

## **'CALL N CONTROL'**

### **MINI PROJECT REPORT**

*Submitted in the partial fulfillment for the award of Degree of  
Bachelor of Technology in Electronics and Communication under  
the University of Kerala*

Submitted by

AJITHA NAFILA P C (10400040)

AKHIL VIJAYAKUMAR (10400003)

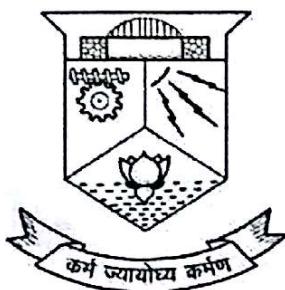
AMAL V (10400004)

S6 EC



**Department of Electronics and Communication  
College of Engineering  
Thiruvananthapuram**

**DEPARTMENT OF ELECTRONICS & COMMUNICATION  
COLLEGE OF ENGINEERING  
THIRUVANANTHAPURAM  
2012-2013**



**CERTIFICATE**

This is to certify that this report entitled "**CALL n CONTROL**" submitted herewith is a bonafide record of the mini project done by **Amal V, Akhil Vijayakumar & Ajitha Nafila P C** in partial fulfillment of requirements for the award of **Bachelor of Technology** in **Electronics & Communication Engineering** under the **University of Kerala**, during the academic year 2012-2013.

**Dr. J David**

HOD & Professor  
Dept. of ECE  
College of Engineering  
Thiruvananthapuram

**Dr. Deepa P Gopinath**

Asst. Professor  
Dept. of ECE  
College of Engineering  
Thiruvananthapuram

**External Examiner**

## **ACKNOWLEDGEMENT**

With prayers to GOD for his grace and blessings, for without his unforeseen guidance, this project would have remained only in dreams.

We express our sincere gratitude to our principal, Dr. Sheela S and our Head of the Department Dr. J David for providing us the ambience for carrying out the work of our project and Ms. Latha V, our staff advisor, without whose help and motivation we could not have completed this project.

We are profoundly indebted to guide Ms. Deepa P Gopinath and all other lecturers in Electronics and Communication Department for their incessant guidance, constructive criticism and encouragement throughout the tenure of the study.

Last, but not the least, we extend our deepest gratitude to our parents and friends without whose support this project wouldn't have become a reality.

Akhil Vijayakumar

Ajitha Nafila P C

Amal V

## **ABSTRACT**

In modern days, we must use various high-tech machineries and equipment to get our jobs done and make the life easier. These machineries should be controlled by the homeowner from any location as the homeowner might be away from home at workplace or traveling in a different place in the weekend. Thus a system of remote monitoring and controlling are very much necessary. Smart home is one of these types of system equipped with home appliances which we wish to control smartly from anywhere.

The objective of this project is to enable users to remotely control their home appliances and systems using a cell phone-based interface. The product uses the advantages of DTMF for the functioning and the DTMF codes are decoded and used for understanding the requirement or the order from the user. This helps to control your home appliances from a distant place using mobile phone by just calling.

## TABLE OF CONTENTS

<u>Serial no.</u>	<u>CHAPTER</u>	<u>Page no.</u>
1	<b>Introduction</b>	1
2	<b>'Call n Control'</b> 2.1 Block diagram 2.2 Circuit diagram & explanation 2.3 Working 2.4 Hardware requirements 2.5 Hardware description 2.6 PCB layout	2 4 7 12 14 21
3	<b>Implementation</b> 3.1 PCB fabrication 3.2 Soldering	24 27
4	<b>Conclusion</b>	24
*	<b>Appendix</b>	
**	<b>References</b>	

## Chapter 1

### **INTRODUCTION**

The aim of the proposed system is to develop a cost effective solution that will provide controlling of home appliances remotely and enable home security against intrusion in the absence of homeowner. The home appliances control system with an affordable cost was thought to be built that should be mobile providing remote access to the appliances and allowing home security. These devices should be controlled as well as turn on/off if required. Most of the time it was done manually. Now it is a necessity to control devices more effectively and efficiently at anytime from anywhere.

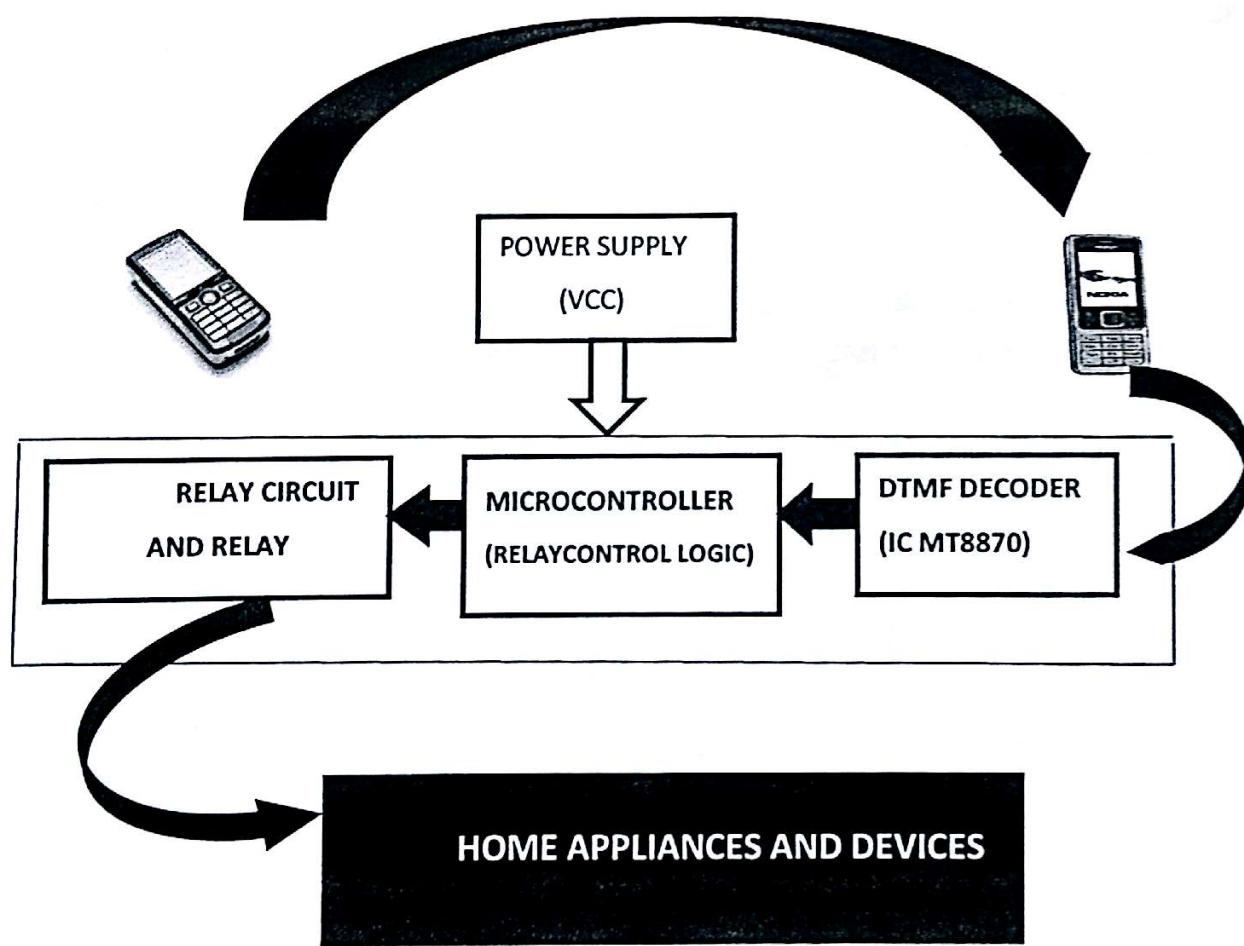
What happened if we forgot to turn off the light when you left home?? Before: Nothing to do... Now a day's the conservation of electrical energy is very important. So to avoid the wastage we should turn on lights only if we require. But during above situations it is very difficult. That's why we say "now it is a necessity to control devices more effectively and efficiently at anytime from anywhere".

