing is done to improved the system. This testing about using an internet application to check the status of the inputs (motion sensor and smoke detector) and also the outputs by xport server web page.

3.6 Summary

This chapter contained the main objectives and requirements in the project, and showed the general block diagram of the project and explain each sub block diagram, and it represented the available design options. Finally it explained how the does system work.

Chapter Four

Detailed Technical Project Design

4.1 Overview.

4.2 Schematic Design.

4.3 Visonic 2-Channell speech dialer DL-125CA.

4.4 XPort Embedded Web Server.

4.5 Summary

Chapter four
Detailed Technical Project Design

4.1 Overview

This chapter will shows the details structure of the system circuit. It describes each system part circuit independently, Atmel32 microcontroller, sensors, camera and other component, also provides the full schematic view to this project for each subsystem in it and the complete system.

4.2 Schematic Design

This chapter illustrates a detailed schematic design for the major components that make up this project as well as the interfacing schematic design for this entire component together.

The components are:

1. Atmel mega32 Microcontroller.
2. Motion sensor.
3. Smoke sensor.
4. XPort embedded web server.
5. Visonic 2- channel speech dialer DL-125CA.

4.2.1 AVR Atmel Mega32 Microcontroller.

Atmel Mega32 is the main part of the system; it should receive the value of environment parameters and send the right decision to the right actuator. Figure 4-1 shows the microcontroller with oscillating circuit and power on reset circuit.

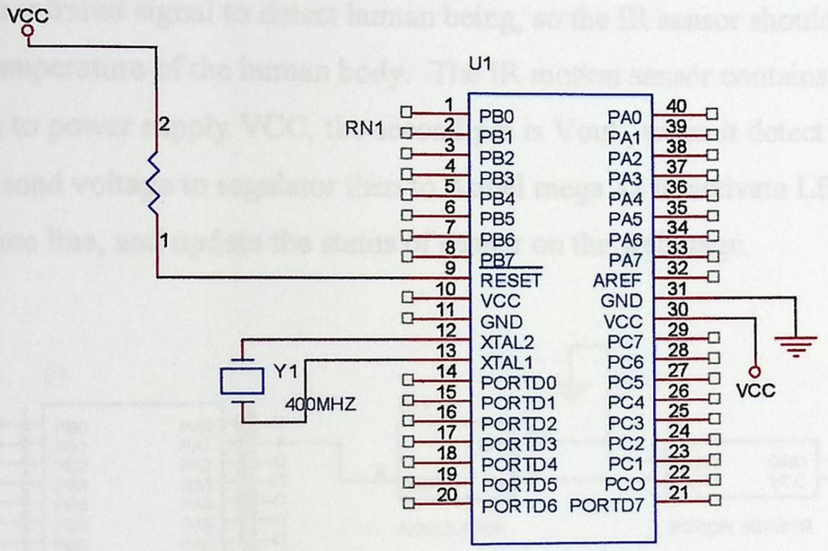


Figure 4.1 Microcontroller with oscillator

4.2.2 Sensing Circuits:

Sensing circuits are designed to realize the environment parameters; these parameters are measured in order to guide the microcontroller to response in a right manner. Sensor used in this project temperature sensor, motion sensor and smoke

sensor. Each sensor has a distinct circuit used to get the desired and functional output.

4.2.2.A Motion Sensor Circuit

The IR motion sensor is used to sensing feature on most security system is a passive system that detects infrared energy. In our project the IR motion sensor send the infrared signal to detect human being, so the IR sensor should be sensitive to the temperature of the human body. The IR motion sensor contains three pin the first pin to power supply VCC, the second pin is Vout , when it detect human motion send voltage to regulator then to Atmel mega 32 to activate LED, buzzer, and phone line, and update the status of sensor on the web bage.

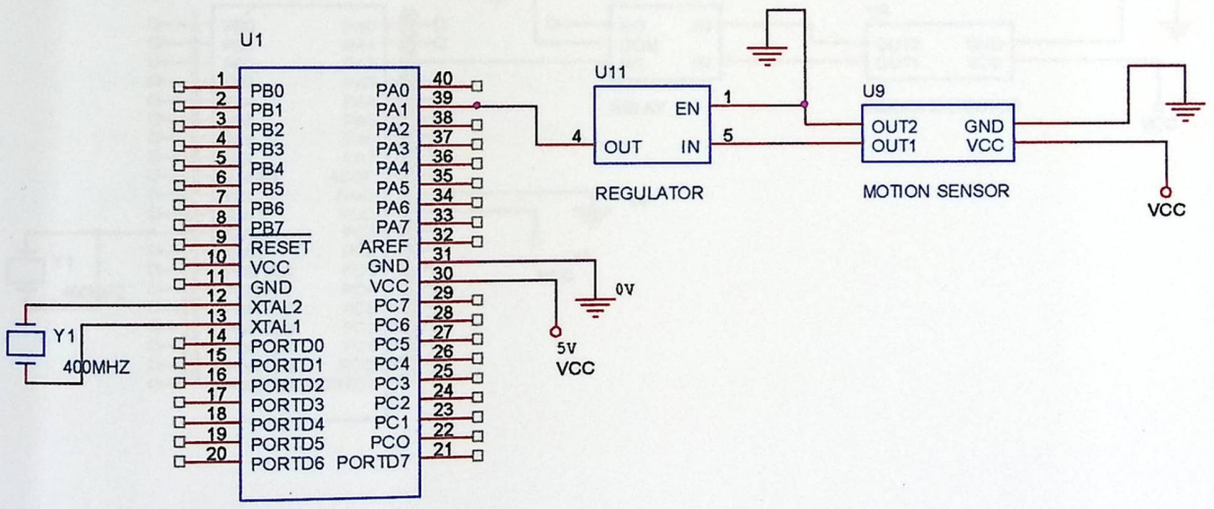


Figure 4.2 motion sensor circuit

4.2.2.B Smoke Detector

The some detector is one of the most popular alarm sensors used nowadays for fire alerting at kitchens, hotels, and hospitals. We choose this sensor because it is easy to use and deal with, and available on demand.

This detector consists of four , one is connected to the power supply (21V), the second edge is connected to the ground, and the other two edge is connected to the coil of the relay , and the NC(NORMALLY CLOSED) line of the relay is connected to the Atmel mega32. The principle on which this detector behaviors works is to output (12 V)for clear environment.

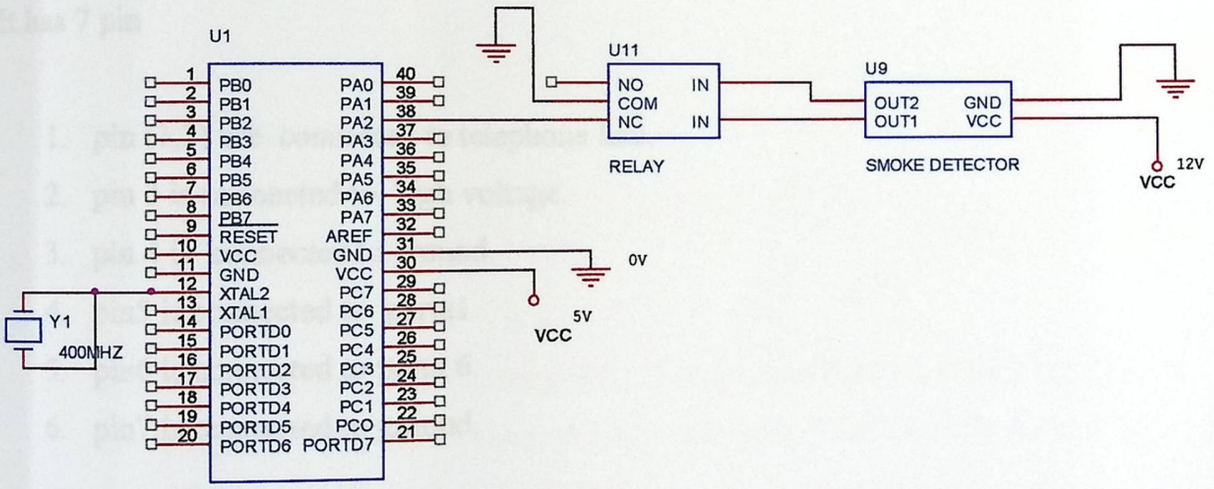


Figure 4.3 Smoke sensor circuits

4.3 Visonic 2- channel speech dialer DL-125CA.

The DL-125C is an programmable speech dialer with two alarm inputs. It is designed for verbal reporting of two separate events, one event per input. Each event can be reported to 4 different remote telephones, or both events can be reported to the same 4 telephones. The telephone numbers of the called parties may be frequently reprogrammed by the user.

A communication session with the first / second group of telephones is initiated by triggering alarm inputs Z-1 / Z-2, or by pressing AL-1 / AL-2 on the front panel, respectively.

It has 7 pin

1. pin (1,2) are connected to telephone line.
2. pin 3 is connected to high voltage.
3. pin 4 is connected to ground.
4. pin5 is connected to porta1
5. pin6 is connected to porta 6.
6. pin7 is connected to ground.

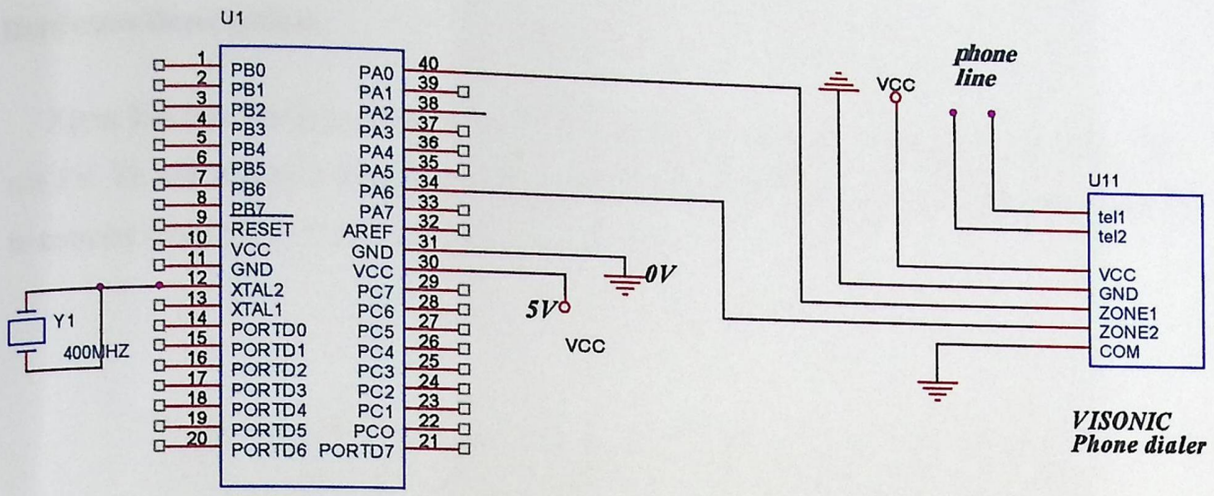


Figure 4.4 Visonic speech dialer DL-125CA

4.4 XPort Embedded Web Server

Simple description of logical structure:

1. The sensors connect to the Atmega32 A/D converter (PORTA PINs 1 and 2).
2. The Output connect to the Atmega32 A/D converter (PORTA PINs 4,5 and 7).
3. Read Sensors and output every 400ms.
4. The results are then sent out through the RS232 serial connection.
5. The serial connection goes to the Xport, which converts serial communication to TCP/IP packets.
6. Then we can connect to the Xport through a java applet that it hosts on its website.
7. The applet receives the serial information as if the Xport were not there, and then interprets it and places it onto a GUI.

Hardware description:

Xport has one serial port and three configurable I/O pins. It operates on 3.3 volts, DC, not 5V. To use it with ATmega32 that have TTL serial capability, we need some circuitry to convert from 5V to 3.3V. as in the circuit below.

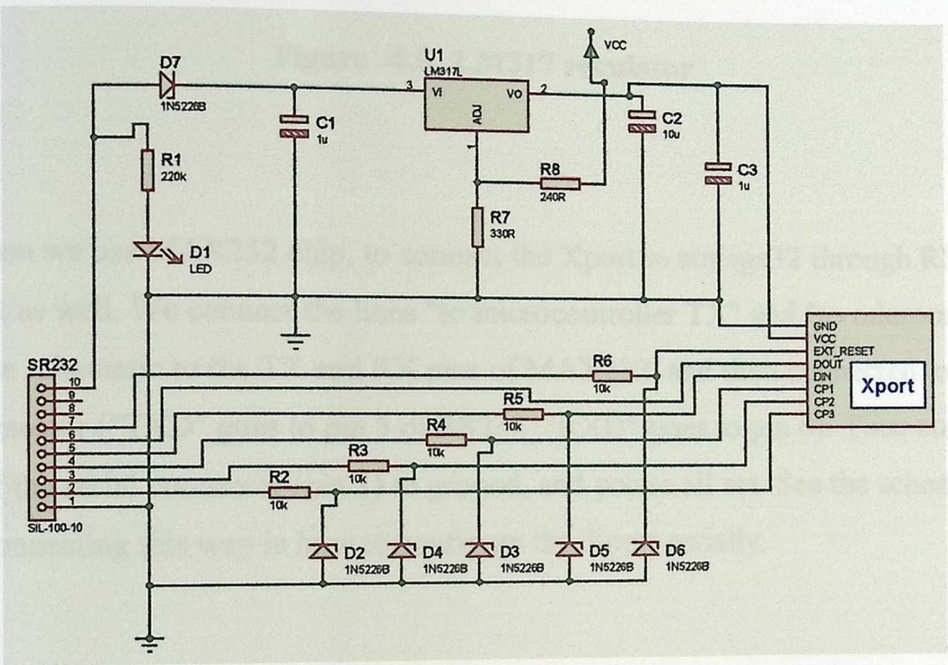
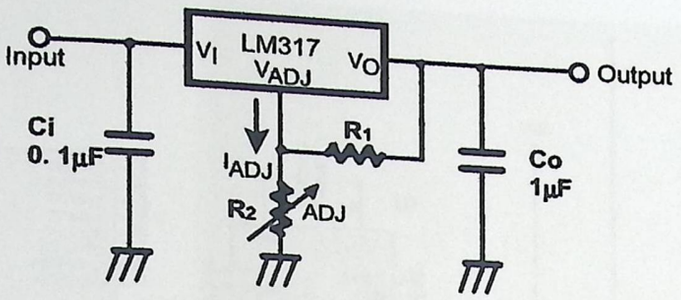


Figure 4.5 general schematic XPort server

For that we didn't find a regulator to convert 5V to 3.3V at the market, so we used LM317 regulator with small circuit see below.



$$V_O = 1.25V\ (1 + R_2 / R_1) + I_{ADJ}\ R_2$$

Figure 4.6 LM317 regulator

Then we use MAX232 chip, to connect the Xport to atmega32 through RS-232 serial port as well. We connect the lines "to microcontroller TX" and "to microcontroller RX" in the schematic to the TX and RX pins of MAX232, and then connected to the serial connector ("TXD" goes to pin 5 of RS232 ,"RXD" goes to pin 6). Then connect the ground of the serial connector (pin 1) to ground, and you're all set. See the schematic below. Connecting this way is how to configure the Xport serially.

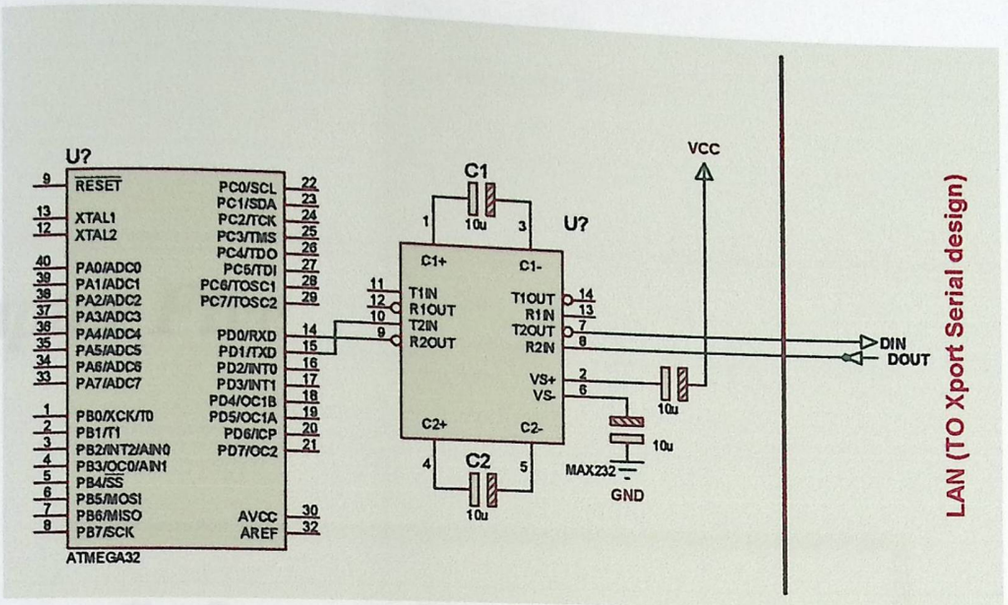


Figure 4.7 Xport with Atmega32 circuit

4.5 summary

At this chapter the hardware system design was discussed and presented. First it showed the main part of the system the microcontroller , then the sensing circuit (smoke and motion) , Visonic 2- channel speech dialer DL-125CA and finally, XPort Embedded Web Server.

Chapter Five

Software System Design

5.1pony prog 2000.

5.2 Code Vision AVR Studio.

5.3 Main program Flow Chart.

5.4 Smoke Sensor Floow Chart.

5.5 Motion Sensor Flow Chart.

5.6 Buzzer Flow Chart.

5.7 Xport Flow Chart.

5.8 Summary.

Chapter Five
Software System Design

This chapter describes the general and detailed program flowcharts which were designed to program the microcontroller of the system. The overall software is programmed using C language. The CodeVisioAVR software is used.

5.1 PonyProg2000:

Being that Atmel microcontroller has the ability to be programmed more than one time, so there must be an interface circuit between Atmel and PC (personal computer) so we can read from and write to it (programming). Also the PC show have software which can program Atmel mega32 through it and that software is PonyProg2000 is free download software used to operate along with the corresponding programmer.

The PonyProg2000support multiple series from the micro chip Atmel (for 16 pin and 40 pin_, there are many devices that are compatible with PonyProg2000, this PonyProg2000 programmer downloads the hex file that represents the program we want to download it on the microcontroller.

PonyProg2000 is very easy to deal with, and use through its friendly UI (user interface). Before that the software should be download to the PC, then the programmer circuits is connected to the PC through USB port, and then we can work easy with programmer.

Studio:

compiler is 32 bit application, runs under Windows 95, 98,
to use Integrated Development Environment and C Compiler
ntation and keywords highlighting. Supported data types: bit,
, float.

tensions for:
ssing the EEPROM & FLASH memory areas
level access to registers
terrupt support

ode Vision advantages:
bility to insert assembler code directly in the C source file.
efficient use of RAM: Constant character strings are stored only in FLASH
mory and aren't copied to RAM, like in other compilers for the AVR
Source level debugging

Using coder vision avr studio help us writing code and testing it to make the project works in a proper way. After building the program (executing the build command in the cod vision avr studio so it compiles the code0 we write for the system. It automatically creates hex file having the same name of the project which includes the hex translated code of the C code.

5.2 Code Vision AVR Studio:

Code Vision AVR compiler is 32 bit application, runs under Windows 95, 98, NT 4.0 and 2000. Easy to use Integrated Development Environment and C Compiler editor with auto indentation and keywords highlighting. Supported data types: bit, char, int, short, long, float.

AVR specific extensions for:

- Accessing the EEPROM & FLASH memory areas
- Bit level access to registers
- Interrupt support

Other Code Vision advantages:

- Possibility to insert assembler code directly in the C source file.
- Very efficient use of RAM: Constant character strings are stored only in FLASH memory and aren't copied to RAM, like in other compilers for the AVR
  C Source level debugging

Using coder vision avr studio help us writing code and testing it to make the project works in a proper way. After building the program (executing the build command in the cod vision avr studio so it compiles the code0 we write for the system. It automatically creates hex file having the same name of the project which includes the hex translated code of the C code.

