

Palestine Polytechnic University



College of Engineering & Technology
Electrical and Computer Engineering Department
Mechanical Engineering Department

Graduation Project

Air Condition Controller Using Mobile Technologies

Project Team
Hamdi Refat Tahboub
Saadi Fareed Abu Zanch

Project Supervisors
Dr. Isahq Sider
Eng. Radwan Tahboub

Hebron – Palestine

May, 2007



Abstract

The smart home offers a new opportunity to augment people's lives with ubiquitous computing technology that provides increased communications, awareness and functionality.

Mobile technology is evolving at such a rapid pace that new doors have been opened in the field of mobility and wireless applications. This project gives an account of the architecture behind the end-to-end control of one of the common household devices such as Air – Condition, and how this was achieved successfully in real-time. Software on the system controller (8051 microcontroller) is used to communicate mobile and control the Air – Condition.

Mobile which grants client remote control to devices and allows the ability of advanced control between the devices and the end-user. Global System for Mobile communications (GSM) is used to link the two ends.

In this project we were faced a problem in the communication between the mobile and the system controller (8051 microcontroller), so we decided to use the computer to overcome this problem and implement the idea of the project. An explanation on how this is accomplished is provided in the following chapters.

1.1	Introduction	1
1.2	Need of this	2
1.3	The Ultimate Design of PBDS and MCU	3
1.4	ATM	4
1.5	Soft Technical Details	5
1.6	Market Programming	6
1.7	System	7
1.8	System Operation	8
1.9	Development on Mobile Cell	9
1.10	MSM	10
1.11	Java ME Platform for Client/Server	11
1.12	AT Command	12
1.13	Bluetooth	13
1.14	Bluetooth Application	14
1.15	New Wireless Technology in the	15
1.16	Conclusion	16

Table of Contents

Cover page	I
Approved page	II
Dedication	III
Acknowledgment	IV
Abstract	V
Table of contents	VI
List of tables	IX
List of figure	X
List of abbreviations.....	XIII
1 Introduction	1
1.1 Introduction	2
1.2 Problem Definition	2
1.3 Project Objectives	4
1.4 Project Benefits	4
1.5 Similar Work	4
1.6 Time Planning	5
1.6.1 The First Semester Time Plan	5
1.6.2 The Second Semester Time Plan	6
2 Theoretical Background	7
2.1 Introduction	8
2.2 The Mobile Phone	8
2.2.1 Mobile Technology	9
2.2.2 GSM Technology	10
2.2.3 Mobile FBUS	11
2.2.4 Nokia FBUS	11
2.2.4.1 The Difference Between FBUS and MBUS	12
2.2.5 SMS	14
2.2.5.1 SMS Technical Details	14
2.2.6 Mobile Programming	16
2.2.6.1 Symbian OS	16
2.2.6.1.1 Symbian Structure	16
2.2.6.1.2 Developing on Symbian OS	18
2.2.6.2 J2ME	19
2.2.6.2.1 Java ME Platform for Converged Services	21
2.2.6.3 AT Commands	22
2.2.7 Bluetooth	23
2.2.7.1 List of Applications	23
2.2.7.2 How Bluetooth Technology Works	23
2.3 The Air - Condition	26

2.3.1	Introduction	26
2.3.2	What is Air Conditioning	26
2.3.3	Individual Room Air Conditioning Systems	27
2.3.3.1	Advantages and Disadvantages.....	28
2.3.4	Air Conditioning Devices	28
2.3.5	Air Conditioning System	29
2.3.6	Components of the Air Conditioning System	30
2.3.6.1	Compressor.....	30
2.3.6.2	Condenser.....	31
2.3.6.3	Expansion Valve.....	31
2.3.6.4	Evaporator.....	31
2.3.6.5	Fans.....	31
2.3.7	Sensors and Transducers	32
2.3.7.1	Temperature Sensors.....	33
2.3.7.2	Transducers or Transmitters.....	34
2.3.8	Controllers	35
2.3.9	Air Conditioning Cycle	36
2.3.10	Split-System Air Conditioning	36
2.3.11	Problems Accruing with Air Conditioning Disadvantages	37
2.3.12	Variable Speed Control	38
2.4	The Microcontrollers and FPGA	39
2.4.1	Microcontrollers	39
2.4.1.1	Microcontroller Architecture.....	40
2.4.1.1.1	Central Processing Unit CPU.....	40
2.4.1.1.2	Program and Data Memory.....	41
2.4.1.1.3	Inputs/Outputs an Peripherals.....	42
2.4.1.1.3.1	Analog to Digital and Digital to Analog Converters.....	42
2.4.1.2	Microcontroller Applications.....	43
2.4.2	Micro Web – Server	43
2.4.3	PIC Microcontroller	44
2.4.3.1	ToothPIC.....	44
2.4.4	FPGA	45
2.4.4.2	Comparison Between FPGA and CPLD.....	49
2.4.4.3	Comparison Between FPGA and ASIC.....	46
2.4.4.4	Hardware Description Languages (HDL).....	50
2.4.4.4.1	HDL and programming languages.....	51
2.4.4.4.2	Verilog HDL.....	52
2.4.4.5	FPGA and CPLD Implementation Procedure.....	52
2.4.4.6	CPLDs and FPGAs Applications.....	54
3	Architectural Design	56
3.1	Introduction	57
3.2	Project Objectives	57
3.3	System Functional Requirements	58
3.4	System Components	58

3.5	General Block Diagram	60
3.5.1	System Block Diagram	61
3.6	How Does the System Work?	62
4	Detailed System Design.....	63
4.1	Introduction	64
4.2	System Controller Design Options	64
4.2.1	FPGA Controller.....	64
4.2.1.1	Altera.....	65
4.2.1.2	Xilinx	66
4.2.2	PIC Controller	67
4.2.2.1	ToothPIC Controller	68
4.3	Mobile to Mobile Design Options	70
4.3.1	Control Using SMS	71
4.3.2	Real Time Calling/Control	72
4.4	Interfacing Circuits	73
4.4.1	Project Schematic.....	80
5	Software System Design	82
5.1	Introduction	83
5.2	Software Requirement Specification	83
5.2.1	Microcontroller Software Requirement Specification.....	84
5.3	Controlling the Air – Condition.....	88
5.3.1	Sending Signal to the Relay	89
5.4	Mobile Communication	92
5.4.1	Sub Code for Sending AT Command to Mobile	92
6	Implementation and Testing.....	95
6.1	Introduction	96
6.2	Air – Condition	96
6.2.1	Air – Condition Implementation	96
6.3	Mobile.....	99
6.3.1	Mobile Implementation	100
6.3.2	Mobile Communication Problem	100
6.4	Microcontroller	103
6.4.1	Microcontroller Implementation	103
6.4.2	Microcontroller Failure.....	105
7	Future Work.....	107
7.1	Introduction	108
7.2	Expected learning Outcomes	108
7.3	Real Learning Outcomes	108

7.4	Future View	109
-----	-------------------	-----

Table 7.1	Future View - Summary	110
Table 7.2	Future View - Summary	111
Table 7.3	Future View - Summary	112
Table 7.4	Future View - Summary	113

List of Tables

Table 1.1 First Semester Time Plan	5
Table 1.2 Second Semester Time Plan	6
Table 1.3 Second Semester Time Plan	6
Table 6.1 8051 Microcontroller Board	105

List of Figures

Figure 1.1 Remote device control using Mobile terminal through GPRS system	3
Figure 1.2 Remote device control using peer Mobile phones through GSM system	3
Figure 2.1 GSM Network	11
Figure 2.2 3310 Nokia FBUS	12
Figure 2.3 FBUS example1	13
Figure 2.4 FBUS example2	13
Figure 2.5 FBUS example3	13
Figure 2.6 FBUS example4	13
Figure 2.7 FBUS example5	13
Figure 2.8 FBUS example6	13
Figure 2.9 Java Platform	21
Figure 2.10 Air - Condition components	30
Figure 2.11 A temperature sensor	33
Figure 2.12 Air - Condition cycle	36
Figure 2.13 Split Air - Condition	37
Figure 2.14 Example on speed control	39
Figure 2.15 Microcontroller General Block Diagram	41
Figure 2.16 ToothPIC with mobile	45
Figure 2.17 PLD Architecture	47
Figure 2.18 CPLD and FPGA Architecture	48
Figure 2.19 FPGA Programming	53
Figure 3.1 Remote device control using peer Mobile phones through GSM system	59
Figure 3.2 General Block Diagram	60
Figure 3.3 System Block Diagram with F- bus	61
Figure 3.4 System Block Diagram with Bluetooth	61
Figure 4.1 FPGA controller	65
Figure 4.2 UP education board block diagram	66
Figure 4.3 Virtex - II prototype platform block diagram	67
Figure 4.4 PIC Controller	68
Figure 4.5 ToothPIC Controller	69
Figure 4.6 PIC18LF6720 with BlueMatik	70
Figure 4.7 Mobile to Mobile Design Options	71
Figure 4.8 Mobile to Mobile by SMS	72
Figure 4.9 Mobile to Mobile by real time calling	73
Figure 4.10 Microcontroller schematic	74
Figure 4.11 10mA relay circuit	75
Figure 4.12 20mA relay circuit	76
Figure 4.13 Relays in the project	77
Figure 4.14 First MAX232	78
Figure 4.15 DLR-3 cable	79
Figure 4.16 Mobile to other devices circuit	79
Figure 4.17 Temperature Sensor Interfacing Circuit	80
Figure 4.18 Project Schematic	81

Figure 5.1 Connection Description Screen	85
Figure 5.2 Connect To Screen.....	85
Figure 5.3 COM1 Proprieties.....	86
Figure 5.4 Assembler.....	87
Figure 5.5 Serial Cable	87
Figure 5.6 DLR - 3 Cable.....	88
Figure 5.7 Flow Chart 1 Cooling mode	90
Figure 5.8 Flow Chart 2 Heating Mode	91
Figure 5.9 Flow chart 3 Mobile Communication.....	94
Figure 6.1 Air - Condition on the Base of metal and wood.....	97
Figure 6.2 Relays and Optocoupler Circuit	98
Figure 6.3 10mA Relay Circuit.....	98
Figure 6.4 20mA Relay Circuit.....	99
Figure 6.5 Nokia 6100 Phone	100
Figure 6.6 Second Project Schematic	102
Figure 6.7 8051 Microcontroller Board	104
Figure 6.8 Third Project Schematic	106

PCD - Project Compiler

DLR - Dynamic Linking Library

DM - Operating System

List of Abbreviations

SMS	: Short Message Service
SMSC	: Short Message Service Center
FPGA	: Field Programmable Gateway Array
GSM	: Global System for Mobile communication
PLA	: Programmable Logic Array
FPD	: Field Programmable Device
PAL	: Programmable Array Logic
CPLD	: Complex Programmable Logic Device
PLD	: Programmable Logic Device
OTP	: One – Time Programmable
HDL	: Hardware Description Language
VLSI	: Very Large Scale Integration
CLB	: Configurable Logic Blocks
LUT	: Look – Up Tables
SPLD	: Single Chip Programmable Logic Device
DLL	: Dynamic Linking Library
OS	: Operating System

Chapter One

Introduction

1 Introduction

1.1 Introduction	1
1.2 Project Definition	2
1.3 Project Objectives	3
1.4 Project Structure	4
1.5 Project Work	4
1.6 Time Planning	5

1.1 Introduction

1

The first part of the book is devoted to the general theory of project management. It discusses the importance of project management in the modern business environment and the role of the project manager. It also covers the basic concepts of project management, such as project objectives, project benefits, and project risks. The second part of the book is devoted to the practical aspects of project management. It discusses the various stages of the project management process, from project initiation to project closure. It also covers the various tools and techniques used in project management, such as Gantt charts, PERT networks, and earned value management. The third part of the book is devoted to the human aspects of project management. It discusses the importance of communication, team building, and conflict resolution in project management. It also covers the various roles and responsibilities of the project manager and the project team members.

Chapter One Introduction

1 Introduction

1.1 Introduction	2
1.2 Problem Definition	2
1.3 Project Objectives	4
1.4 Project Benefits	4
1.5 Similar Work	4
1.6 Time Planning	5

1.1 Introduction

The notion that we could eventually live in so-called "smart homes" domestic environments in which we are surrounded by interconnected technologies that are, more or less, responsive to our presence and actions seems increasingly plausible. Trends such as the proliferation of networkable devices, wireless technologies, and an increasing vendor focus on technologies for the home are driving awareness of the smart home idea. While different initiatives can teach us about what a smart home could provide to its "occupant-users," technical questions remain. In this project we present a hardware control system using mobile technologies that can be used in controlling smart home devices. The air-conditioning system will be considered for implementation in order to test the controlling hardware.

The ability to remotely control household devices using a mobile wireless system has great advantages for the average home-owner. For example, doors can be unlocked when children arrive home early and do not have a key. The project we are presenting boosts the role of remote access by offering such control from the user's mobile. A program on a system controller used to control the device. GSM technology allows the mobile to communicate with its peer mobile at the home end which creates the necessary link required for accessing the control hardware. This pathway can also be utilized for reverse communication between the hardware and the mobile client (Terminal mobile) through the peer mobile at the home end. The Short Message Service (SMS) also provides an alternate path for such communication.

1.2 Problem Definition

There are many remote device control systems using mobile technologies that utilize the GPRS/Internet protocols as shown in figure 1.1. In this project we will use two peer mobile phones which communicate directly with each other using the GSM system to control the home devices through the specified control hardware.

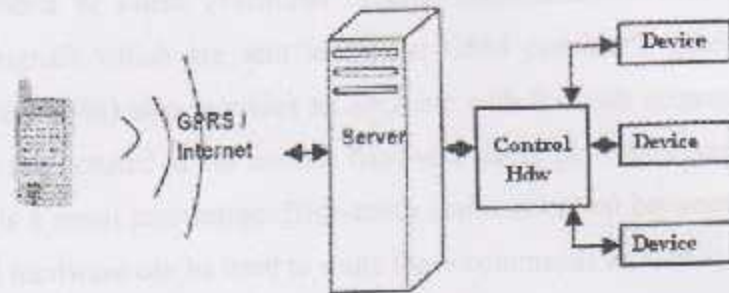


Figure 1.1 Remote device control using Mobile terminal through GPRS system

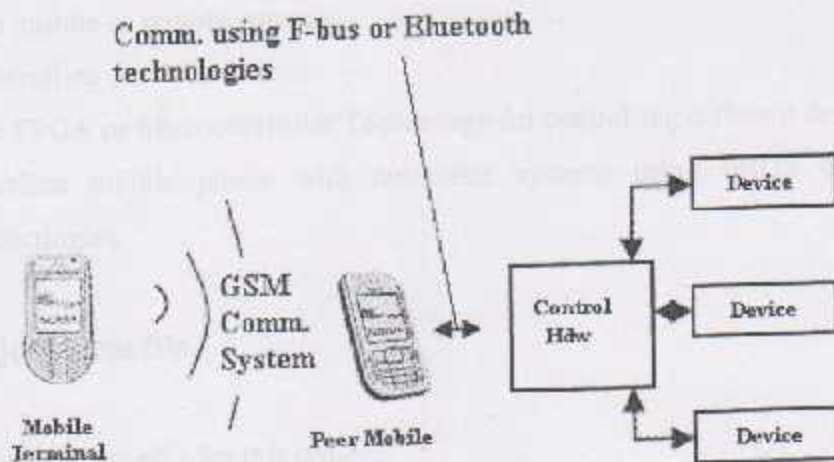


Figure 1.2 Remote device control using peer Mobile phones through GSM system

As can be seen from Figure 1.2, we can group the project components under the following categories: the Client or Remote side (Terminal) using a mobile phone, the Peer Mobile at the home side. The GSM communication system between the Peer mobile phones, the Hardware Controlling System (the Controller) and the controlled devices (the Air-condition in our project.). These main components are linked to each other with the appropriate communication channel.

The mobile phone or client (Terminal mobile) communicates with the peer mobile through data signals which are sent using the GSM provider's gateways (The Short Message Service (SMS) also provides an alternate path for such communication). These commands are then routed to the control hardware using the F-bus standard port on the mobile which is a serial connection. Blue-tooth communication between the peer mobile and the control hardware can be used to route these commands as well.

1.3 Project Objectives

This projects aims at achieving the following objectives:

- Use mobile as remote control.
- Controlling Air – Condition.
- Use FPGA or Microcontroller Technology for controlling different devices.
- Interface mobile phone with controller systems using FBUS or Bluetooth technologies.

1.4 Project Benefits

There are several benefits for this project:

- This project can be used in developing smart houses.
- Using mobile in this project, enable the remote controlling of device.
- Sending and Receiving data to and from Air – Condition.
- Controlling Air – Condition
- Can be used to facilitate bringing other information from mobile.
- Can be used in many security systems.

1.5 Similar Work

Too many projects discussed the idea of smart homes. Some of these projects were done in our university for example Cell Phone – Based Controller for a Toy Car was done in 2003, and many other were published on the internet.

Also this project is a continuation and development of the project Air – Condition Controller done in 2001.

Finally the idea of remotely controlling devices is widely spread, and it grown every day.

1.6 Estimated Development Cost

The following table shows the main and necessary components that are needed for the system.

Component	Required Number	Price (U.S Dollar)
Microcontroller 8051	1	120
Development Board with LCD		
Mobile with SIM Card	1	100
Optocoupler(4N25)	5	3
Relay(+5V) 20mA	1	5
Relay(+5v) 10mA	4	5
Resistors	50	10
Wires and Cable	-	10
Wood base for the Air – condition	1	70
DRL-3P Cable	1	4
Standard 9 pin Serial Cable	1	3

Table 1.1 Estimated Development Cost

1.7 Time Planning

The project plan follows the following time schedule, which includes the related tasks of study and system analysis.

The following time plan is for the first and second semesters.

1.7.1 The First Semester Time Plan

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Collecting Information about the project	█	█	█													
Studying FPGA Technology				█	█											
Studying the Air-Condition						█	█	█								
Studying mobile and PIC Microcontroller							█	█	█							
System Design									█	█	█					
Project Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█		

Table 1.2 First Semester Time Plan

1.7.2 The Second Semester Time Plan

Task/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Interfacing Air - Condition	█	█	█	█	█	█										
Interfacing Mobile	█	█	█	█	█	█										
Making the system							█	█	█							
System testing										█	█	█				
Specifying Future Work													█	█		
Project Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█		

Table 1.3 Second Semester Time Plan

2

Chapter Two

Theoretical Background

2 Theoretical Background

2.1 Introduction	8
2.2 The Mobile Phone	8
2.3 The Air - Condition	26
2.4 The Microcontrollers and FPGA	39

2.1 Introduction

The basic idea behind our project is to design and implement an Air – Condition control system that can be triggered and managed through two peer mobile phones communication through GSM technology. The remote access can be accomplished by offering such control from the user's mobile device (Terminal Mobile). A program on a system controller can be used to control the device (Air – Condition). GSM technology allows the mobile to communicate with its peer mobile at the home end which creates the necessary link required for accessing the control hardware. This pathway can also be utilized for reverse communication between the hardware and the mobile client through the peer mobile at the home end. The Short Message Service (SMS) also provides an alternate path for such communication.

Different technologies and techniques will be used in designing and implementing such a system. This chapter presents most of the required background information required for understanding the theory, terminology and applications used in our project. This includes Mobile phone technologies and programming techniques, Microcontroller and FPGA technologies and design options, and the air-condition systems and devices.

2.2 The Mobile Phone

The Mobile Phone is an important part in the project. It is used to communicate with the Air – Condition via FPGA or microcontroller to send commands and receive data. The following paragraphs are introducing the Mobile Phone.

A mobile or cellular telephone is a long-range, portable electronic device for personal telecommunications over long distances.

2.2.1 Mobile Technology

A cell site is a site where antennae and electronic communications equipment are placed to create a cell in a cellular network for the use of mobile phones.

Mobile phones and the network they operate under vary significantly from provider to provider, and even from nation to nation. However, all of them communicate through electromagnetic radio waves with a cell site base station, the antennas of which are usually mounted on a tower, pole, or building.

The phones have a low-power transceiver that transmits voice and data to the nearest cell sites, usually 5 to 8 miles (approximately 8 to 13 kilometers) away. When the cellular phone or data device is turned on, it registers with the mobile telephone exchange, or switch, with its unique identifiers, and will then be alerted by the mobile switch when there is an incoming telephone call. The handset constantly listens for the strongest signal being received from the surrounding base stations. As the user moves around the network, the mobile device will "handoff" to various cell sites during calls, or while waiting (idle) between calls it will reselect cell sites.

Cell sites have relatively low-power (often only one or two watts) radio transmitters which broadcast their presence and relay communications between the mobile handsets and the switch. The switch in turn connects the call to another subscriber of the same wireless service provider or to the public telephone network, which includes the networks of other wireless carriers.[1]

2.2.2 GSM Technology

The GSM Technology provides the environment in which the two mobiles communicate; the following is an illustration of this technology.

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services. GSM differs from first generation wireless systems in that it uses digital technology and time division multiple access transmission methods. GSM is a circuit-switched system that divides each 200kHz channel into eight 25kHz time-slots. GSM operates in the 900MHz and 1.8GHz bands in Europe and the 1.9GHz and 850MHz bands in the US. The 850MHz band is also used for GSM and 3GSM in Australia, Canada and many South American countries. GSM supports data transfer speeds of up to 9.6 kbit/s, allowing the transmission of basic data services such as SMS (Short Message Service). Another major benefit is its international roaming capability, allowing users to access the same services when traveling abroad as at home. This gives consumers seamless and same number connectivity in more than 210 countries. GSM satellite roaming has also extended service access to areas where terrestrial coverage is not available. [2]

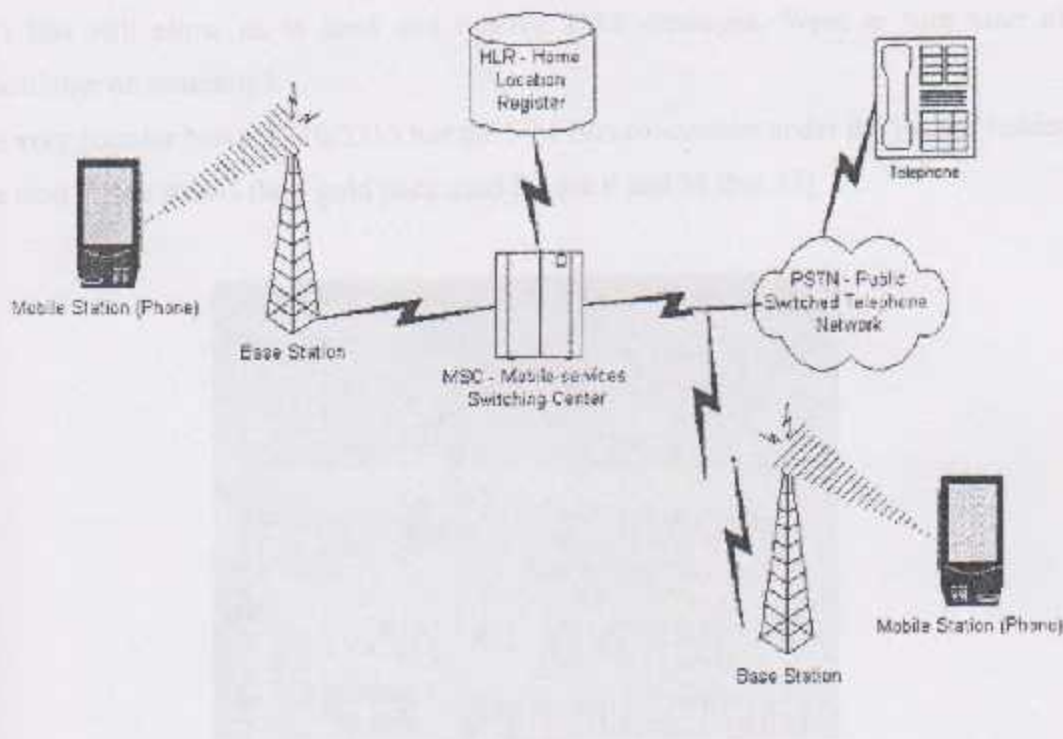


Figure 2.1 GSM Network

2.2.3 Mobile FBUS

The Mobile FBUS is one of the ways that we can exchange information with the mobile. The following paragraphs illustrates the NOKIA FBUS, which it is widely spread, and there are two many documents publish on it. By the way NOKIA did not publish any documentation about it, all the information about the FBUS is a result of experiments and observations on the FBUS; also we can have information about the FBUS from previous projects that use it.

2.2.4 Nokia FBUS

Most Nokia phones have F-Bus and M-Bus connections that can be used to connect a phone to a PC or in our case a microcontroller. The connection can be used for controlling just about all functions of the phone, as well as uploading new firmware etc.

This bus will allow us to send and receive SMS messages. Want to turn your air-conditioner on remotely?

The very popular Nokia 3310/3315 has the F/M Bus connection under the battery holder. The next figure shows the 4 gold pads used for the F and M Bus. [3]



Figure 2.2 3310 Nokia FBUS

2.2.4.1 The Difference Between FBUS and MBUS

M-Bus is a one pin bi-directional bus for both transmitting and receiving data from the phone. It is slow (9600bps) and only half-duplex. Only two pins on the phone are used. One ground and one data. M-Bus runs at 9600bps, 8 data bits, odd parity, one stop bit. The data terminal ready (DTR) pin must be cleared with the request to send (RTS). This powers the electronics in the cable.

F-Bus is the later high-speed full-duplex bus. It uses one pin for transmitting data and one pin for receiving data plus the ground pin. Very much like a standard serial port. It is fast 115,200bps, 8 data bits, no parity, one stop bit. For F-Bus the data terminal ready (DTR) pin must be set and the request to send (RTS) pin cleared. [3]

The following figures show different FBUS for different mobile models.



NOKIA 6610, 7210 Data connector
 1-Charge 2-Charge GND 3-ACI 4- Vout
 5-USB Vbus 6-Fbus_Rx / USB D+
 7-FBus_Tx / USB D- 8-Data GND

Figure 2.3 FBUS example1



NOKIA 51xx,61xx,71xx,62xx,6310
 6-MBUS 7-Rx 8-Tx 9-GND

Figure 2.4 FBUS example2



NOKIA 51xx,61xx,71xx,62xx,6310
 6-MBUS 7-Rx 8-Tx 9-GND

Figure 2.5 FBUS example3



NOKIA 51xx,61xx,71xx,62xx,6310
 6-MBUS 7-Rx 8-Tx 9-GND

Figure 2.6 FBUS example4



ERICSSON 8xx, T10, T18, A1018
 7-Test 8-+5V 9-Tx 10-GND 11-Rx

Figure 2.7 FBUS example5



MOTOROLA V60,V66,V70,T280,T720
 1-GND 2-R 25 k to GND 3-+5V
 4-Tx 5-Rx 6-VBUS 10-GND

Figure 2.8 FBUS example6

2.2.5 SMS

Short Message Service (SMS) is a service available on most digital mobile phones (and other mobile devices, e.g. a Pocket PC, or occasionally even desktop computers) that permits the sending of short messages (also known as text messages, or more colloquially SMSes, texts or even txts) between mobile phones, other handheld devices and even landline telephones. The term text messaging and its variants are more commonly used in North America, the UK, and the Philippines, while most other countries prefer the term SMS. Text messages are often used to interact with automated systems, such as ordering products and services for mobile phones, or participating in contests. There are also many services available on the Internet that allows users to send text messages free of charge.

[4]

2.2.5.1 SMS Technical Details

The Short Message Service - Point to Point (SMS-PP) is defined in GSM recommendation 03.40. GSM 03.41 defines the Short Message Service - Cell Broadcast (SMS-CB) which allows messages (advertising, public information, etc.) to be broadcast to all mobile users in a specified geographical area.