In this system, we are going to develop a cellular phone based home/office appliance. This system is designed for controlling arbitrary devices, it includes a cell phone which is connect to the system via head set. To active the cellular phone unit on the system a call is to be made and as the call is answered, the user can turn on/off the appliances using the numbers in the keypad by just pressing them. The underlying principle mainly relies up on the ability of DTMF (Double Tune Multi Frequency) ICs to generate DTMF corresponding to a number or code in the number pad and to detect the same number or code from its corresponding DTMF.

## Chapter 2

### HOME APPLIANCES CONTROL USING MOBILE PHONE – ‘CALL N CONTROL’

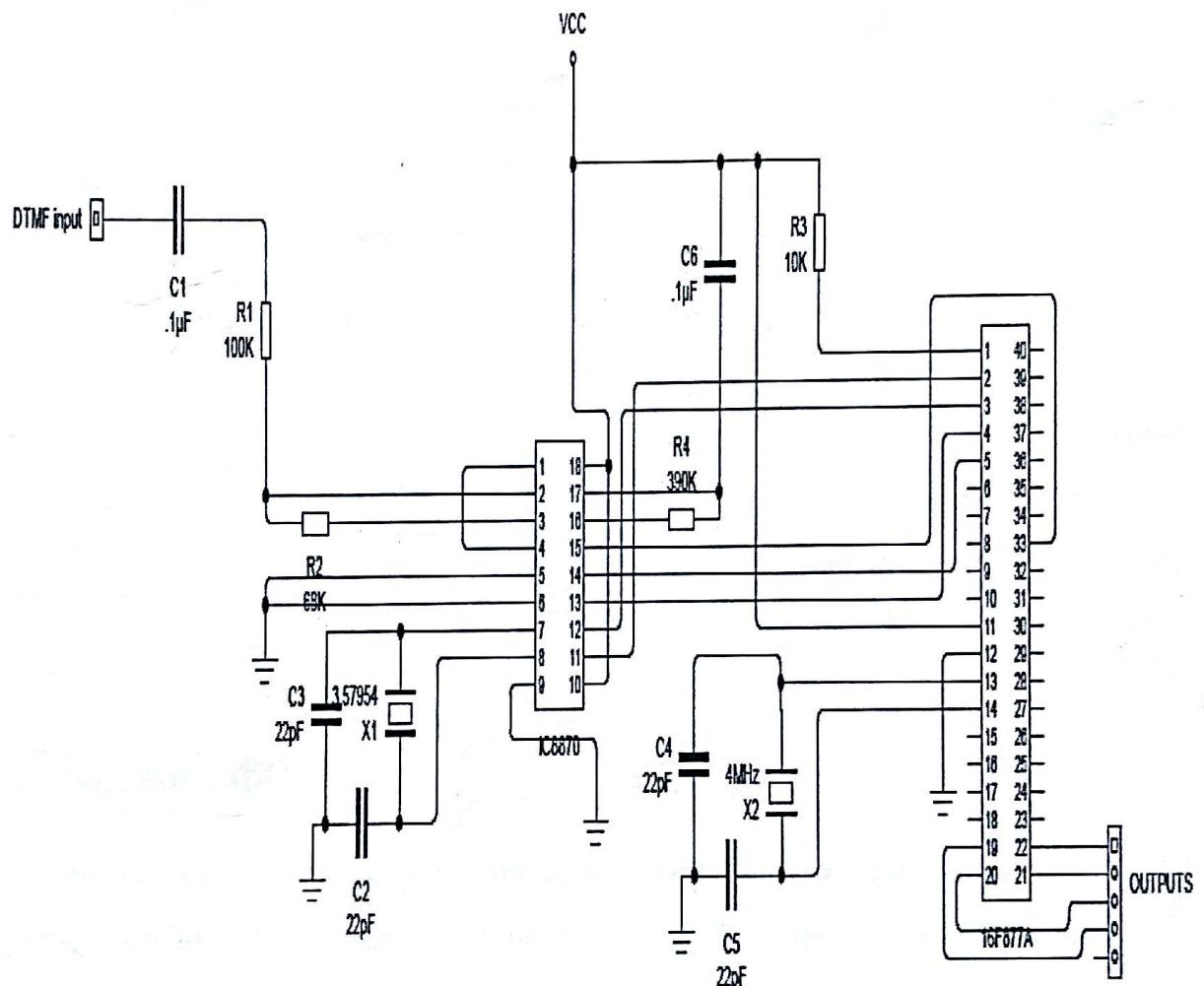
#### 2.1 BLOCK DIAGRAM:



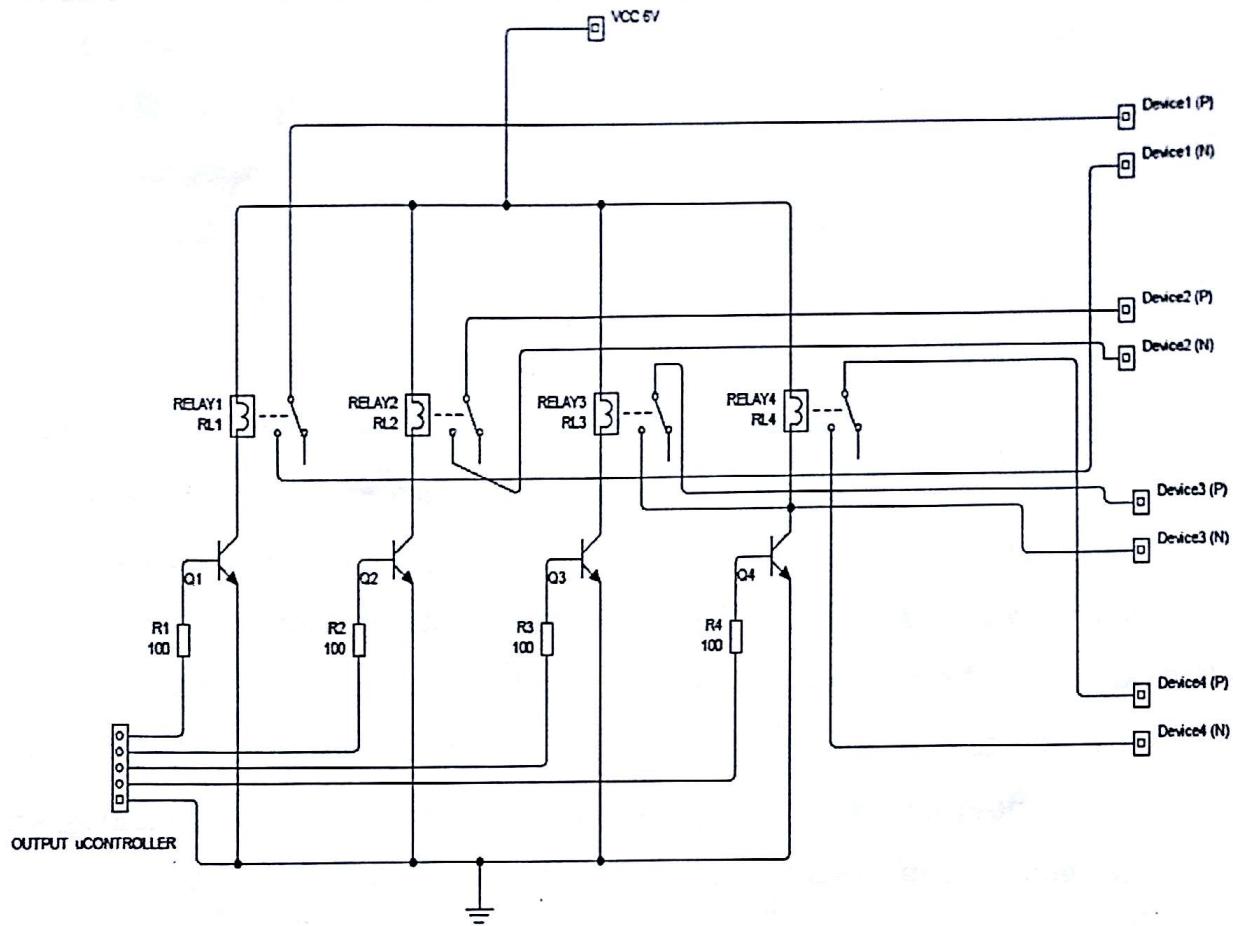
- Whenever you want to turn on/off a device ,make a call from any phone to the mobile phone connected in the device section.
- The device mobile phone which is in auto answer mode so that the call should connect automatically when a call is received in the mobile.
- In order to control the on and turn off of a device user press the keypad key corresponding to the device to be controlled.
- Receiver mobile phone receives the DTMF tone corresponding to the key, it gets decoded. The phone at the receiver is connected to the decoder circuit through microphone/headset and the phone should be in key tone activated profile.
- This signal passed to microcontroller which produce the relay control signal By controlling the relay circuit the device can be turned on/off.

## 2.1CIRCUIT DIAGRAM:

MAIN CIRCUIT (DECODER AND MICROCONTROLLER)



## RELAY CIRCUIT (OUTPUT)



## Circuit diagram explanation:

Whenever you press any key on your mobile phone keypad, the delayed steering (Std) output of the IC goes high on receiving the tone-pair, causing LED5 (connected to pin 15 of IC via resistor R15) to glow. It will be high for a duration de-pending on the values of capacitor and resistors at pins 16 and 17. The optional circuit shown within dotted line is used for guard time adjustment.

The LEDs connected via resistors R11 to R14 at pins 11 through 14, respectively, indicate the output of the IC. The tone-pair DTMF (dual-tone multi-frequency) generated by pressing the telephone button is converted into bi-binary values internally in the IC. The binary values are indicated by glowing of LEDs at the

output pins of the IC. LED1 represents the lowest significant bit (LSB) and LED4 represents the most significant bit (MSB). So, when you dial a number, say, 5, LED1 and LED3 will glow, which is equal to 0101. Similarly, for every other number dialed on your telephone, the corresponding LEDs will glow. Thus, a non-defective IC should indicate proper binary values corresponding to the decimal number pressed on your telephone key-pad.

The user makes a call from any phone to the mobile phone connected in the device section. The device consists of a dedicated mobile phone which is in auto answer mode so that the call should connect automatically when a call is received in the mobile. In order to control the on and turn off of a device user have to press the keypad key corresponding to the device to be controlled. And when the phone at the receiver receives the DTMF tone corresponding to the key, it gets decoded. The phone at the receiver is connected to the decoder circuit through microphone/headset and the phone should be in key tone activated profile. The received tone is decoded by the Decoder IC and corresponding signals produced. This signal passed to microcontroller and microcontroller produce signal required to control the relay and output circuit according to the algorithm written on the micro programme. By controlling the relay circuit the turn on and turn off process of the devices can be done. By using this maximum of 12 devices can be established but in the given product we are only provides the facility to control four devices.

## **2.3 WORKING:**

The device works on the principle of DTMF decoding. The user makes a call from any phone to the mobile phone connected in the device section. The device consists of a dedicated mobile phone which is in auto answer mode so that the call should connect automatically when a call is received in the mobile. In order to control the on and turn off of a device user have to press the keypad key corresponding to the device to be controlled. And when the phone at the receiver receives the DTMF tone corresponding to the key, it gets decoded. The phone at the receiver is connected to the decoder circuit through microphone/headset and the phone should be in key tone activated profile. The received tone is decoded by the Decoder IC and corresponding signals produced. This signal passed to microcontroller and microcontroller produce signal required to control the relay and output circuit according to the algorithm written on the micro programme.

DTMF is a signaling system for identifying the keys or better say the number dialed on a push button or DTMF keypad. The early telephone systems used pulse dialing or loop disconnect signaling. This was replaced by multi frequency (MF) dialing. DTMF is a multi-frequency tone dialing system used by the push button keypads in telephone and mobile sets to convey the number or key dialed by the caller. DTMF has enabled the long distance signaling of dialed numbers in voice frequency range over telephone lines. This has eliminated the need of telecom operator between the caller and the callee and evolved automated dialing in the telephone switching centres. DTMF (Dual tone multi frequency) as the name suggests uses a combination of two sine wave tones to represent a key. These tones are called row and column frequencies as they correspond to the layout of a telephone keypad. A DTMF keypad (generator or encoder) generates a

sinusoidal tone which is mixture of the row and column frequencies. The row frequencies are low group frequencies. The column frequencies belong to high group frequencies. This prevents misinterpretation of the harmonics. Also the frequencies for DTMF are so chosen that none have a harmonic relationship with the others and that mixing the frequencies would not produce sum or product frequencies that could mimic another valid tone. The high-group frequencies (the column tones) are slightly louder than the low-group to compensate for the high-frequency roll off of voice audio systems.

The row and column frequencies corresponding to a DTMF keypad have been indicated in the above figure. DTMF tones are able to represent one of the 16 different states or symbols on the keypad. This is equivalent to 4 bits of data, also known as nibble.

Today, most telephone equipment use a DTMF receiver IC. One common DTMF receiver IC is the Motorola MT8870 that is widely used in electronic communications circuits. The MT8870 is an 18-pin IC. It is used in telephones and a variety of other applications. When a proper output is not obtained in projects using this IC, engineers or technicians need to test this IC separately. A quick testing of this IC could save a lot of time in re-search labs and manufacturing industries of communication instruments. Here's a small and handy tester circuit for the DTMF IC. It can be assembled on a multipurpose PCB with an 18-pin IC base. One can also test the IC on a simple breadboard.