5.3 main program flowchart

In this section the general program of the ATMega microcontroller is explained

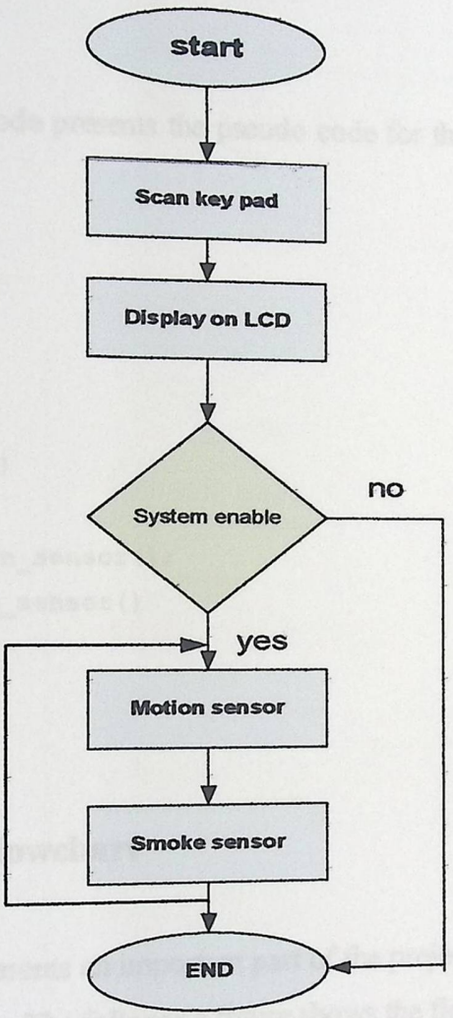
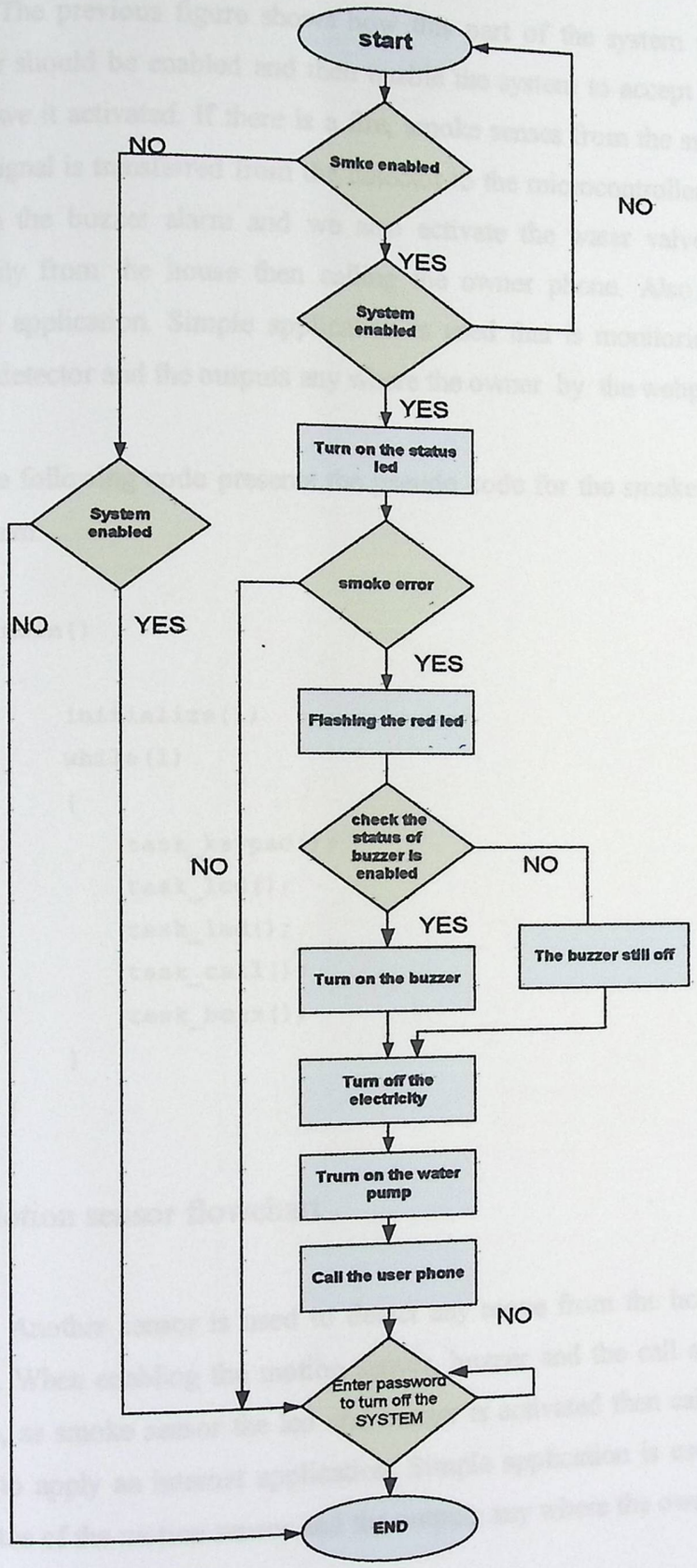


Figure 5.1 General flowchart of the system.

The previous flowchart describes how the microcontroller follows a specific sequence in order to match the desired results. This sequence indicates that the program initializes the microcontroller ports using the build in libraries in the CodeVision AVR software.



5.2 Smoke sensor flowchart

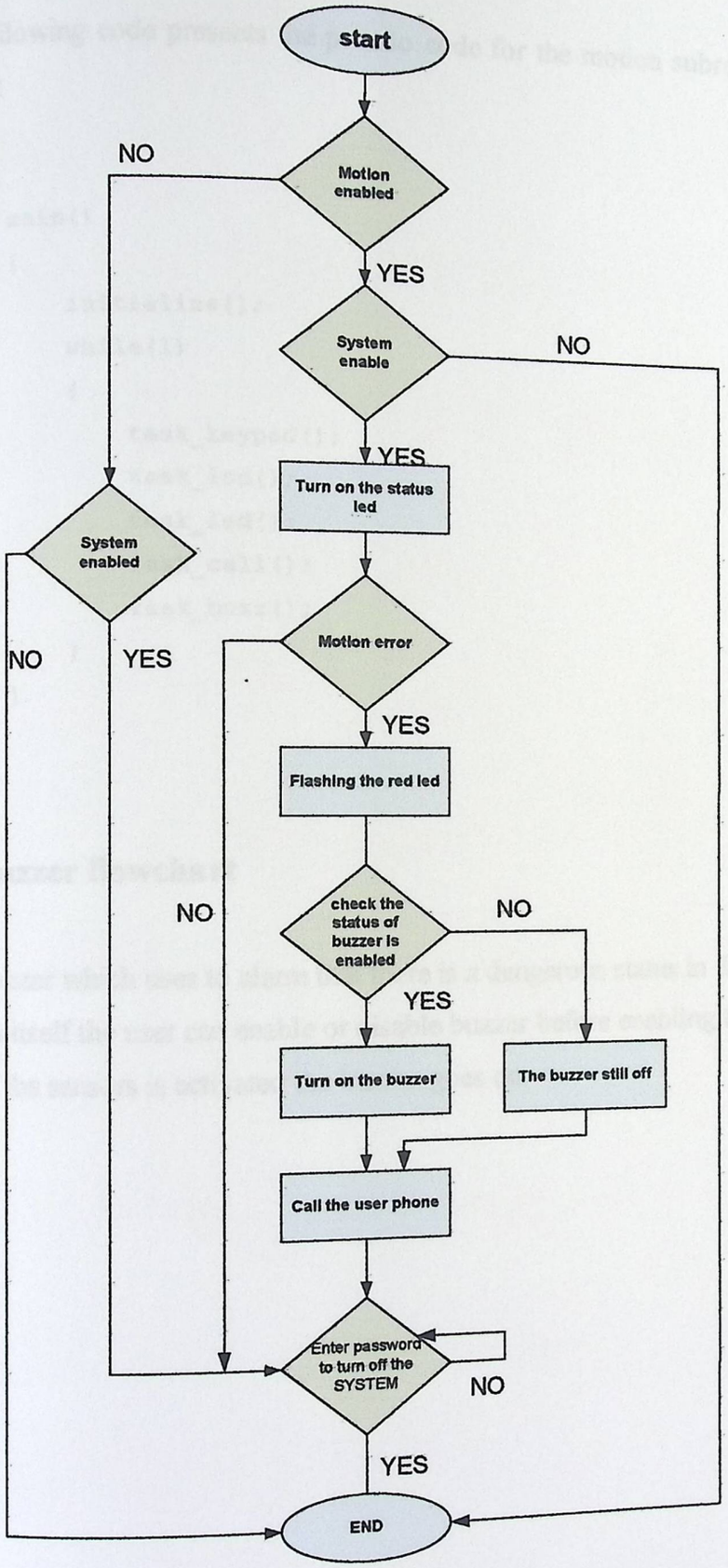
The previous figure shows how this part of the system works. First smoke detector should be enabled and then enable the system to accept the signal from the smoke we it activated. If there is a fire, smoke senses from the smoke detector. This signal signal is transferred from the detector to the microcontroller to activate the led, turn on the buzzer alarm and we also activate the water valve after turn off the electricity from the house then calling the owner phone. Also trying to apply an internet application. Simple application is used that is monitoring the status of the smoke detector and the outputs any where the owner by the webpage.

The following code presents the pseudo code for the smoke subroutine from of the system:

```
main()
{
    initialize();
    while(1)
    {
        task_keypad();
        task_lcd();
        task_led();
        task_call();
        task_buzz();
    }
}
```

5.5 Motion sensor flowchart

Another sensor is used to detect any move from the house which is motion sensor. When enabling the motion sensor, buzzer and the call alarm then enable the system, as smoke sensor the led and buzzer is activated then calling the owner. Also trying to apply an internet application. Simple application is used that is monitoring the status of the motion sensor and the outputs any where the owner by the webpage.



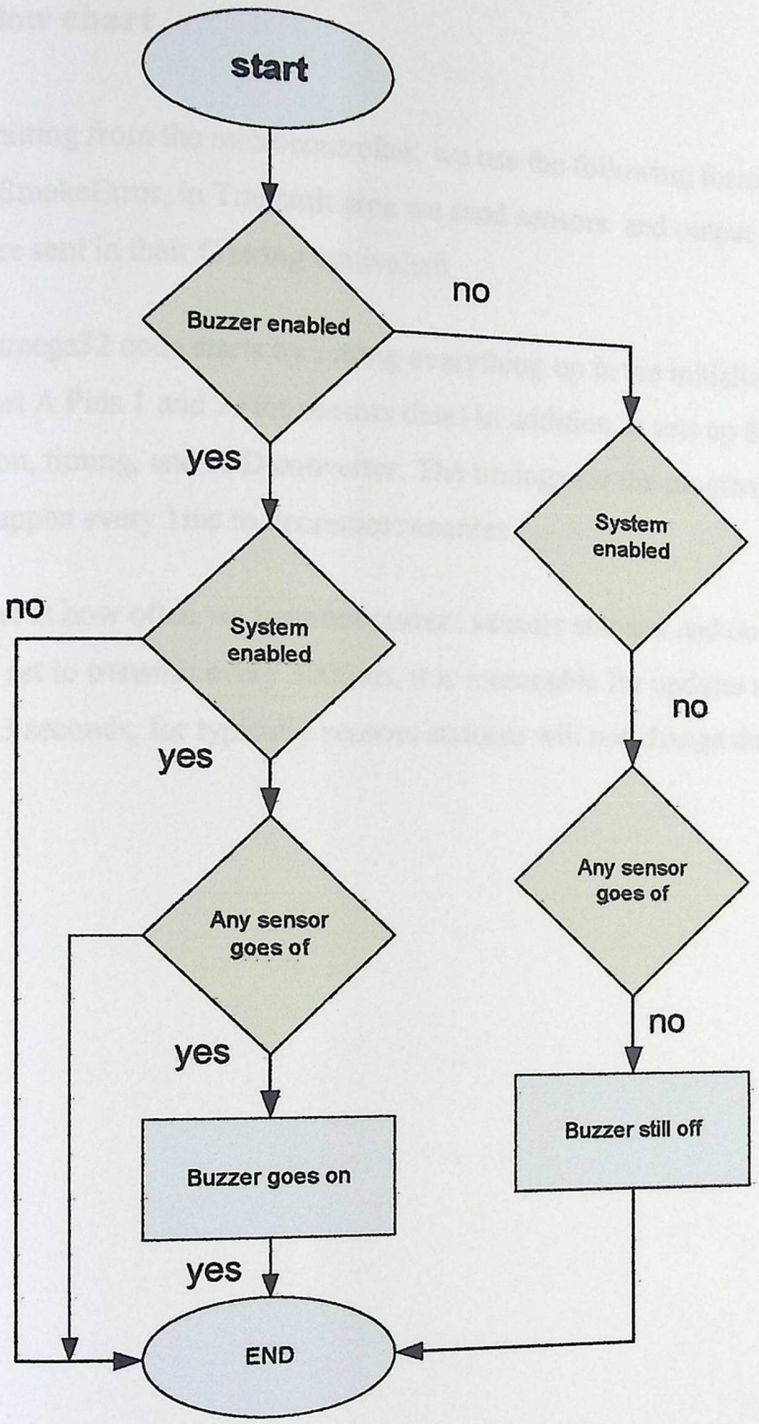
5.3 Motion sensor flowchart.

The following code presents the pseudo code for the motion subroutine from of the system:

```
main()
{
    initialize();
    while(1)
    {
        task_keypad();
        task_lcd();
        task_led();
        task_call();
        task_buzz();
    }
}
```

5.6 Buzzer flowchart

The buzzer which uses to alarm that there is a dangerous status in the house. From the system itself the user can enable or disable buzzer before enabling the system. When any of the sensors is activated the buzzes goes on.



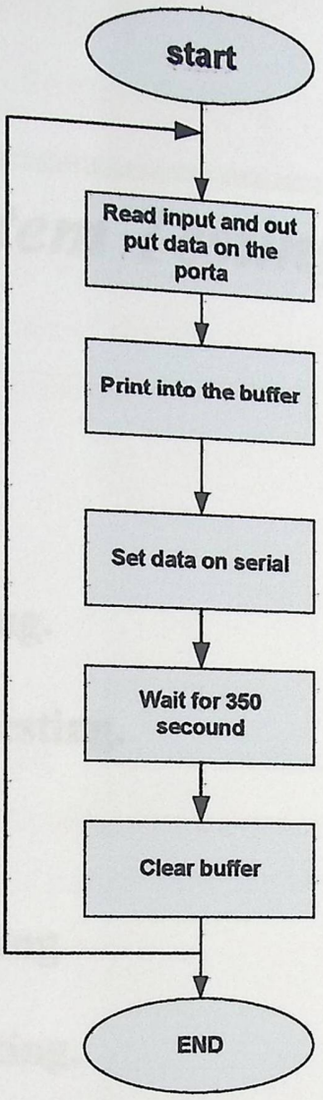
5.4 Buzzer flowchart.

5.7 Xport flow chart

For transmitting from the microcontroller, we use the following format. MotionError, SmokeError, in Transmit area we send sensors and output informations these values are sent in their C string equivalent.

The Atmega32 code starts by setting everything up in the initialize function, which sets Port A Pins 1 and 2 (for sensors data) In addition, it sets up the serial communication, timing, and A/D converter. The timings for the program set an interrupt to happen every 1ms to decrement counter.

The counter is how often we transmit current sensors statuses and output statuses. Currently it is set to transmit every 3000ms, it is reasonable for updates to only happen every 3 seconds, for typically sensors statuses will not change that often



5.5 xport flow chart

5.8 Summary

This chapter explained exactly the sequence of the main program of the ATMega32 microcontroller; also each subroutine was illustrated by its flowchart and pseudo code.

Chapter Six

System Testing

6.1 Atmel Mega32 testing.

6.2 Keypad and LCD testing.

6.3 Buzzer testing.

6.4 Smoke detector testing.

6.5 Motion detector testing.

6.6 Visonic calling alarm testing.

6.7 Other components testing.

Chapter Six
System Testing

This chapter demonstrates how the system was tested. Each subsystem is tested independently. Software and hardware of the system is included in the same subsections. Finally complete system testing is presented

6.1 Atmel mega 32 testing

Atmel Mega32 microcontroller with the oscillating circuit was build as shown in the next figures, this testing needs the use of the programmer in order to download the test program in order to insure that the Atmel Mega32 microcontroller response to its desired function.

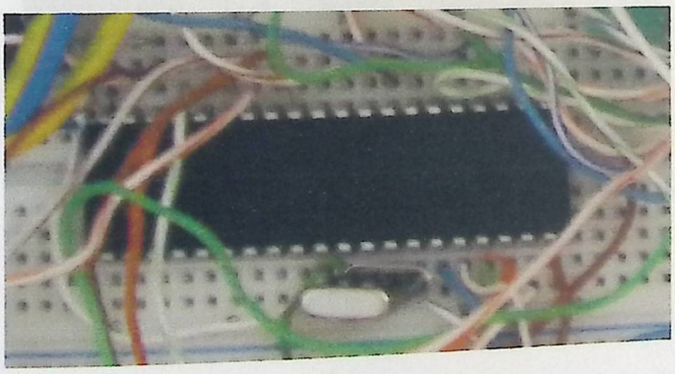


Figure 6.1 Atmel mega 32 testing

Two microcontroller programmers are used through working in this project. The first one was building by the project work team with parallel cable but it did not work well

for long time only for a few week, then using the second one from the university itself with USB port and continue with it to the end of the project.

6.2 Keypad and LCD

LCD is used in this project to display the menu that represent the interface for the user and the keypad is used to navigate through this menu and also to entering the user password. Testing this part of the system is done by pressing on the key and notice the effect on the LCD through the enabling process and to change the current menu on LCD.

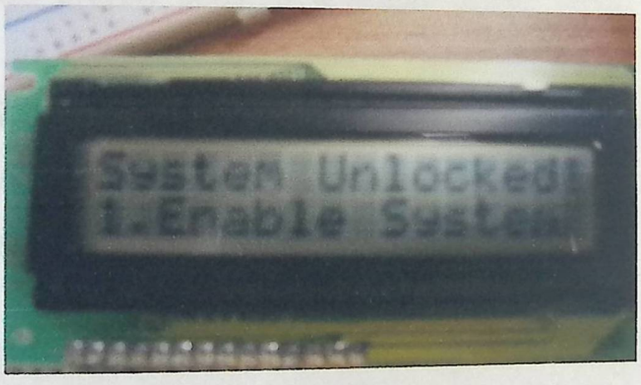


Figure 6.2 LCD testing

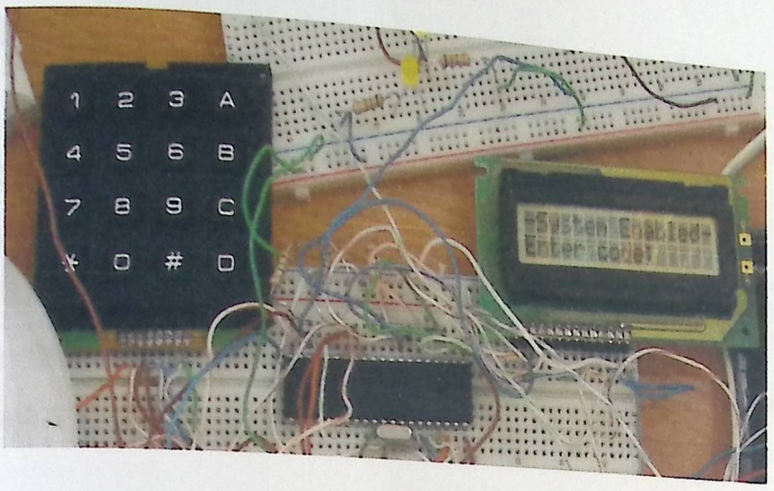


Figure 6.3 LCD with keypad testing

6.3 Buzzer testing

Alarm circuit of this project contains a simple buzzer which tested alone by giving it 5 volts as power in VCC edge and connect the other on ground. In this system, to test it transistor is applied to its circuit that connected to microcontroller then activate the buzzer enable submenu from the main menu on the LCD and then activate smoke detector or the motion sensor to do it alarming.