Messages are sent to a Short Message Service Centre (SMSC) which provides a store-and-forward mechanism. It attempts to send messages to their recipients. If a recipient is not reachable, the SMSC queues the message for later retry. Some SMSCs also provide a "forward and forget" option where transmission is tried only once. Both Mobile Terminated (MT), for messages sent to a mobile handset, and Mobile Originating (MO), for those that are sent from the mobile handset, operations are supported. Message delivery is best effort, so there are no guarantees that a message will actually be delivered to its recipient and delay or complete loss of a message is not uncommon, particularly when sending between networks. Users may choose to request delivery reports, which can provide positive confirmation that the message has reached the intended recipient, but notifications for failed deliveries are unreliable at best.

Transmission of the short messages between SMSC and phone can be done through different protocols such as SS7 within the standard GSM MAP framework or TCP/IP within the same standard. Messages are sent with the additional MAP operation `forward_short_message`, whose payload length is limited by the constraints of the signaling protocol to precisely 140 bytes ($140 \text{ bytes} = 140 * 8 \text{ bits} = 1120 \text{ bits}$). In practice, this translates to either 160 7-bit characters, 140 8-bit characters, or 70 16-bit characters. Characters in languages such as Arabic, Chinese, Korean, Japanese or Slavic languages (e.g. Russian) must be encoded using the 16-bit UCS-2 character encoding. Routing data and other metadata is additional to the payload size.

Larger content (known as long SMS or concatenated SMS) can be sent segmented over multiple messages, in which case each message will start with a user data header (UDH) containing segmentation information. Since UDH is included in the payload, the number of characters per segment is lower: 153 for 7-bit encoding, 134 for 8-bit encoding and 67 for 16-bit encoding. The receiving phone is then responsible for reassembling the message and presenting it to the user as one long message. While the standard theoretically permits up to 255 segments, 6 to 8 segment messages are the practical maximum, and long messages are billed as equivalent to multiple SMS messages.

Short messages can also be used to send binary content such as ring tones or logos, as well as OTA programming or configuration data. Such uses are a vendor-specific extension of the GSM specification and there are multiple competing standards, although Nokia's Smart Messaging is by far the most common.

The SMS specification has defined a way for an external Terminal Equipment, such as a PC or Pocket PC, to control the SMS functions of a mobile phone. The connection between the Terminal Equipment and the mobile phone can be realized with a serial cable, a Bluetooth link, an infrared link, etc. The interface protocol is based on AT commands. Common AT commands include `AT+CMGS` (send message), `AT+CMSS` (send message from storage), `AT+CMGL` (list messages) and `AT+CMGR` (read message).

Some service providers offer the ability to send messages to land line telephones regardless of their capability of receiving text messages by automatically phoning the recipient and reading the message aloud using a speech synthesizer along with the number of the sender.

Today, SMS is also used for machine to machine communication. For instance, there is an LED display machine controlled by SMS, and some vehicle tracking companies like ESITrack use SMS for their data transport or telemetry needs. [4]

2.2.6 Mobile Programming

2.2.6.1 Symbian OS

Symbian OS is an operating system, designed for mobile devices, with associated libraries, user interface frameworks and reference implementations of common tools, produced by Symbian Ltd. It is a descendant of Psion's EPOC and runs exclusively on ARM processors.

Symbian is currently owned by Ericsson (15.6%), Nokia (47.9%), Panasonic (10.5%), Samsung (4.5%), Siemens AG (8.4%), and Sony Ericsson (13.1%). [5]

2.2.6.1.1 Symbian Structure

At its lowest level sit the base components of Symbian OS. This includes the kernel (EKA1 or EKA2), along with the user library which allows user-side applications to request things of the kernel. Symbian OS has a microkernel architecture, which means that the minimum necessary is within the kernel. It contains a scheduler and memory management, but no networking or file system support. These things are provided by user-side servers. The base layer includes the file server, which provides a fairly DOS-like view of the file systems on the device (each drive has a drive letter, and backslashes are used as the directory delimiter). Symbian OS supports various file system types

including FAT32 and Symbian OS-specific NOR flash filing systems. The file system is generally not exposed to the user through the phone user interface.

Immediately above base are a selection of system libraries. These take all shapes and sizes, including for example character set conversion, a DBMS database, and resource file handling.

There is a large networking and communication subsystem, which has three main servers - ETEL (EPOC telephony), ESOCK (EPOC sockets) and C32 (responsible for serial communication). Each of these has a plug-in scheme. For example ESOCK allows different ".PRT" protocol modules, implementing different types of networking protocol scheme. There's lots of stuff relating to short-range communication links too, such as Bluetooth, IrDA and USB.

There's also a large amount of user interface code. Even though the user interfaces themselves are maintained by other parties, the base classes and substructure ("UIKON") for all UIs are present in Symbian OS, along with certain related servers (for example, a view server which controls transitions between different phone user interface screens). There's a lot of related graphics code too - such as a window server and a font and bitmap server.

An application architecture provides for standard application types, embedding, and file and data recognition. There is also a selection of application engines for popular smart phone applications such as calendars, address books, and task lists. A typical Symbian OS application is split up into an engine DLL and a graphical application - the application being a thin wrapper over the engine DLL. Symbian OS provides some of these engine DLLs.

There are, of course, many other things that don't yet fit into this model - for example, SyncML, Java ME providing another set of APIs on top of most of the OS and multimedia. Quite a few of these things are frameworks, and vendors are expected to supply plug-ins to these frameworks from third parties (for example, Helix player for multimedia codecs). This has the advantage that the APIs to such areas of functionality are the same on many phone models, and that vendors get a lot of flexibility, but means that phone vendors need to do a great deal of integration work to make a Symbian OS phone.

Symbian OS device manufacturers also get supplied with an example user interface layer called Tech View. This is very similar to the user interface from a Psion Series 5 personal organizer, so isn't particularly similar to any given phone user interface, but provides a basis to start customization. It is also the environment in which a lot of Symbian OS test code and example code runs. [5]

2.2.6.1.2 Developing on Symbian OS

There are multiple platforms, based upon Symbian OS, that provide an SDK for application developers wishing to target a Symbian OS device - the main ones being UIQ, Series 60, etc. Individual phone products, or families, often have SDKs or SDK extensions downloadable from the manufacturer's website too. The SDKs contain documentation, the header files and library files required to build Symbian OS software, and a Windows-based emulator ("WINS"). Up until Symbian OS version 8, the SDKs also included a version of the GCC compiler (a cross-compiler) required to build software to work on the device.

Symbian OS 9 uses a new ABI and so requires a new compiler - a choice of compilers is available including a new version of GCC (see external links below). In terms of SDKs, UIQ Technology now provides a simplified framework so that the single UIQ SDK forms the basis for developing on all UIQ 3 devices, such as the Sony Ericsson P990 and Sony Ericsson M600.

Symbian C++ programming is commonly done with a commercial IDE. For current versions of Symbian OS, CodeWarrior for Symbian OS is favoured. The CodeWarrior tools will be replaced during 2006 by Carbide.c++, an Eclipse-based IDE developed by Nokia. It is expected that Carbide.c++ will be offered in different versions: a free version may allow users to prototype software on the emulator for the first time in a free product.

Visual Basic, VB.NET, and C# development for Symbian can be accomplished through AppForge Crossfire, a plugin for Microsoft Visual Studio.

There's also a version of a Borland IDE for Symbian OS. Symbian OS development is also possible on Linux and Mac OS X using tools and techniques developed by the community, partly enabled by Symbian releasing the source code for key tools. A plugin that allows development of Symbian OS applications in Apple's Xcode IDE for Mac OS X is available.

Once developed, Symbian OS applications need to find a route to customers' mobile phones. They are packaged in SIS files which may be installed over-the-air, via PC connect or in some cases via Bluetooth or memory cards. An alternative is to partner with a phone manufacturer to have the software included on the phone itself. The SIS file route will be a little more difficult from Symbian OS 9, because any application wishing to have any capabilities beyond the bare minimum must be signed via the Symbian Signed program.

Java ME applications for Symbian OS are developed using standard techniques and tools such as the Sun Java Wireless Toolkit (formerly the J2ME Wireless Toolkit). They are packaged as JAR (and possibly JAD) files. Both CLDC and CDC applications can be created with NetBeans. Other tools include SuperWaba, which can be used to build Symbian 7.0 and 7.0s programs using Java.

Nokia Series 60 phones can also run python scripts when the interpreter is installed, with a custom made API that allows for Bluetooth support and such. There is also an interactive console to allow the user to write python scripts directly from the phone. [5]

2.2.6.2 J2ME

Java ME technology was originally created in order to deal with the constraints associated with building applications for small devices. For this purpose Sun defined the basics for Java ME technology to fit such a limited environment and make it possible to create Java applications running on small devices with limited memory, display and power capacity.

Java ME platform is a collection of technologies and specifications that can be combined to construct a complete Java runtime environment specifically to fit the requirements of a particular device or market. This offers a flexibility and co-existence for all the players in the eco-system to seamlessly cooperate to offer the most appealing experience for the end-user.

The Java ME technology is based on three elements:

- A configuration provides the most basic set of libraries and virtual machine capabilities for a broad range of devices,
- A profile is a set of APIs that support a narrower range of devices, and
- An optional package is a set of technology-specific APIs.

Over time the Java ME platform has been divided into two base configurations, one to fit small mobile devices and one to be targeted towards more capable mobile devices like smart-phones and set top boxes.

The configuration for small devices is called the Connected Limited Device Profile (CLDC) and the more capable configuration is called the Connected Device Profile (CDC).

The figure below represents an overview of the components of Java ME technology and how it relates to the other Java Technologies. [6]

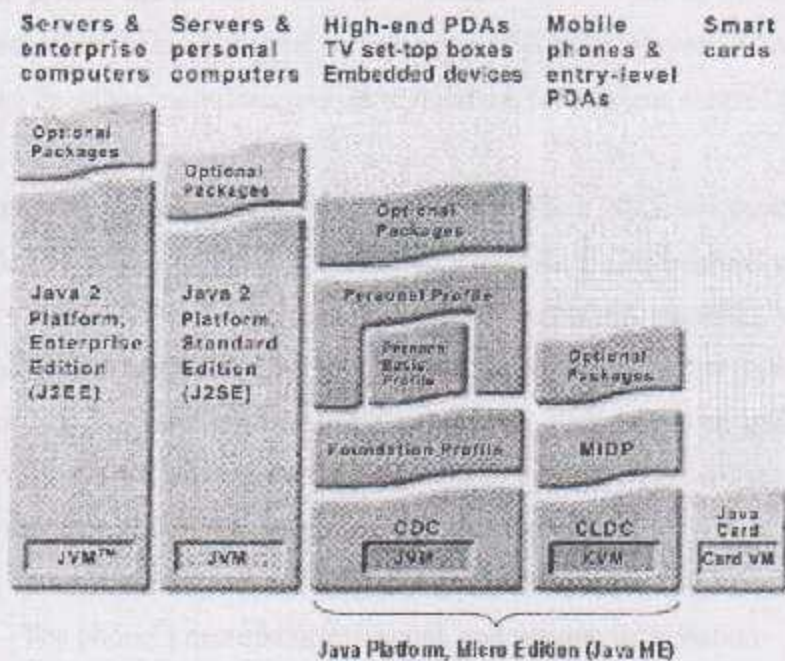


Figure 2.9 Java Platform

2.2.6.2.1 Java ME Platform for Converged Services

The design and set up of the Java ME platform, covering everything from small limited devices with intermittent network connection, to capable on-line mobile devices, the design of the platform makes it flexible and capable to efficiently support the need for services covering all channels for mobility. The basic design makes services easily portable between the different configurations and profiles and the ambition to allow the same service being delivered via different channels can be achieved. [6]

2.2.6.3 AT Commands

AT, or Hayes commands were originally designed by Hayes Microsystems for modem control (Hayes Microsystems 2005). A standard for GSM phones was developed which includes commands to access information such as phone book entries, call logs and SMS messages.

AT commands were designed to control a modem from a PC, and when communicating

with a mobile phone, the phone's internal logic receives and parses the commands. AT commands were originally designed in the early 1980s, by Hayes Microsystems, and were taken up by other manufacturers as a standard for modem control (Durda 2004).

The AT commands specified in the TS 27.007 and TS 27.005 standards theoretically provide access to a large amount of information available in GSM phones and SIM cards (ETSI 2004b, ETSI 2005). The TIA/EIA/IS-707 standard provides a set of AT commands for CDMA phones (TIA 1999). Unfortunately many of the commands cannot be relied upon, as they are optional implementations, and cannot be guaranteed to be present on every phone. In practice however, information such as the following are generally available on GSM phones through AT commands:

- The phone's manufacturer, model, and version information.
- The phone's International Mobile Equipment Identity (IMEI).
- The SIM card's International Mobile Subscriber Identity (IMSI).
- Phone book entries.
- Call log entries.
- Sent and received SMS messages.

AT commands for CDMA phones will generally provide access to a much more limited set of information:

- The phone's manufacturer, model and version information.
- The phone's Electronic Serial Number (ESN). [7]

2.2.7 Bluetooth

Bluetooth is an industrial specification for wireless personal area networks (PANs). Bluetooth provides a way to connect and exchange information between devices such as mobile phones, laptops, PCs, printers, digital cameras and video game consoles via a secure, globally unlicensed short-range radio frequency. [8]

2.2.7.1 List of Applications

- Wireless control of and communication between a cell phone and a hands free headset or car kit. This was one of the earliest applications to become popular.
- Wireless networking between PCs in a confined space and where little bandwidth is required.
- Wireless communications with PC input and output devices, the most common being the mouse, keyboard and printer.
- Transfer of files between devices via OBEX.
- Transfer of contact details, calendar appointments, and reminders between devices via OBEX.
- Replacement of traditional wired serial communications in test equipment, GPS receivers and medical equipment.
- For remote controls where infrared was traditionally used.
- Sending small advertisements from Bluetooth enabled advertising hoardings to other, discoverable, Bluetooth devices.
- Wireless control of a games console – Nintendo's Wii and Sony's PlayStation 3 will both use Bluetooth technology for their wireless controllers.
- Sending commands and software to the LEGO Mindstorms NXT instead of infrared. [8]

2.2.7.2 How Bluetooth Technology Works

Bluetooth wireless technology is a short-range communications system intended to replace the cables connecting portable and/or fixed electronic devices. The key features of Bluetooth wireless technology are robustness, low power, and low cost. Many features of the core specification are optional, allowing product differentiation.

The Bluetooth core system consists of an RF transceiver, baseband, and protocol stack. The system offers services that enable the connection of devices and the exchange of a variety of data classes between these devices.

The Bluetooth RF (physical layer) operates in the unlicensed ISM band at 2.4GHz. The system employs a frequency hop transceiver to combat interference and fading, and provides many FHSS carriers. RF operation uses a shaped, binary frequency modulation to minimize transceiver complexity. The symbol rate is 1 Megasymbol per second (Msps) supporting the bit rate of 1 Megabit per second (Mbps) or, with Enhanced Data Rate, a gross air bit rate of 2 or 3Mb/s. These modes are known as Basic Rate and Enhanced Data Rate respectively.

During typical operation, a physical radio channel is shared by a group of devices that are synchronized to a common clock and frequency hopping pattern. One device provides the synchronization reference and is known as the master. All other devices are known as slaves. A group of devices synchronized in this fashion form a piconet. This is the fundamental form of communication for Bluetooth wireless technology.

Devices in a piconet use a specific frequency hopping pattern which is algorithmically determined by certain fields in the Bluetooth specification address and clock of the master. The basic hopping pattern is a pseudo-random ordering of the 79 frequencies in the ISM band. The hopping pattern may be adapted to exclude a portion of the frequencies that are used by interfering devices. The adaptive hopping technique improves Bluetooth technology co-existence with static (non-hopping) ISM systems when these are co-located.

The physical channel is sub-divided into time units known as slots. Data is transmitted between Bluetooth enabled devices in packets that are positioned in these slots. When circumstances permit, a number of consecutive slots may be allocated to a single packet. Frequency hopping takes place between the transmission or reception of packets. Bluetooth technology provides the effect of full duplex transmission through the use of a time-division duplex (TDD) scheme.

Above the physical channel there is a layering of links and channels and associated control protocols. The hierarchy of channels and links from the physical channel upwards is physical channel, physical link, logical transport, logical link and L2CAP channel.

Within a physical channel, a physical link is formed between any two devices that transmit packets in either direction between them. In a piconet physical channel there are restrictions on which devices may form a physical link. There is a physical link between each slave and the master. Physical links are not formed directly between the slaves in a piconet.

The physical link is used as a transport for one or more logical links that support unicast synchronous, asynchronous and isochronous traffic, and broadcast traffic. Traffic on logical links is multiplexed onto the physical link by occupying slots assigned by a scheduling function in the resource manager.

A control protocol for the baseband and physical layers is carried over logical links in addition to user data. This is the link manager protocol (LMP). Devices that are active in a piconet have a default asynchronous connection-oriented logical transport that is used to transport the LMP protocol signaling. For historical reasons this is known as the ACL logical transport. The default ACL logical transport is the one that is created whenever a device joins a piconet. Additional logical transports may be created to transport synchronous data streams when this is required.

The link manager function uses LMP to control the operation of devices in the piconet and provide services to manage the lower architectural layers (radio layer and baseband layer). The LMP protocol is only carried on the default ACL logical transport and the default broadcast logical transport.

Above the baseband layer the L2CAP layer provides a channel-based abstraction to applications and services. It carries out segmentation and reassembly of application data and multiplexing and de-multiplexing of multiple channels over a shared logical link. L2CAP has a protocol control channel that is carried over the default ACL logical transport. Application data submitted to the L2CAP protocol may be carried on any logical link that supports the L2CAP protocol. [8]

2.3 Air – Condition

2.3.1 Introduction

Air conditioning is a combined process that performs many functions simultaneously. It conditions the air, transports it, and introduces it to the conditioned space. It provides heating and cooling from its central plant or rooftop units. It also controls and maintains the temperature, humidity, air movement, air cleanliness, sound level, and pressure differential in a space within predetermined limits for the comfort and health of the occupants of the conditioned space or for the purpose of product processing.

The term HVAC&R is an abbreviation of heating, ventilating, air conditioning, and refrigerating. The combination of processes in this commonly adopted term is equivalent to the current definition of air conditioning. Because all these individual component processes were developed prior to the more complete concept of air conditioning, the term HVAC&R is often used by the industry. [9]

2.3.2 What is Air Conditioning

Many consider air conditioning just the operation of refrigeration, but actually refrigeration is considered one of Air Conditioning operations. Filtering air and moving air indoor an enclosed conditioned place with a certain speed dividing it in a right way eliminating increased humidity within it and warming it plus increasing humidity percent, is also considered an important and essential aspect of the Air conditioning process.

The air conditioning system also controls the elements related to air conditioning at the same time, and these elements are: air temperature, humidity, air movement and division, dust, microbes, different smells, and harmful gases. The three main elements of air conditioning systems are air temperature and humidity and air movement.

2.3.3 Individual Room Air Conditioning Systems

Individual room or simply individual air conditioning systems employ a single, self-contained room air conditioner, a packaged terminal, a separated indoor-outdoor split unit, or a heat pump. A heat pump extracts heat from a heat source and rejects heat to air or water at a higher temperature for heating. Unlike other systems, these systems normally use a totally independent unit or units in each room. Individual air conditioning systems can be classified into two categories:

- Room air conditioner (window-mounted).
- Packaged Terminal Air Conditioner (PTAC), installed in a sleeve through the outdoor wall.

The major components in a factory-assembled and ready-for-use room air conditioner include the following: An evaporator fan pressurizes and supplies the conditioned air to the space. In tube and- fin coil, the refrigerant evaporates, expands directly indoor the tubes, and absorbs the heat energy from the ambient air during the cooling season; it is called a direct expansion (DX) coil. When the hot refrigerant releases heat energy to the conditioned space during the heating season, it acts as a heat pump. An air filter removes airborne particulates. A compressor compresses the refrigerant from a lower evaporating pressure to a higher condensing pressure. A condenser liquefies refrigerant from hot gas to liquid and rejects heat through a coil and a condenser fan. A temperature control system senses the space air temperature (sensor) and starts or stops the compressor to control its cooling and heating capacity through a thermostat.

The difference between a room air conditioner and a room heat pump, and a packaged terminal air conditioner and a packaged terminal heat pump, is that a four-way reversing valve is added to all room heat pumps. Sometimes room air conditioners are separated into two split units: an outdoor condensing unit with compressor and condenser, and an indoor air handler in order to have the air handler in a more advantageous location and to reduce the compressor noise indoors.

Individual air conditioning systems are characterized by the use of a DX coil for a single room. This is the simplest and most direct way of cooling the air. Most of the individual

systems do not employ connecting ductwork. Outdoor air is introduced through an opening or through a small air damper. Individual systems are usually used only for the perimeter zone of the building. [9]

2.3.3.1 Advantages and Disadvantages

Individual systems have the following disadvantages:

- Temperature control is usually on /off, resulting in space temperature swing.
- Air filters are limited to coarse or low-efficiency filters.
- Local outdoor ventilation air intake is often affected by wind speed and wind direction.
- Noise level is not suitable for critical applications.
- More regular maintenance of coils and filters is required than for packaged and central systems.

Individual room and unitary packaged air conditioning systems are both self-contained, factory made, packaged systems using direct expansion coils for cooling. However, individual systems are single-zone units, whereas a packaged system can be either a single-zone or multi zone unit. Individual systems have no supply and return ducts. Also, individual systems provide far smaller heating-cooling capacities, poorer outdoor ventilation intake, lower-efficiency filters, simpler control system, and far less energy conservation measures than packaged systems. Room air conditioners are often window-mounted. Most of the packaged units are rooftop units. [9]

2.3.4 Air Conditioning Devices

Air conditioning devices have all the necessary devices that work on conditioning the air and taking care of the necessary relaxation needed. A conditioning device has all or some of these units:

- Heating Unit: it works on increasing air temperature by passing air over hot surfaces.
- Cooling Unit: works on decreasing air temperature by passing air over cold surfaces.
- Humidification Unit: works on Increasing dry air humidity by adding a sum of vapor for it to become comfortable for winter.
- Drying Units: works on absorbing vapor from air using chemical objects.
- Filtering unit: is used for cleansing air from dust.
- Ventilation Units: it pushes and mixes air so that it can reach different places in a room or building.
- Control Unit: it organizes temperature, humidity and air speed indoor a conditioned place and observing these elements so that they do not go beyond a certain limit.
- Sterilizing Unit: this unit is usually used in sterilizing the air of operation rooms and central processing room.

Figure 2.10 Air Conditioning components

2.3.5 Air Conditioning System

2.3.5.1 Compressor

The process of Air Conditioning is done through a group of units that work together to provide the right air in order for the human being to feel comfortable, these units make what is called air-conditioning system. To control this system one should first study its main components and the way they work combined. [9]

2.3.6 Components of the Air Conditioning System

2.3.6.1 Compressor

The compressor in the previous figure 2.10 represents the heart of an air conditioning system, a compressor is mainly a pump that is used to increase fluid temperature so that it can be used directly or to be moved and distributed.

An air conditioning system represents an open system which includes two parts one with high pressure and the other low pressure as the compressor is considered an electronic pump it coordinates the flow of the refrigeration fluid between the two parts mentioned earlier, usually the refrigeration fluid is refrigerant gas. After pressurizing the refrigeration fluid it is sent to the condenser.

After pressurizing the refrigeration fluid it is sent to the condenser. The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

The condenser is a coil of pipe through which the refrigerant gas flows. The condenser is usually located outdoors where the air is hot. The condenser is connected to the compressor by a pipe.

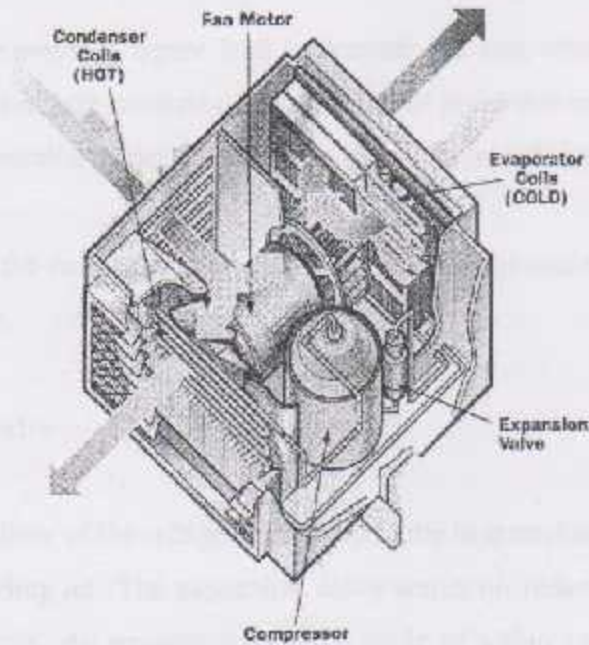


Figure 2.10 Air - Condition components

2.3.6.1 Compressor

The compressor in the previous figure 2.10 represents the heart of an air conditioning system, a compressor is mainly a pump that is used to increase fluid temperature so that it can be used directly or to be moved and distributed. An air conditioning system represents an open system which includes two parts one with high pressure and the other low pressure as the compressor is considered an electronic pump it coordinates the flow of the refrigeration fluid between the two parts mentioned earlier, usually the refrigeration fluid is refrigerant gas. After pressurizing the refrigeration fluid it is sent to the condenser.

2.3.6.1 Evaporator and Condenser

2.3.6.2 Condenser

The evaporator in the previous figure 2.10 represents the area where the heat is sent to the outdoors air.

A condenser consists of a collection of pipes that are organized either in a serial order or a parallel order in order to resist flow and space needed for heat exchange.

After the fluid passes the condenser it becomes liquid at high pressure then it is passed on to the expansion valve.

2.3.6.3 Expansion valve

This part controls the flow of the refrigeration fluid to the heat exchanger for it exchanges heat with the surrounding air.

The expansion valve works on reducing the refrigeration fluid and its temperature. An expansion valve is made of a thin valve with an internal diameter of 1-1, 5 millimeters; Figure 2.10 shows the expansion valve.

2.3.6.4 Evaporator

The condenser in the previous figure 2.10 represents the area in which heat is absorbed from the surrounding air.

The evaporator and condenser are similar in design but the evaporator does the opposite of the condenser and so both the condenser and evaporator can be called heat exchanger.

Both the evaporator and condenser are surrounded with fins from the outdoor to increase heat exchange and the fins are made of aluminum chips.

2.3.6.5 Fans

The air conditioning system contains two fans: the evaporator fan and the condenser fan.

The evaporator fan works at pushing cold air to the conditioned area and the condenser fan pushes hot air to the conditioned area, which is shown in figure 2.10.