For optimum working of telephone equipment, the DTMF receiver must be designed to recognize a valid tone pair greater than 40 ms in duration and to accept successive digit tone-pairs that are greater than 40 ms apart. However, for other applications like remote controls and radio communications, the tone duration may differ due to noise considerations. Therefore, by adding an extra resistor and steering diode the tone duration can be set to different values. The circuit is

configured in balanced-line mode. To reject common-mode noise signals, a balanced differential amplifier input is used. The circuit also provides an excellent bridging interface across a properly terminated telephone line. Transient protection may be achieved by splitting the input resistors and inserting ZENER diodes (ZD1 and ZD2) to achieve voltage clamping. This allows the transient energy to be dissipated in the resistors and diodes, and limits the maximum voltage that may appear at the inputs.

Whenever you press any key on your local telephone keypad, the delayed steering (Std) output of the IC goes high on receiving the tone-pair, causing LED5 (connected to pin 15 of IC via resistor R15) to glow. It will be high for a duration depending on the values of capacitor and resistors at pins 16 and 17. The optional circuit shown within dotted line is used for guard time adjustment.

The LEDs connected via resistors R11 to R14 at pins 11 through 14, respectively, indicate the output of the IC. The tone-pair DTMF (dual-tone multi-frequency) generated by pressing the telephone button is converted into bi-nary values internally in the IC. The binary values are indicated by glowing of LEDs at the output pins of the IC. LED1 represents the lowest significant bit (LSB) and LED4 represents the most significant bit (MSB). So, when you dial a number, say, 5, LED1 and LED3 will glow, which is equal to 0101. Similarly, for every other number dialed on your telephone, the corresponding LEDs will glow. Thus, a non-defective IC should indicate proper bi-nary values corresponding to the decimal number pressed on your telephone key-pad.

To test the DTMF IC 8870/KT3170, proceed as follows:

Connect local telephone and the circuit in parallel to the same telephone line.

Switch on S1. (Switch on auxiliary switch S2 only if keys A, B, C, and D are to be used.)

- Now push key '\*' to generate DTMF tone.
- Push any decimal key from the telephone keypad.
- Observe the equivalent binary as shown in the table.
- If the binary number implied by glowing of LED1 to LED4 is equivalent to the pressed key number (decimal/A, B, C, or D), the DTMF IC 8870 is correct.

Keys A, B, C, and D on the telephone keypad are used for special signaling and are not available on standard pushbutton telephone keypads. Pin 5 of the IC is pulled down to ground through resistor R8. Switch on auxiliary switch S2. Now the high logic at pin 5 enables the detection of tones representing characters A, B, C, and D.

By controlling the relay circuit the turn on and turn off process of the devices can be done. By using this maximum of 12 devices can be established but in the given product we are only provides the facility to control four devices. By simple modification in the software and hardware side, the capability of the device can be extended to its maximum limit. The heart of the circuit is DTMF decoder which decodes the DTMF tone and produces the output binary signals.

DTMF signal transmits along with the modulated speech signal in a mobile and telephone communication. So this helps the receiver to understand what are the keys pressed in the transmitter phone/mobile phone.

---

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

Table showing DTMF  
Low and High frequency tones  
and decoded output of IC8870

## 2.4 HARDWARE REQUIREMENTS

### **DECODER AND CONTROLLER CIRCUIT**

S.No	Component	Specification	Quantity
1	DTMF Decoder	MT8870	1
2	MICROCONTROLLER	16F877A	1
3	CRYSTAL	4 MHz	1
		3.579545 MHz	1
4	CAPACITOR	22Pf	4
		0.1UF	2
5.	RESISTORS	10K	1
		68K	1
		390K	1
		100K	1

### **OUTPUT CIRCUIT/ RELAY CIRCUIT**

S.No	Components	Specification	Quantity
1	TRANSISTOR	PN2222	4
2	RELAY	6V	4
3	RESISTORS	100 ohm	4

### **POWER SUPPLY CIRCUIT**

S.No	Components	Specification	Quantity
1	TRANSFORMER	230:15	1
2	DIODE	1N4001	4
3	CAPACITOR	22pf	1
		0.1uf	1
4	VOLTAGE REGULATOR	7805	1

## 2.5 HARDWARE DESCRIPTION

### 1.DTMF and DTMF Decoder

*“Dual-tone multi-frequency signaling (DTMF) is used for telecommunication signaling over analog telephone lines in the voice-frequency band between telephone handsets and other communications devices and the switching center. The version of DTMF that is used in push-button telephones for tone dialing is known as **Touch-Tone**. It was developed by Western Electric and first used by the Bell System in commerce, using that name as a registered trademark. DTMF is standardized by ITU-T. It is also known in the UK as MF4.”*

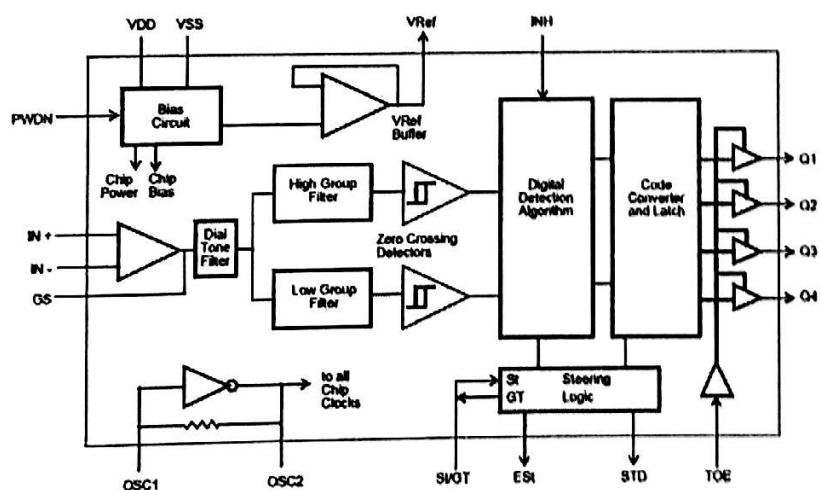
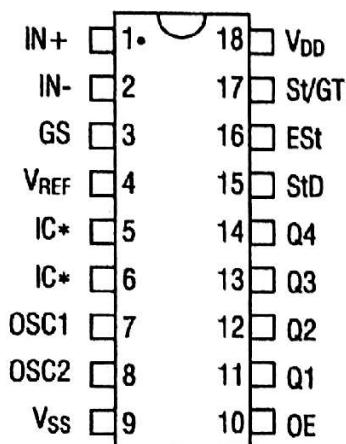
*(Wikipedia-Dual Tone Multi-Frequency)*

High Group Frequencies			
1209 Hz	1336 Hz	1477 Hz	1633 Hz
<b>Low Group Frequencies</b>			
697 Hz	770 Hz	852 Hz	941 Hz
<b>DTMF code/TouchTone corresponding to each key (Last column usually not found on telephones)</b>	<b>1</b>	<b>2</b>	<b>3</b>
	<b>4</b>	<b>5</b>	<b>6</b>
	<b>7</b>	<b>8</b>	<b>9</b>
	<b>*</b>	<b>0</b>	<b>#</b>
			<b>C</b>
			<b>B</b>
			<b>A</b>