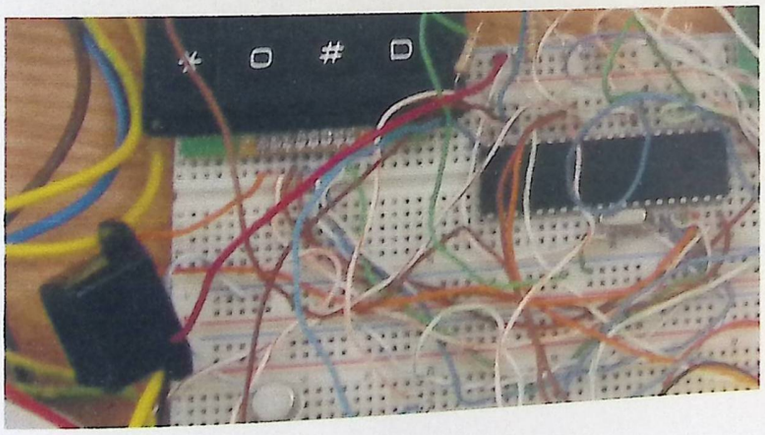


Figure 6.4 Buzzer testing

6.4 Smoke detector testing

Smoke detector circuit as show in figure. This circuit contains a smoke detector has four edges, and a relay also used. The smoke detector is tested by connected it to high voltage (+12 v) and with ground (0v), the other two edge is connected to the relay coil to microcontroller. When there is a fire the detector sense the smoke and turning on the led which build in the smoke and display S on LCD.

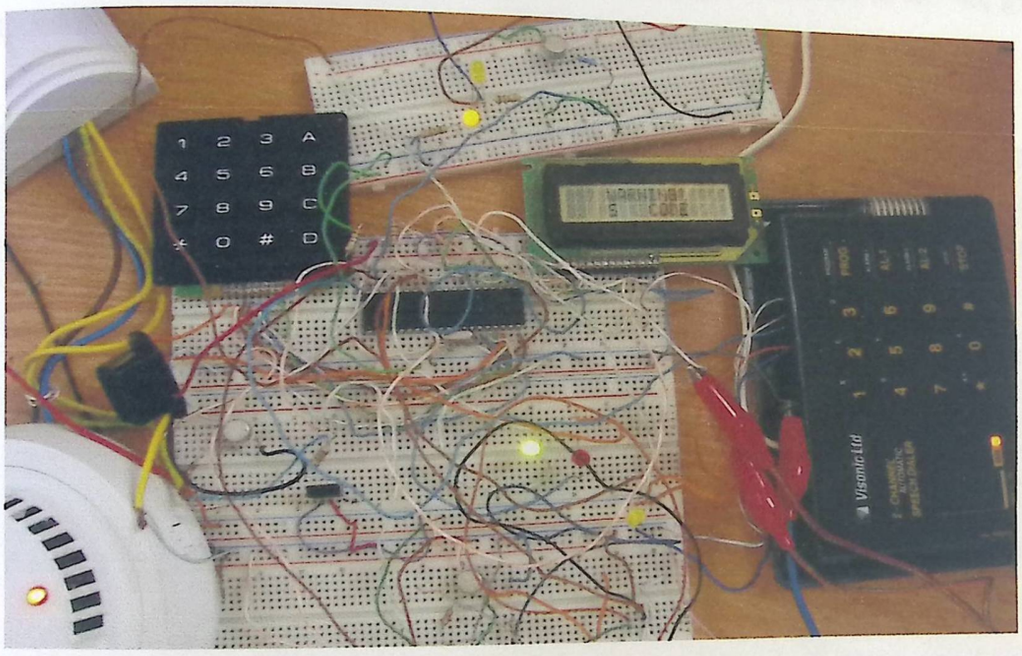


Figure 6.5 Smoke detector testing

6.5 Motion detector testing

Motion detector circuit using the IR motion detector with four edges, and regulator with three edges. The motion detector is tested by connecting the first edge of motion detector is connected to ground (0 V) and the second edge connected to the high voltage (12 V), and the others, one of them is connected to the ground and the other to the input of regulator. When the motion detect any moving the led will light the build in led on, giving 5 volt and appears M on LCD else it stay off and giving o volt.

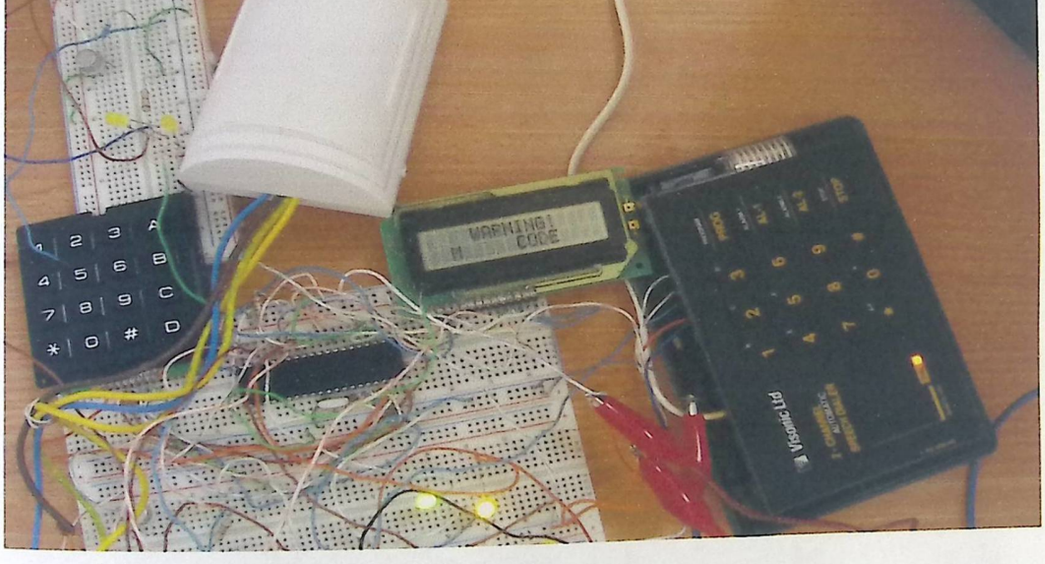


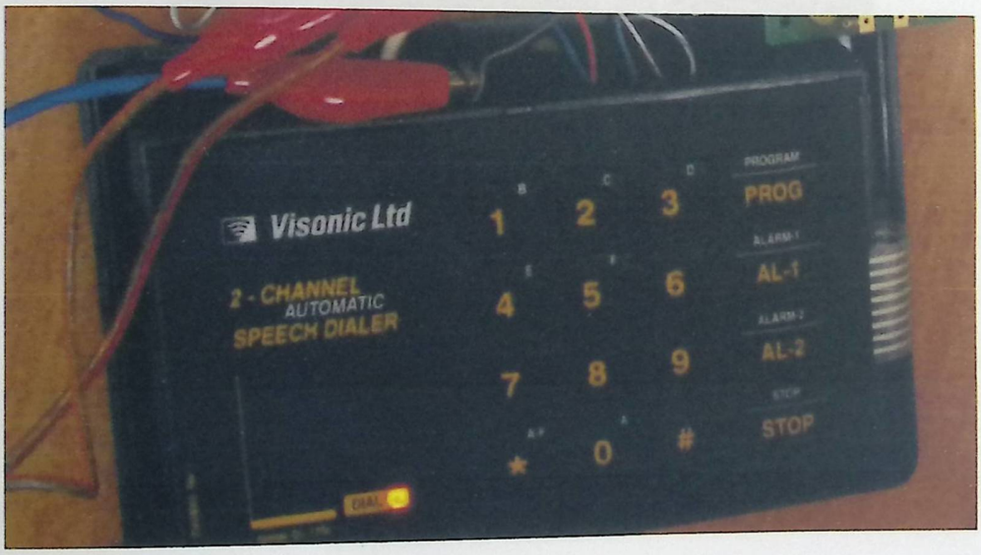
Figure 6.6 Motion Sensor testing

6.6 Visonic calling alarm testing

First this device is needed to be programmed. Programming it means to entering the numbers that user want for each zone separately of the other zone up to 8 numbers for both

first fourth numbers for zone1 and others for zone2 and recording the message which is divided to three parts, the common for zone1 and zone2 , and the other two massages one for zon1 and other for zone 2. Message must be recorded according to the action that related to this zone. It was tested independently by connecting the power line to (12 V), line two to ground, by making a short circuit between zone one and ground will calling the four numbers that was specified. The same for the second zone.

In this project it was tested by connected the zones for tow pins of PORTA when the smoke detector is activated zone one will turn on and calling the numbers to tell that there is a fire in the house. Similar for the motion sensor but telling that there is some one moving in the house.



6.7 Visonic calling alarm testing

6.8 Other components

The other components of the system such as green status led and the red led which display the activation of the sensors is connected as shown in schematic figure of the system is tested by enable the system and the activation of the sensors. These leds are connected to microcontroller and accepting the signal. If the system enabled the green led light also if any sensor activated the red led light.

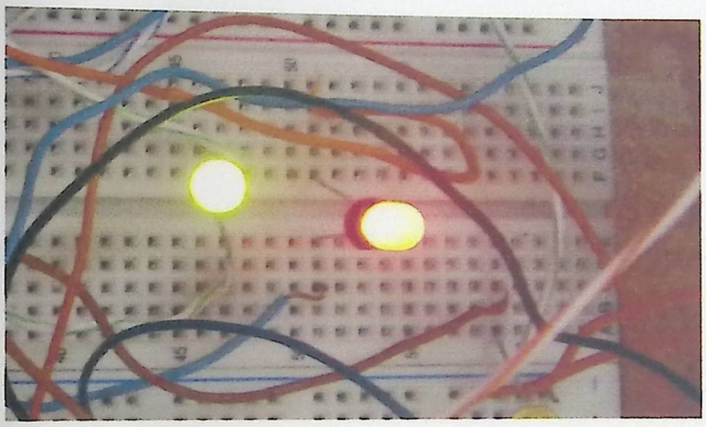


Figure 6.8 Leds Tests

Chapter Seven

Conclusion and Future Work

7.1 Preface.

7.2 Conclusion.

7.3 Problems.

7.3.1 Hardware Problems.

7.3.2 Software Problems.

7.4 Future work and developing suggestion.

Chapter Seven
Conclusion & Future Work

7.1 preface

This chapter describes and gives a complete look over the entire project from the beginning: produced problems faced the implementation, so those who will upgrade and develop this idea can avoid these problems. It also provides suggestions that could be useful for developing the idea in future.

7.2 Conclusion:

Three main goals were considered during the developments " Smart house" : creat integrated system that would monitor the home continuously. Design the system so that affordable and easy enough for the average home owner to build, and enable the system to not only detect problems in the home, but also to correct these problem when wver possible, each of these goal was met during the system design and construction process.

A few different design considerations evaluated, while keeping in mind that the system was supposed to be ado-it yourself project that could be easily recreated. For this reason relay logic was chosen because of the availability of relays, and their ease of use.

Aside from the affordability the ' smart house" also offers the advantage being able to prevent damage, which cannot emergency personnel, valuable time is lost in waiting for them to arrive. The ' smart house' has the ability damage from firing.

7.3 problems:

The problems we faced during this year have two phases hardware and software problems and the sections below illustrate them.

7.3.1 Hardware Problems:

- The first and the most difficult problem was the waiting the components (atmel mega 32. and xport embedded web server) to be available. This late extended from last month of the first semester to the second month on the second semester.

- After atmel mega 32 crash, we wait for another one for at least three weeks, and then we worked together and built our own programmer.

- Also one of the problems, connecting wrong wires in the wrong places, such as swapping the high and low voltage to the circuit, or provides high voltage to the circuits, or providing high voltage or current that the circuit cannot work with it and in some times damaged the components.

7.3.2 Software Problems:

- Since that each atmel microcontroller should have programmer so that atmel could be programmed, there was only on atmel programmer for 3 graduation groups.

- The unavailability of this device caused a great latency in microcontroller programming which was the most important operation in system testing and implementation.
- Programming atmel mega 32 at first was difficult not in using C_ programming language but because of the consideration, that we must study to configure atmel mega 32 to work correctly.

7.4 future work & dèveloping suggestions:

The "smart house " is a fully functional and reliable system. But then are afew improvement and additions that can be made to it. First make control over the internet, secondly, printed circuit in boards could be used for the central unit, which would increase the cost, but would take up less space and allow the central unit to fit into a smaller enclosure, lastly , in the future the sensors could be connected to the system wiressly.

References

A. Internet sites

1. http://eleceng.dit.ie/frank/msp430/microcontrollers.pdf
2. http://avrmicrocontrollers.com/
3. http://www.atmel.com
4. https://knol.google.com/k/leroy-davis/rs232-serial-interface/h0rhqwmul9ql/3
5. http://www.sparkfun.com/commerce/product_info.php?products_id=8845
6. http://folk.uio.no/od/tcp-ip-intro/section3_2.html
7. http://www.drrobot.com/products/item_downloads/CAM3908_1.pdf
8. http://www.national.com/ds/LM/LM35.pdf
9. http://www.glolab.com/pirparts/infrared.html
10. http://jaycar.com.au/products_uploaded/QP-5519-2.pdf
11. http://lisgar.net/weatherstation/BX24_Docs/App_Notes/Keypad_App_Note.pdf
12. http://www.thomasnet.com/products/piezoelectric-buzzers-9386921-1.html
13. http://www.keil.com/dd/docs/datashts/atmel/at89c51_ds.pdf
14. www.nullmodem.com
15. http://www.tslab.ssvl.kth.se/csd/projects/0313/ipsil%20product%20brief%208930.pdf

16. http://www.epanorama.net/links/irremote.html

17. http://www.smart-nl/engles/infonet/projects.html#top

B. Smith, Ralph L. smart House: the Coming Revolution in Housing. Columbia, MD: Gp publishing,1988

C. Allen, Edward, Ed. (1974). The Responsive House. Cambridge, MA: MIt press, 1975.

D. شد محمد عبد المعطي. برمجة c باستخدام لغة atmel avr: شعاع للنشر والعلوم, 2005

E. Bolton, **Measurements and Instrumentation System,** New York, 1996

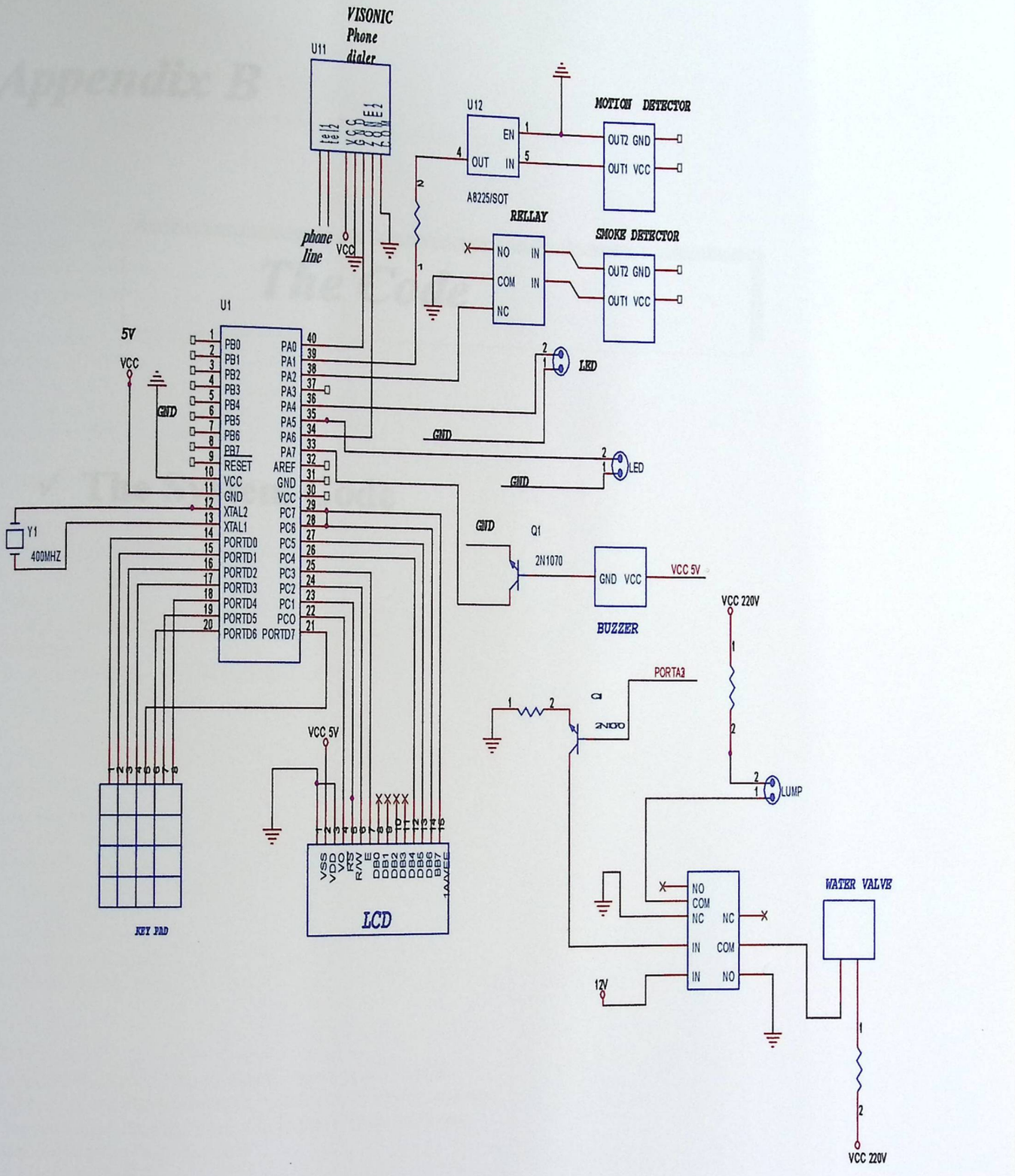
F. Ali Salameh.Omar Al Ja'bari and Mais Tahboub ,**Autonomous Mobile Fire Fighting Robot**, Palestine polytechniqe university,Hebron,Palestine,2007

G. "**Smart Home**" ,Brandgurg Technical University Cotton, Germany,2002

Appendix A

System Schematic Circuit

- ✓ **Schematic Project Design**



Title <Title>		
Size B	Document Number <Doc>	Rev <Rev Code>
Date: Wednesday, June 10, 2009	Sheet 1 of 1	