2.3.7 Sensors and Transducers

A sensor is a device that acts as a component in a control system to detect and measure the controlled variable and to send a signal to the controller. A sensor consists of a sensing element and accessories; the term sensing element often refers to that part of the sensor that actually senses the controlled variable. In HVAC&R systems, the most widely used sensors are temperature sensors, humidity sensors, pressure sensors, and flow sensors. Recently, CO₂ sensors, air quality sensors, and occupancy sensors are being used in many new and retrofit projects. Electronic sensors that send electric signals to electronic controllers, Electric output from the sensors is usually expressed in 0 to 10 V dc or 4 to 20 mA. In the selection of sensors, accuracy, sensitivity of response, reliability, long-term stability or drift, required calibration intervals, maintainability, and especially the possibility of contamination by dust particles due to contact with the control medium should be considered. A calibration period of 1 year and longer is often considered acceptable. Location of the sensor affects the sensor output directly. A sensor should be located in the critical area where the controlled variable needs to be maintained within required limits. The sensing element of an air sensor should be well exposed to the air, so that air can flow through the sensor without obstruction. Air sensors should not be affected directly by the supply air stream nor should space air sensors be affected by the outdoor air stream. Sensors should be shielded from radiation and mounted firmly on a structural member or duct wall, free from vibration. If a space air sensor must be located on a concrete column, thermal insulation should be provided between the sensor and the column. [9]

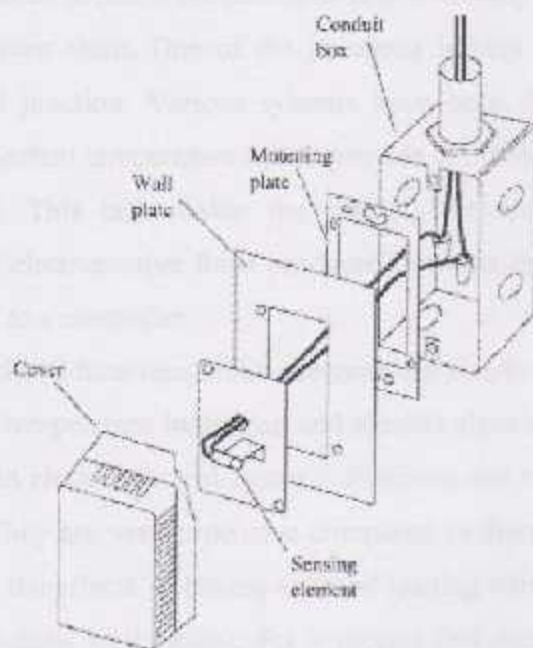


Figure 2.11 A temperature sensor

2.3.7.1 Temperature Sensors

Temperature sensors fall into two categories: those that produce mechanical signals and those that emit electric signals. Bimetal and rod-and-tube sensors that use sensing elements to produce a mechanical displacement during a sensed temperature change either open or close an electric circuit in an electric control system, or adjust the throttling pressure by means of a bleeding nozzle in a pneumatic control system. For a sealed bellows sensor, a change in temperature causes a change in the pressure of a liquid in a remote bulb. The expansion and contraction of vapor then move the mechanism of the controller.

Temperature sensors that produce electric signals, as shown in figure 2.11 are the same as the sensors for temperature measurement and indication.

In addition to resistance temperature detectors (RTDs) and thermostats, sensors sometimes use thermocouples. A thermocouple uses wires of two dissimilar metals, such as copper and constantan, or iron and nickel, connected at two junctions, to generate an

electromotive force between the junctions that is directly proportional to the temperature difference between them. One of the junctions is kept at constant temperature and is called the cold junction. Various systems have been developed to maintain the cold junction at a constant temperature and to provide compensation if the cold junction is not at 32°F (0°C). This task makes the use of thermocouples more complicated and expensive. The electromotive force produced between the two junctions can be used as the signal input to a controller.

Bimetal and rod-and-tube temperature sensors are simple and low in cost. However, they cannot provide temperature indication and electric signals for Direct Digital Control and are often used in electric control systems. Platinum and nickel RTDs are stable, reliable, and accurate. They are very expensive compared to thermostats and need calibration to compensate for the effects of having external leading wires. RTDs are widely adopted in DDC for commercial applications. For a project that needs precise temperature control, RTD is often the choice. High-quality thermostats can provide stable, reliable, and interchangeable temperature sensors, and are also widely used in many commercial applications. [9]

13.7.2 Transducers or Transmitters

A transducer is a device that converts energy from one form to another or amplifies an input or output signal. In HVAC&R control systems, a transducer may be used to convert an electric signal to a pneumatic signal (E/P transducer), e.g., a pneumatic proportional relay that varies its branch air pressure from 3 to 15 psig (20 to 103 kPa/ g) in direct proportion to changes in the electrical input from 2 to 10 V. Also E/P transducers are used between microprocessor-based or electronic controllers and pneumatic actuators. However, a pneumatic signal can be converted to an electric signal in a P/E transducer. For example, a P/E relay closes a contact when the air pressure falls and opens the contact when the air pressure rises above a predetermined value. A transmitter is used to transmit a signal, pneumatic or electric; through air, water, or other fluids. A sensor is also a transmitter.

electromotive force between the junctions that is directly proportional to the temperature difference between them. One of the junctions is kept at constant temperature and is called the cold junction. Various systems have been developed to maintain the cold junction at a constant temperature and to provide compensation if the cold junction is not $\pm 32^{\circ}\text{F}$ (0°C). This task makes the use of thermocouples more complicated and expensive. The electromotive force produced between the two junctions can be used as the signal input to a controller.

Bimetal and rod-and-tube temperature sensors are simple and low in cost. However, they cannot provide temperature indication and electric signals for Direct Digital Control and are often used in electric control systems. Platinum and nickel RTDs are stable, reliable, and accurate. They are very expensive compared to thermostats and need calibration to compensate for the effects of having external leading wires. RTDs are widely adopted in DDC for commercial applications. For a project that needs precise temperature control, RTD is often the choice. High-quality thermostats can provide stable, reliable, and interchangeable temperature sensors, and are also widely used in many commercial applications. [9]

2.3.7.2 Transducers or Transmitters

A transducer is a device that converts energy from one form to another or amplifies an input or output signal. In HVAC&R control systems, a transducer may be used to convert an electric signal to a pneumatic signal (E/P transducer), e.g., a pneumatic proportional relay that varies its branch air pressure from 3 to 15 psig (20 to 103 kPa/ g) in direct proportion to changes in the electrical input from 2 to 10 V. Also E/P transducers are used between microprocessor-based or electronic controllers and pneumatic actuators. However, a pneumatic signal can be converted to an electric signal in a P/E transducer. For example, a P/E relay closes a contact when the air pressure falls and opens the contact when the air pressure rises above a predetermined value. A transmitter is used to transmit a signal, pneumatic or electric; through air, water, or other fluids. A sensor is also a transmitter.

However, the difference between them is that a sensor only senses the signal of the controlled variable and transmits it. [9]

2.3.8 Controllers

A controller receives input from the sensor, compares with the set point or implementation based on its stored computer software, sends an output to or modulates the control device for maintaining a desirable indoor environment. A thermostat is a combination of a temperature sensor and a temperature controller, whereas a humidistat is a combination of a humidity sensor and a humidity controller.

RACs usually use a two-stage thermostat to separate cooling mode and heating mode operations. One of the RAC controls selects the operation mode. Another control adjusts the set point and cycles the compressor on and off to maintain the required set point. The third one changes the fan speed if required.

There is another alternative. The thermostat reduces the indoor fan speed when the space temperature approaches the set point by using a two- or three-stage speed operation. If the space temperature drops further, it cycles the compressor off. If the space temperature drops still further, the indoor fan is finally cycled off.

For a room heat pump system, a two-stage thermostat often selects the heating or cooling mode operation manually. Then it cycles the compressor and evaporator fan separately or simultaneously to maintain a space temperature set point. Usually, there is a 2 to 5°F (1.1 to 2.8°C) difference between heating and cooling set points for room air conditioning systems. If room air conditioners use a two-speed (high-low) or three-speed (high-medium-low) indoor fan and if the fan speed is regulated only a few times a day manually, then the supply volume flow of this RAC is often steady within an hour, and its air system is constant-volume in nature. If the thermostat in an RAC reduces the fan speed when the space temperature approaches its set point automatically, the air system is actually a VAV system. [9]

2.3.9 Air Conditioning Cycle

The air conditioning cycle in figure 2.12 is that same cycle used in refrigeration but can also be used in heating when the objects of the cycle are changed in a reverse way.

The refrigeration cycle starts when the compressor increases pressure and the temperature of the cold refrigerant then passes it to the condenser which sends the heat to the air making the area hot then it changes to liquid at high pressure, after that the gas is passed on a expansion valve, and then the liquid is turned into cold gas at low pressure, then it is passes to the evaporator which absorbs the heat from the air leading to the cooling of surrounding air. The refrigerator returns to the compressor to restart a new cycle.

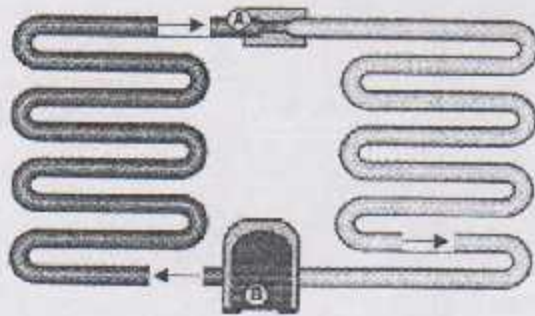


Figure 2.12 Air - Condition cycle

2.3.10 Split-System Air Conditioning

The split air conditioning in figure 2.13 represents a complete air conditioning system and it is the most conditioning type available. The split conditioners are divided into two units: Outdoor unit which contains compressor-the heart of the system, and the indoor unit which switches heat with the air in a room. in the refrigeration cycle the indoor unit represents the heat absorbing unit and the outdoors unit represents the heat rejection unit, in the heating cycle the indoors unit represents the heat rejection unit and the outdoors unit represents the heat absorbing unit.

To reverse the air conditioner from the refrigerator cycle to the heating cycle and vice versa the valve control is used. The control valves job in the air conditioner. These conditioners have many good qualities one which is separating the cooling process from the heating one. The indoor of an air conditioner in a conditioned room and its main components, the indoor unit is put indoors and the outdoor unit works outdoors and both units are connected together with metal tubes.

Variable speed controller

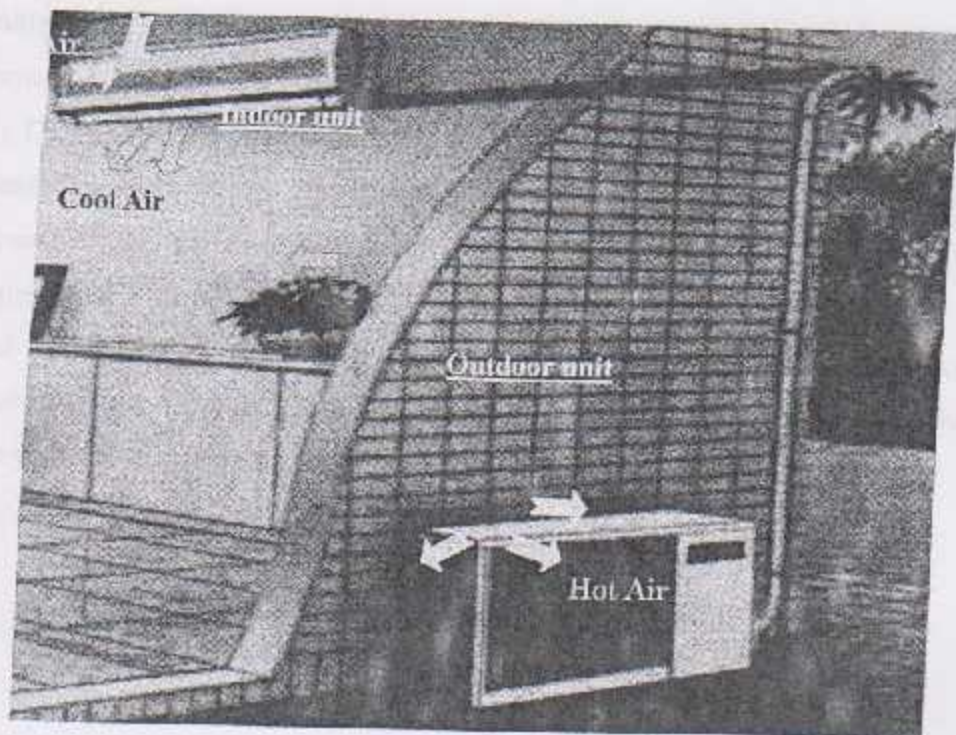


Figure 2.103 Split Air - Condition

2.3.11 Problems Occurring With Air Conditioning (Disadvantages)

One of the most important frequent problems that face the air conditioning system especially when using the system in the winter is that when turning the system on the evaporator is used by the outdoor unit and the condenser is used by the indoor.

After a certain period of time where the system has been used a layer of ice appears on the surface of the evaporator in the outdoor unit as a result of the cold weather outdoors.

Even with there being a sensor on the evaporator for when the temperature reaches a certain degree the cycle reverses in order for the ice to be melted but most of the time the layer of ice is not melted even after reversing the cycle which leads to there being more than one layer of ice on the evaporator which in conclusion leads to a lack of air conditioning efficiency.

2.3.12 Variable Speed Control

Variable speed control of induction motors of the sizes found in screw compressors is performed by an inverter. Inverters work by rectifying the incoming AC mains to a steady DC voltage and then digitally 'chopping' it at high frequency to produce a variable frequency, variable voltage output waveform. This variable frequency waveform allows the speed of the motor to be varied efficiently to match the air demand, and the microprocessor controls allow fast and accurate pressure control.

Special 'switched reluctance drives' are used with switched reluctance motors, but while the technology is different; both approaches use variable speed control and share the same core benefits. Figure 2.14 shows variable speed control [10]

2.4 The 32-bit microcontroller and FPGA

2.4.1 Microcontrollers

A microcontroller is a highly integrated, single-chip microcomputer. Microcomputers include a central processor, instruction program memory, data memory, and other information system, timing, and bus-related features to support a wide range of applications.

Microcontrollers can do the work of many different kinds of logic circuit. Microcontrollers are programmed when to perform the functions that were designed to do. If the design requirements are changed slightly, or even a whole new design is needed, the microcontroller can be reprogrammed to accommodate the change. With a microcontroller, changing the logic function, even changes can be made simply by reprogramming the microcontroller. That is, the software (program) is changed rather than the hardware.

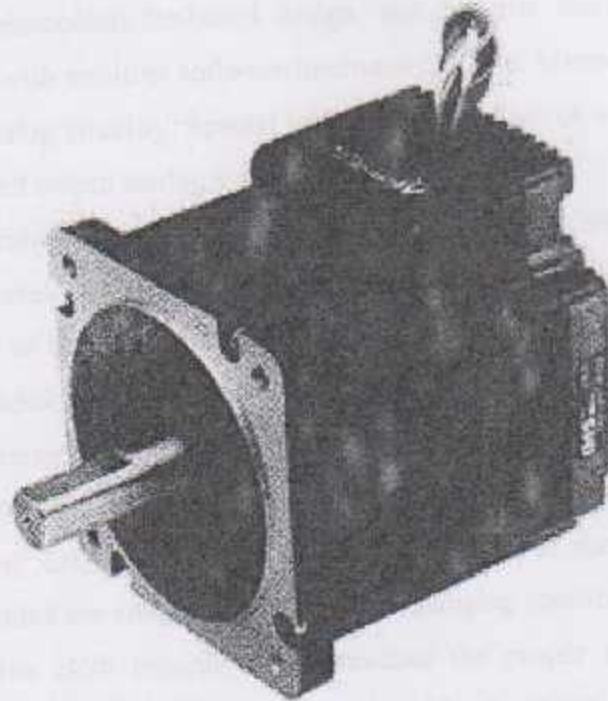


Figure 2.14 Example on speed control

2.4 The Microcontrollers and FPGA

2.4.1 Microcontrollers

Microcontroller is a highly integrated single-chip microcomputer. Microcontrollers include a CPU to process information, program memory to store instructions, data memory to store information, system timing, and input/output sections to communicate with the outdoor world.

A microcontroller can do the work of many different types of logic circuits. Discrete logic circuits are permanently wired to perform the function they were designed to do. If the design requirements are changed slightly, an entire printed circuit board or many boards may have to be redesigned to accommodate the change. With a microcontroller performing the logic functions, most changes can be made simply by reprogramming the microcontroller. That is, the software (program) is changed rather than the hardware

(logic circuits). This makes the microcontroller a very attractive building block in any digital system. With a microcontroller-based design, the designer can simply add a feature set to the product with minimal software/hardware changes. Microcontrollers can also be used to replace analog circuitry. Special interface circuits can be used to enable a microcontroller to input and output analog signals

Microcontrollers can efficiently be used as single-chip replacements in such applications with significantly lower development cost and fast time to market.

One of the great benefits of the electronic revolution is that it brought intelligence, and with it adaptability, to traditional electro-mechanical devices. By continuously gathering information on the performance of the device, its operating environment, and other factors, microcontroller intelligence can determine a new and proper strategy and then command the surrounding device to react. The most important of these are the new intelligent applications which are adapted in real time to changing conditions, such as the iron that senses when the cloth beneath it has reached the proper temperature and moisture, or the vacuum cleaner that adjusts its speed and brush height automatically to changes in carpet nap. [11]

Figure 2.18 Microcontroller Block Diagram

2.4.1.1 Microcontroller Architecture

Figure shows the general block diagram of a microcontroller. In general any microcontroller architecture contains the following main components: [11]

2.4.1.1.1 Central Processing Unit CPU

Central Processing Unit (CPU) is the heart of a microcontroller where all of the arithmetic and logical operations are performed. This is the calculator part of the microcontroller. The CPU gets program instructions from the program memory. [11]

2.4.1.1.2 Program and Data Memory

Program Memory contains a set of CPU instructions organized into a particular sequence to do a particular task. Program Memory is referred to as Read Only Memory (ROM) or OTP/EPROM. OTP or "One-Time Programmable" can be programmed only once and the program is stored permanently, even when the microcontroller power is turned off.

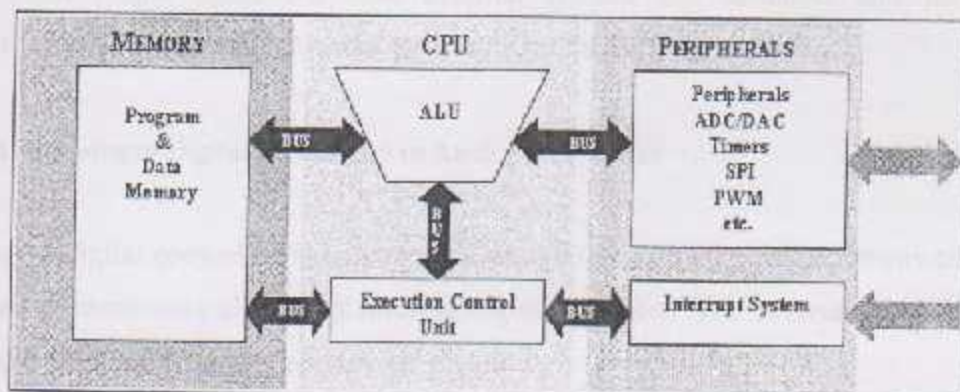


Figure 2.15 Microcontroller General Block Diagram

Program memory enables the microcontroller to immediately begin running its program as soon as it is turned on.

Data Memory can be both read and written and is required for the program stack, data storage, and program variables. This type of memory is commonly referred to as Random Access Memory (RAM). Each memory location has a unique address which the CPU uses to find the information it needs. A typical microcontroller contains both ROM and RAM type memory.

For the storage of data one microcontroller may use different types of memory:

- On-chip and/or external RAM (volatile and/or non-volatile)
- On-chip and/or external FLASH
- EEPROM. [9]

2.4.1.1.3 Inputs/Outputs and Peripherals

Microcontrollers require interface sections to communicate with external circuitry. Input ports allow data and status conditions to be read into the microcontroller while the output ports allow the microcontroller to affect external logic systems. The interfaces between the microcontroller and the outdoor world vary with the application, and may include display units, keypads, switches, sensors, relays, motors, and so on. Peripherals are used for the input output operations with external sensors and actuators and for the communication with external networks over standard buses. [11]

2.4.1.1.3.1 Analog to Digital and Digital to Analog Converters

An analog-to-digital converter (abbreviated ADC, Λ/D or A to D) is an electronic circuit that converts continuous signals to discrete digital numbers. The reverse operation is performed by a digital-to-analog converter (DAC).

Typically, an ADC is an electronic device that converts an input analog voltage to a digital number. The digital output may be using different coding schemes, such as binary and two's complement binary. However, some non-electronic or only partially electronic devices, such as rotary encoders, can also be considered ADCs.

The resolution of the converter indicates the number of discrete values it can produce over the range of voltage values. It is usually expressed in bits.

In electronics, a digital-to-analog converter (DAC or D-to-A) is a device for converting a digital (usually binary) code to an analog signal (current, voltage or electric charge). Digital-to-analog converters are the interface between the abstract digital world and the analog real life. Simple switches, a network of resistors, current sources or capacitors may implement this conversion. [12]

A DAC usually only deals with pulse-code modulation (PCM)-encoded signals. The job of converting various compressed forms of signals into PCM is left to codecs.

The DAC fundamentally converts finite-precision numbers (usually fixed-point binary numbers) into a physical quantity, usually an electrical voltage. Normally the output

voltage is a linear function of the input number. Usually these numbers are updated at uniform sampling intervals and can be thought of as numbers obtained from a sampling process. [13]

2.4.1.2 Microcontroller Applications

Microcontroller's applications are more or less limited only by the user imagination. Microcontrollers now reside in our televisions, keyboards, modems, printers, telephones, cars, household appliances, and every other part of home and work life. The market for microcontrollers continues to expand rapidly, encompassing a wide range of consumer, industrial, automotive, and telecommunications applications. In fact, a typical home today contains over 35 microcontroller-based products. The emergence of new low cost microcontrollers offers a wealth of benefits for today's consumer applications and represents an entirely new profit source for manufacturers. In the past, the high cost of electronics limited the use of microcontrollers to "high tech" applications such as video recorders, stereo systems, and high-end durable goods such as washing machines. Today, the application base has broadened to include systems such as coffee machines, irons, shavers, and cleaners, where the introduction of electronics helps to provide product differentiation and allows the inclusion safety features. [11]

2.4.2 Micro Web – Server

Micro web-servers are microcontroller systems with an integrated chip that contains a processor and ports. It also contains the required TCP/IP protocol stack to establish an Internet connection. Micro web-servers can be embedded in remote control systems to network-enable remote applications.

Many web-server options are available to construct, achieve the connectivity and controlling tasks. Pico web-servers and Micro Web-servers are candidate categories to be used in such systems.

Micro web server applications now include remote monitoring, industrial automation and process control, home automation, lighting control, environmental monitoring, remote telemetry and test and lab equipment monitoring.

While both the low-end and high-end solutions have their place, the reality is that in the embedded world most applications have the following requirements: Low-cost, Single-chip solution, and 8-bit or 16-bit performance. So, the requirement here is for a high-performance chip that is very code efficient. [11]

2.4.3 PIC Microcontroller

PIC is a family of RISC microcontrollers made by Microchip Technology, regarded that PIC stands for Peripheral Interface Controller, the 8-bit PIC was developed in 1975 to improve performance of the overall system by offloading I/O tasks from the CPU. The PIC used simple microcode stored in ROM to perform its tasks.

PICs use a RISC instruction set, which varies in length from about 35 instructions for the low-end PICs to about 70 instructions for the high-end PICs. The instruction set includes instructions to perform a variety of operations on the accumulator and a constant or the accumulator and a memory location, as well as for conditionally executing code and jumping/calling other parts of the program and returning from them, and specific hardware features like interrupts and one low-power mode called sleep. Microchip provides a freeware IDE package called MPLAB, that also includes a software simulator as well as an assembler. [14]

2.4.3.1 ToothPIC

ToothPIC is a PIC microcontroller and BlueMatik radio combination, ToothPIC combines a PIC18LF6720 microcontroller and a BlueMatik Bluetooth radio, and is preloaded with ToothPIC Services including FlexiPanel user interface server.

The FlexiPanel User Interface server takes advantage of FlexiPanel Ltd's unique user interface protocol. Remote devices such as PCs, PDAs and cellphones can connect to

ToothPIC using a freely available FlexiPanel Client software layer. Once connected, the FlexiPanel Server tells the remote device what controls to display on its user interface. Both the user and ToothPIC can then modify the user interface controls as required. [15]

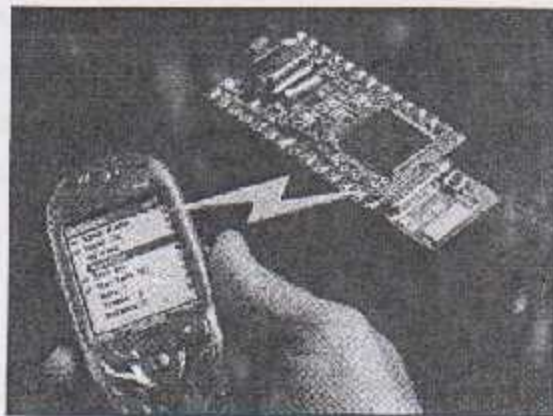


Figure 2.16 ToothPIC with mobile

2.4.4 FPGA

Field-Programmable Device (FPD) refers to any type of integrated circuit used for implementing digital hardware, where the chip can be configured by the end user to realize different designs. Programming of such devices often involves placing the chip into a special programming unit, but some chips can also be configured in-system.

The use and development of new types of sophisticated field-programmable devices (FPDs), has changed the process of designing digital hardware dramatically over the past few years. In the previous, design was done using board-level designs using SSI chips and basic gates. Today virtually every digital system design consists of high density devices. This includes almost all types of digital devices like processors, memory, logic circuits such as state machine controllers, counters, registers, and decoders. When such circuits are destined for high-volume systems they have been integrated into high-density gate arrays. However, gate array costs are too expensive and gate arrays take too long to manufacture for prototyping. For these reasons, most prototypes, and also many production designs are now built using FPDs. The most compelling advantages of FPDs are instant manufacturing turnaround, low start-up costs, low financial risk and ease of

design changes. The market for FPDs has grown dramatically over the past decade to the point where there is now a wide assortment of devices to choose from.

A Programmable Logic Array (PLA) is a relatively small FPD that contains two levels of logic, an AND-plane and an OR-plane, where both levels are programmable. PAL on the other hand is a Programmable Array Logic (PAL) which has a programmable AND-plane followed by a fixed OR-plane. An SPLD refers in general to PLA or PAL.

Now days more complex programming array technologies are used, this includes CPLDs and FPGAs. CPLD, a more Complex PLD, consists of an arrangement of multiple SPLD-like blocks on a single chip.

Field-Programmable Gate Arrays, FPGAs, on the other hand are FPDs featuring a general structure that allows very high logic capacity. Whereas CPLDs feature logic resources with a wide number of inputs (AND planes), FPGAs offer more narrow logic resources. FPGAs also offer a higher ratio of flip-flops to logic resources than do CPLDs.

Both FPGAs and CPLDs uses Logic Blocks, a relatively small circuit block that is replicated in an array, to implement circuits after decomposing of each circuit and mapping it into a logic block. These logic blocks are usually connected using an interconnection wiring and Programmable Switch resources. The amount of digital logic, logic blocks, that can be mapped into a single FPGA is called Logic Capacity. This is usually measured in units of equivalent number of gates in a traditional gate array. Logic Density is the amount of logic per unit area.

Speed / Performance are measures of the maximum operable speed of a circuit when implemented in an FPGA or CPLD. For combinational circuits, it is set by the longest delay through any path, and for sequential circuits it is the maximum clock frequency for which the circuit functions properly. [16]

2.4.4.1 Evolution of CPLD and FPGA

Manufacturing standard digital systems usually takes a lot of time because each design change required that the wiring be redone which usually meant building a new printed circuit board. This is problem can be solved by placing an unconnected array of AND-OR gates in a single chip called a programmable logic device (PLD).

The PLD contained an array of fuses that could be blown open or left closed to connect various inputs to each AND gate. PLD can be programmed with a set of Boolean sum-of-product equations so it would perform the logic functions in a system. Hence no need to change a printed circuit board connections, Figure 2.17.