**DTMF** is a signaling system for identifying the keys or better say the number dialed on a **pushbutton or DTMF keypad**. The early telephone systems used pulse dialing or loop disconnect signaling. This was replaced by multi frequency (MF) dialing. DTMF is a multi-frequency tone dialing system used by the push button keypads in telephone and mobile sets to convey the number or key dialed by the caller. DTMF has enabled the long distance signaling of dialed numbers in voice frequency range over telephone lines. This has eliminated the need of telecom operator between the caller and the callee and evolved automated dialing in the telephone switching centers'. **DTMF (Dual tone multi frequency)** as the name suggests uses a **combination of two sine wave tones** to represent a key. These tones are called row and column frequencies as they correspond to the layout of a telephone keypad. A **DTMF keypad** (generator or encoder) generates a sinusoidal tone which is **mixture of the row and column frequencies**. The row frequencies are low group frequencies. The column frequencies belong to high group frequencies. This prevents misinterpretation of the harmonics. Also the frequencies for **DTMF** are so chosen that none have a harmonic relationship with the others and that mixing the frequencies would not produce sum or product frequencies that could mimic another valid tone. The high-group frequencies (the column tones) are slightly louder than the low-group to compensate for the high-frequency roll off of voice audio systems.

The row and column frequencies corresponding to a **DTMF keypad** have been indicated in the above figure. **DTMF tones** are able to represent one of the 16 different states or symbols on the keypad. This is equivalent to 4 bits of data, also known as nibble.

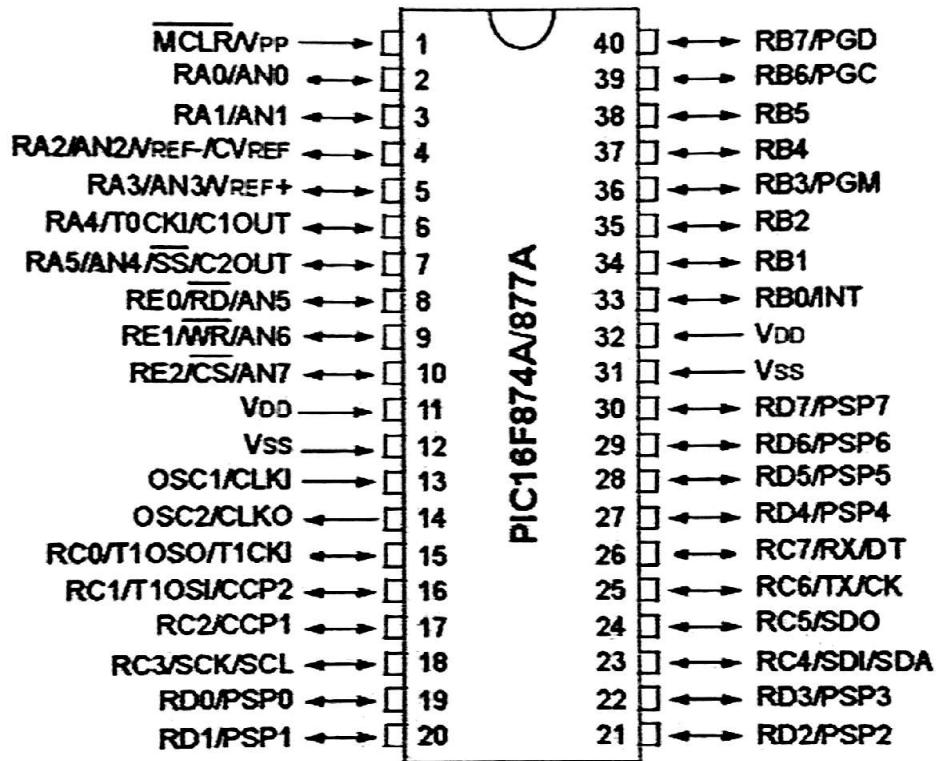
## DTMF DECODER IC- 8870



## 2.PIC MICRO CONTROLLER

### PIN DIAGRAM

40-Pin PDIP



### Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for programbranches which are two cycle
- Operating speed: DC - 20 MHz clock inputDC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
- Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory

- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming□ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V

### **Peripheral Features:**

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns- Compare is 16-bit
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter

- Synchronous Serial Port (SSP) with SPI (Mastermode) and I2C (Master/Slave)
- Universal Synchronous Asynchronous ReceiverTransmitter (USART/SCI) with 9-bit addressdetection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

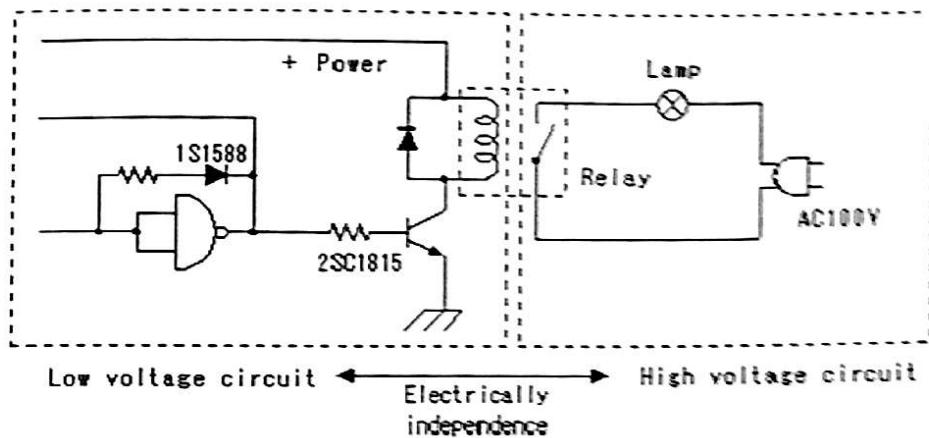
### 3.RELAY

The relay takes advantage of the fact that when electricity flows through a coil, it becomes an electromagnet. The electromagnetic coil attracts a steel plate, which is attached to a switch. So the switch's motion (ON and OFF) is controlled by the current flowing to the coil, or not, respectively.

A very useful feature of a relay is that it can be used to electrically isolate different parts of a circuit. It will allow a low voltage circuit (e.g. 5VDC) to switch the power in a high voltage circuit (e.g. 100 VAC or more).

The relay operates mechanically, so it can not operate at high speed.

There are many kind of relays. You can select one according to your needs. The various things to consider when selecting a relay are its size, voltage and current capacity of the contact points, drive voltage, impedance, number of contacts, resistance of the contacts, etc. The resistance voltage of the contacts is the maximum voltage that can be conducted at the point of contact in the switch. When the maximum is exceeded, the contacts will spark and melt, sometimes fusing together. The relay will fail. The value is printed on the relay.



#### 4.VOLTAGE REGULATOR (IC7805)

It is a three pin IC used as a voltage regulator. It converts unregulated DC current into regulated DC current.