Appendix B

The Code

- ✓ **The System Code**

/******** the system code **********/

```
#include <mega32.h>
#include <stdio.h>
#include <stdlib.h>
#include <delay.h>

//timeout values for each task
#define t1 100
#define t2 120          //task_keypad
#define t3 250          //task_led
#define t4 200      //task_call
#define t5 200      //task_lcd
#define t6 1            //task_buzz

//LCD
#define LCDwidth 16 //characters

#asm
   .equ __lcd_port=0x15
#endasm
#include <lcd.h>     // LCD driver routines



//functions
void initialize(void);
void task_keypad(void);
void getKey(void);
char mapKey(int myNum);
void task_led(void);
void task_lcd(void);
void task_call(void);
void task_buzz(void);


//variables
unsigned char time1, time2, time3, time4, time5, time6;
signed char securityEnabled, motionEnabled, smokeEnabled;
char tempError, motionError, smokeError, magneticError;
char callEnabled, buzzEnabled;
char securityCode[4];
char tempCode[4];
char adc_ch, currMenu, currKey, numDigEntered;
char buzzFreq;

//keypad
char butnum, key, prvbutnum;
char lcd_buffer[5]; // LCD display buffer
```

```
flash unsigned char keytbl[16]={0xee, 0xde, 0xbe, 0x7e, 0xed, 0xdd, 0xbd, 0x7d, 0xeb, 0xdb, 0xbb, 0x7b,
0xe7, 0xd7, 0xb7, 0x77};


int i;
int buzzCounter;
int currMotion;



//*********************************************************
//timer 0 compare ISR
interrupt [TIM0_COMP] void timer0_compare(void)
{
        //Executed every 0.001 second
        if (time1>0)        --time1;
        if (time2>0)        --time2;
        if (time3>0)        --time3;
        if (time4>0)        --time4;
        if (time5>0)        --time5;
        if (time6>0)        {--time6; buzzCounter ++; if(buzzCounter == 8000) buzzCounter = 0;}
}


//*********************************************************
//ADC input ISR
interrupt [ADC_INT] void adc_isr(void)
{
  unsigned int adc_data;
  adc_data = ADCW;

    if(adc_ch == 0xC1) //motion sensor
  {
    currMotion = adc_data;
    if(motionEnabled == 1)
    {
      if(adc_data  > 5)
        motionError = 1;
      else
        motionError = 0 | motionError;
    }
    else
      motionError = 0;
    ADMUX = 0xC2;
  }
  if(adc_ch == 0xC2) //smoke detector
  {
    if(smokeEnabled == 1)
    {
      if(adc_data > 10)
        smokeError = 1;
      else
        smokeError = 0 | smokeError;
```

```
    }
    else
      smokeError = 0;
    ADMUX = 0xC1;
    ADMUX = ADMUX | 0x40;
  }
  adc_ch = ADMUX;
  ADCSRA = ADCSRA | 0x40;
}

//***********************************************
//update LCD screen
void task_lcd(void)
{
  time5 = t5;
  if(securityEnabled == 1)
  {
    if(motionError == 1|| smokeError == 1)
    {
      lcd_clear();
      lcd_gotoxy(0,0);
      lcd_putsf("   WARNING!   ");
      lcd_gotoxy(0,1);
      if(motionError == 1) //motion error
      {
        lcd_gotoxy(2,1);
        lcd_putsf("M");
      }
      if(smokeError == 1) //smoke error
      {
        lcd_gotoxy(4,1);
        lcd_putsf("S");
      }
      lcd_gotoxy(8,1);
      lcd_putsf("CODE");
      for(i=0; i<numDigEntered; i++)
      {
        lcd_gotoxy(12+i, 1);
        lcd_putsf("*");
      }
      return;
    }

    lcd_gotoxy(0,0);
    lcd_putsf("-System Enabled-");
    lcd_gotoxy(0,1);
    lcd_putsf("Enter code:    ");

    for(i=0; i<numDigEntered; i++)
    {
      lcd_gotoxy(11+i, 1);
      lcd_putsf("*");
    }
    return;
  }
```

```
if(securityEnabled == -1)
{
  lcd_clear();
  lcd_gotoxy(0,0);
  lcd_putsf("-Enter New Code-");
  for(i=0; i<numDigEntered; i++)
  {
    lcd_gotoxy(5+i, 1);
    lcd_putsf("*");
  }
  return;
}
if(securityEnabled == 0)
{
  lcd_gotoxy(0,0);
  lcd_putsf("System Unlocked!");

  lcd_gotoxy(0,1);
  if(currMenu == 0)
  {
    lcd_putsf("UP & DOWN to NAV");
  }
  if(currMenu == 1)
  {
    lcd_putsf("1.Enable System ");
  }
  if(currMenu == 2)
  {
    lcd_putsf("2.(");
    lcd_gotoxy(3,1);
    if(motionEnabled == 1)
      lcd_putsf("*");
    else
      lcd_putsf(" ");
    lcd_gotoxy(4,1);
    lcd_putsf(")Mot. Sensor");
  }
  if(currMenu == 3)
  {
    lcd_putsf("3.(");
    lcd_gotoxy(3,1);
    if(smokeEnabled == 1)
      lcd_putsf("*");
    else
      lcd_putsf(" ");
    lcd_gotoxy(4,1);
    lcd_putsf(")Smk. Sensor");
  }
  if(currMenu == 4)
  {
    lcd_putsf("4.(");
    lcd_gotoxy(3,1);
    if(callEnabled == 1)
      lcd_putsf("*");
    else
      lcd_putsf(" ");
```

```
        lcd_gotoxy(4,1);
        lcd_putsf(")call Alarm");
      }
      if(currMenu == 5)
      {
        lcd_putsf("5.(");
        lcd_gotoxy(3,1);
        if(buzzEnabled == 1)
          lcd_putsf("*");
        else
          lcd_putsf(" ");
        lcd_gotoxy(4,1);
        lcd_putsf(")Buzz Alarm ");
      }
      if(currMenu == 6)
      {
        lcd_putsf("6.New Pswd:    ");
        for(i=0; i<numDigEntered; i++)
        {
          lcd_gotoxy(11+i, 1);
          lcd_putsf("*");
        }
      }
      if(currMenu == 7)
      {
        lcd_putsf("A.Buzz Freq:   ");
        sprintf(lcd_buffer,"%1i",buzzFreq);
        lcd_gotoxy(13,1);
        lcd_puts(lcd_buffer);
      }

    }
}

//***********************************************
//led status and etc.
void task_led(void)
{
  time3 = t3;
  if(securityEnabled == 1)
  {
    PORTA.4 = 1;              //green led
    if( motionError == 1|| smokeError == 1)
      PORTA.5 = !PORTA.5;    // red led
    else
      PORTA.5 = 0;
  }
  else
    PORTA = PORTA & 0xcF;
}

//***********************************************
//calling
void task_call(void)
{
  if( smokeError == 1)
```

```
    {
      if(securityEnabled == 1 && callEnabled == 1)
        PORTA.0=0;

    }
    if( motionError == 1)
    {
      if(securityEnabled == 1 && callEnabled == 1)
        PORTA.6=0;
    }
}

//***********************************************
//buzz alarm
void task_buzz(void)
{
  //buzzFreq 1~9
  //execute every 0.5 mSec
  time6 = 10-buzzFreq;

  if(butnum != 0)
  {
    PORTA.7 = !PORTA.7;
    return;
  }
  if(tempError == 1 || motionError == 1
    || smokeError == 1|| magneticError == 1)
  {
    if(securityEnabled == 1 && buzzEnabled == 1)

        PORTA.7 = !PORTA.7;
    else
        PORTA.7 = 0;

  }
  else
    PORTA.7 = 0;
}



//***********************************************
//keypad and take action
void task_keypad(void)
{
  time2 = t2;
  if(securityEnabled == 1)
  {
    getKey();
    if(butnum != 0)
    {
      if(mapKey(butnum) == 'S')
      {
        if(numDigEntered > 0)
          numDigEntered--;
        else
```

```
            return;
        }
        else
        {
          tempCode[numDigEntered] = mapKey(butnum);
          numDigEntered++;
        }
      }
      if(numDigEntered == 4)
      {
        if(tempCode[0] == securityCode[0] && tempCode[1] == securityCode[1]
        && tempCode[2] == securityCode[2] && tempCode[3] == securityCode[3])
        {
          securityEnabled = 0;
          motionError = 0;
          smokeError = 0;
          currMenu = 0;
          PORTA.0 = 1;
          PORTA.6 = 1;
        }
        numDigEntered = 0;
      }
      return;
    }
    if(securityEnabled == -1) //new pswd enter mode
    {
      getKey();
      if(butnum != 0)
      {
        if(mapKey(butnum) == 'S')
        {
          if(numDigEntered > 0)
            numDigEntered--;
          else
            return;
        }
        else
        {
          securityCode[numDigEntered] = mapKey(butnum);
          numDigEntered++;
        }
      }
      if(numDigEntered == 4)
      {
        securityEnabled = 0;
        currMenu = 0;
        numDigEntered = 0;
      }
      return;
    }
    if(securityEnabled == 0)
    {
      if(currMenu == 0)
      {
        getKey();
        currKey = mapKey(butnum);
```

```
    if(currKey == 'U')
      currMenu = 7;
    if(currKey == 'D')
      currMenu = 1;
    return;
  }
  if(currMenu == 1)
  {
    getKey();
    currKey = mapKey(butnum);
    if(currKey == 'E')
    {
      securityEnabled = 1;
      motionError = 0;
      smokeError = 0;
    }
    if(currKey == 'U')
      currMenu = 7;
    if(currKey == 'D')
      currMenu = 2;
    return;
  }
  if(currMenu == 2)
  {
    getKey();
    currKey = mapKey(butnum);
    if(currKey == 'E')
    {
      if(motionEnabled == 0)
        motionEnabled = 1;
      else
        motionEnabled = 0;
    }
    if(currKey == 'U')
      currMenu = 1;
    if(currKey == 'D')
      currMenu = 3;
    return;
  }
  if(currMenu == 3)
  {
    getKey();
    currKey = mapKey(butnum);
    if(currKey == 'E')
    {
      if(smokeEnabled == 0)
        smokeEnabled = 1;
      else
        smokeEnabled = 0;
    }
    if(currKey == 'U')
      currMenu = 2;
    if(currKey == 'D')
      currMenu = 4;
    return;
  }
```

```
if(currMenu == 4)
{
  getKey();
  currKey = mapKey(butnum);
  if(currKey == 'E')
  {
    if(callEnabled == 0)
      callEnabled = 1;
    else
      callEnabled = 0;
  }
  if(currKey == 'U')
    currMenu = 3;
  if(currKey == 'D')
    currMenu = 5;
  return;
}
if(currMenu == 5)
{
  getKey();
  currKey = mapKey(butnum);
  if(currKey == 'E')
  {
    if(buzzEnabled == 0)
      buzzEnabled = 1;
    else
      buzzEnabled = 0;
  }
  if(currKey == 'U')
    currMenu = 4;
  if(currKey == 'D')
    currMenu = 6;
  return;
}
if(currMenu == 6)
{
  getKey();
  currKey = mapKey(butnum);
  if(currKey == 'E')
  {
    securityEnabled = -1;
  }
  if(currKey == 'U')
    currMenu = 5;
  if(currKey == 'D')
    currMenu = 7;
  return;
}
if(currMenu == 7)
{
  getKey();
  currKey = mapKey(butnum);
  if(currKey == '*')
  {
    if(buzzFreq > 1)
      buzzFreq--;
```

```
        }
        if(currKey == '#')
        {
          if(buzzFreq < 9)
            buzzFreq++;
        }
        if(currKey == 'U')
          currMenu = 6;
        if(currKey == 'D')
          currMenu = 1;
        return;
      }

    }
}

//************************************************
//decode button pressed
void getKey(void)
{

  //get lower nibble
  DDRD = 0x0f;
  PORTD = 0xf0;
  delay_us(5);
  key = PIND;

  //get uppwer nibble
  DDRD = 0xf0;
  PORTD = 0x0f;
  delay_us(5);
  key = key | PIND;

  //find matching keycode in keytbl
  if (key != 0xff) //so we got something!
  {
    for (butnum=0; butnum<16; butnum++)
      if(keytbl[butnum] == key)
        break;
    if(butnum == 16)
      butnum = 0;
    else
      butnum++; //adjust by one to make range 1-16
  }
  else
    butnum = 0;

  //button debounce scheme
  //if a button is still pressed
  if(prvbutnum != 0 && butnum != 0)
  {
    butnum = 0;
    return;
  }
  //if a button just pressed
  if(prvbutnum == 0 && butnum != 0)
```

```
    {
      prvbutnum = butnum;
      return;
    }
    //if a button just released
    if(prvbutnum != 0 && butnum == 0)
    {
      prvbutnum = 0;
      return;
    }

  }

  char mapKey(int myNum)
  {
    switch(myNum)
    {
      case 1:
        return '1';
      case 2:
        return '2';
      case 3:
        return '3';
      case 4:
        return 'U';
      case 5:
        return '4';
      case 6:
        return '5';
      case 7:
        return '6';
      case 8:
        return 'D';
      case 9:
        return '7';
      case 10:
        return '8';
      case 11:
        return '9';
      case 12:
        return 'S';
      case 13:
        return '*';
      case 14:
        return '0';
      case 15:
        return '#';
      case 16:
        return 'E';
      default:
        return '0';
    }
  }
```

```
//***********************************************
//Usual initialization stuff
void initialize(void)
{
  //PORTA: 1-2 input, 0,4-7 output;
  DDRA = 0x70;
  PORTA = PORTA | 0x40;
  //PORTC: LCD
  //PORTD: keypad
  DDRD = 0xff;
  PORTC = 0;

  //set up timer 0
  TIMSK = 0x02;
  OCR0 = 125;
  TCCR0=0b00001011; //prescaler 64, clear on match

  //init the task timers
  time1 = t1;
  time2 = t2;
  time3 = t3;
  time4 = t4;
  time5 = t5;

  //init ADC
  ADMUX = 0;
  ADMUX = ADMUX | 0xC0;
  adc_ch = 0;
  ADCSRA = 0xCF; //ADC on, /128, interrupt unmasked, and started

   //init lcd
  lcd_init(LCDwidth);       //initialize the display
  lcd_clear();              //clear the display

  //init variables

  securityEnabled = 0;
  motionEnabled = 1;
  smokeEnabled = 1;
  motionError = 0;
  smokeError = 0;
  callEnabled = 1;
  buzzEnabled = 1;
  buzzFreq = 7;
  for(i=0; i<4; i++)
  securityCode[i] = i+0x31;
  currMenu = 0;


  //crank up the ISRs
  #asm("sei")

}

void main(void)
```

```
{
  initialize();

  while(1)
  {
    if(time2 == 0)   task_keypad();
    if(time3 == 0)   task_led();
    if(time4 == 0)   task_call();
    if(time5 == 0)   task_lcd();
    if(time6 == 0)   task_buzz();
  }
}
```

Appendix C

Data Sheets

- ✓ **TF Smoke & Heat Detector.**
- ✓ **Atmega 32 Data Sheet.**
- ✓ **Visonic Programmable 2- Channel Speech Dialer.**
- ✓ **Keypad Data Sheet.**
- ✓ **16x2 Character LCD.**

TF Smoke & Heat Detector
INSTALLATION INSTRUCTIONS

Description

The TF base is a common base for Ionisation (TFI-330), Optical (TFO-440) smoke detector and Fix Temperature @ 65° C Heat Detector (TFH-220) operating on 12-28V DC supply. These professional smoke and Heat detectors are applicable at both Fire and Alarm systems.

Installation Instructions

- Before wiring the smoke detector, verify the wiring terminals sign
- When using a non-*EasyLoader* panel verify compatibility

Step One: Mounting

Remove the smoke base cover by inserting a tiny screwdriver (there are three clips) between the clips and base case to free the clips. Remove the base, attach to ceiling with two screws, enter the wires via the central round hole. Install one smoke per 20 M^2, at 3 meters ceiling high. Heat detectors install at one to two meters from the heat source, but all according to the installation design and engineer.

Step Two: Wiring

The detector drives -V (negative voltage) during alarm at the tag marked as LAMP, the (-) is the negative side. DC supply connects to LINE IN, the DC OUT is not used (connects the DC supply to next detector).

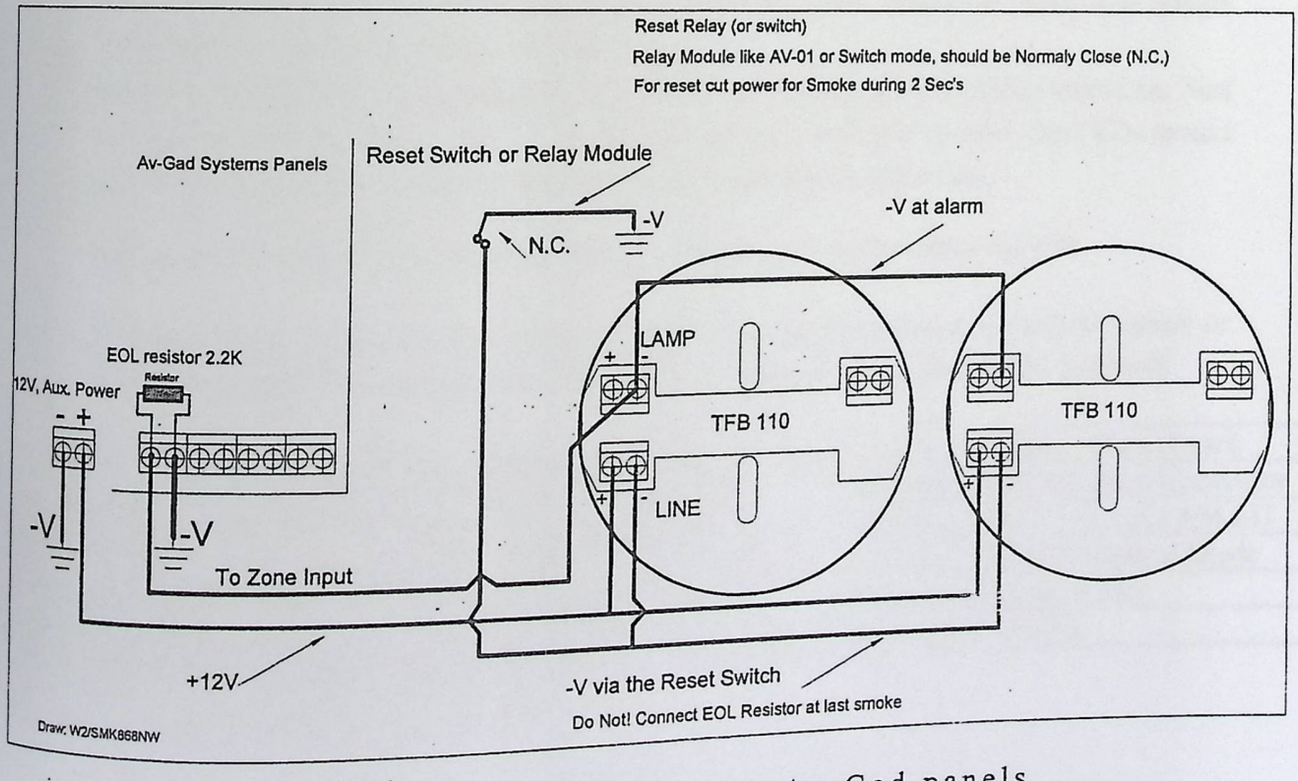


Figure 1: TF base smoke wiring to Av-Gad panels

Connect the EOL (end of line) resistor (2.2K-0.25W) at the last detector. It's recommended to connect no more than five detectors to each zone.
For RESET the smoke use a 12V relay (like AV01 module), with N.O. and N.C. contacts. Add relay coil surge protection.

Connect the relay common terminal to +12V Aux. Power, the N.C. contact to the smoke + Power input terminal. The relay coil is energized by applying –12V from the alarm panel SLO terminal and constant +12V, refer to Fig. 1 drawing. To reset the smoke interfaced to EasyLoader panels, hold-down key 9, or use a manual N.C. switch (replace the relay). During reset, the power to the smoke is disconnected for few seconds to clear alarm and rest the LED (to off).

Step Three: Control Panel Setting

Refer to the control panel manual:
Program the zone/s as N.O. and Fire type.
Program the SLO output to be a Smoke Reset output.
When alarm is ON, the smoke will sink -V at LAMP tag, this -V will trig zone to alarm.
To reset the detector apply -V to the AV-01 relay coil. Resetting with Av-Gad panels: Hold-down key '9' at keypad.
When the AV-2008ELT is used, Tamper alarm is available when the smoke head is removed.

Step Four: Testing

Verify that the wiring is correctly wired, attach the base cover by aligning the three clips, note there is **one wider clip,** when 3 clips meet push to insert. Rotate and insert the detector head, power-up the control panel; the zone status should be clear, use Smoke Tester Spray to test each connected detector.
Make sure all smokes are operative and after alarm test, the smoke LED is turned on. Test the reset by holding-down key '9', or the manual reset switch if so used, the LEDs should go off and zone cleared, reset the panel alarm by entering the user code.

Safety: Clean and test the smoke frequently. Never open or paint the head unit.

It's recommended to send the smokes to approved cleaning laboratory every two years or earlier if installation is at dusty site. **Do not try to open or clean the unit by yourself.**

TFB-110 base and Ionization Optic & Heat Smoke Specifications		
Operating Temp: -10 to +55 C°	DC Input: 12 to 28V DC	Activity (TFH220): Semi Conductor
Humidity: 80%	DC Current on standby: 40 uA	Activity (TFI330): 0.7 uCi AM 241
Wiring: Screw Wire Terminals	DC current on alarm: 100 mA	Activity (TFO440): Optical Diode
Color: Off White	Alarm type: Red LED	Weight with base: 0.2 kg
EMI and RFI Filter: Varistor	Area cover: 20 m^2 at 3m high	Size: 120 x 40 m"m

AV-GAD systems ltd.