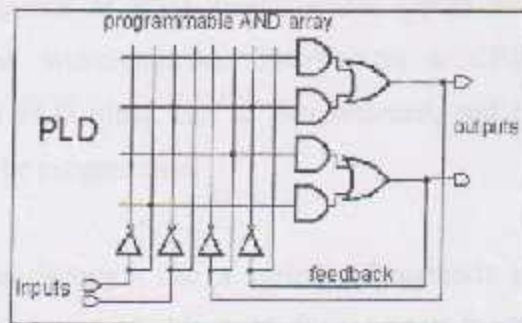


Figure 2.17 PLD Architecture

Simple PLDs can only handle up to 10–20 logic equations, not fitting a very large logic design into just one of them. Hence the designer should figure out how to break larger designs apart and fit them into a set of PLDs. This is a time-consuming and needs the interconnection of the PLDs with wires. Wiring is a big problem because eventually some design change that couldn't be handled just by reprogramming the PLDs and then building a new circuit board. Building much larger programmable chips called complex programmable logic devices (CPLDs) and field-programmable gate arrays (FPGAs) could essentially solve the problem and the system on a single chip idea can be more feasible, Figure 2.18.

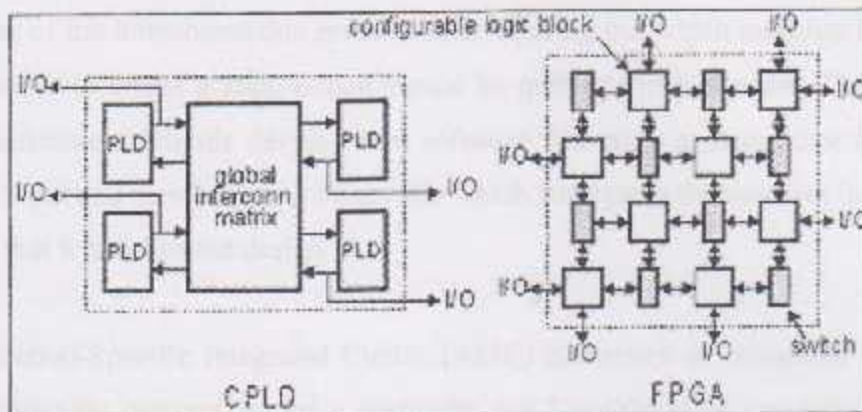


Figure 2.18 CPLD and FPGA Architecture

A CPLD contains a bunch of PLD blocks whose inputs and outputs are connected together by a global interconnection matrix. So a CPLD has two levels of programmability: each PLD block can be programmed, and then the interconnections between the PLDs can be programmed

CPLD and FPGA manufacturers use a variety of methods to make the connections between logic blocks. Some make chips with fuses or anti-fuses that are programmed by passing a large current through them. These types of CPLDs and FPGAs are one-time programmable (OTP) because you can't rewire them internally once the fuses are blown. Other manufacturers make the connections using pass transistors that are opened or closed by storing a charge on their gate electrodes using a high-voltage pulse. This type of programmable device resembles an EPROM or EEPROM that can be erased and then placed in a special programmer socket and reprogrammed, unless the CPLD or FPGA chips are soldered into a circuit board.

Other manufacturers use static RAM or Flash bits to control the pass transistors for each interconnection. By loading each bit with a 1 or a 0, by controlling whether the switch is closed or opened and, therefore, whether two logic elements are connected or not. CPLDs and FPGAs built using RAM/Flash switches can be reprogrammed without removing them from the circuit board. They are often said to be in-circuit or "in-system" reconfigurable or programmable.

Regardless of the interconnection method used, figuring out which switches to open and close in order to create a logic circuit would be quite difficult process. That's why the chip manufacturers provide development software that takes a description of the logic design as input and then outputs a binary file which configures the switches in a CPLD or FPGA so that it acts like the design.

An Application-Specific Integrated Circuit (ASIC) comprises an integrated circuit (IC) with functionality customized for a particular use (equipment or project), rather than serving for general-purpose use. For example, a chip designed solely to run a cash register is an ASIC. In contrast, a microprocessor is not application-specific, because users can adapt it to many purposes. [16]

2.4.4.2 Comparison Between FPGA and CPLD

- FPGAs are "fine-grain" devices. That means that they contain a lot of tiny blocks of logic with flip-flops (up to 100000). CPLDs are "coarse-grain" devices. They contain relatively few large blocks of logic with flip-flops(a few 100's max).
- FPGAs are RAM based. They need to be "downloaded" (configured) at each power-up. CPLDs are EEPROM based and are active at power-up.
- CPLDs have a faster input-to-output timings than FPGAs.
- FPGAs have special routing resources to implement efficiently binary counters and arithmetic functions (adders, comparators...).
- FPGAs can contain very large digital designs, while CPLDs can contain small designs only. [16]

2.4.4.3 Comparison Between FPGA and ASIC

- The blocks that are used in FPGA are similar in structure to the gate arrays used in some ASIC's, however standard gate arrays are configured and fixed during



manufacture, the configurable logic blocks in new FPGA's can be rewired and reprogrammed repeatedly and very fast.

- FPGA is generally slower than their ASIC counterparts, and draws more power. However, they have several advantages such as a shorter time-to-market, and lower development costs.
- An ASIC can be made that is a so-called hard copy of an FPGA. Hence an integrated circuit with the same functionality as the FPGA, but faster and consuming less power. [16]

2.4.4.4 Hardware Description Languages (HDL)

Hardware description language or HDL is any language from a class of computer languages for formal description of electronic circuits. It can describe circuit's operation, its design, and tests to verify its operation by means of simulation.

A HDL is a standard text-based expression of the temporal behavior and/or (spatial) circuit structure of an electronic system. In contrast to a software programming language, an HDL's syntax and semantics include explicit notations for expressing time and concurrency which is the primary attributes of hardware. Languages whose only characteristic is to express circuit connectivity between hierarchies of blocks are properly classified as netlist languages.

A simulation program, designed to implement the underlying semantics of the language statements and the progression of time, provides the hardware designer with the ability to model a piece of hardware before it is created physically. This execution capability makes it seem like the language is used to program something. Simulators are capable of supporting discrete event (digital), and continuous time (analog) modeling exist and HDL's targeted for each are available.

HDLs are used to design two kinds of systems. First, they are used to design a dedicated integrated circuit, such as a processor or other kind of digital logic chip. In this case,

HDL specifies a model for the expected behavior of a circuit before that circuit is designed and built. The end result is a silicon chip that would be manufactured.

The second use involves programming the programmable logic devices, such as the FPGAs. The HDL code is fed into a logic compiler, and the output is uploaded into the device. The unique property of this process and of the programmable logic in general, is that it is possible to alter the code many times, compile it, and upload into the same device for testing.

An HDL program may be tested in hardware, such as by uploading it into a programmable logic device or even by producing a chip based on its specification. However, this is generally a very time-consuming and costly process, and generally the bulk of testing and debugging is done using a program called simulator. The simulator maintains a resettable clock, similar to the real clock of a digital device, and allows the designer to print out the values of various registers over time in order to debug the code.

[16]

2.4.4.4.1 HDL and Programming languages

An HDL is analogous to a software programming language, but with subtle differences. Both types of language are processed by a compiler. An HDL compiler often works in several stages, first producing a logic description file in a proprietary format, then converting that to a logic description file in the industry-standard EDIF format, then converting that to a JEDEC-format file. The JEDEC file contains instructions to a PLD programmer for building logic. HDLs used for digital circuit design include: Verilog HDL, VHDL, Advanced Boolean Expression Language ABEL HDL, Altera HDL, AHDL, and CUPL. The current trend is to move away from proprietary HDLs and towards the two leading standards, VHDL and Verilog HDL. [16]

2.4.4.4.2 Verilog HDL

Verilog is one of the major Hardware Description Languages (HDL) used by hardware designers in industry and academia. VHDL was made an IEEE Standard in 1987, and Verilog in 1995. Verilog is very C-like language. Verilog HDL was a proprietary language of Cadence.

Verilog HDL allows a hardware designer to describe designs at a high level of abstraction such as at the architectural or behavioral level as well as the lower implementation levels (i.e., gate and switch levels) leading to Very Large Scale Integration (VLSI) Integrated Circuits (IC) layouts and chip fabrication. [16]

2.4.4.5 FPGA and CPLD Implementation Procedure

Implementing a logic design with the FPGA or CPLD development software usually consists of the following steps as shown in Figure 2.19.

Figure 2.19 FPGA Programming

- 1. The logic design designer begins by writing down using any of the HDL languages like Verilog. Usually design can be described using any of the languages.
- 2. A logic synthesizer program is used to transform the HDL code into a gate-level netlist. The netlist is just a collection of logic functions, gates, and how they are interconnected.
- 3. Different implementation tools can be used to map the logic gates and interconnections into the FPGA. The manufacturer's logic blocks offered in the FPGA can be further decomposed into look-up tables (LUTs) that perform logic operations. The LUTs and LUTs are loaded with various routing resources. The mapping tool defines the routing paths and groups them into the LUTs and then the place-and-route tool assigns the logic cells to the LUTs and the routing resources in the routing resources to connect the logic elements.

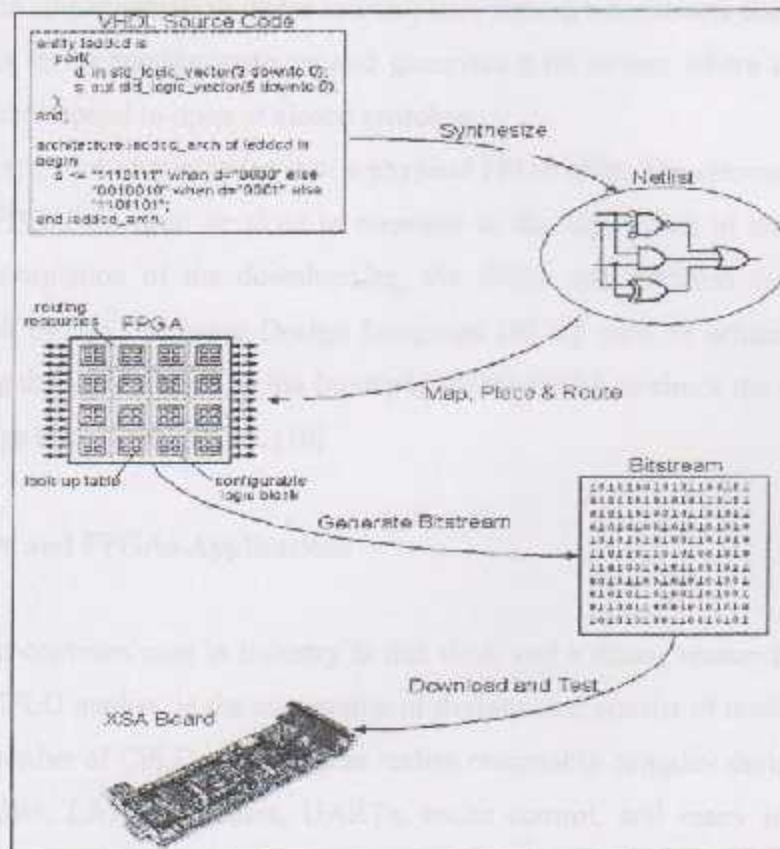


Figure 2.19 FPGA Programming

- The logic circuit design or behavior is first done using any of the HDL languages like Verilog. Also this design can be described using a schematic editor.
- A logic synthesizer program is used to transform the HDL or schematic into a netlist. The netlist is just a description of the various logic gates in the design and how they are interconnected.
- Different implementation tools can be used to map the logic gates and interconnections into the FPGA. The configurable logic blocks (CLBs) in the FPGA can be further decomposed into look-up tables (LUTs) that perform logic operations. The CLBs and LUTs are mixed with various routing resources. The mapping tool collects the netlist gates into groups that fit into the LUTs and then the place & route tool assign the gate collections to specific CLBs while opening or closing the switches in the routing matrices to connect the gates together.

- Once the implementation phase is complete, a program extracts the state of the switches in the routing matrices and generates a bit stream where the ones and zeroes correspond to open or closed switches.
- The bit stream is downloaded into a physical FPGA chip. The electronic switches in the FPGA are open or close in response to the binary bits in the bit stream. Upon completion of the downloading, the FPGA will perform the operations specified by the Hardware Design Language (HDL) code or schematic. Hence applying the input signals to the Input pins of the FPGA to check the operation of the design at the Output pins. [16]

2.4.4.6 CPLDs and FPGAs Applications

One of the most common uses in industry at this time, and a strong reason for the large growth of the CPLD market, is the conversion of designs that consist of multiple SPLDs into a smaller number of CPLDs. CPLDs can realize reasonably complex designs, such as graphics controller, LAN controllers, UARTs, cache control, and many others. As a general rule-of-thumb, circuits that can exploit wide AND/OR gates, and do not need a very large number of flip-flops are good candidates for implementation in CPLDs. A significant advantage of CPLDs is that they provide simple design changes through re-programming. With in system programmable CPLDs it is even possible to re-configure hardware without power-down.

FPGAs have gained rapid acceptance and growth over the past decade because they can be applied to a very wide range of applications. A list of typical applications includes: random logic, integrating multiple SPLDs, device controllers, communication encoding and filtering, small to medium sized systems with SRAM blocks, and many more.

Other interesting applications of FPGAs are prototyping of designs later to be implemented in gate arrays, and also emulation of entire large hardware systems.

Another promising area for FPGA application, which is only beginning to be developed, is the usage of FPGAs as custom computing machines. This involves using the

programmable parts to “execute” software, rather than compiling the software for execution on a regular CPU. [16]

Chapter Three

Architectural Design

3 Architectural Design

3.1 Introduction	51
3.2 Design Objectives	57
3.3 System Partitioning/Requirements	28
3.4 System Capabilities	35
3.5 System Block Diagram	40
3.6 How Does the System Work?	49

3

Chapter Three

Architectural Design

3 Architectural Design

3.1 Introduction	57
3.2 Project Objectives	57
3.3 System Functional Requirements	58
3.4 System Components	58
3.5 General Block Diagram	60
3.6 How Does the System Work?	62

3.1 Introduction

Nowadays the Air – Condition becomes one of the most important devices in the home, office, factory, etc.

The system which we were designed must achieve the efficiency and easiest in use to be a step in the development of our new world.

The number of people who use mobile phones and technologies are increasing every day, so the systems designers must take this point when designing remote control systems.

Also the FPGA and microcontroller are a main block in any control system, the range of things that FPGA or microcontroller provides to us enables to fully control any device.

The system which we were designed will use specific commands to communicate with Air – Condition through microcontroller from mobile which reflex the user needs. The communication method that may be used to communicate with mobile is SMS through F-bus.

The user needs/functions to control the Air-Condition can be classified as:

- Remotely Turn-On the Air-Condition.
- Remotely Turn-Off the Air-Condition.
- Remotely Set the temperature.
- Remotely Read the temperature.

3.2 Project Objectives

This projects aims at achieving the following objectives:

- Use mobile as remote control.

There are many ways in which we can use the mobile as remote control, like GPRS, SMS, Bluetooth, etc. in this system we will use SMS.

- Controlling Air – Condition.

In controlling the Air – Condition we deal with elements in the Air – Condition that enables us to achieve the user requirement specified in this system. An example on elements is relays, sensors, etc.

- Use Microcontroller Technology for controlling different devices.

In this system we choose the Air – Condition as an example to control.

- Interface mobile phone with controller systems using FBUS technology.

There are two ways in which the system controller can be interfaced to the mobile, one of them is serially (F-bus), and the other is wirelessly (Bluetooth). In this system we were chosen the FBUS.

3.3 System Functional Requirements

The system must be able to do the following:

- Remotely Switch on the Air – Condition
- Remotely Switch off the Air – Condition
- Remotely Set the temperature.
- Remotely Read the temperature.

3.4 System Components

In this project we will use two peer mobile phones which communicate directly with each other using the GSM system to control the home devices through the specified control hardware.

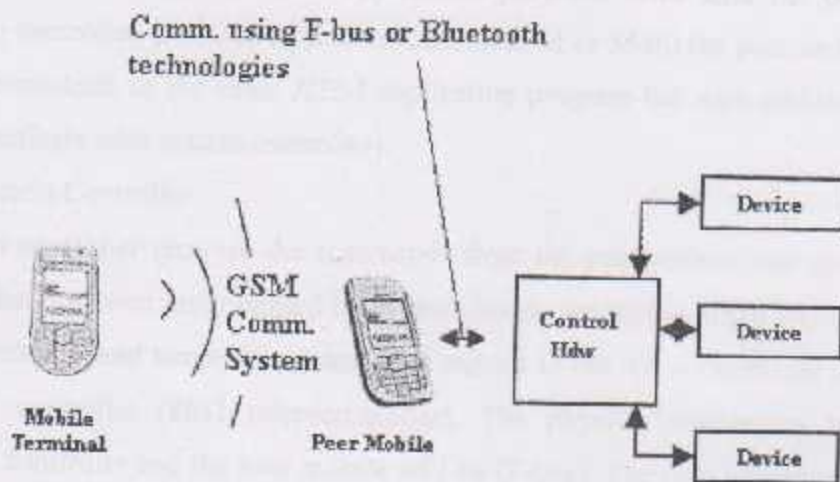


Figure 3.1 Remote device control using peer Mobile phones through GSM system

As can be seen from Figure 3.1, we can group the project components under the following categories: the Client (or Remote side) using a mobile phone, the Peer Mobile at the home side. The GSM communication system between the Peer mobile phones, the Hardware Controlling System (the Controller) and the controlled devices (the Air-condition in our project). These main components are linked to each other with the appropriate communication channel. In the following paragraphs we will describe the general function of each of these components, and in the next chapter a more comprehensive design options and explanations will be presented.

- Client Mobile (Mobile Terminal)

The client mobile is a mobile which is used to remote control the system. It is also reflex the user needs.

This mobile use the GSM network to control the system by developing a program that use the text messaging service (SMS) to send commands controlling the system.

- Peer Mobile

The peer mobile is directly attached to the system controller. It receives the commands from the client mobile (Mobile Terminal) by GSM to control the system by SMS text messaging. If the on – line method is used to control the system then the

peer mobile must have a J2EM application program. And send the commands to system controller. In the two previous cases (J2EM or SMS) the peer mobile may use (AT commands or the same J2EM application program but with additional tasks to communicate with system controller).

- System Controller

System controller receives the commands from the peer mobile and according to a logic that has been programmed by it specifies the operation (Turn on, Turn off, Set temperature, Read temperature) and send signals to the Air - Condition to do it. The system controller (8051 microcontroller). The physical connection between the system controller and the peer mobile will be (F-bus). The next chapter considers all design options using different controllers and communication techniques.

- Air - Condition

The Air- Condition is the device that has been chosen to be controlled, it can be replaced by any other device in the smart home, but as said before because of its importance in the smart home we choose it. The Air - Condition receive signals from the system controller and the system controller read signals from it.

3.5 General Block Diagram

The general block diagram shows the main blocks in the project, as seen from the next figure that there are three blocks the GSM Network block that includes the two mobiles the peer and terminal mobile, the system controller (8051 microcontroller) which is responsible for communicating the peer mobile and the device which is the Air - Condition, and the last block is the Air - Condition which the device to be controlled. While the next sub section shows the system block diagram which is a more detailed diagram shows more specific information about the project.



Figure 3.2 General Block Diagram

3.5.1 System Block Diagram

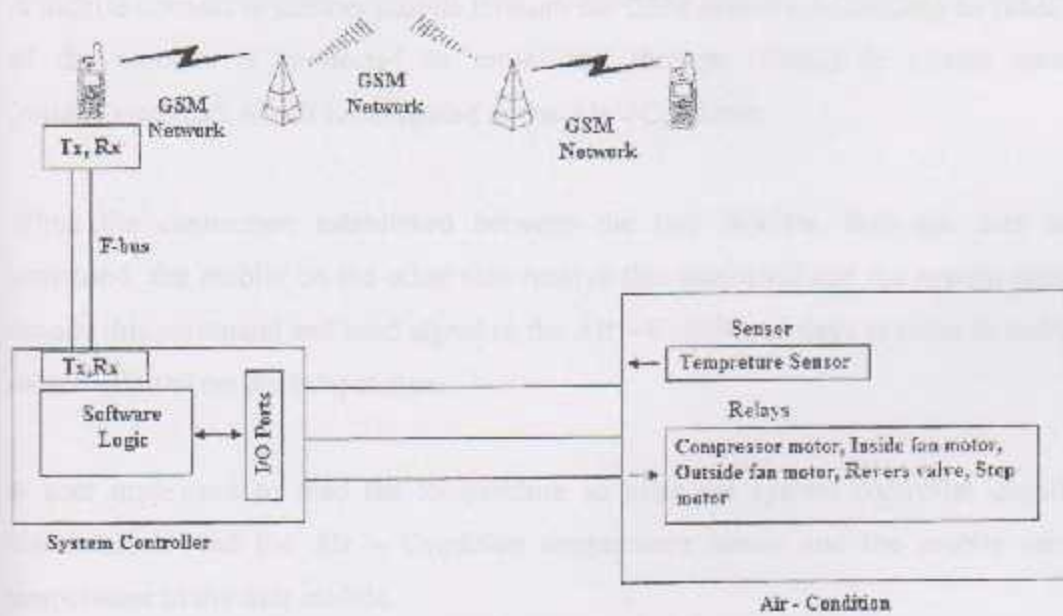


Figure 3.3 System Block Diagram with F-bus

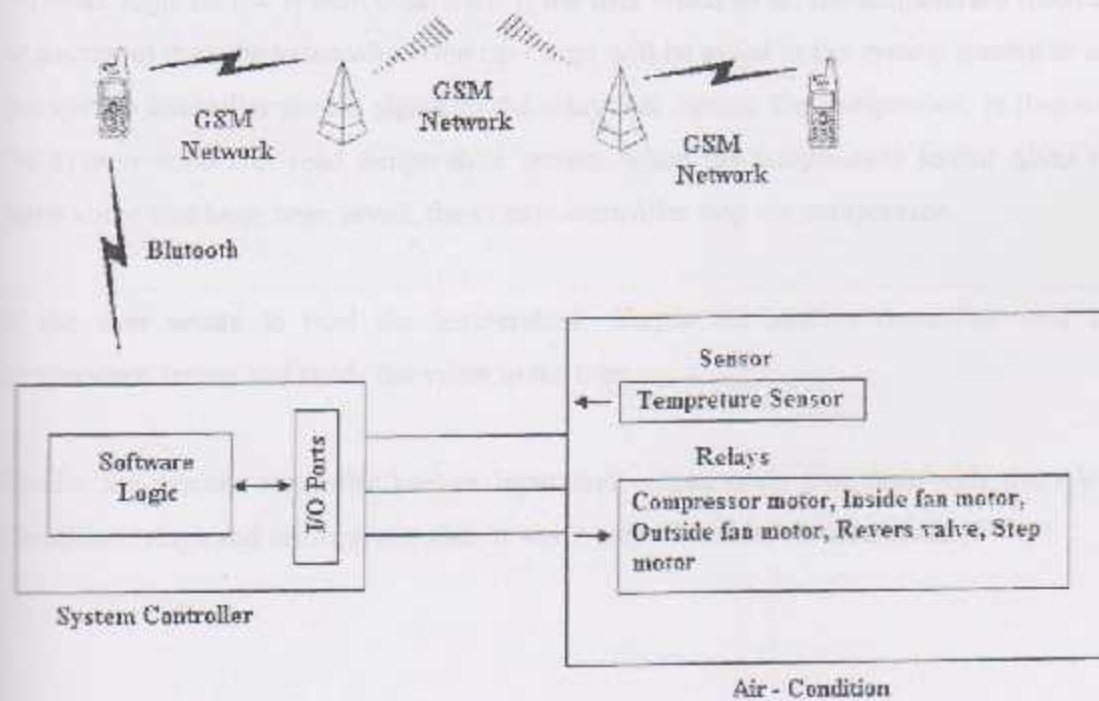


Figure 3.4 System Block Diagram with Bluetooth

3.6 How Does the System Work?

A mobile connect to another mobile through the GSM network by sending an (SMS), one of the mobiles is connected in some way through (F-bus) to system controller (microcontroller), which is connected to the Air – Condition.

When the connection established between the two mobiles, then the user send a command, the mobile on the other side receive this command and the system controller decode this command and send signal to the Air – Condition relays in order to switch on, switch off, and set the temperature.

A user may need to read the temperature so after the system controller decode the command, it read the Air – Condition temperature sensor and the mobile send the temperature to the user mobile.

A system controller (microcontroller) receives input from the sensor. According to the software logic on the system controller, if the user wants to set the temperature (increase or decrease) then the value which the user send will be saved in the system controller and the system controller send a signal to the relay that control the compressor, in that time the system controller read temperature sensor, when the temperature sensor gives the same value that have been saved, the system controller stop the compressor.

If the user wants to read the temperature, simply the system controller read the temperature sensor and sends the value to the user.

Finally the system controller has an input and output ports that deal with the Air – Condition relays and sensors, and also in some way connected to the mobile (F-bus).

4

Chapter Four

Detailed System Design

4 Detailed System Design

4.1 Introduction	64
4.2 System Controller Design Options	64
4.3 Mobile to Mobile Design Options	70
4.4 Interfacing Circuits.....	73

4.1 Introduction

Different design options can be considered in order to achieve our goals in controlling the Air – Condition system.

All design options will use the same GSM technology protocol options in sending or receiving the commands to control the system. These options are explained in section 4.2; section 4.3.1 to section 4.3.2 contains different design options for the system.

Also this chapter contains the Interfacing circuits and project schematic, while choosing the system controller the 8051 microcontroller and SMS to send the commands from the user to system controller.

4.2 System Controller Design Options

There are different design options for the system controller; this section illustrates each one of them.

4.2.1 FPGA Controller

One of the options is to use the FPGA as a controller for the system; the following figure shows the FPGA attached to the system.

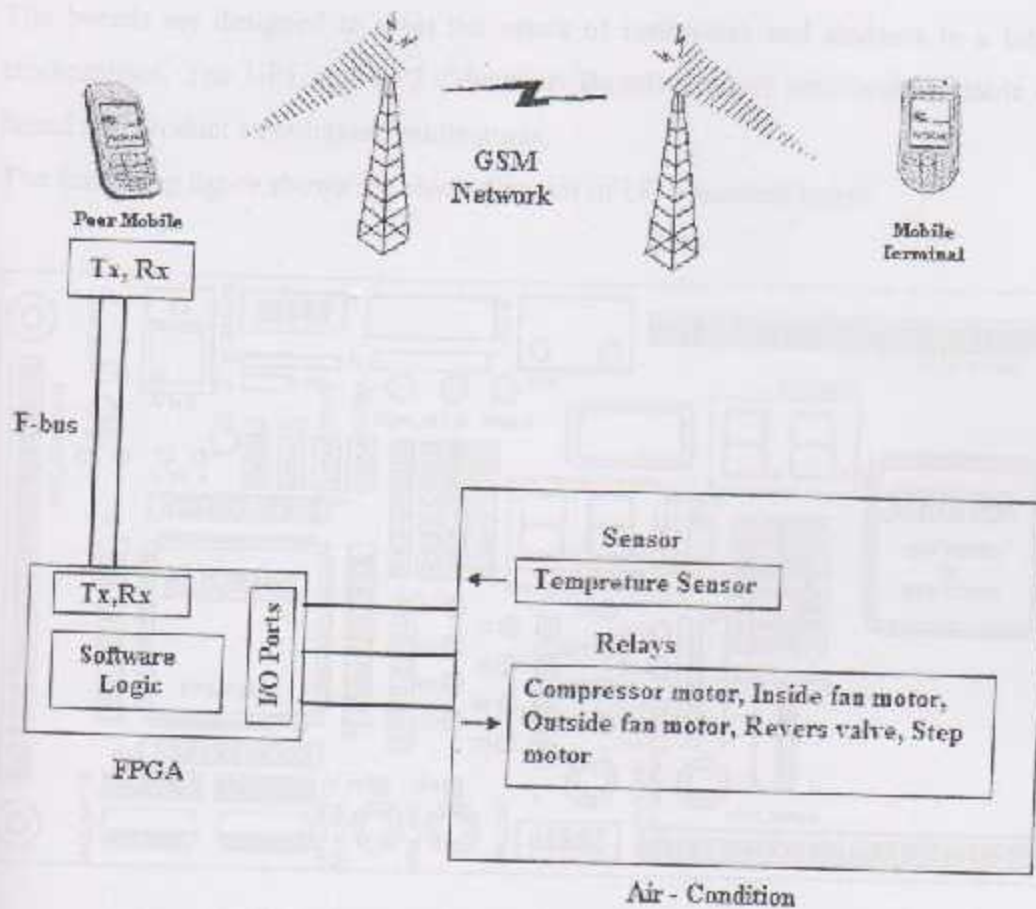


Figure 4.1 FPGA controller

The following paragraphs described two kinds of FGPA boards which can be used to control the system.