Normally we get fixed output by connecting the voltage regulator at the output of the filtered DC (see in above diagram). It can also be used in circuits to get a low DC voltage from a high DC voltage (for example we use 7805 to get 5V from 12V). There are two types of voltage regulators 1. fixed voltage regulators (78xx, 79xx) 2. variable voltage regulators(LM317) In fixed voltage regulators there is another classification

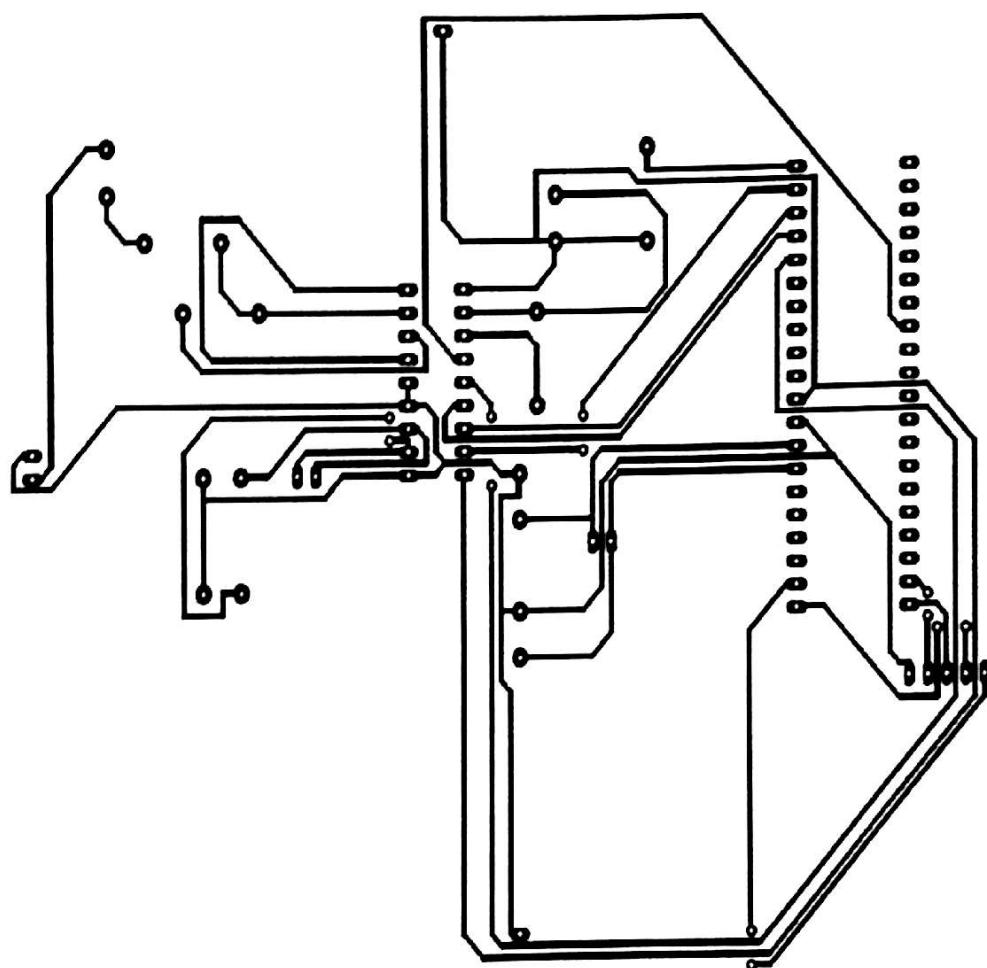
1. +ve voltage regulators

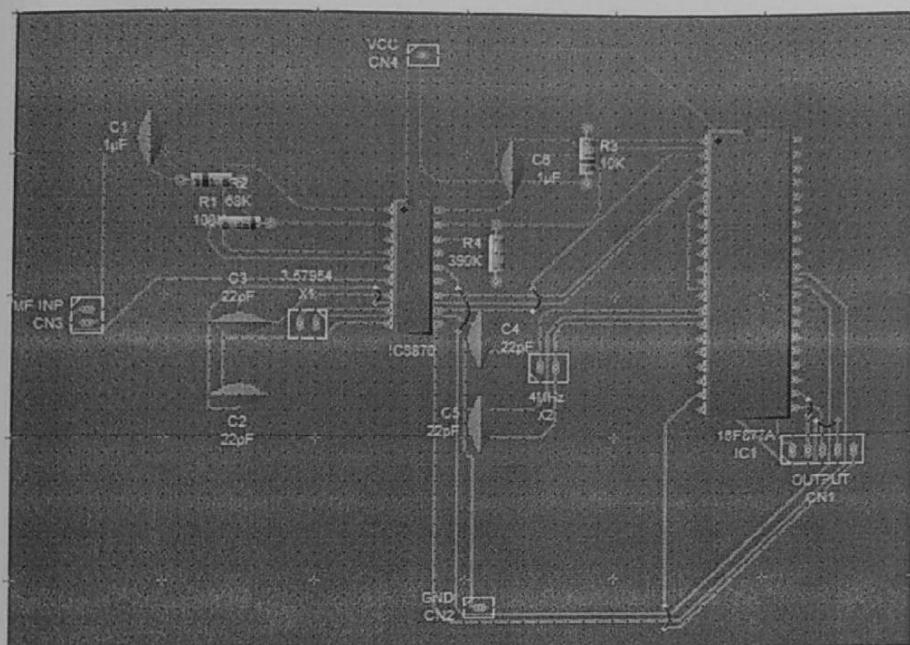
2. -ve voltage regulators

**Positive voltage :** This include 78xx voltage regulators. The most commonly used ones are 7805 and 7812. 7805 gives fixed 5V DC voltage if input voltage is in (7.5V, 20V).