AV-GAD Systems Ltd. Panarama House
84, Ben-Zvi Road Tel Aviv 68 104, Israel
P.O.B. 49 080, Tel Aviv 61 490, Israel
Tel: 972-3-681 67 67, Fax: 972-3-683 55 05

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 32K Bytes of In-System Self-programmable Flash program memory
  - 1024 Bytes EEPROM
  - 2K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels in TQFP Package Only
    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega32L
  - 4.5 - 5.5V for ATmega32
- Speed Grades
  - 0 - 8 MHz for ATmega32L
  - 0 - 16 MHz for ATmega32
- Power Consumption at 1 MHz, 3V, 25°C for ATmega32L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA

8-bit AVR® Microcontroller with 32K Bytes In-System Programmable Flash

ATmega32
ATmega32L

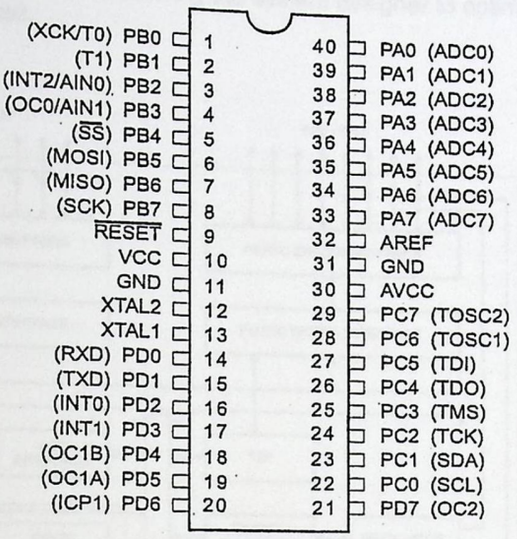
Summary

Note: Not Recommended for new designs.

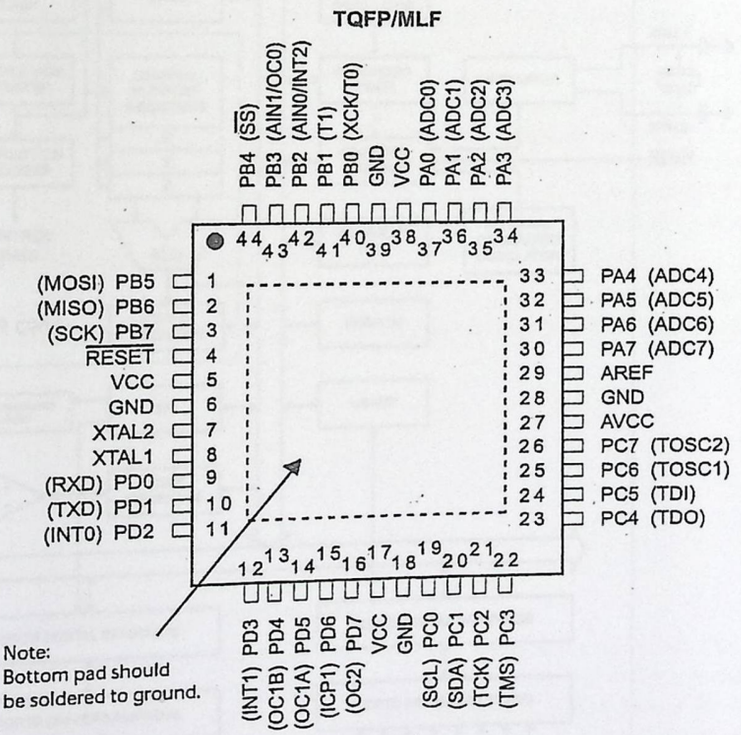
Pin Configurations

Figure 1. Pinout ATmega32

PDIP



TQFP/MLF

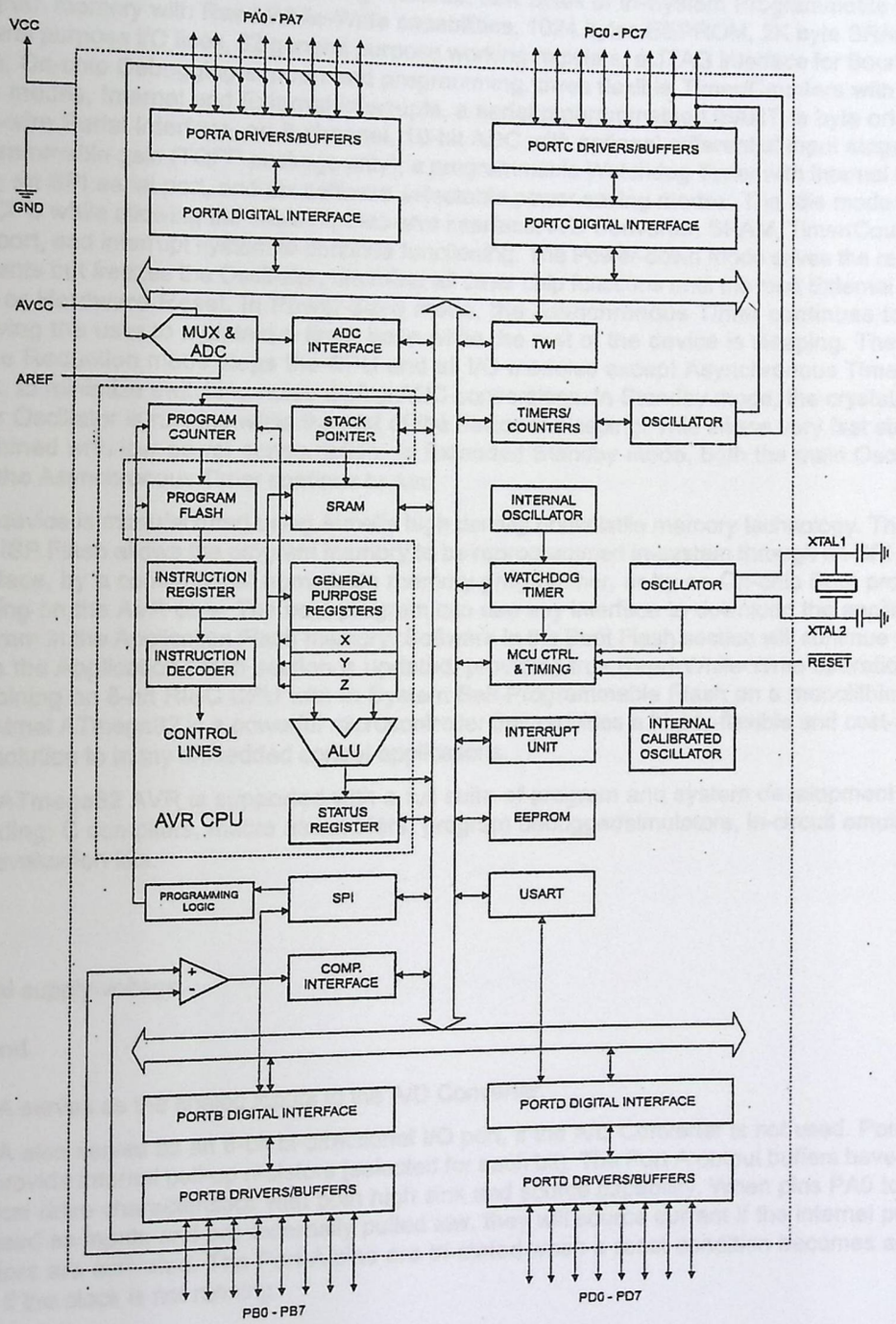


Overview

The ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega32 provides the following features: 32K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1024 bytes EEPROM, 2K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega32 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega32 as listed on page 57.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

The TD0 pin is tri-stated unless TAP states that shift out data are entered.

Port C also serves the functions of the JTAG interface and other special features of the ATmega32 as listed on page 60.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega32 as listed on page 62.

$\overline{\text{RESET}}$

Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 37. Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF

AREF is the analog reference pin for the A/D Converter.

Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	10
$3E ($5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	12
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
$3C ($5C)	OCR0	Timer/Counter0 Output Compare Register								82
$3B ($5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	47, 67
$3A ($5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	68
$39 ($59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	82, 112, 130
$38 ($58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	83, 112, 130
$37 ($57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	248
$36 ($56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	177
$35 ($55)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 66
$34 ($54)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	40, 67, 228
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	80
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								82
$31[1] ($51)[1]	OSCCAL	Oscillator Calibration Register								30
	OCDR	On-Chip Debug Register								224
$30 ($50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	56,85,131,198,218
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	107
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	110
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								111
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								111
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								111
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								111
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								111
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								111
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								111
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								111
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	125
$24 ($44)	TCNT2	Timer/Counter2 (8 Bits)								127
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								127
$22 ($42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	128
$21 ($41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	42
$20[2] ($40)[2]	UBRRH	URSEL	–	–	–	UBRR[11:8]				164
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	162
$1F ($3F)	EEARH	–	–	–	–	–	–	EEAR9	EEAR8	19
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								19
$1D ($3D)	EEDR	EEPROM Data Register								19
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	19
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	64
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	64
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
$0F ($2F)	SPDR	SPI Data Register								138
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	138
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	136
$0C ($2C)	UDR	USART I/O Data Register								159
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	160
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	161
$09 ($29)	UBRRL	USART Baud Rate Register Low Byte								164
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	199
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	214
$06 ($26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	216
$05 ($25)	ADCH	ADC Data Register High Byte								217
$04 ($24)	ADCL	ADC Data Register Low Byte								217
$03 ($23)	TWDR	Two-wire Serial Interface Data Register								179
$02 ($22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	179

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$01 ($21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	178
$00 ($20)	TWBR	Two-wire Serial Interface Bit Rate Register								177

Notes:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCDR Register.
2. Refer to the USART description for details on how to access UBRRH and UCSRC.
3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
4. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 – Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← Stack	None	4
RETI		Interrupt Return	PC ← Stack	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	If (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	If (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	If (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	If (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	If (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	If (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	If (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	If (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	If (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	If (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	If (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	If (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	If (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	If (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	If (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	If (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	If (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	If (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	If (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	If (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	If (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	If (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	If (V = 0) then PC ← PC + k + 1	None	1 / 2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch If Interrupt Enabled	If (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch If Interrupt Disabled	If (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	Stack ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← Stack	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-Chip Debug Only	None	N/A

DL-125C, DL-125CA

Programmable 2-Channel Speech Dialer

Visonic®

Installation Instructions

1. INTRODUCTION

1.1 General description

The DL-125C and DL-125CA are automatic programmable speech dialer designed for verbal reporting of events.

Features

- 2 alarm inputs (Z-1 & Z-2).
- When each input is triggered, a specific prerecorded verbal message is reported to 4 remote telephones.
- When an event occurs, it can be reported to different 4 telephones or (if both inputs are connected in parallel) to different 8 telephones.
- The target telephone numbers may be frequently reprogrammed by the user.
- If the telephone line is disconnected, the two output terminals LF (solid state switch) are shorted or open (programmable). These terminals can be used to send a signal to the alarm system for telephone line failure indication.
- The alarm logic of each input can be programmed by the installer (activation when alarm input is "open" or "closed").
- Selectable Pulse or DTMF dialing method.
- Certain functions can be performed in response to DTMF control commands received from remote telephones.
- Whenever a message is acknowledged by the called party, a highly sensitive microphone is activated, to allow the called party to listen and hear sounds in the installation site.
- The "listening-in" period is limited in time, but the called party can send a specific DTMF command to prolong it.
- Programmed data is retained in an EEPROM, unaffected by power failures.
- A communication session with the first / second group of telephones is initiated by triggering alarm inputs Z-1 / Z-2, or by pressing AL-1 / AL-2 on the front panel, respectively.
- Powered by an external source and can be backed up by a rechargeable battery.

Dialer Types

Type	Dialing Can be Stopped by:	
	Pressing STOP button	Manual or Automatic Power Disconnection
DL-125C	YES (*)	YES
DL-125CA	NO (**)	YES

* Provided that momentary alarm contacts are used.

** This feature is sometimes required by regulatory authorities.

1.2 Applications

- Upgrading alarm control panels that do not have a dialer. Two different events can be reported to remote telephones.
- Stand-alone 2-input 24-hour alarm system, triggered directly by a smoke/shock detectors or a panic button (loop response time 200 ms).
- Looking after infants or old, sick and disabled people. The dialer delivers a distress message and then allows the called party to "listen in".
- Supervising unattended technical devices or processes, with verbal reporting of equipment failures or process anomalies.
- Transmitting numeric reports to numeric pagers or voice messages to voice pagers.

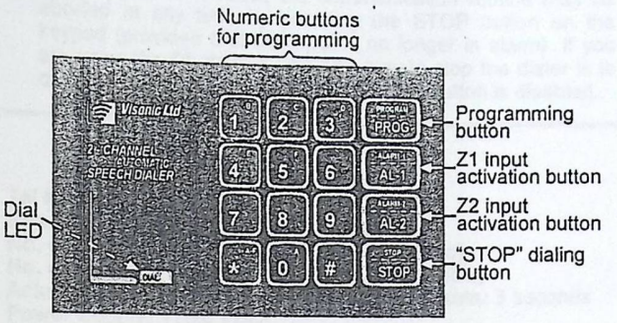


Figure 1. Front Panel

1.3 Message Structure

The overall length of the speech message that can be recorded is limited to 20 seconds (see par. 5,4) . Within this limit, the message can be composed of two pre-recorded segments:

- The identification segment, common to both alarm inputs. This segment usually identifies the user or the protected premises.
- The alarm type segment, associated with a specific alarm input. This segment is used to describe the type of event reported ("fire", "intrusion", "panic", etc.).

A transmission initiated by a specific event (one of the two alarms) is composed of the **identification** segment and one of two **alarm type** segments. The order of transmission of the two segments can be selected. For instance, you can select: "The Smith residence, 25 Scarecrow Drive – Fire Alarm", or you can select: "Fire Alarm – the Smith residence, 25 Scarecrow Drive".

1.4 Communication Routine

Note: In this section, location numbers identify "memory cells" that retain programmed parameters (see Para. 5.8).

Once triggered into action, the dialer introduces a programmed pre-dialing pause (see Location 14 in Para. 5.8). Then it disconnects the local telephone set and engages the telephone line. The DIAL LED lights and the process continues as follows:

A. The dialer starts dialing if uninterrupted tone is detected for 2 seconds (see C below). If 5 seconds elapse with no dial tone - the dialer disengages the line, waits 5 seconds and tries again. If another 5 seconds go by without dial tone, the dialing procedure starts anyway (see B below).

B. The dialer checks whether a letter is programmed as a prefix to the first telephone number. Letter prefixes impose an additional delay before dialing (see Para. 5.2). The dialer introduces the required delay (if any) and then starts dialing.

C. The dialer dials the programmed number. During dialing, the LED either remains lighted (DTMF dialing) or flashes (pulse dialing), depending on the dialing method selected. After dialing, the dialer pauses for 5 seconds and transmits the message prepared for the called party associated with the input that had been triggered.

D. The dialer now waits 3 seconds for the called party to acknowledge (the acknowledge signal is DTMF "1").

E. Upon receiving the acknowledge signal, the dialer removes the presently contacted telephone from its task list for the current event. If the "listen in" function is permitted (see Location 10 in Para. 5.8) it will continue as in Paragraphs F and G below. If not, the dialer will go "on hook" and proceed to dial the next number.

Note: Without an acknowledgement, the message will be repeated until the maximum number of message repeats is reached (see Location 20 in Para. 5.8). The dialer will call the remaining numbers and will then repeatedly retry the number that didn't acknowledge, until the maximum number of dialing attempts is reached (see Locations 12 and 13).

F. After acknowledgement, the dialer enables the "listen in" function for a preprogrammed period.

G. At the end of the listen-in period, a short beep sounds. If the called party keys "1" within 10 seconds, a new listen-in period begins. Otherwise", the dialer will go "on hook". The listen-in period may be prolonged as many times as necessary or terminated at any time by keying "9" twice in succession.

H. Upon conclusion of the communication session with the first telephone, the procedure in A through G above will be repeated for all remaining telephone numbers in the relevant group (provided that the "non-backup mode" has been selected in Location 24).

Note: Location 24 allows selection of "backup" or "non-backup" mode. In the backup mode, acknowledgement from one telephone is enough to close the event. In the "non-backup" mode, acknowledgement must be obtained from all telephones in the group.

I. Once the entire communication cycle is concluded, the dialer disengages the line and reverts to the standby state. If you are using the DL-125C, the communication routine may be aborted at any time by pressing the STOP button on the keypad (provided that the input is no longer in alarm). If you are using the **DL-125CA**, the only way to stop the dialer is to disconnect the power, because the STOP button is disabled.

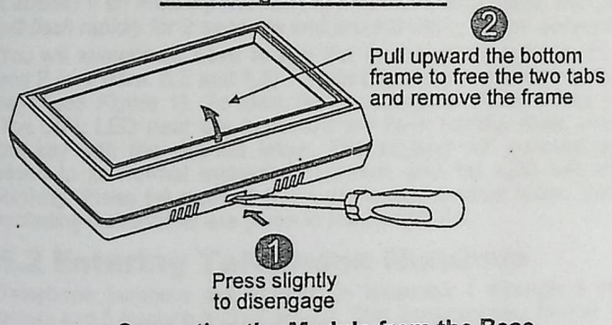
2. SPECIFICATIONS

Input Circuits: Two N.O. or N.C. inputs (programmable)
LF Alarm Output Type: Solid-state relay, N.O. or N.C (programmable), up to 100 mA / 30 V, ~30 Ω internal resistance. (note: this output is comprised of two terminals that have no polarity)
Alarm Logic: Alarm upon circuit closure or upon circuit opening (programmable)
Dialing Method: Pulse or DTMF (programmable)
Tel. Line Impedance: 600Ω, or customized to meet local requirements in country of use.
Reporting Destinations: Two groups of telephone numbers, 4 telephones in each group. Reporting to one pager requires the memory space dedicated to two telephone numbers.
Tel No. Length: 20 digits maximum.
Speech Message Duration: 20 seconds max.
No. of Dialing Attempts: 1 – 15 (programmable)
No. of Message Repeats: 1 – 255 (programmable)
Acknowledge Pause between Message Repeats: 3 seconds
Power Supply: 11-28 VDC
Maximum Current Drain: 20 mA (standby), 105 mA (operation)
Operating Temperatures: 0°C to 50°C (32°F to 122°F)
Size: 150 x 105 x 35 mm (5-7/8 x 4-1/8 x 1-3/8 in.)
Weight: 235 g (8.3 oz)
Standards: Compatible with RTTE requirements - Directive 1999/5/EC of the European Parliament and of the Council of 9 March 1999. EN50131 Grade 2 Class II.

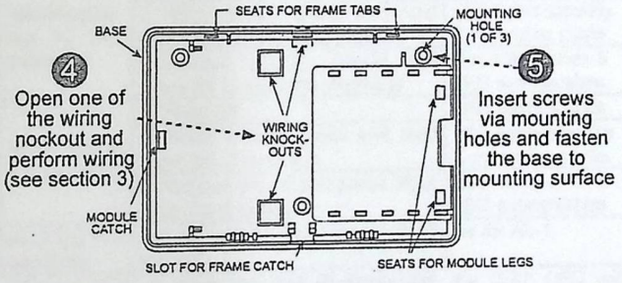
3. MOUNTING

The dialer may be installed as a stand-alone unit or within the housing of a host system such as an alarm control panel.

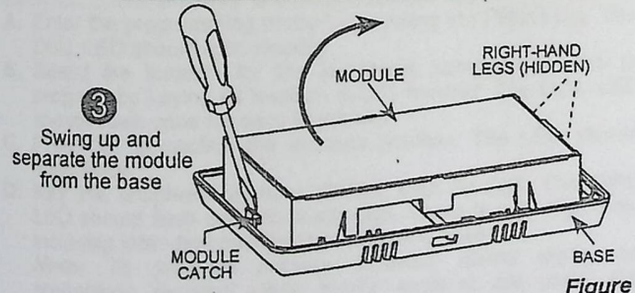
Removing the Front Panel



Securing the Base to a Wall or Host System Cabinet



Separating the Module from the Base



Remounting the Module

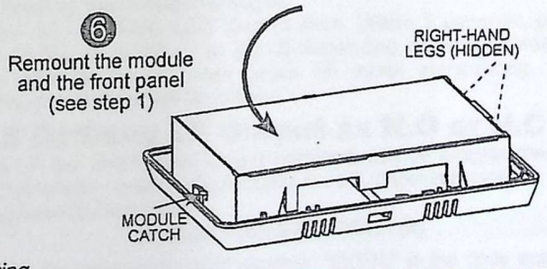


Figure 2. Mounting

4. Wiring

The dialer can be powered by constant power supply or by a switched power supply. When the dialer receives switched power from an alarm system, it will be disabled upon stopping the alarm, since the alarm relay cuts off the power. This type of wiring is ideal when the dialer is mounted in a locked box that prevents access to the STOP pushbutton.