4.2.1.1 Altera

Altera is the name of company that produces education board based on FPGA technology. We think of UP2 education board as design option for the system controller. The UP1 and UP2 Education Boards are stand-alone experiment boards. When used with the MAX-PLUS II University software, the boards provide a superior platform for learning digital logic design using industry standard development tools and PLDs.

The boards are designed to meet the needs of instructors and students in a laboratory environment. The UP1 and UP2 Education Boards support both look-up table (LUT)-based and product term-based architectures.

The following figure shows the block diagram of UP education board.

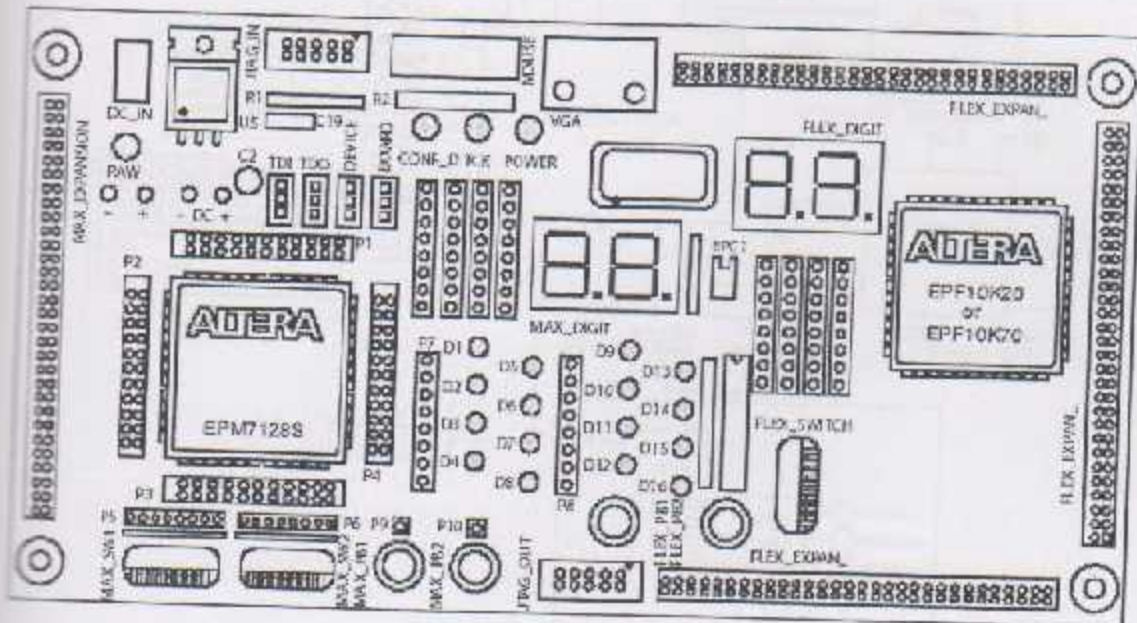


Figure 4.2 UP education board block diagram

4.2.1.2 Xilinx

Xilinx is the name of company that produces education board based on FPGA technology. We think of Virtex – II education board as design option for the system controller.

Virtex-II Prototype Platforms are prototype and demonstration boards that allow designers to investigate and experiment with the features of Virtex-II series FPGAs (chains of FPGAs and serial PROMs). The following figure shows the block diagram of Virtex – II education board.

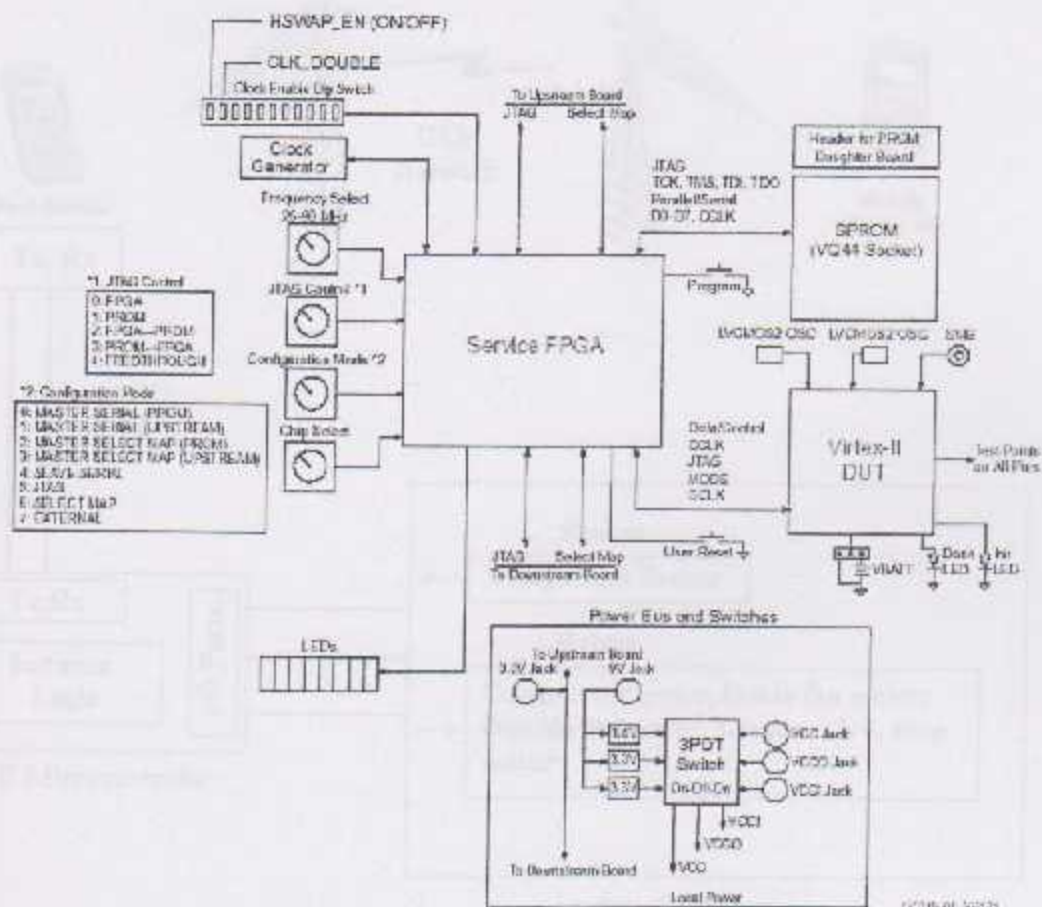


Figure 4.3 Virtex - II prototype platform block diagram

4.2.2 PIC Controller

The PIC microcontroller can be used as a system controller; also there are different ways to configure the PIC to the system.

The next figure is a diagram of PIC microcontroller connected to the system.

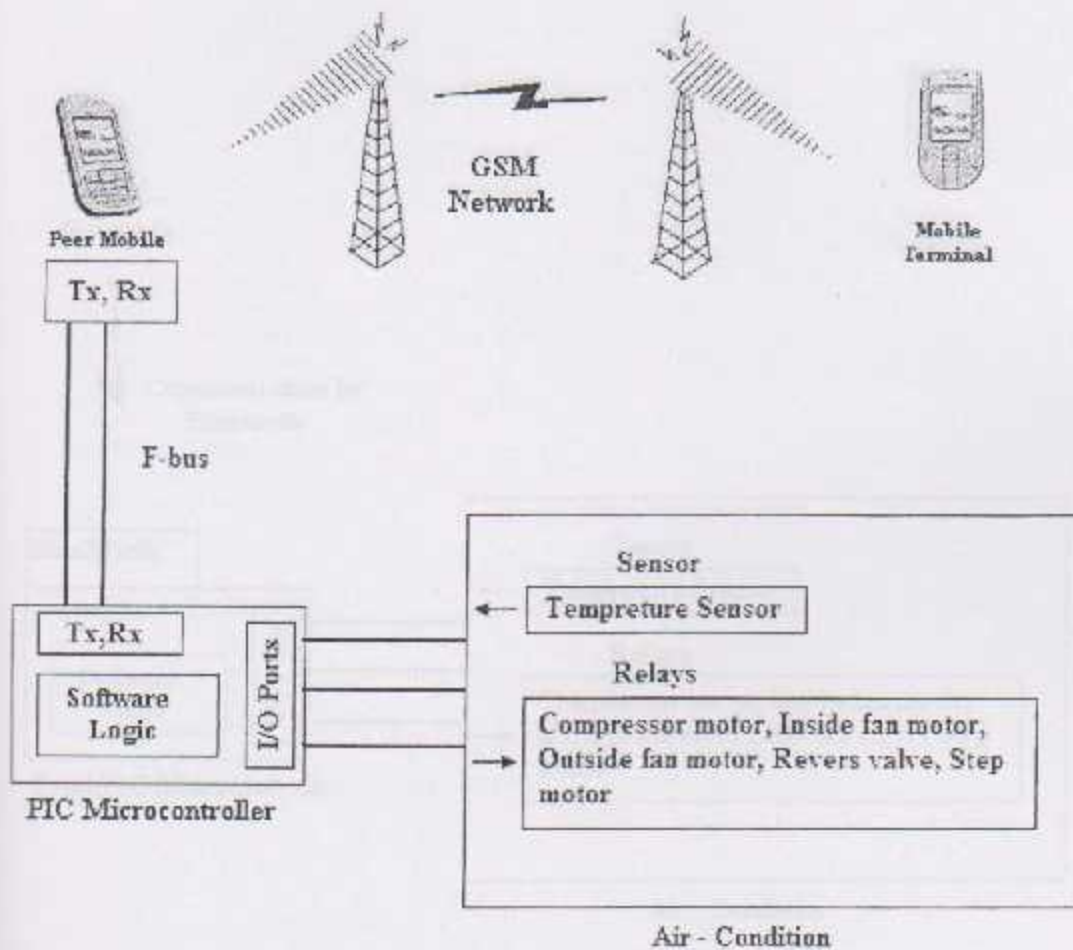


Figure 4.3 ToothPIC Controller

Figure 4.4 PIC Controller

4.2.2.1 ToothPIC Controller

The idea of this design option is to use Bluetooth to communicate the mobile to the PIC microcontroller; one of the most important benefits of using this microcontroller is that the communication between the mobile and the PIC microcontroller is reliable.

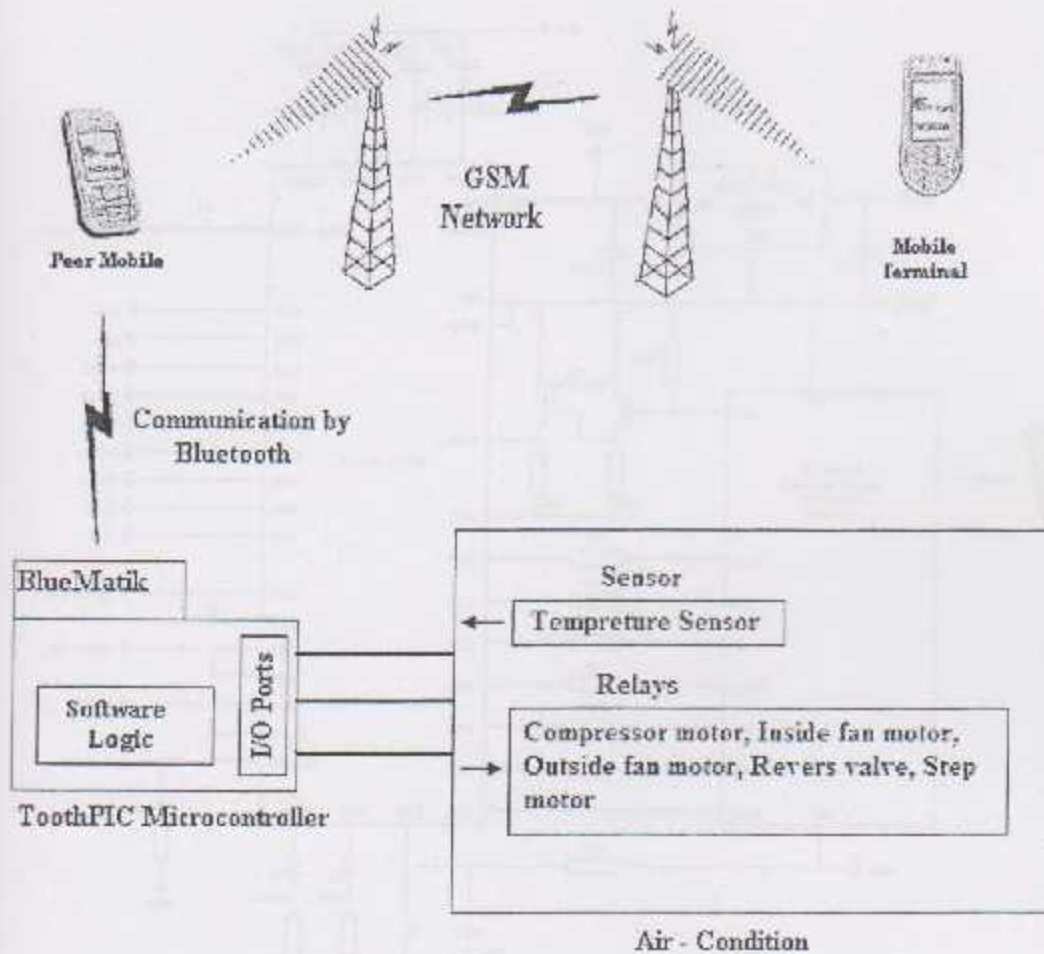


Figure 4.5 ToothPIC Controller

A device called BlueMatik used to provide Bluetooth to the PIC microcontroller, the following figure shows the PIC18LF6720 microcontroller attached to BlueMatik.

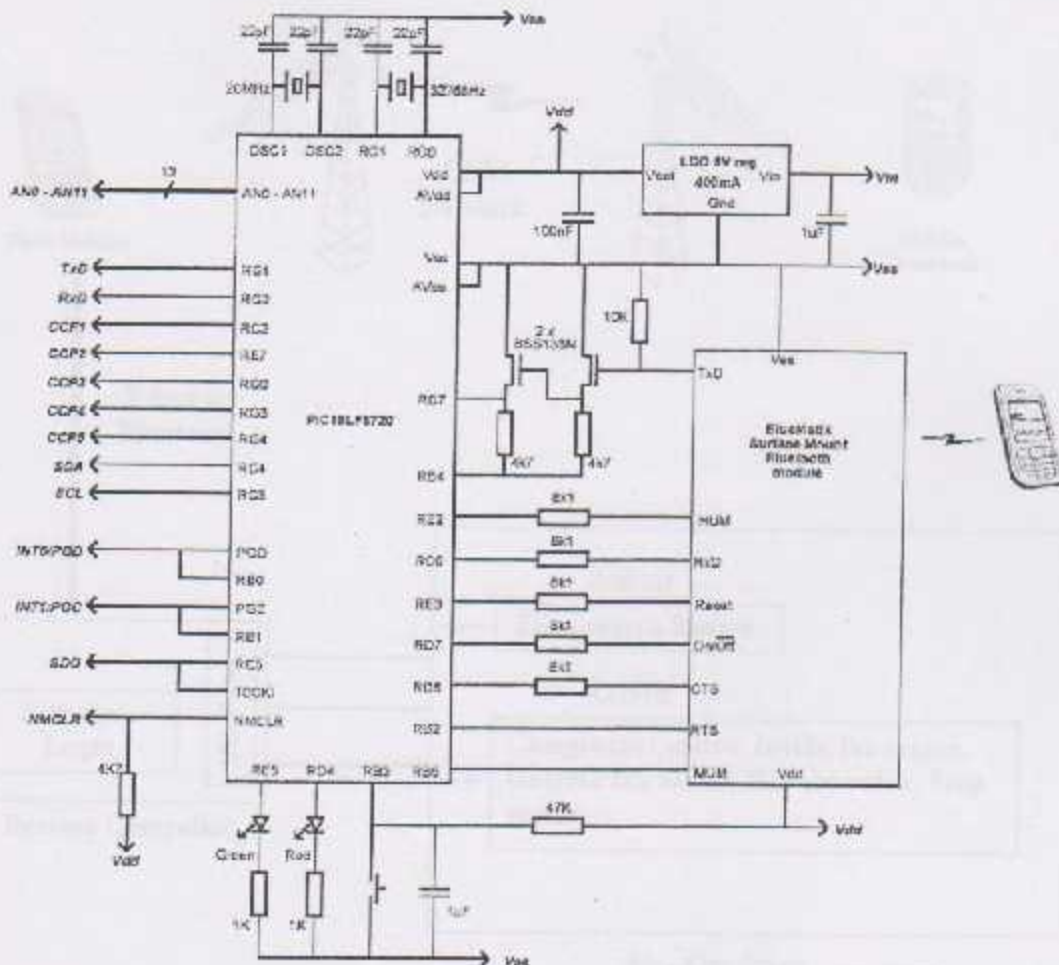


Figure 4.6 PIC18LF6720 with Blue.Matik

4.3 Mobile to Mobile Design Options

Although the communication between the peer mobile and the client mobile is done through the same GSM technology protocol, there are different ways to do that.

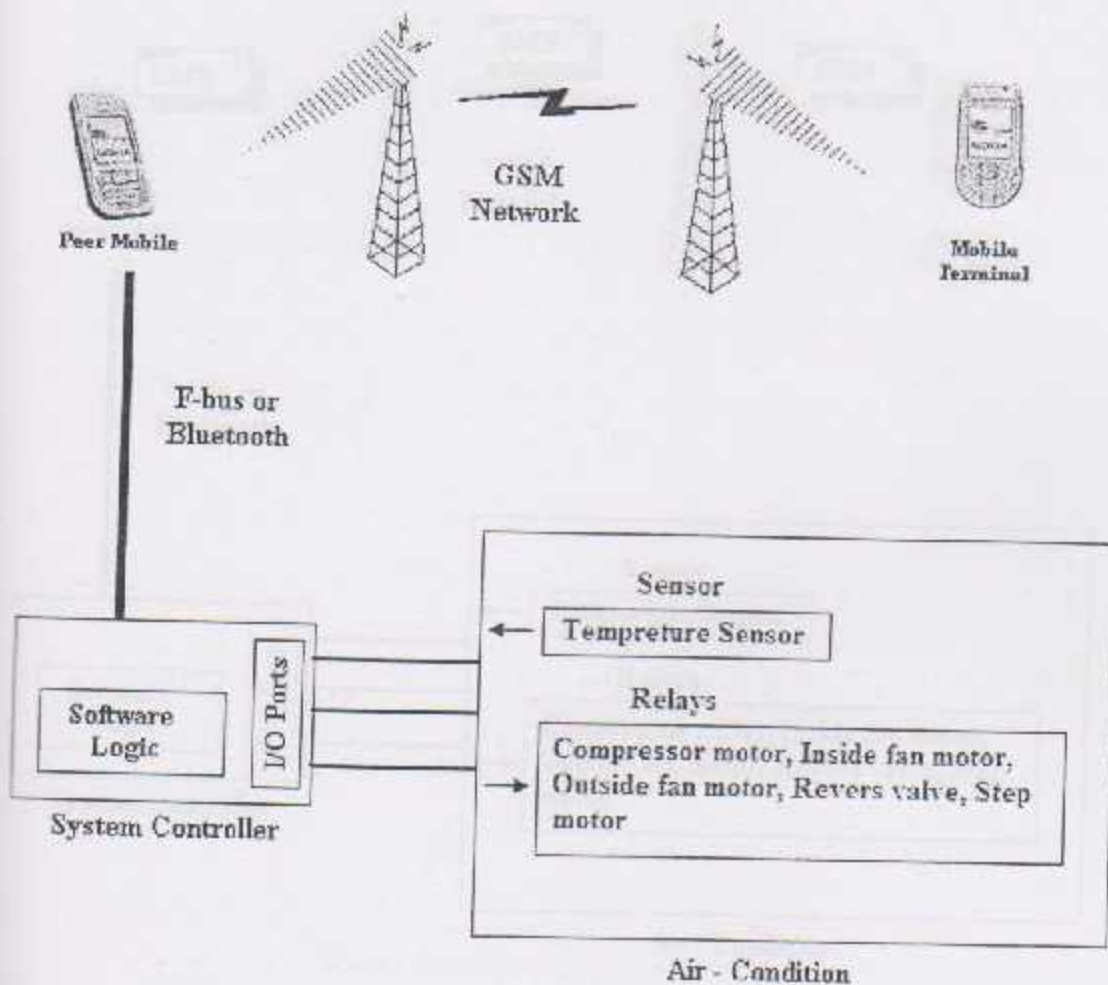


Figure 4.7 Mobile to Mobile Design Options

4.3.1 Control Using SMS

Text messaging service (SMS) is one of the ways that user client mobile can send/receive commands to/from system. In this option predefined commands are given to the user, the user send the command to the system controller which decode the command and execute the desired operation (Switch on, Switch off, etc).

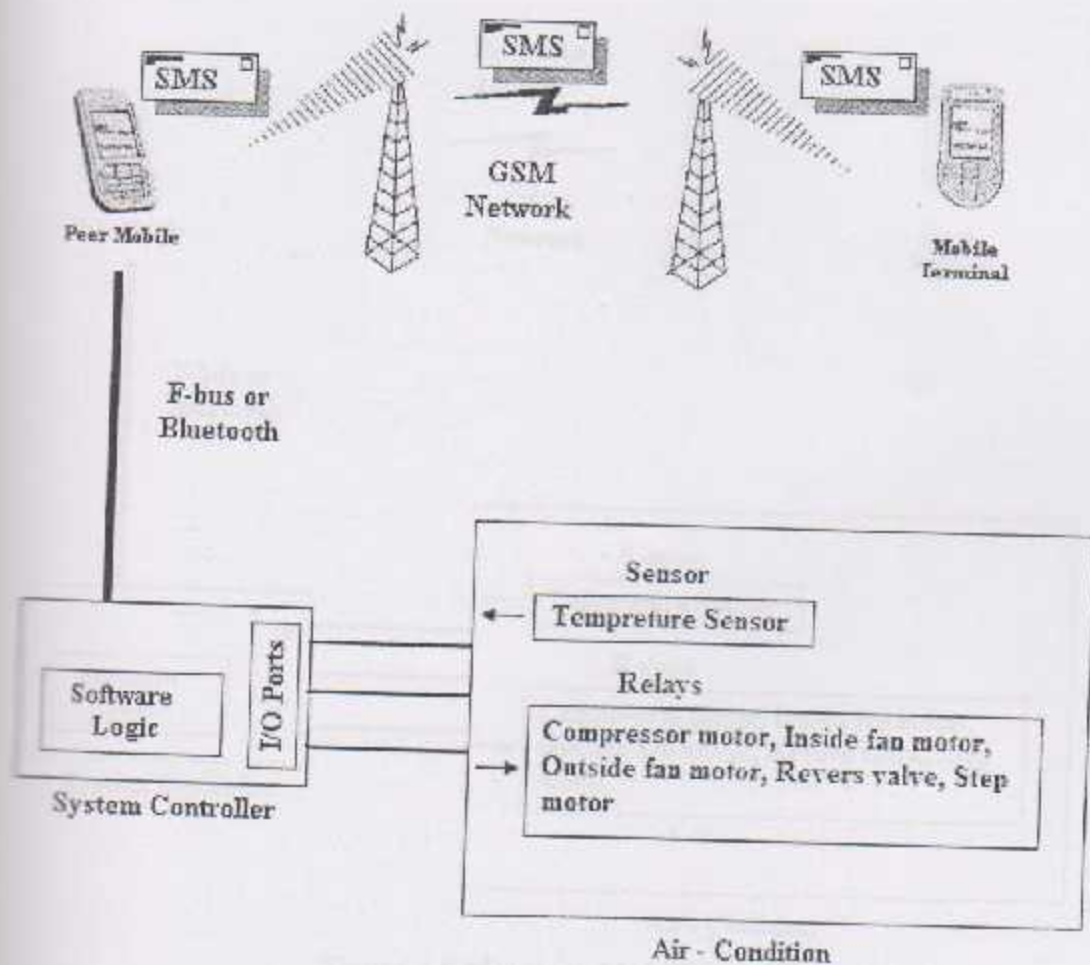


Figure 4.8 Mobile to Mobile by SMS

4.3.2 Real Time Calling/Control

Another design option for communicates the two mobiles is to use real time calling, in which the user simultaneously communicate with system controller in order to control the Air-Condition. The advantage of this option is that the user presses a key and the result displayed on the mobile screen.

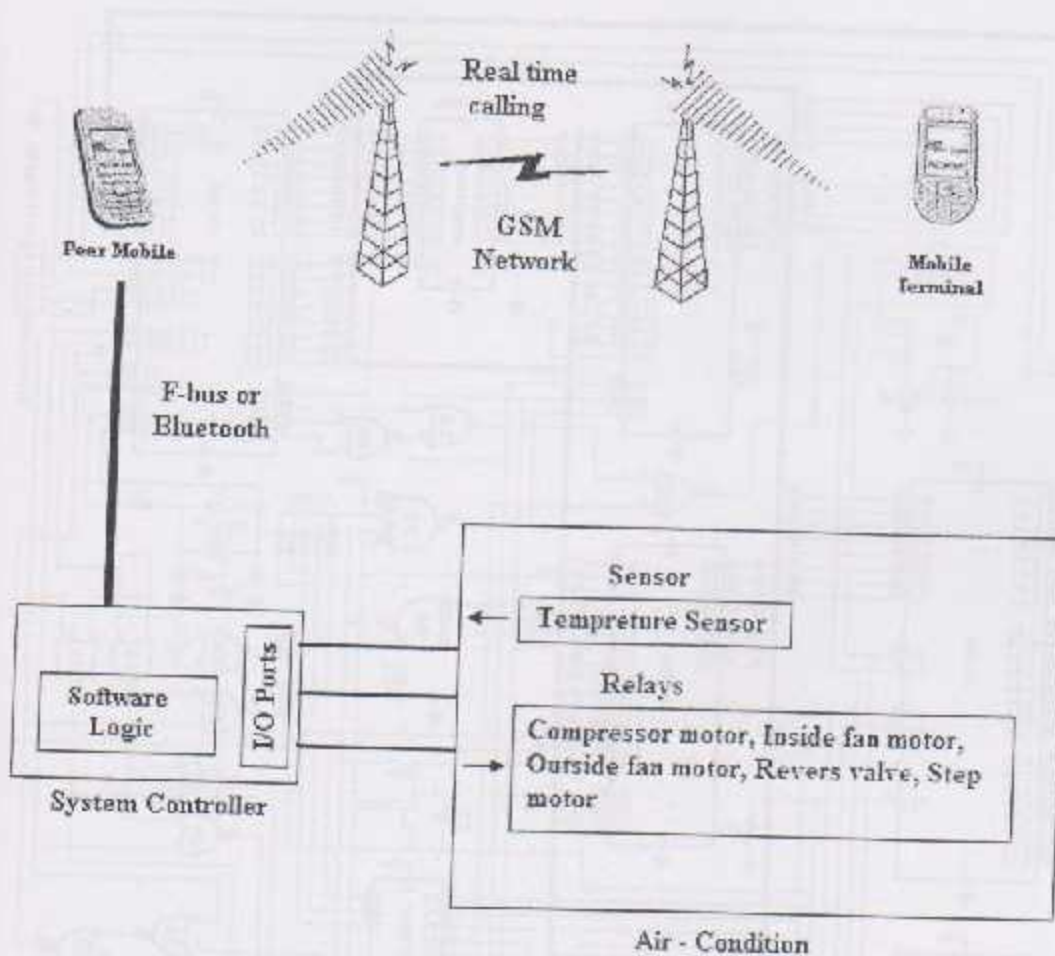


Figure 4.9 Mobile to Mobile by real time calling

4.4 Interfacing Circuits

In this section, we will explain the interfacing circuits that were been built, in order to connect different parts of the system.