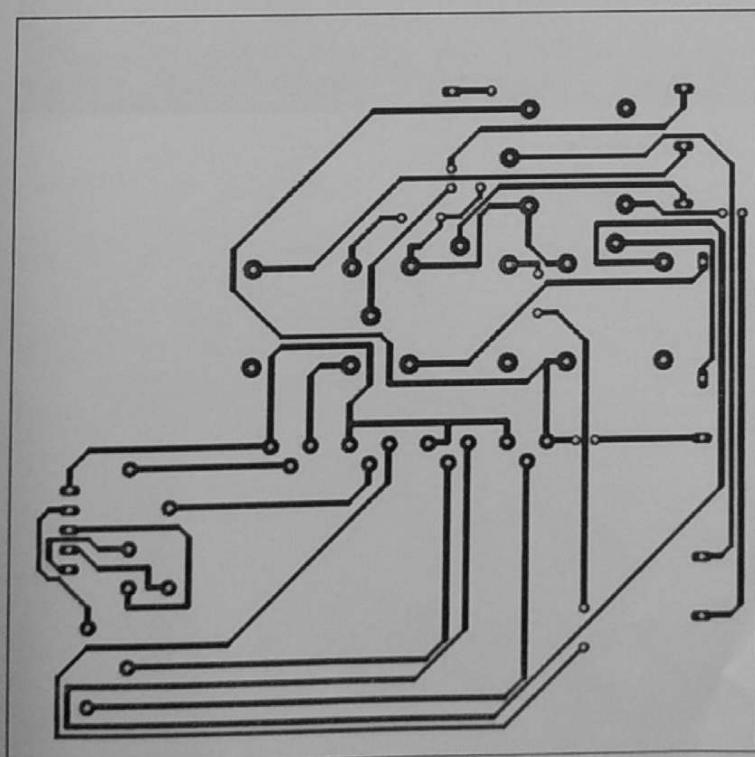
## 2.6 PCB LAYOUT

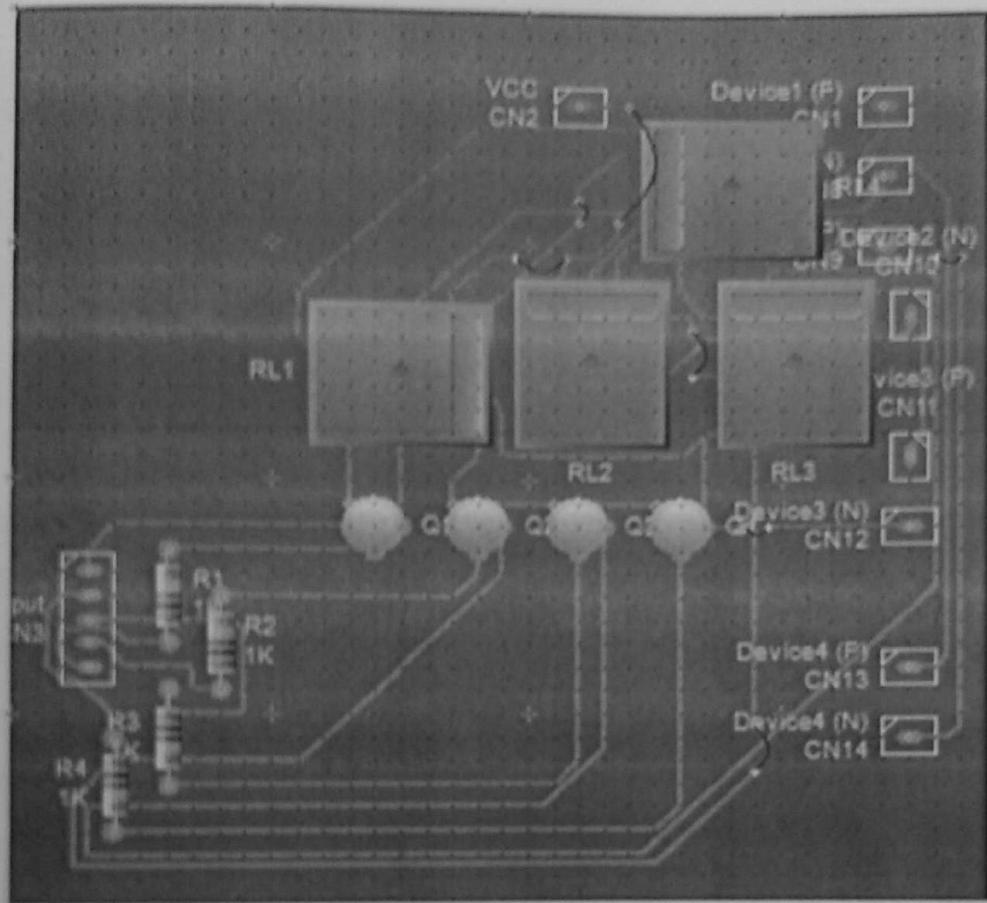
MAIN CIRCUIT (ARTWORK AND REAL WORLD)





### RELAY CIRCUIT (ART WORK AND REAL WORLD)





## Chapter 3

### IMPLEMENTATION

#### 4.1 PCB FABRICATION:

A printed circuit board is defined as an insulating material onto which an electronic circuit has been printed or etched. PCB is developed by etching process. First of all, the required circuit is designed and a layout is prepared. The layout can either be done with hand or by using PCB designing software like “WINBOARD” or “ORCAD”. Since layout preparation using software is much less tedious and more flexible, the layout for the circuits was designed using ORCAD. The layout of the circuit is transferred to the PCB with proper interconnections. The layout is made on the PCB on the component side as well as on the copper clad side. Spaces are provided for drilling holes for inserting the respective components of the circuit.

Now the instrumental interconnections are to be done.

Printed wiring is of two types:

- o Rigid type
- o Flexible type

The process begins with copper foil bonded to an insulating copper clad material. The bonding occurs through an electroplating operation. Following suitable cleansing and other preparations, a pattern of desired circuit configuration with interconnections are marked first with a scribbler. Then the stencil of the above layout is made on a plastic transparent sheet and is used as mask. The resist or the

etchant resistant material is applied to the copper foil as a part of the photo printing process. The resist is a material such as mask or ink, which prevents those parts of the copper foil from etching away. The resist is deposited on the copper cladding in the exact pattern of the circuit. The clad is etched with  $\text{FeCl}_3$  solution containing  $\text{H}_2\text{SO}_4$  or HCl for increasing the activeness of  $\text{FeCl}_3$  in etching out the Cu out of the clad. After etching only those portions protected by the resist have copper foil remaining. The rest of the copper has been removed.

After etching, the PCB is stripped or subjected to removal of resist and cleaned to ensure that no etchant remains. The board is now ready for fabrication, drilling & trimming. The required holes are drilled now using the drill (twist drill)..

#### **Advantages of PCB's over normal wiring are:**

PCBs are necessary for interconnecting a large number of electronic components in a very small area with minimum parasitic writing effects.

1. PCBs are stimulated for mass production with less chance of wiring error.
2. Small components are easily mounted.
3. Wiring micro phony is avoided.
4. Serving is simplified.

#### **Practical Hints (for software design of PCB)**

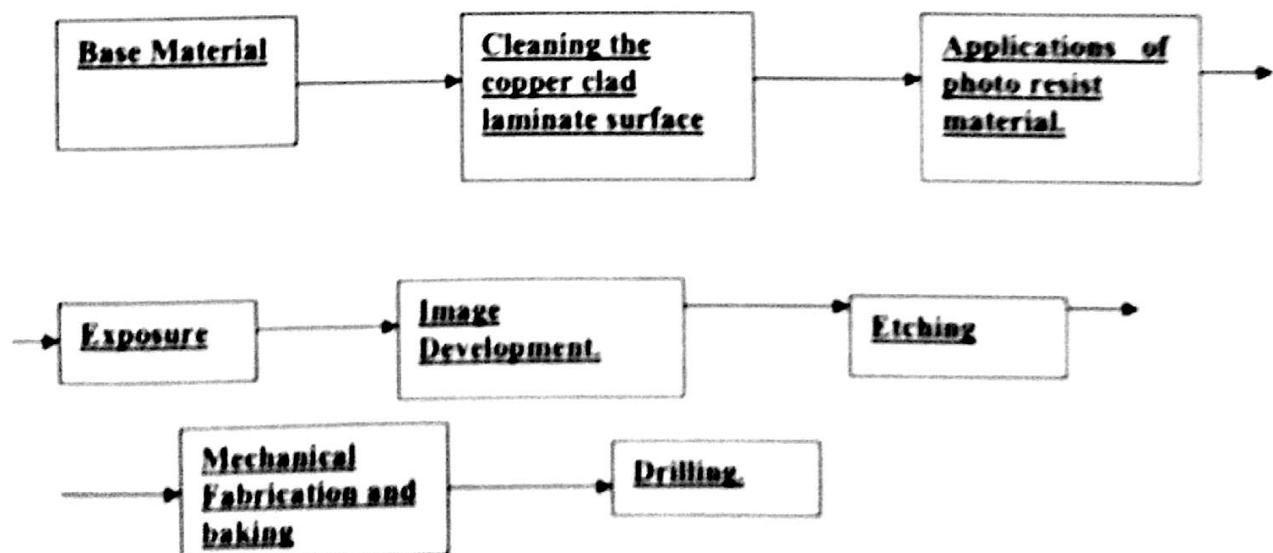
Before proceeding with the design of PCB, certain predesign effort is necessary. It includes:

1. At least a hand-drawn schematic of the circuit.
2. Component details include those of connectors.
3. Mechanical specifications, including board size, shape, size/position of mounting holes, location of connectors of PCB, etc.