Note: Both Z-1 and Z-2 inputs can be programmed as N.O. (normally open) or N.C. (normally closed) inputs (see Locations 22 & 23 in Para. 5.8). With N.O. inputs selected, a short circuit across the relevant input will activate the dialer.

With N.C. inputs selected, an open circuit across the relevant input will activate the dialer.

Use 15 AWG or larger conductor to connect the EARTH terminal to the nearest electrical ground, preferably a ground rod.

Failure to earth the unit compromises safety!

The phone connected to the **SET** terminals will be automatically disconnected from the line whenever the dialer goes into action.

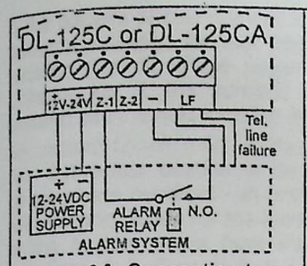


Figure 3A. Connection to an Alarm System with Constant Power Supply (Manual Dialing Stop)

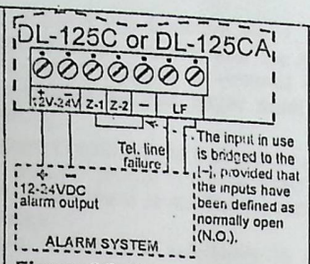


Figure 3B. Connection to an Alarm System's Bell/Alarm Output

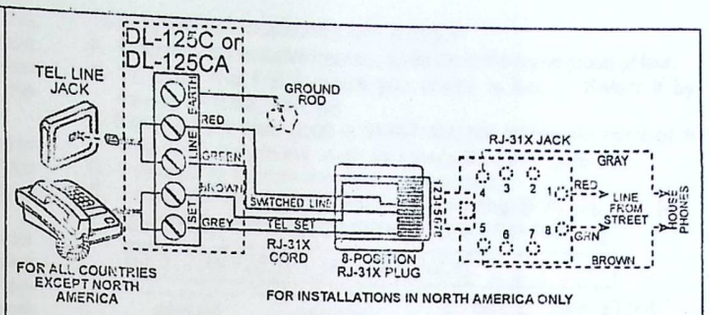


Figure 4. Telephone Wiring

5. PROGRAMMING

5.1 Programming Fundamentals

The dialer employs a non-volatile EEPROM, that stores programmed data and keeps it intact even during power failures. Programming is carried out from the keypad by entering the desired variables or by setting logic flags. Every variable is programmed into a specific location in the memory, and each location is identified by a LOCATION NUMBER. A set of default parameters is programmed at the factory and saved in the **EEPROM,** but you may change these at will (see Para. 5.8). The programming format consists of the following successive entries:

[PROG] <LOC> [#] <VAL> [#]

[PROG] and **[#]** are keys provided on the keypad. **[PROG]** starts the programming sequence and **[#]** confirms the preceding entry.

<LOC> is the location number. A leading zero may be ignored, so Location 06 may be entered as a single digit – **<6>.**

<VAL> is the value or code entered into the selected location.

Refer to the PROGRAMMING CHART (Para. 5.8) for a full list of locations, permissible entries and function details.

Caution! *If an invalid parameter is entered at any stage, the LED will flash rapidly for 2 seconds and programming will be aborted.*

You will sometimes have to key the hexadecimal digits B, C, D and E (see Para. 5.2 and 5.6). These digits are marked on certain keys (see Figure 1). To start the hexadecimal mode, press **[*]**. The DIAL LED near the keyboard will flash rapidly. Next, press the key with the desired letter. The keypad will automatically revert to its normal numerical function, and the LED will stop flashing. Press **[*]** again if you wish to key another letter. Voice recording instructions are given in Paragraph 5.4.

5.2 Entering Telephone Numbers

Telephone numbers are entered in locations 1 through 4 (1st group) and 5 through 8 (2nd group). The programming format is:

[PROG] <LOC> [#] <NUM.> [#]

A. Enter the programming mode by pressing the PROG key. The DIAL LED should light steadily.

B. Select the location for the telephone number you wish to program by keying its location (LOC) number. The DIAL LED should flash once for each keystroke.

C. Press [#] to confirm the location number. The LED should flash twice.

D. Key the telephone number (NUM), digit by digit. The DIAL LED should flash once for each digit. There is a 20-digit limit, including inter-digit pauses (see following note).
Note: *To program pauses between dialed digits, as sometimes required when PABX systems are used, the following entries are available:*

Code Letter	*Key Strokes*	*Resultant Function*
B	*[*][1]*	*wait 5 seconds or wait for dial tone, whichever comes first, and continue dialing.*
C	*[*][2]*	*wait 10 seconds and continue dialing.*
D	*[*][3]*	*wait 5 seconds for dial tone and disengage the line if none is received.*

After pressing [*], the LED indicator flashes until a letter key is pressed.

E. Having entered the last digit, finish off by keying [#] .The DIAL LED indicator will extinguish.

F. To program another telephone number, repeat the procedure outlined in steps A to E above.

5.3 Deleting Telephone Numbers

A telephone number location will "blank out" if you go through the programming process as in B above but skip the telephone number. The deleting format is therefore:

[PROG] <LOC> [#] [#]

Note: *The number already programmed into any location between 1 and 8 may be verified by using the following format:*

[PROG] <LOC> [*]

This initiates a communication session with the particular telephone, and provides a chance to verify correctness of the programmed phone number.

5.4 Recording and Erasing

A. Recording Procedure

Message Segment	Actions Required	Response
Identification (up to 14.5 seconds)	1. Press [#]	LED flashes once.
	2. Within 2 sec., press and hold down [AL-1]+[AL-2] and talk	LED lights steadily & recording starts.
	3. Release [AL-1]+ [AL-2] and press [#] to save the message	Recording ends & LED extinguishes
AL-1 (up to 2.5 seconds)	1. Press [#]	LED flashes once.
	2. Within 2 sec., press and hold down [AL-1] and talk	LED lights steadily & recording starts.
	3. Release [AL-1] and press [#] to save the message	Recording ends & LED extinguishes
AL-2 (up to 2 seconds)	As for AL-1, but press AL-2 instead	As for AL-1

Please note: *If recording time is exceeded, the DIAL LED will flash rapidly*

B. Erasing Recorded Messages

Press [#] - the DIAL LED flashes once. Within 2 seconds, press [AL-1]+[AL-2], or [AL-1] or [AL-2] depending on which message you wish to erase. Then press [#] again immediately. The previous message will be erased.

5.5 Defining LF Output as N.O or N.C

The LF (tel. line failure) output terminals can be programmed as N.O (Normally open, default state) or N.C. (Normally closed). The programming format is:

[PROG] <16> [#] <CODE> [#]

"16" is the memory location number. "CODE" is the code entered into location 16; "0" for N.O., "1" for N.C.

5.6 Dealing with Pagers

You may program the dialer to dial a pager's phone number and send a numeric or verbal message. Communication with a single pager requires two consecutive locations in the dialer memory - one for the pager's phone number and another for the numeric data sent to the pager. Since each dialer input has 4 memory locations for phone numbers, each input can report to one pager and 2 regular telephones or just two pagers.

If the pager's phone number is entered into Location No. 1, the numeric message for that pager must be entered into the next location (No. 2). If the pager's phone number is entered into Location No. 2, the numeric message for that pager must be entered into the next location (No. 3).

Important! Location No. 4 (the last location in the first group) and No. 8 (the last location in the second group) can't be used for pager phone numbers - in both cases there is no "next location" with memory space for the message.

In the backup mode (see Para. 1.4H), it is advisable to use Locations 1 & 2 or 5 & 6 for pager data and the remaining locations for telephone numbers. The dialer will therefore call the pager first and, since the pager doesn't acknowledge, at least one of the telephone numbers will also be called.

Numeric pagers accept both the subscriber ID (PIN number) and a numeric message which is registered and forwarded to the subscriber. Voice pagers accept the subscriber ID (PIN number), record a verbal message and relay it to the subscriber.

***Note:** Some pagers have a special phone number assigned to each specific subscriber. This type of pager does not require a PIN number.*

Some pagers require an asterisk (*) as a separator between the subscriber code and the message. Other pagers require the pound symbol (#). Correct programming is totally dependent on your ability to make the dialer "talk" to the paging company's computer in a language it "understands" (Fig. 5).

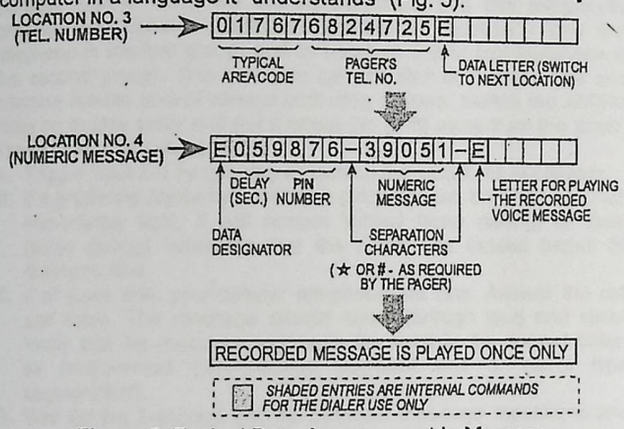


Figure 5. Typical Data Arrangement in Memory

Contact the paging company for specific interface details (**PIN** or no **PIN**, separating characters required etc.).

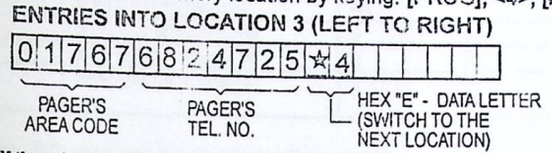
Figure 5 depicts a typical arrangement of data in the dialer memory for reporting an event to a pager. In this example, the phone number is entered into memory location No. 3, and the numeric message is entered into the memory location No. 4.

A hexadecimal E at the end of the phone number tells the dialer move to the next memory location and send the data stored in it.

A hexadecimal E at the beginning of a memory location identifies the contents of this location as pager data. A hexadecimal E after the data serves as a cue for playing the voice message.

To program communication with a pager:

A. Reserve 2 consecutive memory locations in the same group of four.
B. Suppose the first location you chose is No. 3. Select it by keying [PROG], <3>, [#] .
C. Suppose the area code is 01767 and the telephone number is 682-4725. Key in the data as exemplified in Figure 6.
D. Press [#] to confirm the data just entered.
E. Select the next memory location by keying: [PROG], <4>, [#]



[*][4] produces a Hexadecimal E, a cue for data in the next memory location.

***Figure 6.** Programming the Pager's 1st Location*

F. Suppose the target pager requires a 5 second interval between the end of dialing and the beginning of the message. Also suppose that the PIN number is 9876 and the message is **39051**. Key the data as shown in Fig.7.

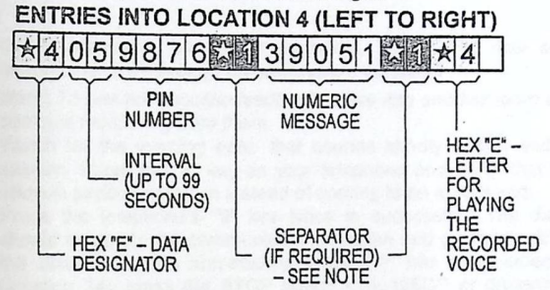


***Figure 7.** Programming the Pager's 2nd Location*

***Notes:** If the required separator is # - key [*][1]*

*If the required separator is * - key [*][2].*

In case of a pager that does not forward voice messages - skip the last [*][4].

If you need a 5-second delay anywhere within the message, key [*] [3].

G. Press [#] to confirm the data just entered.

5.7. Programming Summary

The dialer can be programmed equally well in the installer's office or at the installation site. Follow the PROGRAMMING CHART (Para. 5.8) row by row, and enter the appropriate variables. The code options are explained in the second column, and the full programming sequence for each variable is given in the fourth column. Each programming step is enclosed in square brackets, and may include more than one keystroke. The fifth column shows the default values, and the last column (Prog) is left blank for you to fill in your own programmed values.

***Note:** If no key is pressed for 30 seconds, programming will be aborted and the selected location will revert to the previously saved value.*

To quit programming at any stage, press the [STOP] key.

5.8 Programming Chart

Loc. No.	Description of Parameters & Code Options	Entry Limits	Programming Format	Factory Default	Prog. Record
1	1st telephone number associated with input Z-1	20 digits	[PR] [1] [#] [Num] [#]**	None	
2	2nd telephone number associated with input Z-1	20 digits	[PR] [2] [#] [Num] [#]**	None	
3	3rd telephone number associated with input Z-1	20 digits	[PR] [3] [#] [Num] [#]**	None	
4	4th telephone number associated with input Z-1	20 digits	[PR] [4] [#] [Num] [#]**	None	
5	1st telephone number associated with input Z-2	20 digits	[PR] [5] [#] [Num] [#]**	None	
6	2nd telephone number associated with input Z-2	20 digits	[PR] [6] [#] [Num] [#]**	None	
7	3rd telephone number associated with input Z-2	20 digits	[PR] 7 [#] [Num] [#]**	None	
8	4th telephone number associated with input Z-2	20 digits	[PR] [8] [#] [Num] [#]**	None	
9	Inaccessible to installers or users	–			
10	Inhibit or permit the listen-in function: **0 - inhibited; 1 - permitted**	0 or 1	[PR] [10] [#] [Code] [#]	1	
11	Select Dialing method: **0 - DTMF; 1 - Pulse**	0 or 1	[PR] [11] [#] [Code] [#]	0	
12	No. of dialing attempts for alarms at input Z-1	1 - 15*	[PR] [12] [#] [Num] [#]	4	
13	No. of dialing attempts for alarms at input Z-2	1 - 15*	[PR] [13] [#] [Num] [#]	4	
14	Delay (in seconds) between trigger and action (to permit the user to clear a false alarm)	1 - 255*	[PR] [14] [#] [Sec] [#]	3	

DE5817

Loc. No.	Description of Parameters & Code Options	Entry Limits	Programming Format	Factory Default	Prog. Record
15	Order of transmission of message segments: **0 - alarm type segment first; 1 - identification segment first**	0 or 1	[PR] [15] [#] [Code] [#]	1	
16	LF output logic: **0 - N.C.; 1 - N.O.**	0 or 1	[PR] [16] [#] [Num] [#]	1	
17-19	Inaccessible to installers or users	–			
20	Number of recorded message repeats	1 - 255*	[PR] [20] [#] [Num] [#]	4	
21	Listen-in duration (in seconds)	1 - 255*	[PR] [21] [#] [Sec] [#]	60	
22	Z-1 input definition (Z-1 logic): **0 - N.O.; 1 - N.C.**	0 or 1	[PR] [22] [#] [Code] [#]	0	
23	Z-2 input definition (Z-2 logic): **0 - N.O.; 1 - N.C.**	0 or 1	[PR] [23] [#] [Code] [#]	0	
24	Selection of Backup or Non-backup reporting method: **0 - non-backup; 1 - backup (see note)**	0 or 1	[PR] [24] [#] [Code] [#]	1	

* The "00" value is illegal in this memory location

** When programming a 20-digit number, the LED will go off by itself after the 20th digit and the number will be saved.

Note: In the Backup reporting mode, receiving an acknowledge signal from a single telephone in a group of 4 is sufficient to consider the current event closed and call off the communication session. The remaining 3 telephones are there for backup purposes only. In the Non-Backup mode, an acknowledge signal must be received from each telephone in the group of 4 before the current event is considered reported and closed.

6. TEST

After installation, programming, and message recording, correct function should be verified.

Testing can be made easier if you possess a cellular telephone and a portable AM/FM radio. For testing purposes, you can temporarily program your cellular telephone's number in Location 1 (the first telephone in the first group) and in Location 5 (the first telephone in the second group). This way you can monitor both messages and exercise remote control without bothering anyone. Switch the AM/FM radio on to play softly and put it about 2m (6 ft) away from the dialer. Then proceed as follows:

A. Trigger input Z-1 by opening or closing the circuit, as necessary.
B. If a predialing pause has not been programmed, the DIAL LED will immediately light. It will remain lighted (tone dialing) or flash (pulse dialing) indicating that the dialer has indeed begun its dialing routine.
C. If all goes well, your cellular telephone will ring. Answer the call and listen. The message should come through loud and clear. Verify that the message segments are read in the correct order, as programmed (identification segment first or alarm type segment first).
D. Wait for the 3-second interval between message repetitions and press the "**1**" key on your telephone. After that, the message should not be repeated any more.
E. If the listening-in function is permitted, you should now start hearing the radio through the telephone's earpiece.

   Note: *To prevent acoustic feedback, move into another room and continue monitoring from there.*
F. Watch for the warning beep that sounds shortly before end of session. Press the "**1**" key on your telephone and verify that the listen-in period carries on instead of coming to an abrupt end.
G. Press the telephone's "**9**" key twice in succession. The dialer should terminate the communication session and go "on hook". If the dialer is in the non-backup mode ("0" has been selected Location 24), press the **STOP** button ["DL-125C") or disconnect the power ("DL-125CA") to prevent further dialing.
H. Reconnect the power (DL-125CA only) and trigger input Z-2 by opening or closing the circuit, as necessary.
I. Repeat steps B through G above for this input too. If all goes well, reprogram the telephone numbers in Locations 1 and 5 as requested by the user.

Note: *Testing can be carried out without a cellular telephone, provided that you secure cooperation of the called parties. You must warn them in advance that you are about to test the system, explain briefly what they have to do and have them report to you later whether all went well.*

7. SPECIAL NOTES

FCC Requirements

1. The Federal Communications Commission (FCC) has established Rules which permit this device to be directly connected to the telephone network. Standardized jacks are used for these connections. This equipment should not be used on party lines or coin lines.
2. If this device is malfunctioning, it may be causing also harm to the telephone network; this device should be disconnected until the source of the problem can be determined, and until repair has been made. If this is not done, the telephone company may temporarily disconnect service.
3. The telephone company may make changes in its technical operations and procedures; if such changes affect the compatibility or use of this device, the telephone company is required to give adequate notice of the changes.
4. If the telephone company requests information on what equipment is connected to their lines, inform them of:
   (a) The telephone number that this unit is connected to,
   (b) The ringer equivalence number (0.0B)
   (c) The USOC jack required (RJ-31X), and
   (d) The FCC registration number

Items (b) and (d) are indicated on the label. The ringer equivalence number (REN) is used to determine how many devices can be connected to your telephone line. In most areas, the sum of the RENs of all devices on any one line should not exceed five (5.0). If too many devices are attached, they may not ring properly.

APPENDIX A. USER INFORMATION

A.1 User Guidance

We recommend to photo-copy this section for all the dialer users - the proprietor of the protected premises and all called parties.

If programmed correctly, the dialer will operate automatically without user's intervention. However, the user can initiate an alarm or stop operation manually.

- ALARM-1 / AL-1 — Pressing AL-1 will cause the dialer to call the first group of phone numbers and send them the relevant verbal message.
- ALARM-2 / AL-2 — Pressing AL-2 will cause the dialer to call the second group of phone numbers and send them the relevant verbal message.
- STOP / STOP — Pressing STOP (DL-125C only) will cause the dialer to stop communicating, disengage the line and check both inputs. If an input is "in alarm", a new communication session will start. If both inputs are "normal", the dialer will standby. Note: This function is disabled in the DL-125CA.

While listening to the incoming verbal message, the called party can exercise some control over the dialer by sending DTMF (touchtone) codes over the telephone line:

- 1 — Serves as an acknowledgement. The dialer will stop sending the message and will permit "listening in". Before the listen-in period ends, the dialer beeps once. Pressing [1] again will start another listen-in period.
- 9 9 — Serves as an acknowledgement and causes the dialer to stop sending the message and go "on hook". The dialer will then call the remaining numbers (if programmed to do so).

A.2 Data Record

AL-1 DATA	AL-2 DATA
Message	Message .....
.....	.....
1st called party	1st called party .....
2nd called party	2nd called party .....
3rd called party	3rd called party .....
4th called party	4th called party .....