The following circuits were used to connect microcontroller to other components in the system. Before showing the circuits, we will show in the next figure the microcontroller schematic (the system controller which we used in the project), the system controller is the 8051 microcontroller.

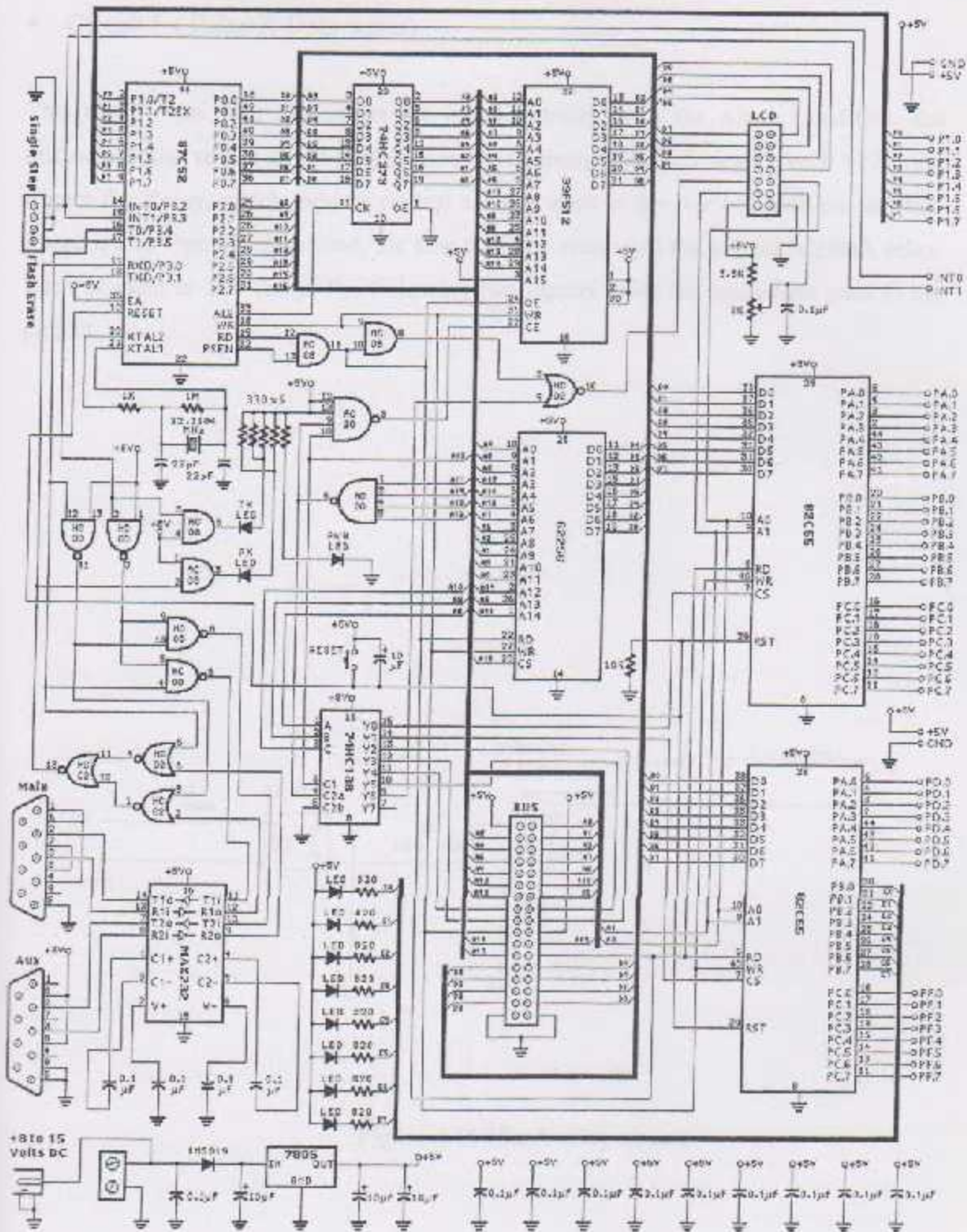


Figure 4.10 Microcontroller schematic

- **Circuit I (Relay & Optocoupler)**

This circuit was used to connect the microcontroller with the Air – Condition, the microcontroller send a signal with +5V to the Optocoupler, which operate with +12V and trigger the Relay, which used to control a component in the Air – Condition like fan. There is two type of relays used, the first is 10mA relay, and the second is 20mA relay. Each of them is +5V relay. The following two figures show the two relays used in the project.

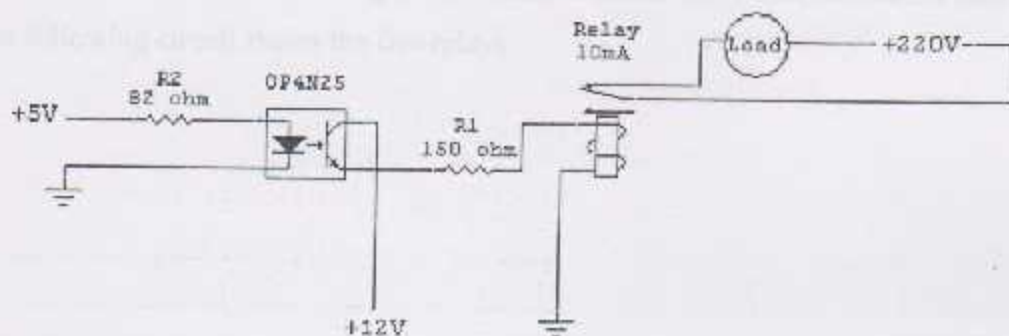


Figure 4.11 10mA relay circuit

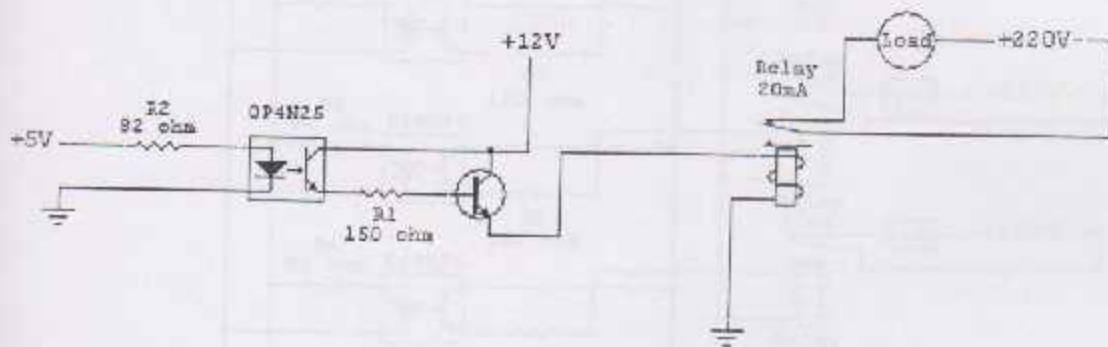


Figure 4.12 20mA relay circuit

In the project we use four 10mA relays and one 20mA relay, to control the Air - Condition components according to commands from the 8051 microcontroller board. The following circuit shows the five relays.

Figure 4.11 relays in the project

4.1.2.1. ENABLE (MAX232)

The MAX232 was used for voltage level shifting between the microcontroller and the modem. We used two MAX232 chips (figure 4.11) to make the voltage conversion between the PC and microcontroller. The first MAX232 provides a positive voltage of the microcontroller (5V) to +12V of the second MAX232 used to provide the voltage level +12V to +220V which is the voltage level of the PC's modem.

The first MAX232 provides two 5V levels to the microcontroller. The second circuit is used to provide the +12V to +220V which we can use to make

The first MAX232 circuit which was used for the microcontroller is shown in the next figure.

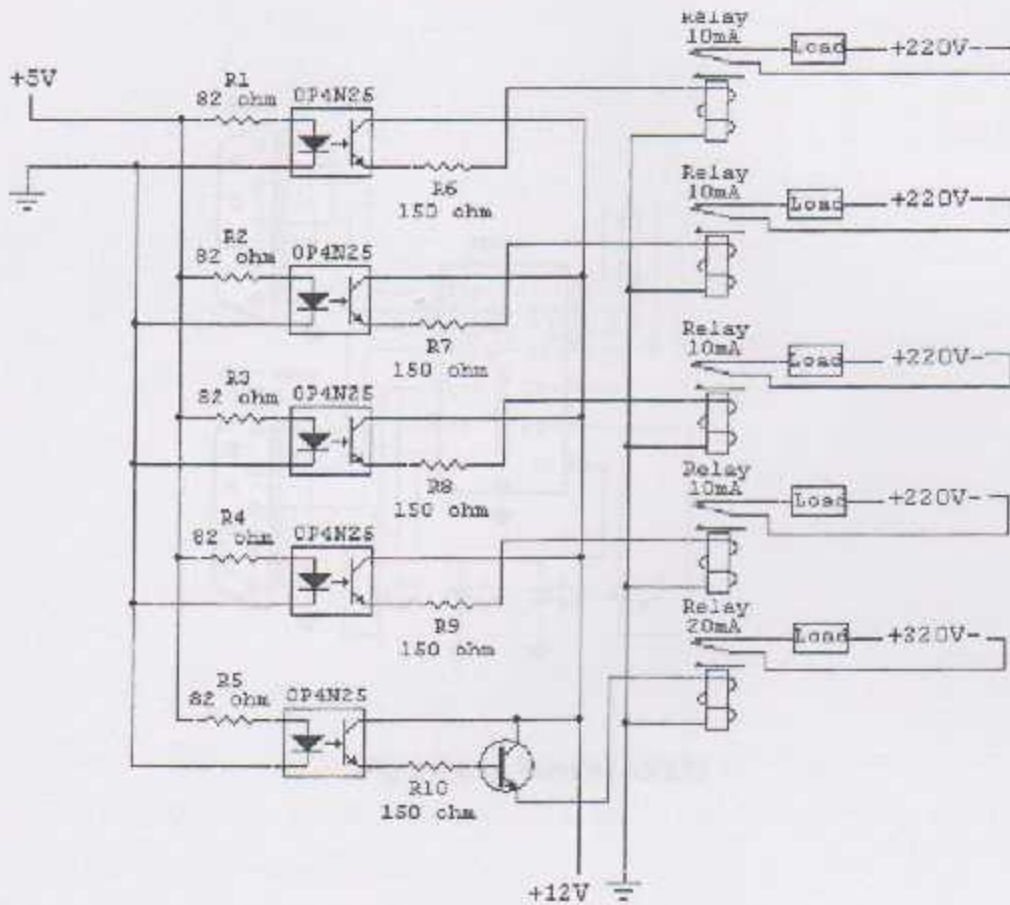


Figure 4.13 Relays in the project

- Circuit II (MAX232)

The MAX232 was used for voltage level shifting between the microcontroller and the mobile; we need two MAX232 chips in order to make the voltage coincidence between mobile and microcontroller. The first MAX232 chip used to minimize the voltage of the microcontroller from +15V to +8V, the second MAX232 used to minimize the voltage from +8V to +3.5V which is the voltage that mobile FBUS operate with it.

The first MAX232 circuit was built with microcontroller board, while the second circuit is built indoor the DRL-3P cable which we can buy or make.

The first MAX232 circuit which was built into the microcontroller is shown in the next figure.

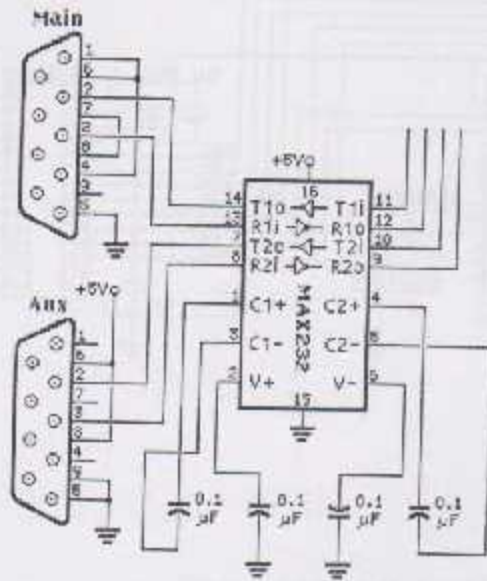


Figure 4.14 First MAX232

The pins 11 and 12 on the MAX232 chip are used to send and receive from the main port of the microcontroller, while the pins 10 and 9 are used to send and receive from the Auxiliary port.

We need a second max chip in order to connect the microcontroller to the mobile. The following figure is the internal diagram of the DLR-3 cable, which is contain the max chip used for the previous purpose.

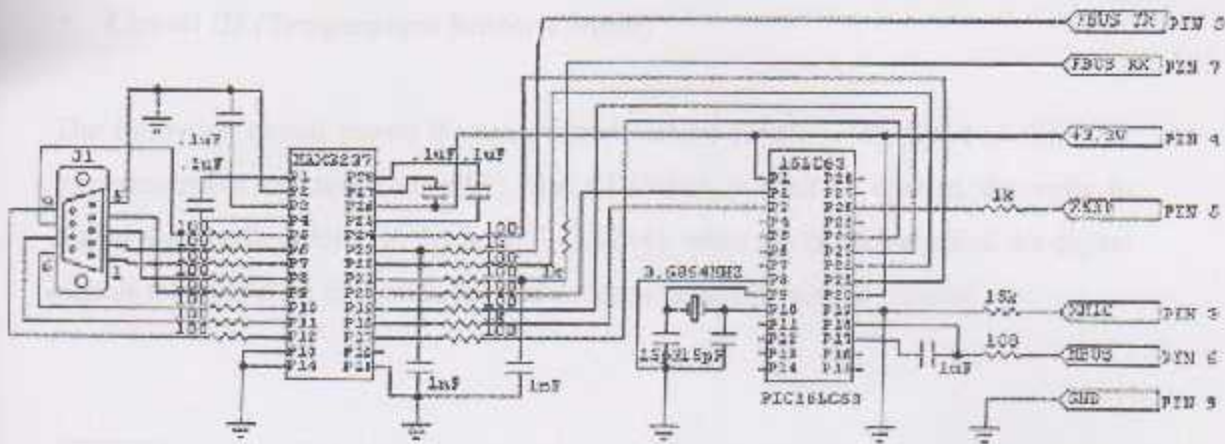


Figure 4.15 DLR-3 cable

Although there is a DLR-3 cable you can buy, which is in the previous figure. We can build it; the following figure shows the components and a circuit of the cable you can build in order to connect the mobile to other devices.

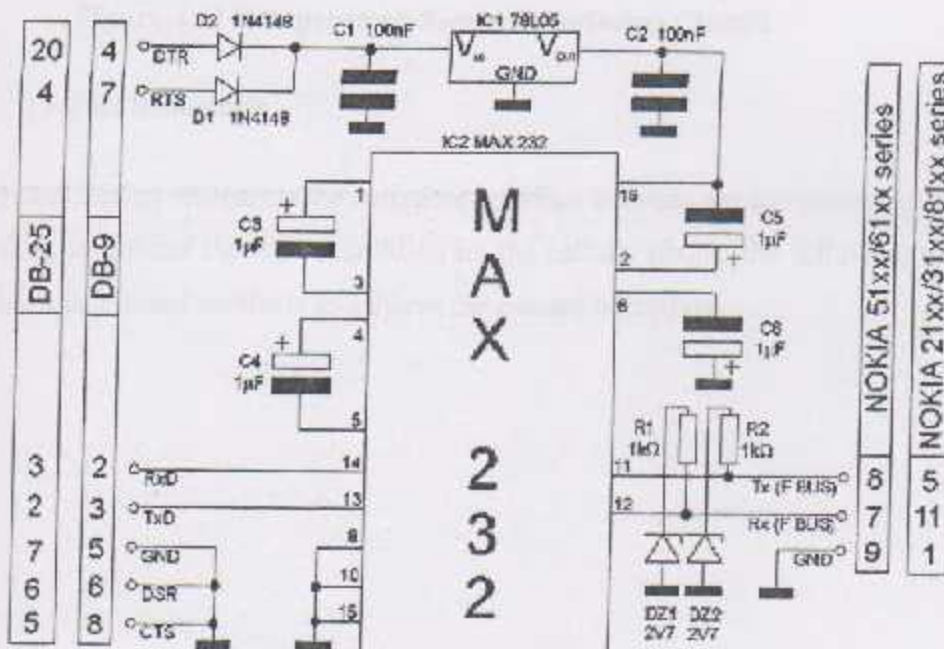


Figure 4.16 Mobile to other devices circuit

- **Circuit III (Temperature Sensor Circuit)**

The following circuit shows the temperature sensor (LM35), interfaced to the 8051 microcontroller (system controller), the ADC0804 is used to convert the volts to digital signals then stored in the latch (74LS244), when the latched enabled the digital signals transferred to the controller and decodes to temperature in Celsius.

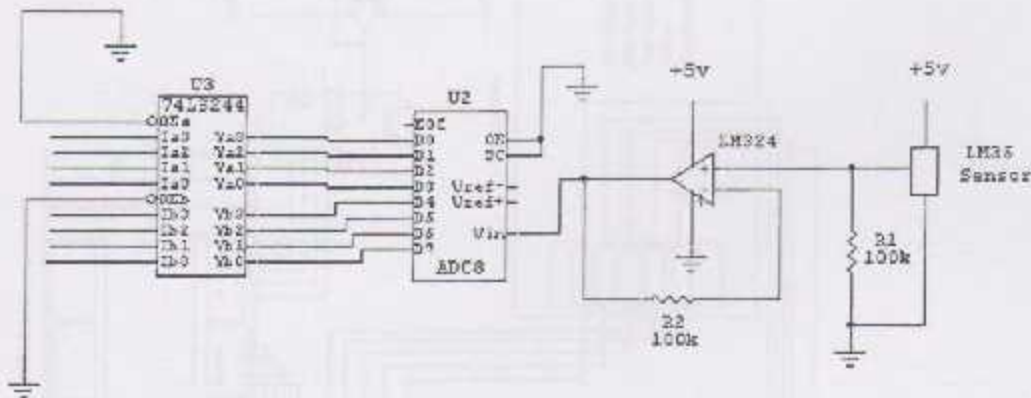


Figure 4.17 Temperature Sensor Interfacing Circuit

4.4.1 Project Schematic

The system design represents the complete interface between the components. Therefore, to control the Air – condition by the cellular phone, the following design has to be built and tested carefully to achieve the needed objectives.

Figure 4.18 Project Schematic

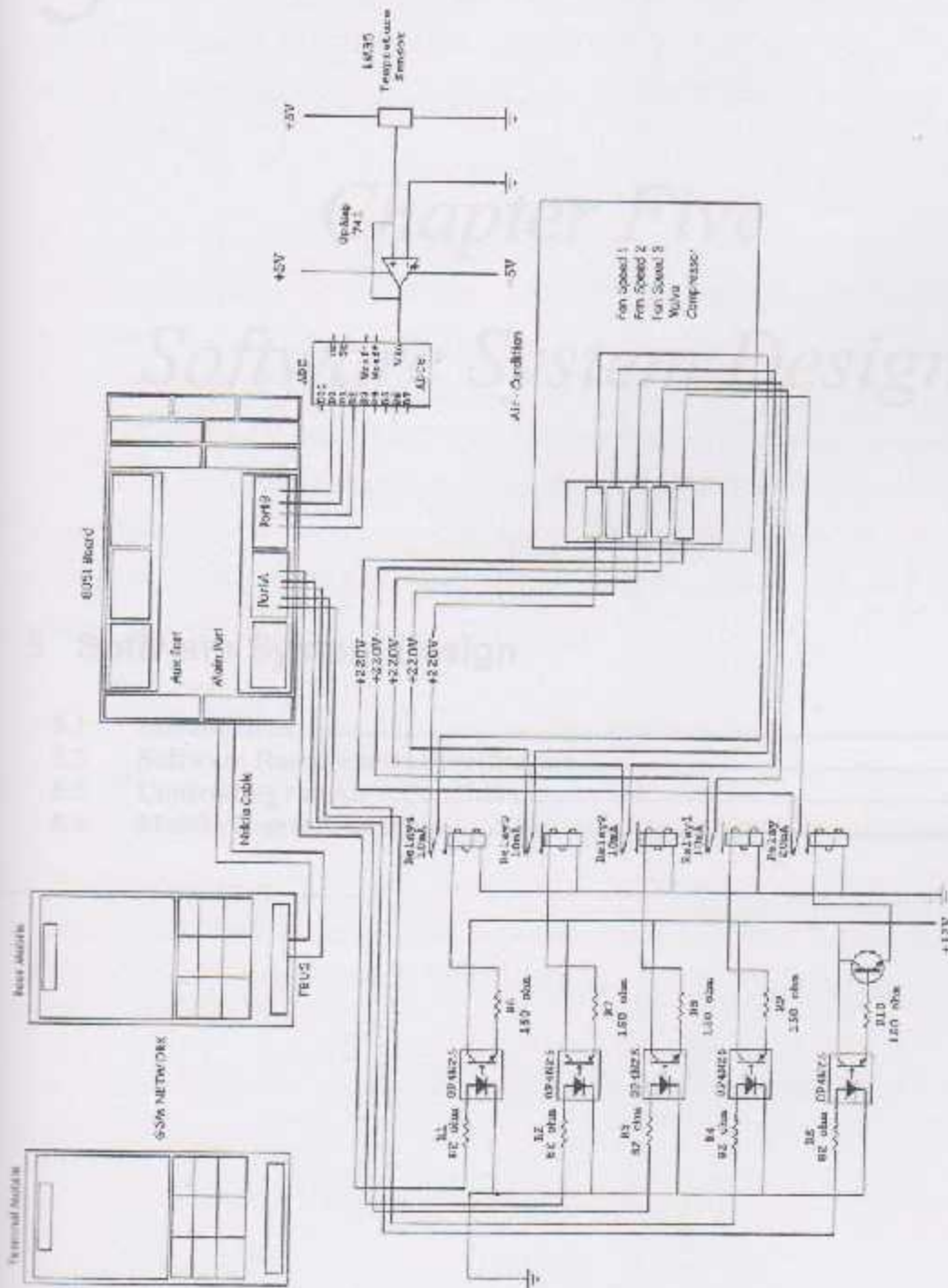


Figure 4.18 Project Schematic

5.1 Introduction

5

Chapter, the detailed description of the software system, the program which will receive and control the messages and the hardware which will be used to control the system. The program will be used to control the system and the hardware will be used to control the system.

Chapter Five

Software System Design

5 Software System Design

5.1	Introduction.....	83
5.2	Software Requirement Specification.....	83
5.3	Controlling the Air – Condition.....	88
5.4	Mobile Communication	92

5.1 Introduction

5.2 Software Requirement Specification

For the software system, there are two main categories of requirements: functional and non-functional. The functional requirements are those that describe what the system should do, and the non-functional requirements are those that describe how the system should do it.

5.1 Introduction

In this chapter, the detailed description for the microcontroller program, the program which will receive and decode the message send from terminal mobile user to peer mobile connected to the system controller, and also will initiates specific operation on the Air - Condition.

The program will communicate with the peer mobile, sends an AT command and receive the message from the user; the message will contain a number that specify the operation to be done.

There are several AT commands for the SMS messaging that enables us to get whatever information we need from the peer mobile, for example to read message either in PDU format or in the text format, according to specific parameter we send in AT commands through the FBUS of the peer mobile we will have the desired output that contain the needed information.

The microcontroller program will be responsible for the following, send an At command to the peer mobile from the microcontroller serial port, receive the response from the mobile, decode the received response, and send a signal to specific microcontroller port that controls the Air – Condition.

5.2 Software Requirement Specification

For the microcontroller software, there are needs; the following paragraph describes them.

5.2.1 Microcontroller Software Requirement Specification

- Win 95/98/ME/NT/2000/XP

The 8051 based system (microcontroller) needs to work on one of these Windows version in order to write the code in any text editor, also used the HyperTerminal program for transfer the code to the microcontroller.

- HyperTerminal

HyperTerminal program, which comes with pervious Windows versions, can be used to obtain information from any external device such as microcontroller or modem, or to send any data to these external devices; in this system it is used to establish the connection between the PC and the microcontroller, where is used to transfer assembled programs to microcontroller.

To setup the connection of the HyperTerminal, the following procedure should be performed.

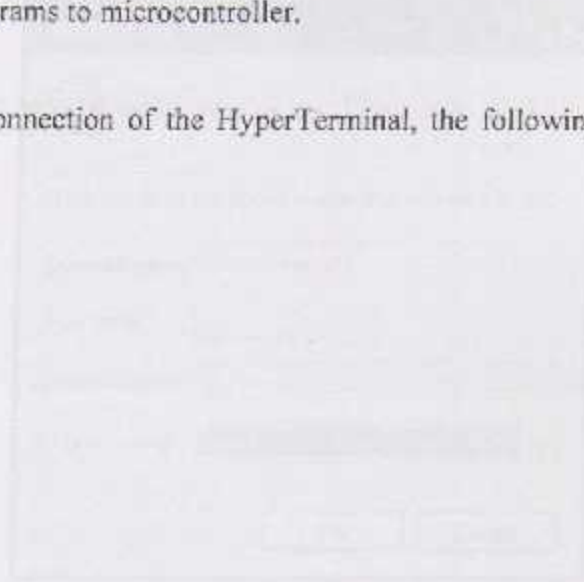


Figure 5.2 Connect to HyperTerminal



Figure 5.1 Connection Description Screen

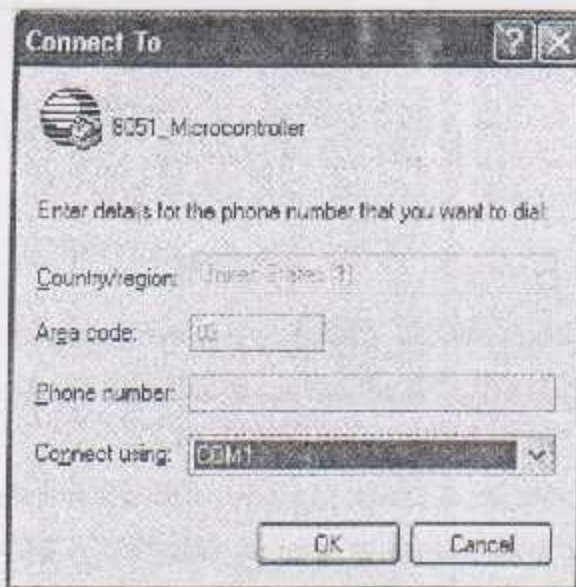


Figure 5.2 Connect To Screen

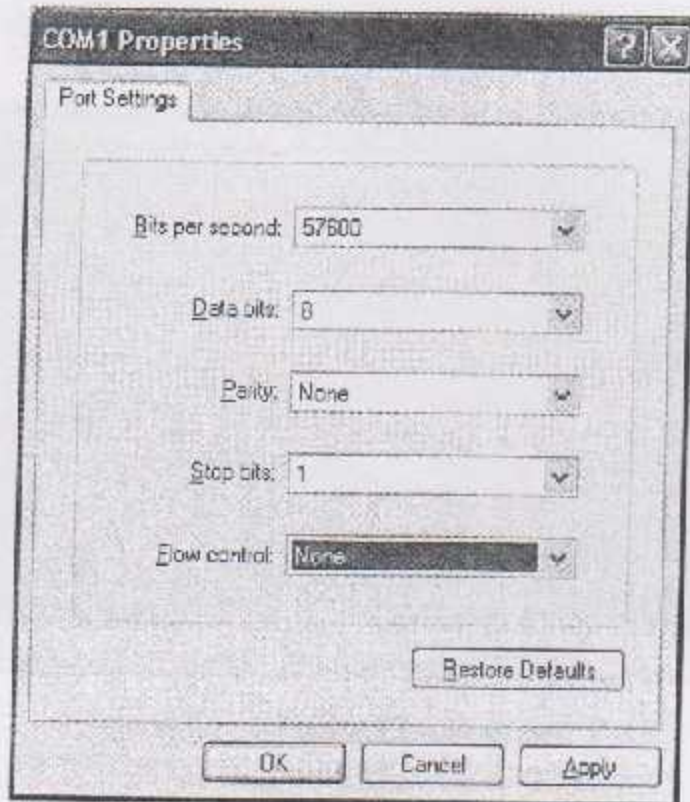


Figure 5.3 COM1 Connection Proprieties

- **Assembler or C Compiler**

The 8051 microcontroller programming is dealing with the hexadecimal code, so there is a need for the C compiler or assembler to convert the code from language code to machine code "hexadecimal".