Visonic Ltd

Declaration of Conformity

In Accordance with R & TTE Directive of 1999/5/EC

We, the undersigned,
Company Visonic Ltd
Address: 24, Habarzel Street. , Tel-Aviv 61220
Country: Israel
Telephone number. +972 3 6456789
Fax number: +972 3 6456788

certify and declare under our sole responsibility that the following equipment

Type	Product description / Supplementary info
DL 125 C	Programmable 2-Channel Speech Dialer

is tested and conforms with the following standards:

Standard
Telephony:
TBR21: Terminal Equipment (TE); Attachment requirements for pan-European approval for connection to the analogue Public Switched Telephone Networks (PSTNs) of TE
Safety Of Information Technology:
EN 60950+ Am1(93), Am2(93), Am3(95), Am4(97). Safety of Information technology equipment including electrical business equipment

and therefore complies with the essential requirements and provisions of the Directive 1999/5/EC of the European Parliament and of the council of 9 march 1999 on Radio equipment and Telecommunications Terminal Equipment and the mutual recognition of their conformity and Annex III (Conformity Assessment procedure referred to in article 10(4)).

The following Notified Bodies and/or independent laboratories have been consulted in the Conformity Assessment procedure:

Type	Number	Name and address
A2LA Certified Lab		ITL, 26 Hacharoshet St. P.O.Box 211 , Or Yehuda 60251, Israel

The technical documentation as required by the Conformity Assessment procedure is kept at the following address.

Company: Visonic Ltd,
Address: Fraser Road
Priory Business Park
Bedford. MK44 3WH
Country: UK
Telephone number: 0870 7300800
Fax number: 0870 7300801

VISONIC LTD. PO BOX 22020 TEL-AVIV 61220 ISRAEL

Drawn up in:

Tel-Aviv, Israel	on	30/05/2004		Avi Shachrai	
(place)		(dd-mm-yy)	(signature)	(name and function)	(company stamp)

Visonic Ltd 24, Habarzel Street. , Tel-Aviv 61220,Israel Fax:+972 3 6456788 Phone:+972 3 6456789

WARRANTY

Visonic Limited (the "Manufacturer") warrants this product only (the "Product") to the original purchaser only (the "Purchaser") against defective workmanship and materials under normal use of the Product for a period of twelve (12) months from the date of shipment by the Manufacturer.
This Warranty is absolutely conditional upon the Product having been properly installed, maintained and operated under conditions of normal use in accordance with the Manufacturers recommended installation and operation instructions. Products which have become defective for any other reason, according to the Manufacturers discretion, such as improper installation, failure to follow recommended installation and operational instructions, neglect, willful damage, misuse or vandalism, accidental damage, alteration or tampering, or repair by anyone other than the manufacturer, are not covered by this Warranty.
The Manufacturer does not represent that this Product may not be compromised and/or circumvented or that the Product will prevent any death and/or personal injury and/or damage to property resulting from burglary, robbery, fire or otherwise, or that the Product will in all cases provide adequate warning or protection. The Product, properly installed and maintained, only reduces the risk of such events without warning and it is not a guarantee or insurance that such events will not occur.
THIS WARRANTY IS EXCLUSIVE AND EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, OBLIGATIONS OR LIABILITIES, WHETHER WRITTEN, ORAL, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR OTHERWISE. IN NO CASE SHALL THE MANUFACTURER BE LIABLE TO ANYONE FOR ANY CONSEQUENTIAL OR INCIDENTAL DAMAGES FOR BREACH OF THIS WARRANTY OR ANY OTHER WARRANTIES WHATSOEVER, AS AFORESAID.
THE MANUFACTURER SHALL IN NO EVENT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES OR FOR LOSS, DAMAGE, OR EXPENSE, INCLUDING LOSS OF USE, PROFITS, REVENUE, OR GOODWILL, DIRECTLY OR INDIRECTLY ARISING FROM PURCHASER'S USE OR INABILITY TO USE THE PRODUCT, OR FOR LOSS OR DESTRUCTION OF OTHER PROPERTY OR FROM ANY OTHER CAUSE, EVEN IF MANUFACTURER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
THE MANUFACTURER SHALL HAVE NO LIABILITY FOR ANY DEATH, PERSONAL AND/OR BODILY INJURY AND/OR DAMAGE TO PROPERTY OR OTHER LOSS WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR OTHERWISE, BASED ON A CLAIM THAT THE PRODUCT FAILED TO FUNCTION.
However, if the Manufacturer is held liable, whether directly or indirectly, for any loss or damage arising under this limited warranty, THE MANUFACTURER'S MAXIMUM LIABILITY (IF ANY) SHALL NOT IN ANY CASE EXCEED THE PURCHASE PRICE OF THE PRODUCT, which shall be fixed as liquidated damages and not as a penalty, and shall be the complete and exclusive remedy against the Manufacturer.
When accepting the delivery of the Product, the Purchaser agrees to the said conditions of sale and warranty and he recognizes having been informed of.
Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so these limitations may not apply under certain circumstances.
The Manufacturer shall be under no liability whatsoever arising out of the corruption and/or malfunctioning of any telecommunication or electronic equipment or any programs.
The Manufacturers obligations under this Warranty are limited solely to repair and/or replace at the Manufacturer's discretion any Product or part thereof that may prove defective. Any repair and/or replacement shall not extend the original Warranty period. The Manufacturer shall not be responsible for dismantling and/or reinstallation costs. To exercise this Warranty the Product must be returned to the Manufacturer freight pre-paid and insured. All freight and insurance costs are the responsibility of the Purchaser and are not included in this Warranty.
This warranty shall not be modified, varied or extended, and the Manufacturer does not authorize any person to act on its behalf in the modification, variation or extension of this warranty. This warranty shall apply to the Product only. All products, accessories or attachments of others used in conjunction with the Product, including batteries, shall be covered solely by their own warranty, if any. The Manufacturer shall not be liable for any damage or loss whatsoever, whether directly, indirectly, incidentally, consequentially or otherwise, caused by the malfunction of the Product due to products, accessories, or attachments of others, including batteries, used in conjunction with the Products. This Warranty is exclusive to the original Purchaser and is not assignable.
This Warranty is in addition to and does not affect your legal rights. Any provision in this warranty which is contrary to the Law in the state or country were the Product is supplied shall not apply.
Warning: The user must follow the Manufacturer's installation and operational instructions including testing the Product and its whole system at least once a week and to take all necessary precautions for his/her safety and the protection of his/her property.

1/08

W.E.E.E. Product Recycling Declaration
For information regarding the recycling of this product you must contact the company from which you orignially purchased it. If you are discarding this product and not returning it for repair then you must ensure that it is returned as identified by your supplier. This product is not to be thrown away with everyday waste.
Directive 2002/96/EC Waste Electrical and Electronic Equipment.

Visonic®

VISONIC LTD . (ISRAEL): P.O.B 22020 TEL-AVIV 61220 ISRAEL. PHONE: (972-3) 645-6789, FAX: (972-3) 645-6788
VISONIC INC. (U.S.A.): 65 WEST DUDLEY TOWN ROAD, BLOOMFIELD CT. 06002-1376. PHONE: (860) 243-0833, (800) 223-0020 FAX: (860) 242-8094
VISONIC LTD. (UK): 7 COPPERHOUSE COURT, CALDECOTTE, MILTON KEYNES. MK7 8NL. PHONE: (0870) 7300800 FAX: (0870) 7300801
INTERNET: www.visonic.com

DL-125C, DL-125CA DE5817- (REV. 3, 8/08)

MADE IN ISRAEL

DE5817

A.2 Data Record

AL-1 DATA	AL-2 DATA
Message	Message .....
.....	.....
1st called party	1st called party .....
2nd called party	2nd called party .....
3rd called party	3rd called party .....
4th called party	4th called party .....

Visonic Ltd

Declaration of Conformity

In Accordance with R & TTE Directive of 1999/5/EC

We, the undersigned,
Company Visonic Ltd
Address: 24, Habarzel Street. , Tel-Aviv 61220
Country: Israel
Telephone number. +972 3 6456789
Fax number: +972 3 6456788

certify and declare under our sole responsibility that the following equipment

Type	Product description / Supplementary info
DL 125 C	Programmable 2-Channel Speech Dialer

is tested and conforms with the following standards:

Standard
Telephony:
TBR21: Terminal Equipment (TE); Attachment requirements for pan-European approval for connection to the analogue Public Switched Telephone Networks (PSTNs) of TE
Safety Of Information Technology:
EN 60950+ Am1(93), Am2(93), Am3(95), Am4(97). Safety of Information technology equipment including electrical business equipment

and therefore complies with the essential requirements and provisions of the Directive 1999/5/EC of the European Parliament and of the council of 9 march 1999 on Radio equipment and Telecommunications Terminal Equipment and the mutual recognition of their conformity and Annex III (Conformity Assessment procedure referred to in article 10(4)).

The following Notified Bodies and/or independent laboratories have been consulted in the Conformity Assessment procedure:

Type	Number	Name and address
A2LA Certified Lab		ITL, 26 Hacharoshet St. P.O.Box 211 , Or Yehuda 60251, Israel

The technical documentation as required by the Conformity Assessment procedure is kept at the following address.

Company: Visonic Ltd.
Address: Fraser Road
Priory Business Park
Bedford. MK44 3WH
Country: UK
Telephone number: 0870 7300800
Fax number: 0870 7300801

VISONIC LTD. PO BOX [illegible] TEL-AVIV [illegible]

Drawn up in:

Tel-Aviv, Israel on	30/05/2004		Avi Shachrai	
(place)	(dd-mm-yy)	(signature)	(name and function)	(company stamp)

Visonic Ltd 24, Habarzel Street. , Tel-Aviv 61220,Israel Fax:+972 3 6456788 Phone:+972 3 6456789

WARRANTY

Visonic Limited (the "Manufacturer") warrants this product only (the "Product") to the original purchaser only (the "Purchaser") against defective workmanship and materials under normal use of the Product for a period of twelve (12) months from the date of shipment by the Manufacturer.
This Warranty is absolutely conditional upon the Product having been properly installed, maintained and operated under conditions of normal use in accordance with the Manufacturers recommended installation and operation instructions. Products which have become defective for any other reason, according to the Manufacturers discretion, such as improper installation, failure to follow recommended installation and operational instructions, neglect, willful damage, misuse or vandalism, accidental damage, alteration or tampering, or repair by anyone other than the manufacturer, are not covered by this Warranty.
The Manufacturer does not represent that this Product may not be compromised and/or circumvented or that the Product will prevent any death and/or personal injury and/or damage to property resulting from burglary, robbery, fire or otherwise, or that the Product will in all cases provide adequate warning or protection. The Product, properly installed and maintained, only reduces the risk of such events without warning and it is not a guarantee or insurance that such events will not occur.
THIS WARRANTY IS EXCLUSIVE AND EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, OBLIGATIONS OR LIABILITIES, WHETHER WRITTEN, ORAL, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR OTHERWISE. IN NO CASE SHALL THE MANUFACTURER BE LIABLE TO ANYONE FOR ANY CONSEQUENTIAL OR INCIDENTAL DAMAGES FOR BREACH OF THIS WARRANTY OR ANY OTHER WARRANTIES WHATSOEVER, AS AFORESAID.
THE MANUFACTURER SHALL IN NO EVENT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES OR FOR LOSS, DAMAGE, OR EXPENSE, INCLUDING LOSS OF USE, PROFITS, REVENUE, OR GOODWILL, DIRECTLY OR INDIRECTLY ARISING FROM PURCHASER'S USE OR INABILITY TO USE THE PRODUCT, OR FOR LOSS OR DESTRUCTION OF OTHER PROPERTY OR FROM ANY OTHER CAUSE, EVEN IF MANUFACTURER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
THE MANUFACTURER SHALL HAVE NO LIABILITY FOR ANY DEATH, PERSONAL AND/OR BODILY INJURY AND/OR DAMAGE TO PROPERTY OR OTHER LOSS WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR OTHERWISE, BASED ON A CLAIM THAT THE PRODUCT FAILED TO FUNCTION.
However, if the Manufacturer is held liable, whether directly or indirectly, for any loss or damage arising under this limited warranty, **THE MANUFACTURER'S MAXIMUM LIABILITY (IF ANY) SHALL NOT IN ANY CASE EXCEED THE PURCHASE PRICE OF THE PRODUCT,** which shall be fixed as liquidated damages and not as a penalty, and shall be the complete and exclusive remedy against the Manufacturer.
When accepting the delivery of the Product, the Purchaser agrees to the said conditions of sale and warranty and he recognizes having been informed of.
Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so these limitations may not apply under certain circumstances.
The Manufacturer shall be under no liability whatsoever arising out of the corruption and/or malfunctioning of any telecommunication or electronic equipment or any programs.
The Manufacturers obligations under this Warranty are limited solely to repair and/or replace at the Manufacturer's discretion any Product or part thereof that may prove defective. Any repair and/or replacement shall not extend the original Warranty period. The Manufacturer shall not be responsible for dismantling and/or reinstallation costs. To exercise this Warranty the Product must be returned to the Manufacturer freight pre-paid and insured. All freight and insurance costs are the responsibility of the Purchaser and are not included in this Warranty.
This warranty shall not be modified, varied or extended, and the Manufacturer does not authorize any person to act on its behalf in the modification, variation or extension of this warranty. This warranty shall apply to the Product only. All products, accessories or attachments of others used in conjunction with the Product, including batteries, shall be covered solely by their own warranty, if any. The Manufacturer shall not be liable for any damage or loss whatsoever, whether directly, indirectly, incidentally, consequentially or otherwise, caused by the malfunction of the Product due to products, accessories, or attachments of others, including batteries, used in conjunction with the Products. This Warranty is exclusive to the original Purchaser and is not assignable.
This Warranty is in addition to and does not affect your legal rights. Any provision in this warranty which is contrary to the Law in the state or country were the Product is supplied shall not apply.
Warning: The user must follow the Manufacturer's installation and operational instructions including testing the Product and its whole system at least once a week and to take all necessary precautions for his/her safety and the protection of his/her property.

1/08

W.E.E.E. Product Recycling Declaration
For information regarding the recycling of this product you must contact the company from which you orignially purchased it. If you are discarding this product and not returning it for repair then you must ensure that it is returned as identified by your supplier. This product is not to be thrown away with everyday waste.
Directive 2002/96/EC Waste Electrical and Electronic Equipment.

RoHS CE

Visonic®

VISONIC LTD . (ISRAEL): P.O.B 22020 TEL-AVIV 61220 ISRAEL. PHONE: (972-3) 645-6789, FAX: (972-3) 645-6788
VISONIC INC. (U.S.A.): 65 WEST DUDLEY TOWN ROAD, BLOOMFIELD CT. 06002-1376. PHONE: (860) 243-0833, (800) 223-0020 FAX: (860) 242-8094
VISONIC LTD. (UK): 7 COPPERHOUSE COURT, CALDECOTTE, MILTON KEYNES. MK7 8NL. PHONE: (0870) 7300800 FAX: (0870) 7300801
INTERNET: www.visonic.com

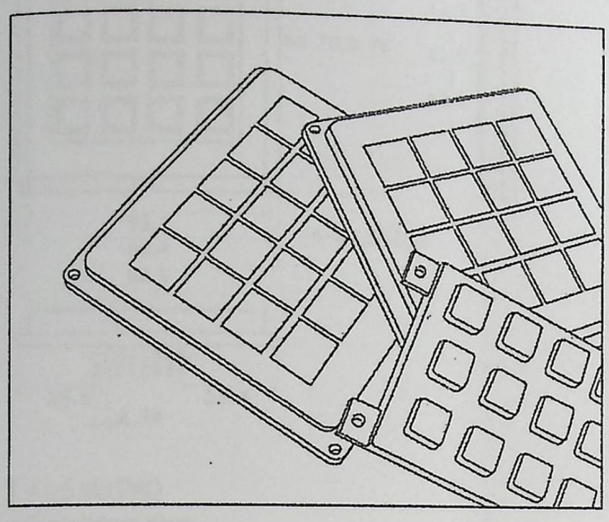
DL-125C, DL-125CA DE5817- (REV. 3, 8/08)

MADE IN ISRAEL

DE5817

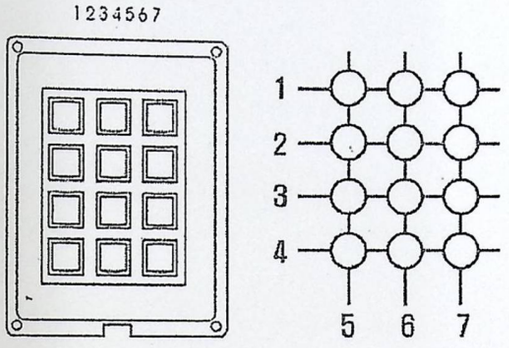
KEYPAD

Keypad Switch

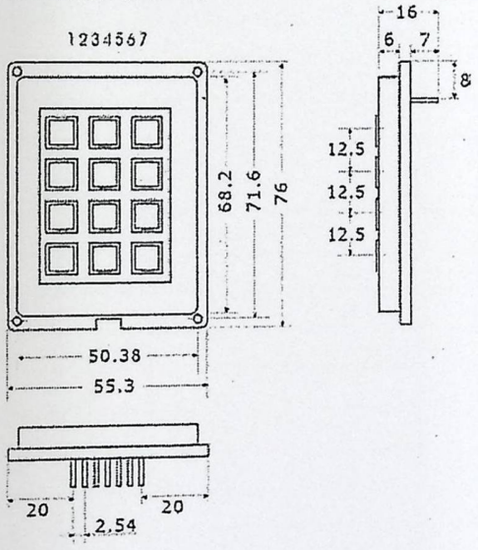


3 x 4 FUNCTION KEYPAD

- Customize Keys
- Pin Layout

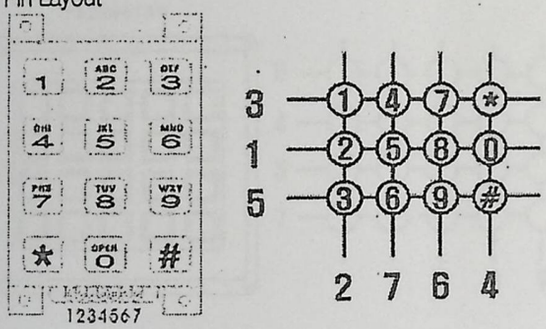


- Dimensions (Units : mm)

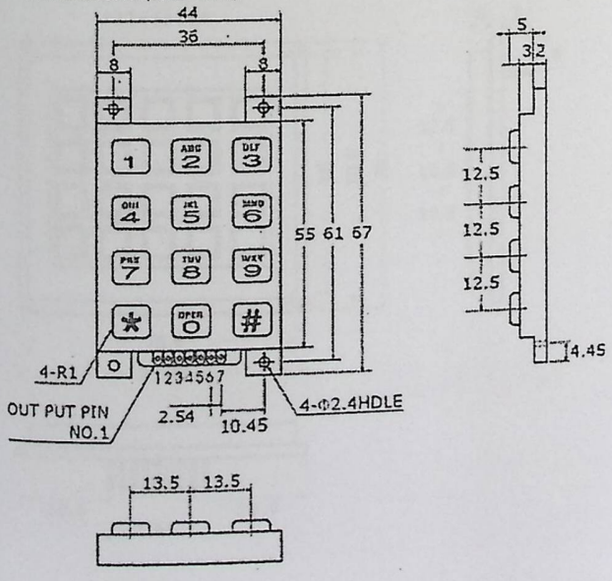


3 x 4 KEYPAD

- Pin Layout

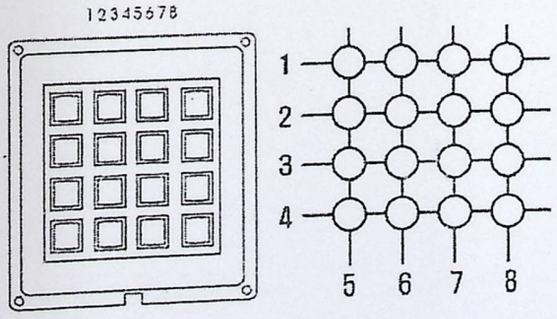


- Dimensions (Units : mm)



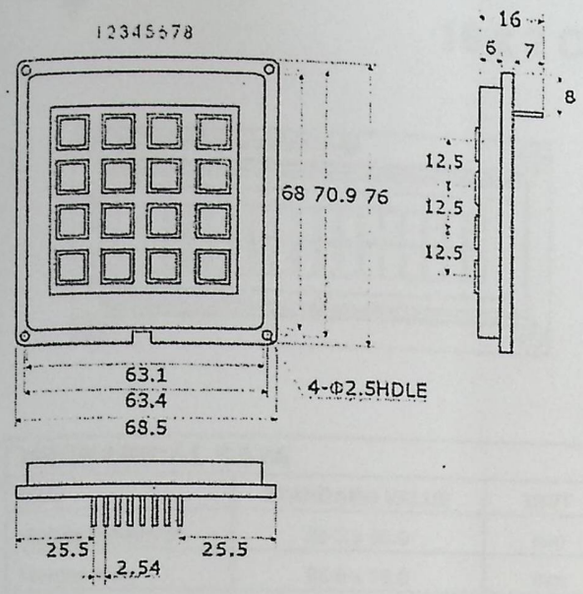
4 x 4 KEYPAD

- Customize Keys
- Pin Layout



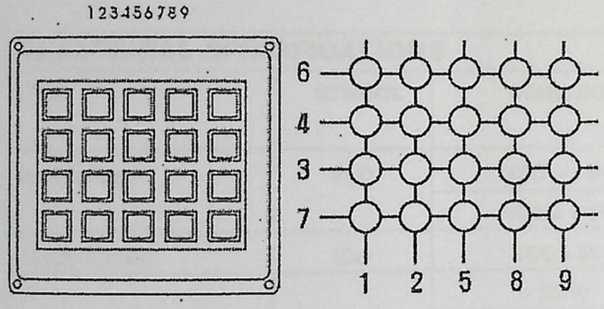
Comfile Technology Inc.
www.comfiletech.com (888)9CUBLOC Fax (650) 638-1360

- Dimensions (Unit : mm)

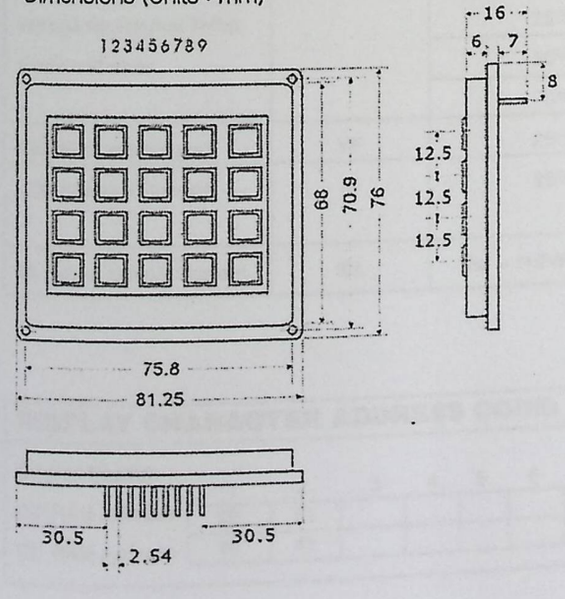


4 x 5 KEYPAD

- Customized Keys
- Pin Layout

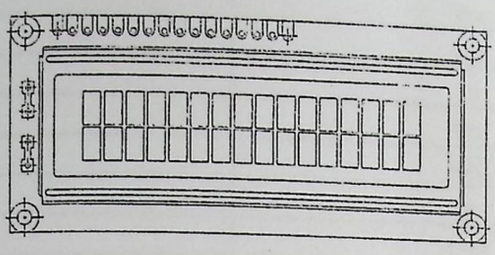


- Dimensions (Units : mm)



Comfile Technology Inc.
www.comfiletech.com (888)9CUBLOC Fax (650) 638-1360

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	–	7.0	V
Input Voltage	VI	- 0.3	–	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS

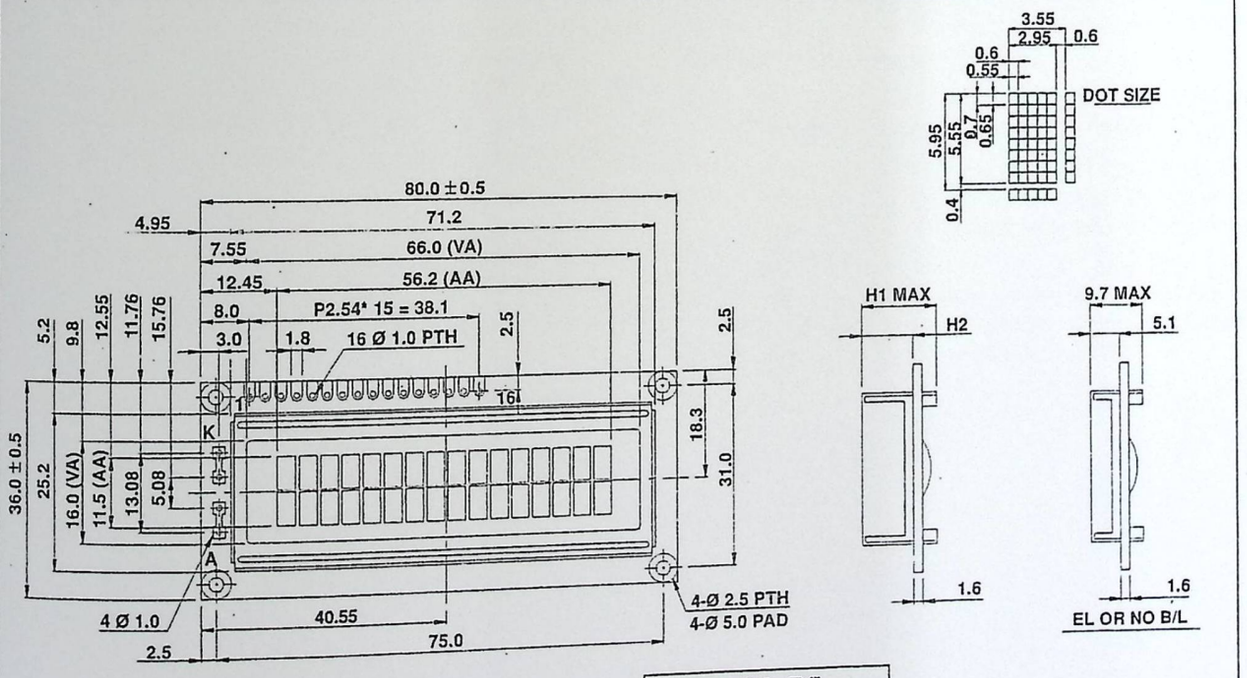
ITEM	SYMBOL	CONDITION		STANDARD VALUE			UNIT
				MIN.	TYP.	MAX.	
Input Voltage	VDD	VDD = + 5V		4.7	5.0	5.3	V
		VDD = + 3V		2.7	3.0	5.3	V
Supply Current	IDD	VDD = 5V		–	1.2	3.0	mA
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C		–	–	–	V
		0°C		4.2	4.8	5.1	
		25°C		3.8	4.2	4.6	
		50°C		3.6	4.0	4.4	
		70°C		–	–	–	
LED Forward Voltage	VF	25°C		–	4.2	4.6	V
LED Forward Current	IF	25°C	Array	–	130	260	mA
			Edge	–	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz		–	–	5.0	mA

DISPLAY CHARACTER ADDRESS CODE:

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or + 5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/$\overline{W}$	H/L Read/Write Signal
6	E	H →L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)

DIMENSIONS in millimeters



LED - H/L B/L	HIGH	LOW
H1	13.2	12.1
H2	8.6	7.5

JOHNSON CONTROLS

Installation Instructions MRC
Issue Date July 19, 2002

Modular Room Control
MRC19-PIR Series Motion Detector Sensors

Application

The MRC19-PIR is a Passive Infrared (PIR) motion detector sensor (wall mount or ceiling mount). The device adds occupancy sensing capability to the Modular Room Control (MRC) integrated room control system. It is used in conjunction with a door contact device and a room controller to determine occupancy in the guest room. This determination contributes to enhanced security, energy management, and overall operating efficiency for a hotel.

Mounting

Location Considerations

MRC19-PIRW (Wall Mount)

Place the motion detector on the wall, as close to the ceiling as possible, and preferably in a corner of the room that overlooks the bed and the entrance area. The field of view for the sensor is 90° horizontal and 45° vertical. Mount at a slight downward angle to maximize coverage. The down side of the PIR is the end with the small window in it.

CAUTION: Risk of Shock.
Disconnect the power supply before mounting the motion detector to prevent electrical shock or possible damage to the equipment.

To mount the motion detector:

1. Remove the cover of the motion detector.
2. Drill a hole in the top part of the plastic, and fish the four-conductor cable through the hole.
3. Mount the detector on the wall using two screws. These screws can be pushed through the two marked circles in the plastic housing. Tighten the screws.

MRC19-PIRC (Ceiling Mount)

Mount the motion detector on a firm section of ceiling, at a height of between 8 and 15 feet and well away from Neon or Fluorescent lights. Keep cables for the PIR away from electrical wires by a minimum of 12 in. (305 mm).

Select the mounting position for optimum coverage, noting that the coverage provided by the detector is slightly elliptical. The field of view is 360° horizontal and 45° vertical. The Light-Emitting Diode (LED) position and the keyhole on the circuit board can identify the correct position.

CAUTION: Risk of Shock.
Disconnect the power supply before mounting the motion detector to prevent electrical shock or possible damage to the equipment.

To mount the motion detector:

1. Remove the motion detector cover by turning the cover in a counterclockwise direction.
2. Screw the base to the ceiling at the required location, passing the wire through the cable entry hole. If the circuit board was removed, replace it at this time.
3. Use a silicon gel to seal any holes. This prevents air drafts and foreign matter, which may cause false alarms, from entering the PIR.

Part No. 24-9778-49, Rev. —

Wiring

MRC19-PIRW (Wall Mount)

CAUTION: Risk of Equipment Damage.
Before applying power, make all wiring connections and check the connections. Short-circuited or improperly connected wires may result in permanent damage to the unit.

IMPORTANT: Make all wiring connections in accordance with the National Electrical Code (NEC) and all local regulations.

Refer to Figure 1 for wiring connections and dimensions. To wire the motion detector:

1. Use a four-conductor, 24-gauge cable from the location of the motion sensor to the MRC thermostat. Strip the insulation from the four wires 1/4 in. (6.35 mm).
2. Turn the four screws on the detector's terminal strip counterclockwise to open them to accept the wires (see Figure 1).
3. Allow at least 4 in. (102 mm) of slack on the cable from the motion detector at the detector's end, to allow for ease of maintenance in the future.
4. Connect the two leads, bringing the 12 VDC power to the terminals marked + and -. Observe polarity. Tighten the screws and tug on the wires to verify that a good connection has been made. Do not exert excessive force when tightening the screws, as this will deform the terminals.
5. Connect the second pair of leads to the terminals marked C and NC. There is no need to observe polarity. Tighten the screws and tug on the wires to verify that a good connection has been made. Do not exert excessive force when tightening the screws, as this will deform the terminals.
6. The onboard LED has a dual function. It will show a short flash for a pre-alarm condition and a long flash (5 seconds) for an alarm condition. If the flashing is not desirable, removing the LED (LK3) link will disable the LED.
7. Install the detector cover.
8. Apply power.
9. Perform a walk test as described in the *Setup and Adjustments* section.

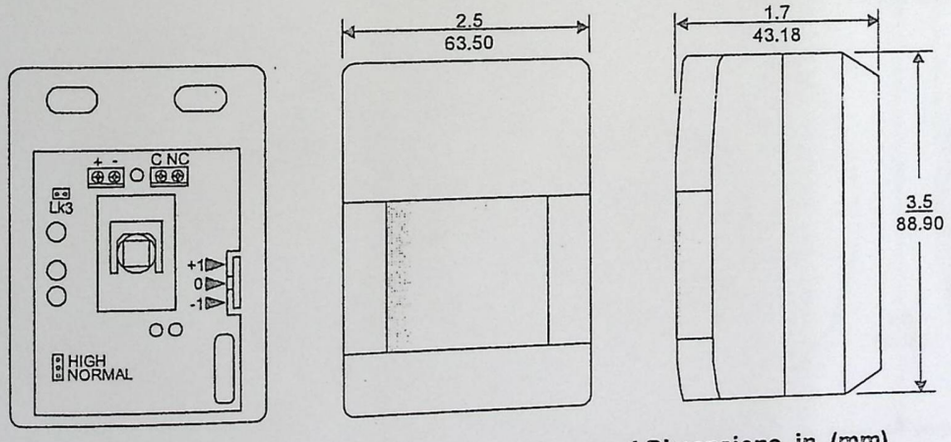


Figure 1: MRC19-PIRW Wiring Connections and Dimensions, in. (mm)

MRC19-PIRC (Ceiling Mount)

CAUTION: Risk of Equipment Damage. Before applying power, make all wiring connections and check the connections. Short-circuited or improperly connected wires may result in permanent damage to the unit.

IMPORTANT: Make all wiring connections in accordance with the National Electrical Code (NEC) and all local regulations.

Refer to Figure 2 for wiring connections and dimensions. To wire the motion detector:

1. Use a four-conductor, 24-gauge cable from the location of the motion sensor to the MRC thermostat. Strip the insulation from the four wires 1/4 in. (6.35 mm).
2. Allow at least 4 in. (102 mm) of slack on the cable from the motion detector at the detector's end, to allow for ease of maintenance in the future.
3. Connect the two leads, bringing the 12 VDC power to the terminals marked 12 V and 0 V. Observe polarity. Tighten the screws and tug on the wires to verify that a good connection has been made. Do not exert excessive force when tightening the screws, as this will deform the terminals.
4. Connect the second pair of leads to the terminals marked N/C. There is no need to observe polarity. Tighten the screws and tug on the wires to verify that a good connection has been made. Do not exert excessive force when tightening the screws, as this will deform the terminals.
5. These are normally closed relay contacts that open when motion is detected. Do not wire the Tamper (not used in this application).
6. Set the range and pulse links according to the required application. It is recommended that the pulse link be in the On position and the range link in the Off position under normal circumstances.
7. Replace the cover.
8. Apply power and wait for the LED to extinguish (approximately 30 seconds).
9. Perform a walk test as described in the *Setup and Adjustments* section.

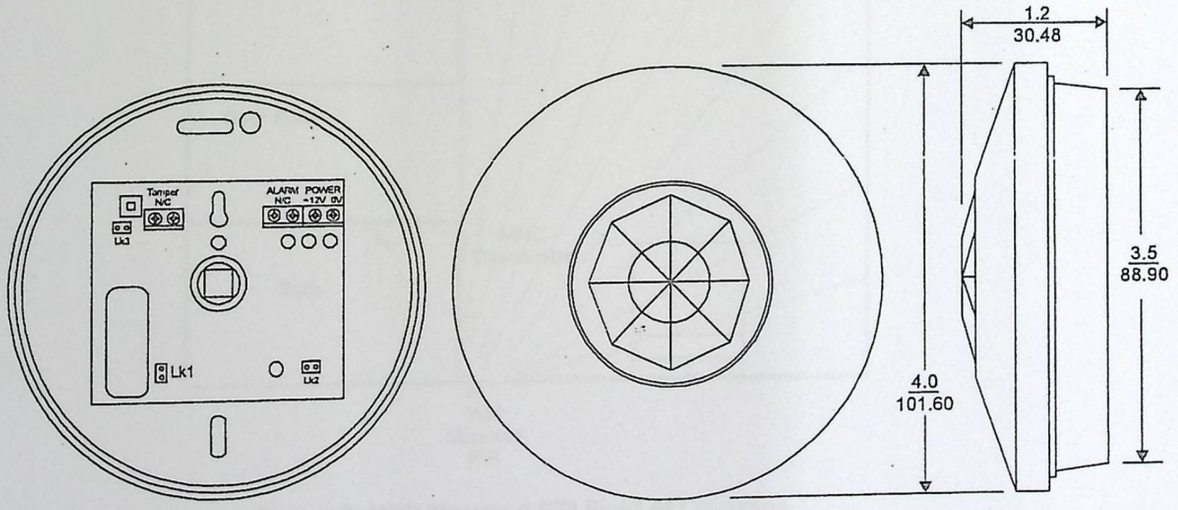


Figure 2: MRC19-PIRC Wiring Connections and Dimensions, in. (mm)

Ceiling Mount PIR Options

Alarm LED LK3

The onboard LED is dual function. It will show a short flash for a pre-alarm condition and a long flash (5 seconds) for an alarm condition.

If the flashing LED is not desirable, removing the LED link will disable the LED.

Pulse Count LK2

It may be unavoidable to install the detector in a bad environment. Pulse count is a means of adjusting the number of pulses received before an alarm is generated. In a normal environment with the jumper on, 1-2 pulses should cause an alarm. For harsher environments, remove the jumper and 3-4 pulses will cause an alarm.

Range LK1

The range link is a means of setting the level of infrared required to generate a pulse from the detector, which will directly affect the detector's sensitivity. The correct setting will depend on the size of the area to be covered.

- With the range link on, the detector will detect the full range of motion.
- With the range link removed, the detector will detect a much smaller area.

Note: The range of the PIR is determined by height of mounting and whether LK1 is on or off.

Setup and Adjustments

Walk Test

By walking throughout the room and observing the LED on the PIR, the covered range and pattern can be verified (see Figure 3 and Figure 4). An alarm should occur at a maximum of 3 to 4 normal steps.

Stop and wait for the LED to turn off before continuing the walk test, continuous motion in the protected area will keep the alarm LED activated.

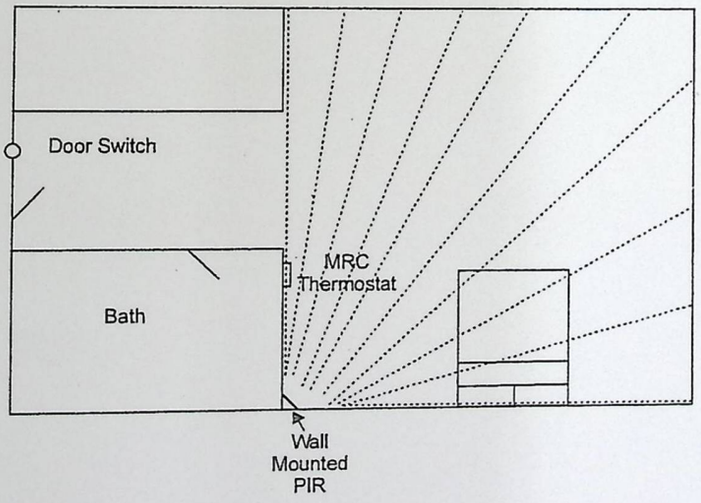


Figure 3: Wall Mounted PIR Field of Coverage

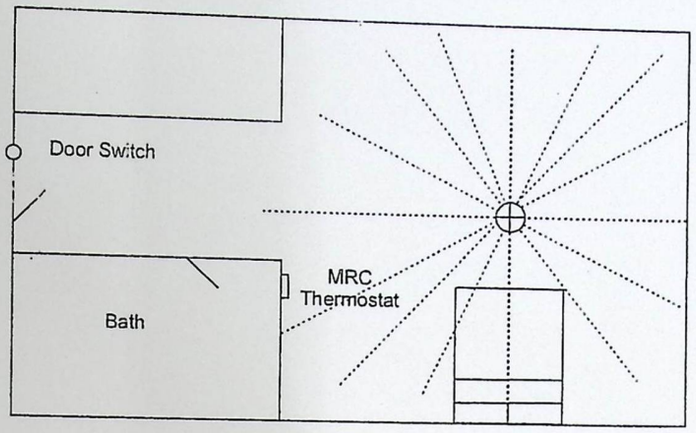


Figure 4: Ceiling Mount PIR Field of Coverage

Technical Specifications

Product	MRC19-PIRW Motion Detector Sensor	MRC19-PIRC Motion Detector Sensor
Power Input	12 VDC	12 VDC
Dimensions: H x L x W	See Figure 1.	See Figure 2.
Shipping Weight	2.8 ounces	2.5 ounces

The performance specifications are nominal and conform to acceptable industry standards. For application at conditions beyond these specifications, consult the local Johnson Controls office. Johnson Controls, Inc. shall not be liable for damages resulting from misapplication or misuse of its products.

JOHNSON CONTROLS

Controls Group
507 E. Michigan Street
P.O. Box 423
Milwaukee, WI 53201

Printed in U.S.A.
www.johnsoncontrols.com