The system programmed in assembler language related to the requirements of the project, where the SDCC and AS31 compilers are used to convert from assembler code into the hexadecimal code.

The following figure illustrates how to run the assembler program to assembly a program that was written in notepad on the location c:\ and its name is microcontroller_recieve.asm

```
C:\WINDOWS\system32\cmd.exe
C:\>cd sdcc
C:\sdcc>cd bin
C:\sdcc\bin>as31 c:\microcontroller_receive.asm
AS31 2.0b3 (beta), March 20, 2001
Please report problems to: paul@pjrc.com
Begin Pass #1
Begin Pass #2
C:\sdcc\bin>
```

Figure 5.4 Assembler

• **Microcontroller communication**

In order to download the program to the microcontroller, connect it to the computer, by using a standard 9 pin serial cable; the serial cable is connected between the computer COM port and the microcontroller main port, which is used for programming the microcontroller. The next figure shows the serial cable used to download the program to the microcontroller.



Figure 5.5 Serial Cable

- **Mobile Communication**

Firstly we need a mobile, which support SMS AT commands to communicate with. And SIM card into it, to send and receive messages from the user terminal mobile.

Secondly we need a DRL-3 cable which used to connect the mobile to computer or to other devices like microcontroller in order to communicate with. By using this cable we can send an AT commands through the FBUS of the mobile. The next figure shows the DRL-3 cable, it is known also as FBUS cable.



Figure 5.6 DLR - 3 Cable

5.3 Controlling the Air – Condition

Controlling the Air – Condition will be done through a simple procedure, the microcontroller will receive the user message and decode it, then if the temperature is more than 20°C (heating mode) send signal to specific relay, each relay will be responsible for controlling a component in the Air – Condition, for example to control the

compressor. By the way the Temperature 20°C and the heating mode were chosen for the implementation.

5.3.1 Sending Signal to the Relay

The microcontroller has ports that deal with the connected devices to the microcontroller. Because the relays in the market is operating at (+12V), we will use an Optocouple that operates at (+5V) that is the signal the microcontroller send it, and the Optocoupler will then make the relay operates.

▪ Sub Code For Sending Signal to Relay

The algorithm in which we can send an signal to relay is very easy, all what you need is to specify the port address after setting the PPI chip ports to be output, and then send the data from the accumulator to the address was specified in the form of byte, decimal, or hexadecimal as desired.

The following is a sub code; the complete code is in the appendix.

```
mov  dptr,#0xF800
mov  a,#11111111b
movx @dptr,a
```

• Flow Chart

The following two flowcharts describes how the Air – Condition controller acts in cooling and heating modes.

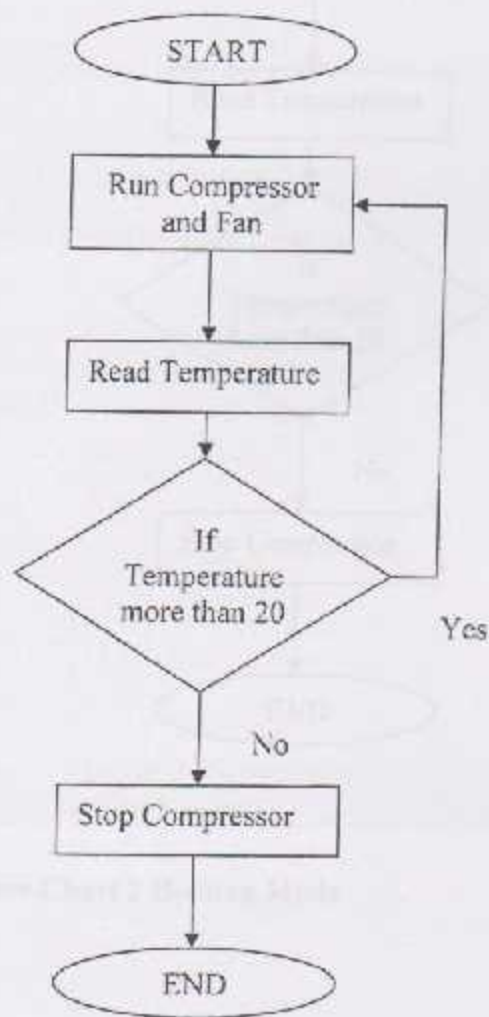


Figure 5.7 Flow Chart 1 Cooling mode

5.1. Mobile Compressor Unit

Mobile compressor is a compressor that is used to compress the refrigerant gas by all such facilities. The following is the flow chart of the mobile compressor.

5.1.1. Sub Cycle for Heating AT

In order to heat the AT, the compressor is run continuously. The refrigerant gas is compressed and the pressure is increased. The temperature of the refrigerant gas is also increased. The temperature of the refrigerant gas is measured and compared with the set point. If the temperature is less than the set point, the compressor is run continuously. If the temperature is greater than the set point, the compressor is stopped.

Flowchart:
1. Start
2. Run Compressor and Fan
3. Read Temperature
4. If Temperature Less than 20
5. Yes
6. No
7. Stop Compressor
8. End

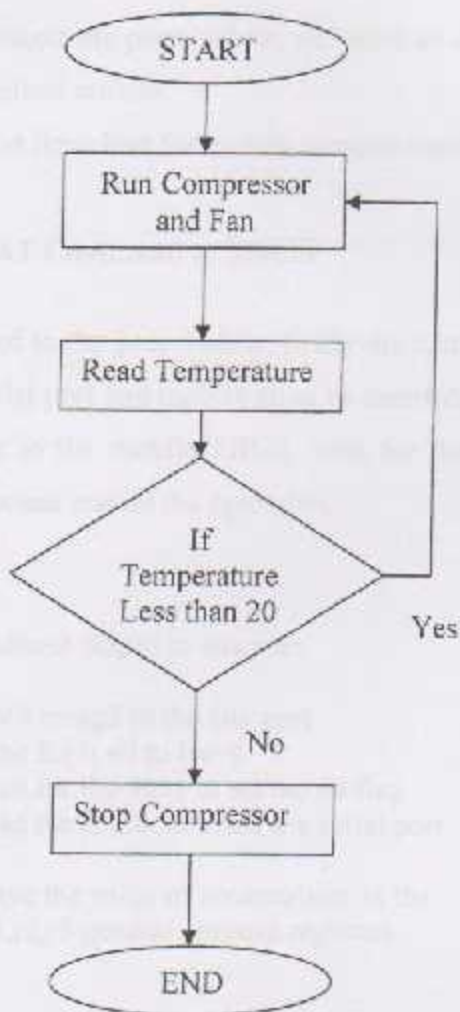


Figure 5.8 Flow Chart 2 Heating Mode

5.4 Mobile Communication

Mobile communication is to connect the peer mobile, and send an AT command to read the message sent by the user terminal mobile.

The following is the sub code and flowchart for mobile communication.

5.4.1 Sub Code for Sending AT Command to Mobile

In order to send an AT command to the peer mobile, firstly the communication between the microcontroller auxiliary serial port and mobile must be enabled, then send character by character from serial buffer to the mobile FBUS, wait for the response from the mobile, decode the received response and do the operation.

```
mainloop:
    clr    p3.4        ;redirect output to aux port
    mov    dptr, #mesg2
    lcall  pstr        ;send mesg2 to the aux port
    jnb    ti, *       ;wait for it all to leave
    jnb    ri, *       ;wait for the 8051 to set the RI flag
    mov    a,sbuf      ;read the character from the serial port

    mov    r1,a        ; save the value of accumulator in the
    mov    r2,a        ; r1,r2,r3 general purpose registers
    mov    r3,a

    anl   a,1         ; compare if response equal 1
    jnz   On

    mov   a,r1
    anl  a,2         ; compare if response equal 2
    jnz  Off

    mov  a,r2
    anl  a,3         ; compare if response equal 3
    jnz  SetTemp

    mov  a,r3
    anl  a,4         ; compare if response equal 4
    jnz  GetTemp
ljmp  mainloop
```

• Flow Chart

The following flow chart describes how the microcontroller acts when communicating with the peer mobile.

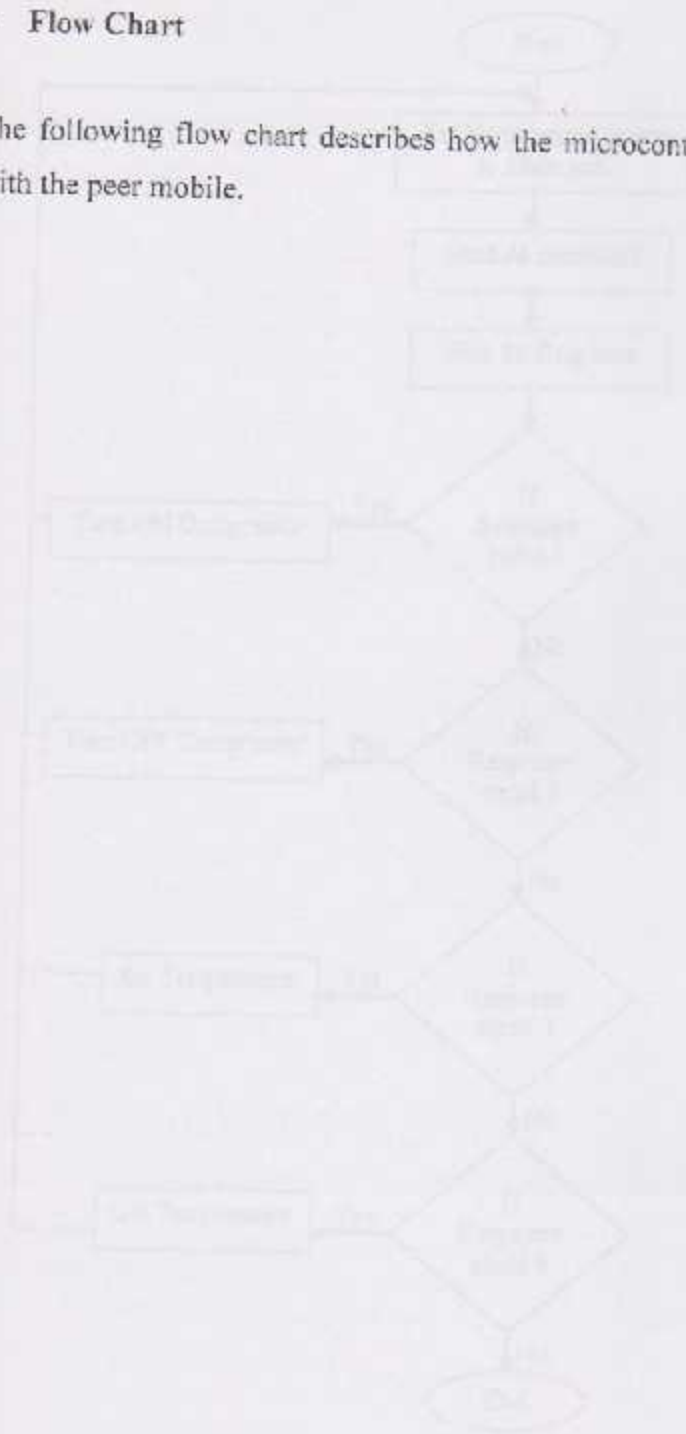


Figure 5.2 Flow chart of buzzer's operation

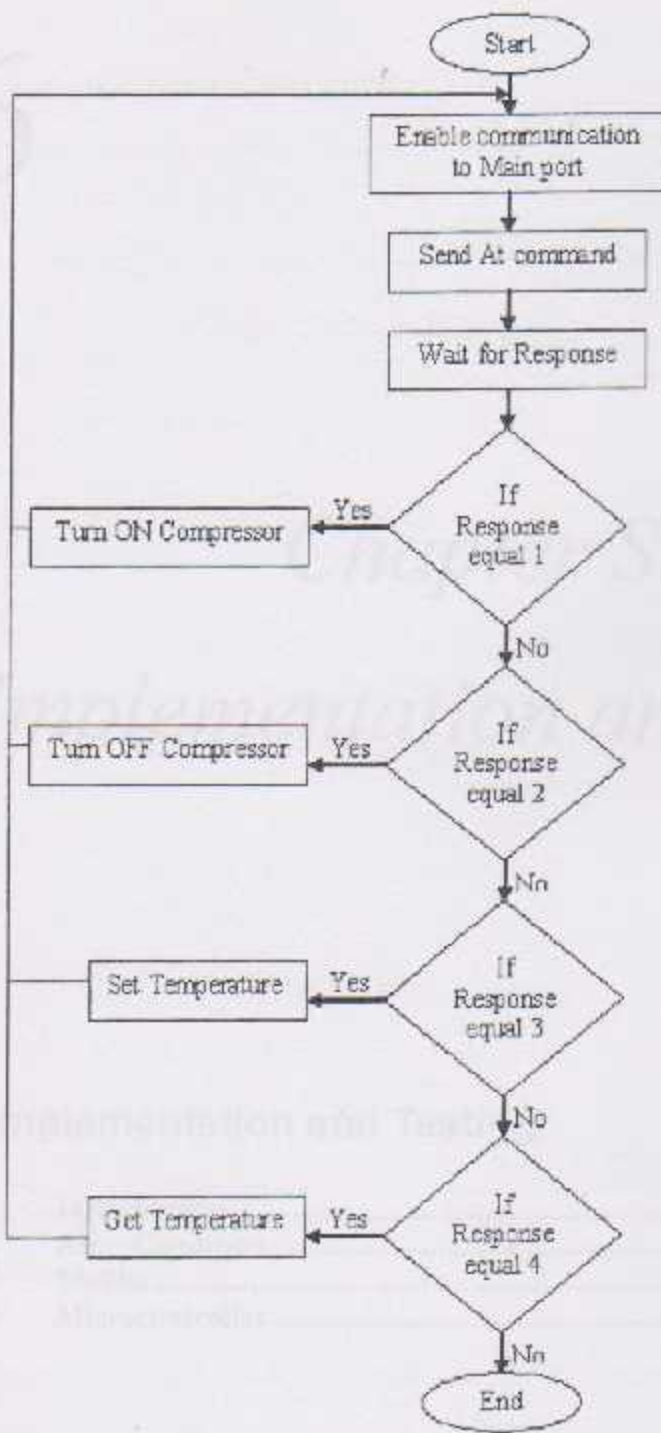


Figure 5.9 Flow chart 3 Mobile Communication

6

Chapter Six

Implementation and Testing

6 Implementation and Testing

6.1	Introduction.....	96
6.2	Air - Condition.....	96
6.3	Mobile.....	99
6.4	Microcontroller.....	103

6.1 Introduction

The whole system will be described in this chapter, that's mean how the system is implemented and tested will be discussed.

An implementation has been done for each part of the system separately, the interface was done between those parts and the main brain of the project which is the 8051 microcontroller, the interface of the system was controlled by the user command determined by the SMS message as environment of sending commands.

6.2 Air - Condition

An Air - Condition is one of the main parts in the project. It contains two units an indoor unit and an outdoor unit. The outdoor unit was brought in from the refrigerant workshops at the university; the indoor unit was brought in from the refrigerant and air conditioning labs.

6.2.1 Air - Condition Implementation

In the Outdoor unit the compressor was not working and it was replaced with a different compressor but with the same characteristics, and so on the indoor unit and out side unit were connected on a base of metal and wood so that the connecting process would be easy, and the controlling unit would be connected in the same way. The next process was to fill the Air - Condition refrigerant R22. Secondly the Air - Condition was tested manually and it worked.

An interface between the Air - Condition and the microcontroller was done because we found that the microcontroller gives a +5 volt output, where as the Air - Condition parts work on a +220 volt and so the circuit was done to help with the difference in voltage the circuit interface contains Relays, Optocoupler, and resistance. The relay of the compressor is different from the other relays that are used in the Air - Condition parts, because it is 20mA while the others are 10mA.

The following figure shows the Air – Condition indoor and outdoor units connected on a based of metal and wood.

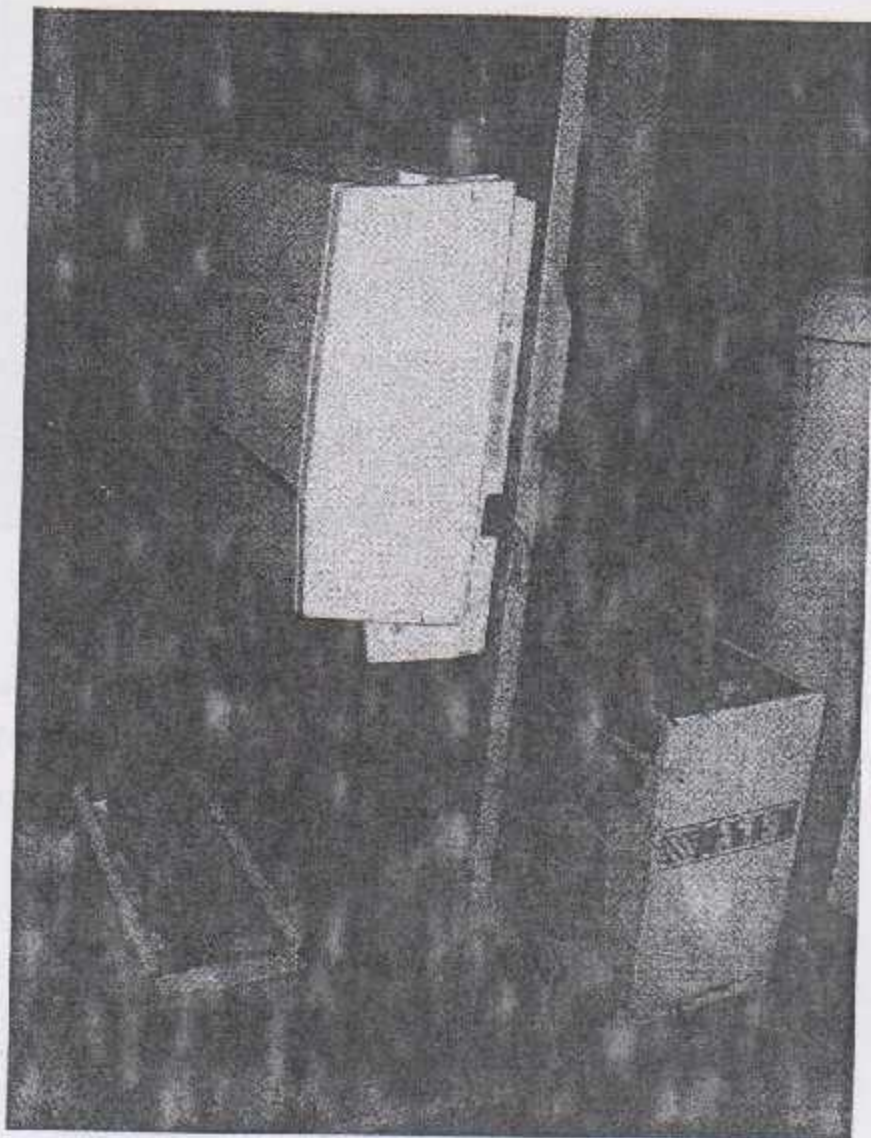


Figure 6.1 Air - Condition on the Base of metal and wood

In order to implement the design in chapter four sending signal from the microcontroller to the Air – Condition, the relays and Optocoupler circuit was made. The following figure shows the relays and Optocoupler circuit.

Figure 6.2 Relay and Optocoupler Circuit

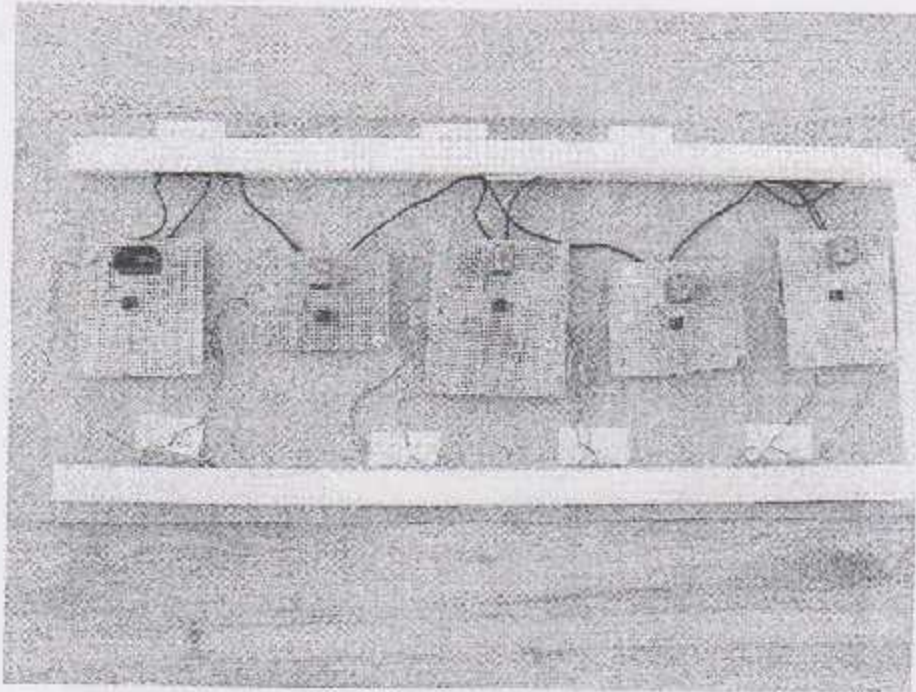


Figure 6.2 Relays and Optocoupler Circuit

The previous circuit was built to control the components of the Air - Condition. It consists of five relays, five Optocouplers, and necessary resistors. Four of the relays are 10mA the next figure shows one of them.

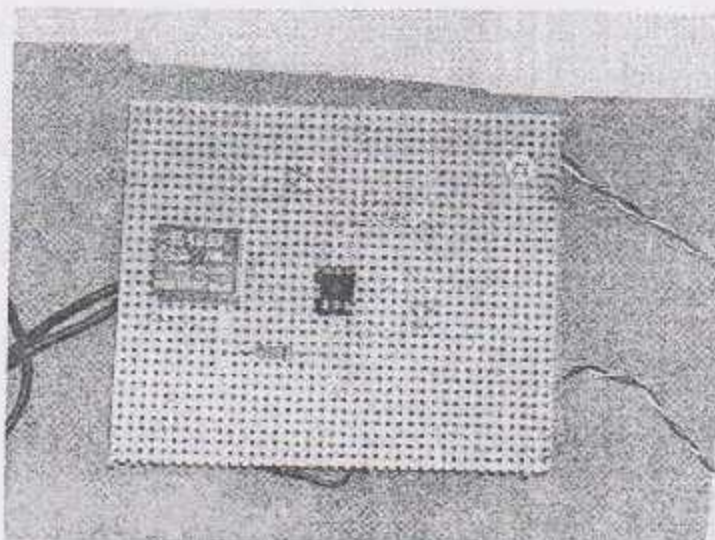


Figure 6.3 10mA Relay Circuit

The last relay is 20mA with a 2N2219A transistor; the next figure shows this relay circuit.

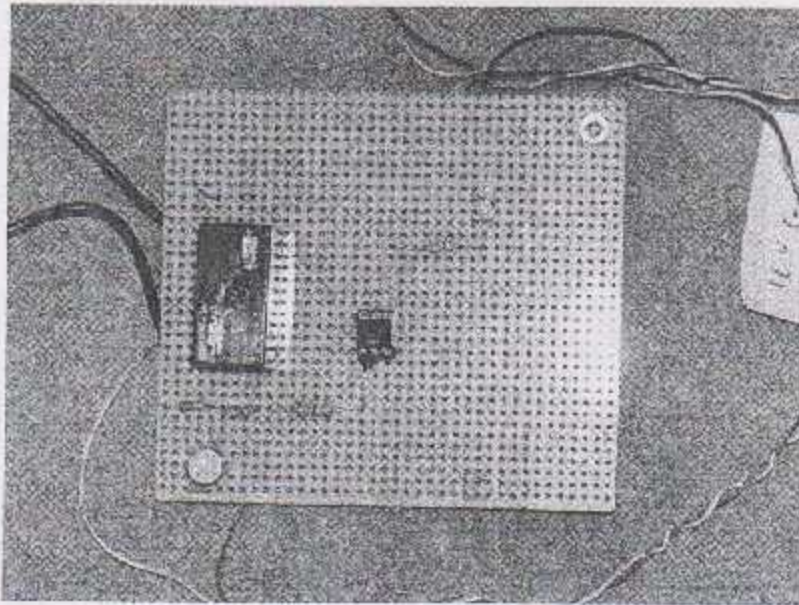


Figure 6.4 20mA Relay Circuit

The relay and Optocoupler circuit first is built and tested with light (220V), after that it tested with signal from the microcontroller on the Air - Condition and it was worked.

6.3 Mobile

The mobile provide the connection between the user terminal mobile and the peer mobile connected to 8051 microcontroller. Nokia 6100 mobile was chosen for the implementation, the following figure shows the Nokia 6100 mobile.

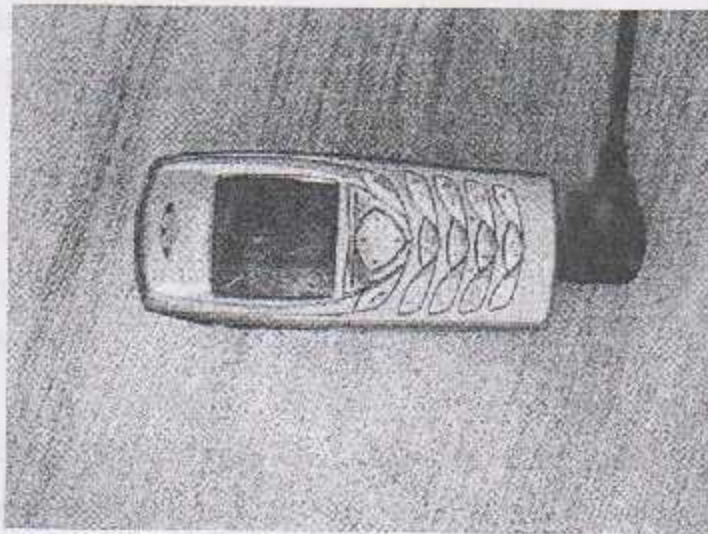


Figure 6.5 Nokia 6100 Phone

Nokia 6100 contains an FBUS for serial communication, also the Nokia 6100 support the AT commands for the short messages service (SMS), because of the previous reasons we chose the Nokia 6100.

6.3.1 Mobile Implementation

To implement the design in chapter four, we connect the DRL-3 cable with mobile FBUS and connect the transmit (Tx), receive (Rx), and ground (GND) of the other side of the cable with microcontroller auxiliary serial port, but we didn't have a response from the mobile. This problem discussed in the next section.

6.3.2 Mobile Communication Problem

When the previous problem arises, we firstly connect other pins from the DLR-3 cable to the microcontroller auxiliary serial port like DTR, but we also didn't have a response from the mobile. Secondly we made the following test, we connect the DRL-3 cable to the microcontroller auxiliary serial port and after the cable we see the signal on the oscilloscope, we find that it is equal to the range that the mobile FBUS operates with (-3.5V to +3.5).

After that we decided to test the cable with computer HyperTerminal program, so we test the DRL-3 cable with the HyperTerminal, but we cannot also have any response from the mobile.

The last test was to connect the MAX232 on the microcontroller to other MAX232, the second MAX232 was connected to the mobile FBUS, and also this test fails to have response from the mobile.

To be sure that we are in the right way, we made the loop test, which means that to connect the transmit and receive of the microcontroller auxiliary serial port, then send a message from auxiliary serial port and we receive it back. Also we send a message

Our expectation about the mobile communication problem from the 8051 microcontroller is the DLR-3 cable in the market is a commerce cable, also it contains a microcontroller in it and we didn't now how it works. Although we search for the schematic of the cable in the internet but we cannot analysis its functionality, Also may be it is one of the Nokia secrets.

At that time we decided to search for a cable driver to try to connect to connect the mobile to HyperTerminal program, then we find the PC Suit software that has Nokia Cable Driver with its package, then we can get response from the mobile.

Because of the previous problem, we add the computer to the project schematic. The new schematic will contain the terminal mobile which communicate with the peer mobile through the GSM network, the peer mobile connected to the computer Com1 port, the microcontroller connected to the computer Com2 port, and the microcontroller connected to the Air - Condition. To do the previous we think to make a Visual Basic.Net program to send the received message from the computer to the microcontroller specially that there are project on the internet do that. The new project schematic is in the next figure.

Figure 4.8 Schematic Project Schematic

6.4 Microcontroller

The parts used in the BSC1 project include the BSC1 board, a computer, a peer-to-peer hub, a GSA network, a fan speed controller, and an ABS responsive heater. The BSC1 board is connected to the computer via a serial cable. The peer-to-peer hub is connected to the BSC1 board via a BNC cable. The GSA network is connected to the BSC1 board via a BNC cable. The fan speed controller is connected to the BSC1 board via a 4-pin connector. The ABS responsive heater is connected to the BSC1 board via a 4-pin connector.

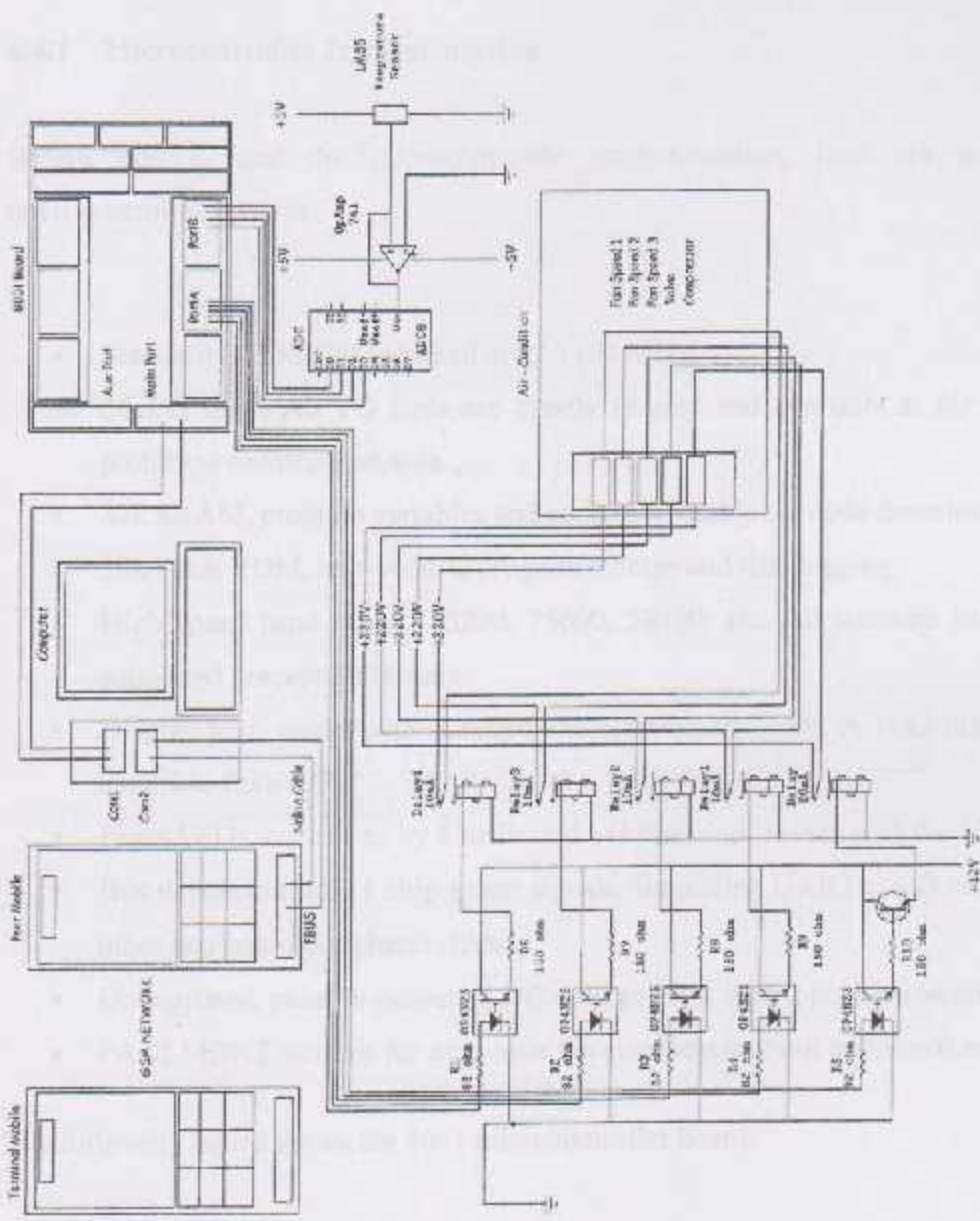


Figure 6.6 Second Project Schematic

6.4 Microcontroller

The system needs the 8051 microcontroller in order to navigate the whole operations of the Air – Condition, and communicate with the peer mobile. We chose the 8051 microcontroller because of its wide range of features that enable us to communicate with serial port and controlling the Air – Condition.

6.4.1 Microcontroller Implementation

Before talk a bout the microcontroller implementation, here are some of the microcontroller features.

- Standard 87C52 CPU clocked at 22.1184 MHz.
- 50 I/O lines. All I/O lines are clearly labeled and available at the edge of the prototype construction area.
- 32k SRAM, program variables and code (24k usable for code download)
- 30k Flash ROM, non-volatile program storage and data logging
- High speed baud rates: 115200, 75600, 38400, etc. All standard baud rates are supported (except 300 baud).
- Display port, works with standard character-based LCDs. A 16x2 display will be available from PJRC
- Eight LEDs, controlled by 8 dedicated I/O lines (not shared with the 50 I/O lines)
- Bus expansion with 4 chip select signals, for adding UARTs, A/D converters and other bus-based peripheral chips.
- Unregulated, polarity-protected DC voltage input with 2 position terminal block.
- PAULMON2 monitor for easy code development without additional equipment.

The following figure shows the 8051 microcontroller board.

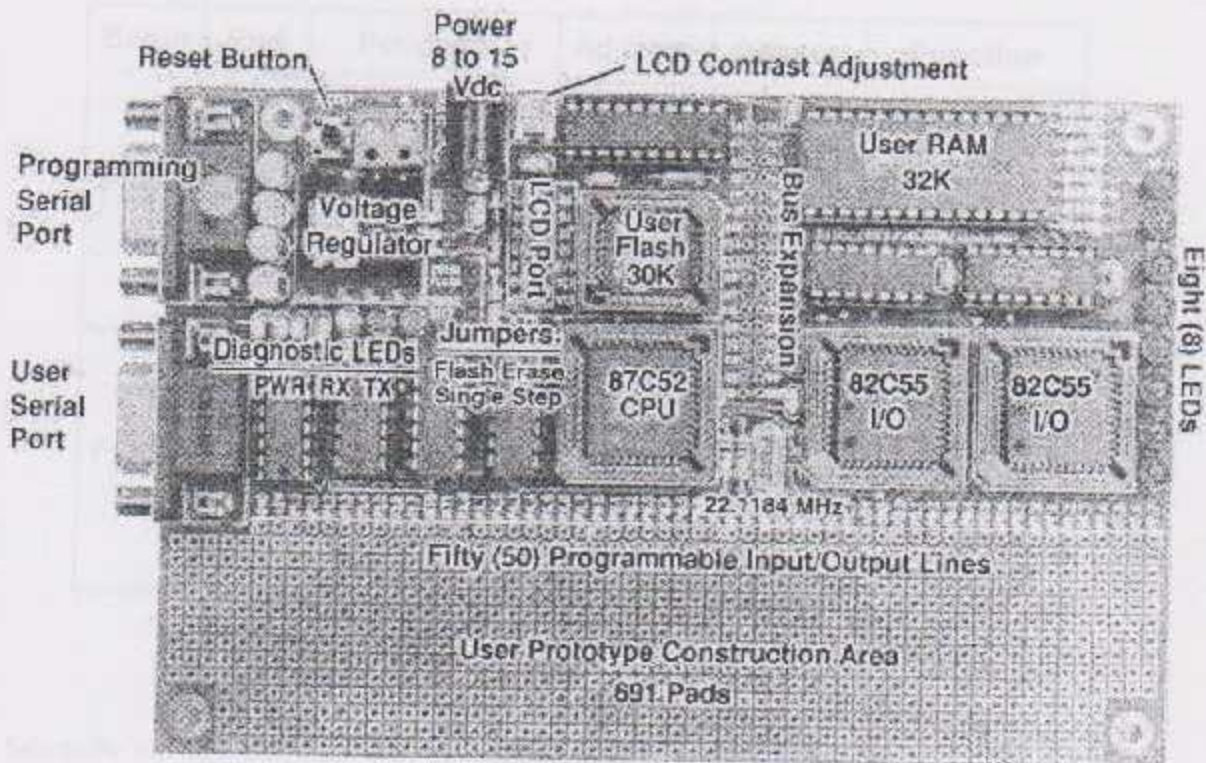


Figure 6.7 8051 Microcontroller Board

About the implementation. We firstly test the example programs on the internet like the blink led example which make the eight leds on the microcontroller on and off one after another. After that we test the ports on the microcontroller that are controlled by the two PPI chips on the board by switching on and off a led connected to specific pin port. The following is the memory map for the two PPI chips and its functionality.

Begin	End	Peripheral	Address	Access	Function
F800	F8FF	82C55 (A, B, C)	F800	RD / WR	Port A
			F801	RD / WR	Port B
			F802	RD / WR	Port C
			F803	WR Only	Config A,B,C
F900	F9FF	82C55 (D, E, F)	F900	RD / WR	Port D
			F901	RD / WR	Port E (LEDs)
			F902	RD / WR	Port F
			F903	WR Only	Config D,E,F

Table 6.1 8051 Microcontroller Board

Secondly we communicate with the mobile, and the problem described in section 6.3.2 were arisen we solve it as we explain previously.

6.4.2 Microcontroller Failure

During the work on 8051 microcontroller board, the processor chip and the two PPI chips were damaged, may be it was need additional regulator circuit than the one on the board, but we didn't expect that, may be this is was happened because we use the microcontroller board every day and night.

So to implement the system, we use the same project schematic in section 6.3.2 but with some editing, we use the computer parallel port to implement the idea, so the computer running a Visual Basic.Net program which receive the message by the Com port which is connected to peer mobile FBUS by the DLR-3 cable from the user, and from the computer parallel port send signal to relay and Optocoupler circuit to control the Air - Condition components. The new project schematic is in the next figure.

Figure 6.3 Final Project Schematic

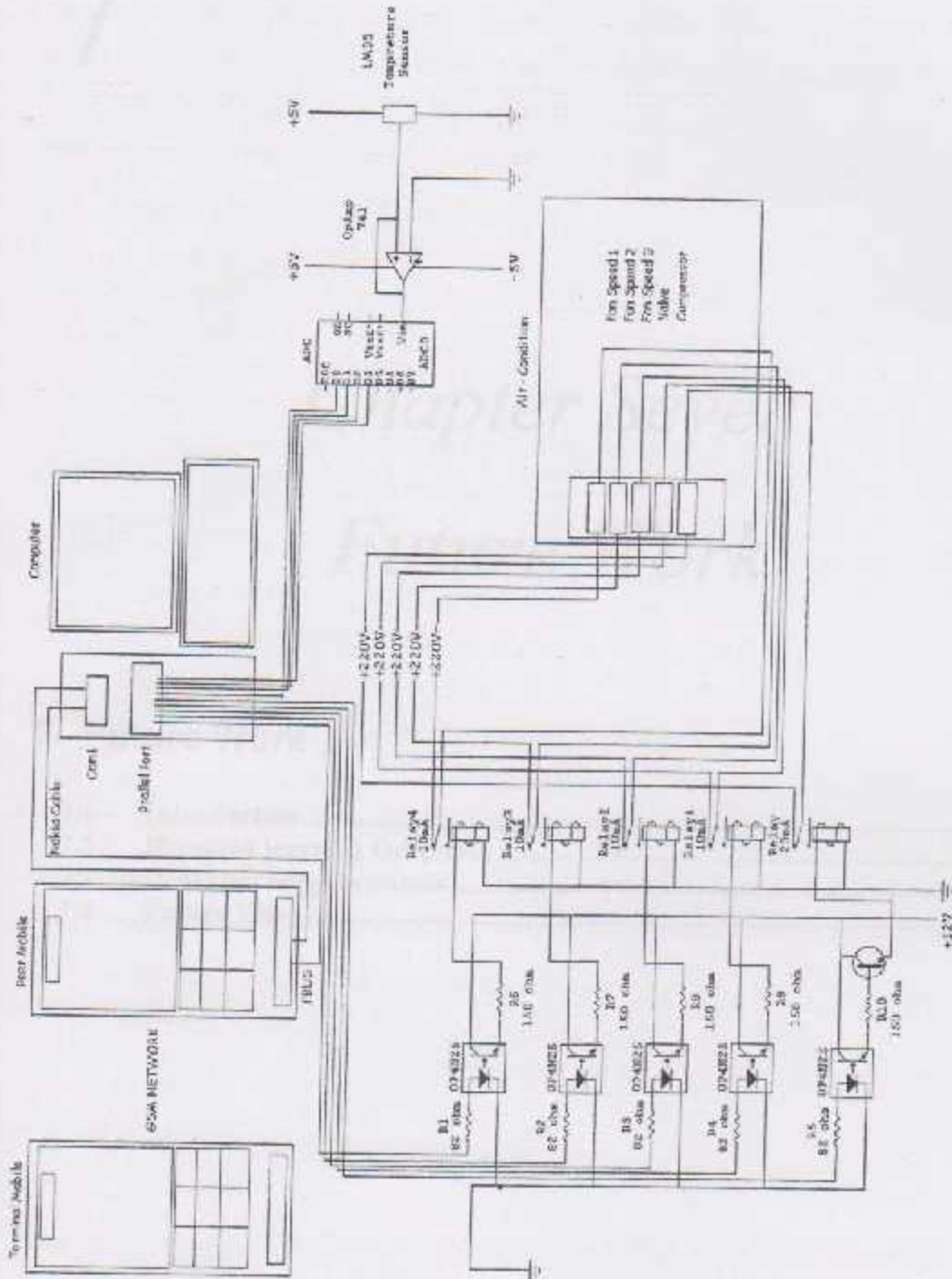


Figure 6.8 Third Project Schematic

7.1 Introduction

7 This chapter introduces the course with a view to developing the skills of formulating the design specifications for Air – Condition. Also the project sets a good way to developing the design. It will be the first stage of communication with experienced people. The design process will be discussed.

7.2 Expected Learning Outcomes

We are expecting to learn the following things before the beginning of implementing the project.

Chapter Seven

Future Work

7 Future Work

7.1	Introduction.....	108
7.2	Expected learning Outcomes	108
7.3	Real Learning Outcomes.....	108
7.4	Future View	109

Design – Condition – Your Day

Design – An system of control – describing the Air – Condition

7.2 Real Learning Outcomes

After the lecture students of the project will have an experience the following things.

- Learn how to use the program with the computer
- How to identify the real design

7.1 Introduction

The project that has been done was a step for developing the idea of remote control for devices, specially the Air – Condition. Also the project was a good step in developing smart houses. Meanwhile we have some recommendations and suggestions for the future work. The following section will discuss them.

7.2 Expected learning Outcomes

We were expecting to learn the following things before the beginning of implementing the project.

- Learn how to use and program one of the following :
 1. Field Programmable Gateway Array (FPGA).
 2. ToothPIC wireless microcontroller.
 3. Program PIC microcontroller.
 4. 8051 microcontroller.
- How to remotely control devices.
- Learn how to communicate the mobile FBUS serial.
- Learn how to access and have specific information from the mobile by the AT commands.
- Learn how to control digital Air – Conditions.
- How to overcome the problem of the accumulative of ice on the outdoor unit of the Air - Condition in winter days.
- Design the system of remotely controlling the Air – Condition.

7.3 Real Learning Outcomes

After the implementation of the project we have an expert in the following things:

- Learn how to use and program 8051 microcontroller.
- How to remotely control devices.

- Learn how to control traditional Air – Condition, because the digital one is for the university labs.
- Learn how to access and have specific information from the mobile by the AT commands.
- Faces many problem with communicating with the mobile and try to solve.
- Design the system of remotely controlling the Air – Condition, we were hoping to have this design with the microcontroller, but because of the problems that were arisen, we do it with PC parallel port.

7.4 Future View

After our work on this project and after facing many problems during the implementation, we as a project team, see that the following points may be a good improvement for this project, and also to avoid the problems were arisen.

- The Bluetooth technology is useful for develop this project, because there is nowadays microcontrollers like the one we mention in chapter 2 (PIC Tooth), which can be programmed from any computer that has a wireless communication easily, also this microcontroller make the communication between it and the peer mobile more easily. And finally this microcontroller can be used for remote controlling indoor the home.
- In this project we were used the short messages service to send the user command from the terminal mobile, but if the online calling was used, it could be an improvement for the project, by using DTMF integrated circuits connected to peer mobile that determine the button pressed on the terminal mobile. So the user online will control the device and see the effect.
- For the Air – Condition, adding a sensor to the outdoor unit, to avoid the accumulation of ice in winter days, will be an important improvement for this project.

- The humidity is one of the most important things that affect our comfortable, so adding a sensor for the humidity attached to the Air – Condition will be a good idea to develop this project.
- The project can be developed and reprogrammed to present other goals. For example it can be designed for controlling many things in the house like switching on and off the lights.
- Finally when you send an AT command to the mobile to read the message, the response will contain information, like the sender number, date and time of sending, this information may be used to add security for the system. Or may be the message sent by the user terminal mobile encrypted in some way to provide security for the system. The security for this system is very important because you remotely access and control devices.

References

- [1] http://en.wikipedia.org/wiki/Mobile_phone.
- [2] <http://www.gsmworld.com/technology/what.shtml>.
- [3] <http://www.embedtronics.com/nokia/fbus.html>.
- [4] <http://en.wikipedia.org/wiki/SMS>
- [5] http://en.wikipedia.org/wiki/Symbian_OS
- [6] <http://java.sun.com/javame/technology/index.jsp>.
- [7] University of south Australia, a thesis for bachelor of computer and information science, Paul McCarthy.
- [8] <http://www.bluetooth.com/Bluetooth/Learn/Works/>.
- [9] Hand book of air conditioning and refrigeration, Shan k. Wang, 2nd edition.
- [10] Air compressor with integral variable speed control, Energy Efficiency Association, 3rd edition.
- [11] http://www.mouthshut.com/productreviews/Microcontroller_Projects_Using_the_Basic_Stamp_-_Al_Williams-925058969.html.
- [12] http://en.wikipedia.org/wiki/Analog_to_digital_converter
- [13] http://en.wikipedia.org/wiki/Digital_to_analog_converter
- [14] http://en.wikipedia.org/wiki/PIC_microcontroller
- [15] <http://www.flexipanel.com/WirelessProducts.htm>.
- [16] <http://cegt201.bradley.edu/projects/proj2001/upmemory/FPGAtutorial.html>.

Vizualizācija programā

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

1998. gada dati
1999. gada dati
2000. gada dati

Diagrama, kurā ir attēloti visi dati
Diagrama, kurā ir attēloti visi dati
Diagrama, kurā ir attēloti visi dati
Diagrama, kurā ir attēloti visi dati

1998. gada dati

Diagrama, kurā ir attēloti visi dati
Diagrama, kurā ir attēloti visi dati
Diagrama, kurā ir attēloti visi dati

Appendices

Diagrama, kurā ir attēloti visi dati
Diagrama, kurā ir attēloti visi dati

1998. gada dati
1999. gada dati

2000. gada dati

Diagrama, kurā ir attēloti visi dati

1998. gada dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

1998. gada dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Diagrama, kurā ir attēloti visi dati

Visual Basic.Net program

```
Imports System.Windows.Forms
```

```
Public Class Form1
```

```
    Dim str As String  
    Dim str1 As String  
    Dim i As Integer
```

```
    Dim returnStr As String = "hello"
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
        ' Show all available COM ports.
```

```
        For Each sp As String In My.Computer.Ports.SerialPortNames  
            ListBox1.Items.Add(sp)
```

```
        Next
```

```
    End Sub
```

```
Function ReceiveSerialData() As String
```

```
    ' Receive strings from a serial port.
```

```
    Dim returnStr1 As String = "com1"
```

```
    Using com1 As IO.Ports.SerialPort =
```

```
        My.Computer.Ports.OpenSerialPort(returnStr1)
```

```
        com1.ReadTimeout = 10
```

```
        Dim incoming As String = com1.ReadExisting
```

```
        If incoming Is Nothing Then
```

```
            TextBox1.Text = "empty"
```

```
        Else
```

```
            TextBox1.Text &= incoming
```

```
            returnStr1 &= incoming
```

```
        End If
```

```
        com1.Close()
```

```
    End Using
```

```
    Return returnStr1
```

```
End Function
```

```
Sub SendSerialData(ByVal data As String)
```

```
    ' Send strings to a serial port.
```

```
    Dim mystr As String = "com1"
```

```
    Using com1 As IO.Ports.SerialPort =
```

```
        My.Computer.Ports.OpenSerialPort(mystr, 57600)
```

```
        Try
```

```
            com1.RtsEnable = True
```

```
            'com1.DtrEnable = True
```

```
            com1.Write(data)
```

```
            'com1.WriteTimeout = 250
```

```
            com1.ReadTimeout = 100
```

```

        'com1.RtsEnable = False
        'TextBox1.Text = com1.RtsEnable
        'com1.StrEnable = True

        Dim incoming As String = com1.ReadByte
        Dim incoming2 As String = com1.ReadLine
        Dim incoming3 As String = com1.ReadLine
        Dim incoming4 As String = com1.ReadLine
        'Dim incoming5 As String = com1.ReadLine
        'Dim incoming6 As String = com1.ReadLine

        TextBox1.Text = incoming4

        If incoming4 = 1 Then
            Out(Val("&H" & 378), Val(1))
            TextBox2.Text = "Air - Condition is now on"
            i = i + 1
        End If

        If incoming4 = 2 Then
            Out(Val("&H" & 378), Val(0))
            TextBox2.Text = "Air - Condition is now off"
            i = i + 1
        End If

        Catch exp As Exception
            TextBox2.Text = "No Message"
            com1.Close()
            Timer1.Enabled = True

        End Try

        com1.Close()
        'returnStr = "at+cmgr=1" + vbCrLf
        'SendSerialData(returnStr)
        Timer1.Enabled = True

    End Using
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    'Out(Val("&H" & 378), Val(1))
    i = 1
    str = i

    returnStr = "at+cmgr=" + str + vbCrLf
    SendSerialData(returnStr)

```

```

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

    str = i
    TextBox3.Text = i
    TextBox1.Text = " "
    returnStr = "at+cmgr=" + str + vbCrLf
    SendSerialData(returnStr)

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Out(Val("&H" & 37E), Val(0))
End Sub

End Class

```

```

1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100

```

8051 Microcontroller program

```
.equ    locat, 0x2000        ;Location for this program

.equ    cout, 0x0030
.equ    cin, 0x0032
.equ    pstr, 0x0038
.equ    newline, 0x0048

.equ    port_a, 0xF800
.equ    port_b, 0xF801
.equ    port_c, 0xF802
.equ    port_abc_pgm, 0xF803
.equ    port_d, 0xF900
.equ    port_e, 0xF901
.equ    port_f, 0xF902
.equ    port_def_pgm, 0xF903

.org    locat

.db     0xA5,0xE5,0xE0,0xA5    ;signature bytes
.db     35,255,0,0            ;id (35=prog)
.db     0,0,0,0              ;prompt code vector
.db     0,0,0,0              ;reserved
.db     0,0,0,0              ;reserved
.db     0,0,0,0              ;reserved
.db     0,0,0,0              ;user defined
.db     255,255,255,255      ;length and checksum (255=unused)
.db     "mobile communication",0;max 31 characters, plus the zero

.org    locat+64            ;executable code begins here

mov     dptr, #port_abc_pgm
mov     a, #128
movx    @dptr, a
mov     dptr, #port_def_pgm
mov     a, #128
movx    @dptr, a
```

```

mainloop:
clr    p3.4      ;redirect output to aux port
mov    dptr, #msg2
lcall  pstr      ;send msg2 to the aux port
jnb    ti, *     ;wait for it all to leave
jnb    ri, *     ;wait for the 8051 to set the RI flag
mov    a, sbuf   ;read the character from the serial port

mov    r1, a     ; save the value of accumulator in the
mov    r2, a     ; r1, r2, r3 general purpose registers
mov    r3, a

anl    a, 1      ; compare if response equal 1
jnz    On

mov    a, r1
anl    a, 2      ; compare if response equal 2
jnz    off

mov    a, r2
anl    a, 3      ; compare if response equal 3
jnz    SetTemp

mov    a, r3
anl    a, 4      ; compare if response equal 4
jnz    GetTemp

On:
mov    dptr, #0xF900
mov    a, #0000001b ;turn on the Air - Condition
movx   @dptr, a
ljmp  mainloop

off:
mov    dptr, #0xF900
mov    a, #00000010b ;turn off the Air - Condition
movx   @dptr, a
ljmp  mainloop

SetTemp:
ljmp  mainloop

GetTemp:
ljmp  mainloop

ret

msg2:
.db "AT", 13, 10, 0

```