4. PCB type: single/double-sided, PTH or non-PTH.
5. Pattern, size, shape of parts and wires.
6. Defining restrictions, spacing, etc.
7. Defining source files.
8. Selecting measurement units.
9. Creating board outline.
10. Setting system grid.
11. Defining manufacturing technology.
12. Selecting colour scheme.

### **Basic steps in PCB manufacturing.**

The base material is selected according to application. It is mechanically and chemically cleaned. Then the photo resist solution is uniformly applied. Photo resist is an organic solution which when exposed to light of proper wavelength, change the solubility of in the developers.



**Fig:steps for PCB manufacture**

Laminate coating of photo resist is done by:

1. Spray coating
2. Dip coating
3. Roller coating

The coated copper clad laminate film negatives are kept in glass frames having intimate contact with each other. The assembly is exposed to UV light for three minutes. The exposed board is raised in the developer tank. Proper developer has to be used for a particular photo resist. Then the PCB is dyed in a tray.

### **3.2 SOLDERING**

Soldering is the process of joining metals by using lower melting point metal to wet or alloy with the joined surfaces. Solder is the joining material. It melts below  $427^{\circ}\text{C}$ . Soldered joints in electronic circuits will establish strong electrical connections between component leads. The popularly used solders are alloys of tin and lead that melt below the melting point of tin.

In order to make the surfaces accept the solder readily, the component terminals should be cleaned chemically or by abrasion using blades or knives. Small amount of lead coating can be done on the cleaned portion of the leads using soldering iron. This process is called tinning. Zinc chloride or ammonium chloride separately or in combination are the most commonly used fluxes. These are available in petroleum jelly as paste flux. A solder joint can at first glance to be okay, but under close examination it could be a 'Dry Joint'. A dry joint is when either the circuit board or the leg of the component has not been properly heated to allow the solder to flow between the surfaces freely. This creates an intermittent or no electrical connection. This can also be caused by a lack of flux or if you reuse

old solder. Quite often, reheating a bad join will cure the problem but in a lot of cases, the old solder will need to be removed and some new solder applied. The residues, which remain after the soldering, may be washed out with more water, accompanied by brushing.

Soldering iron is the tool used to melt the solder and apply at the joints in the circuit. It operates in 230V mains supply. The iron bit at the tip of it gets heated up within a few minutes. 50W and 25W soldering irons are commonly used for soldering purposes.

### **Procedure:**

1. Make a layout of the circuit.
2. Straighten and clean the component leads using blade or knife. Apply a little flux on the leads. Take a little solder on soldering iron and apply the molten solder on the leads.
3. Mount the components on the PCB by bending the leads of the components using nose-pliers.
4. Apply flux on the joints and solder the joints. Soldering must be done in minimum time to avoid dry soldering and heating up of components.
5. Wash the residues using water and brush

## Chapter 4

### **CONCLUSION**

Our product 'call n control' will enable users to remotely control their home appliances and systems using a cell phone-based interface. To access the control unit, the user should send an authentication code along with the required/desired function/action to his/her home control system via a mobile phone. Upon being properly authenticated, the cell phone-based interface at home (control unit) would relay the commands to a microcontroller that would perform the required function/action.

'call n control' circuit consist of discrete components of low cost. So whole product is cost effective . It is also very user friendly. This is an efficient system built with simple components and less complexity. The product is efficiently designed for easy handling, user-friendliness and compactness. We can guarantee that 'call n control' will be a first step for your dream for a smart home.

---

**APPENDIX**

# PROGRAM FOR PIC IC 16877A

```
/******
***** File name:      homeauto.c
***** Date:           Friday, June 21, 2013 17:50:08
***** */

#define MX_PIC

//Defines for microcontroller
#define P16F877A
#define MX_EE
#define MX_EE_TYPE2
#define MX_EE_SIZE 256
#define MX_SPI
#define MX_SPI_C
#define MX_SPI_SDI 4
#define MX_SPI_SDO 5
#define MX_SPI_SCK 3
#define MX_UART
#define MX_UART_C
#define MX_UART_TX 6
#define MX_UART_RX 7
#define MX_I2C
#define MX_MI2C
#define MX_I2C_C
#define MX_I2C_SDA 4
#define MX_I2C_SCL 3
#define MX_PWM
#define MX_PWM_CNT 2
#define MX_PWM_TRIS1 trisc
#define MX_PWM_1 2
#define MX_PWM_TRIS2 trisc
#define MX_PWM_2 1

//Functions
#define MX_CLK_SPEED 4000000
#ifndef _BOOSTC
#include <system.h>
#endif
#ifndef HI_TECH_C
#include <pic.h>
#endif
```

```
//Configuration data
#ifndef _BOOSTC
#pragma DATA 0x2007, 0x3f3a
#endif
#ifndef HI_TECH_C
__CONFIG(0x3f3a);
#endif

//Internal functions
#include "C:\Program Files (x86)\Matrix Multimedia\Flowcode
v4\FCD\internals.h"

//Macro function declarations
void FCM_codecheck();

//Variable declarations
char FCV_FLAG1;
char FCV_FLAG2;
char FCV_FLAG3;
char FCV_INPUT;
char FCV_FLAG4;

//Defines:
***** Macro Substitutions *****
portd = LED Port Register
trisd = LED Data Direction Register
4 = LED Pin Mask
1 = LED Active Polarity
***** */

//LED(0): //Macro function declarations

void FCD_LED0_LEDOn();
void FCD_LED0_LEDOff();
//Defines:

***** Macro Substitutions *****
portd = LED Port Register
trisd = LED Data Direction Register
8 = LED Pin Mask
1 = LED Active Polarity
***** */
```

```

//LED(1): //Macro function declarations
void FCD_LED1_LEDOn();
void FCD_LED1_LEDOff();
//Defines:
***** Macro Substitutions *****
portd = LED Port Register
trisd = LED Data Direction Register
16 = LED Pin Mask
1 = LED Active Polarity
***** */

//LED(2): //Macro function declarations
void FCD_LED2_LEDOn();
void FCD_LED2_LEDOff();
//Defines:
***** Macro Substitutions *****
portd = LED Port Register
trisd = LED Data Direction Register
32 = LED Pin Mask
1 = LED Active Polarity
***** */

//LED(3): //Macro function declarations
void FCD_LED3_LEDOn();
void FCD_LED3_LEDOff();

//LED(0): //Macro implementations

void FCD_LED0_LEDOn()
{
    #ifdef MX_10F_TRIS
        tvar = tvar & ~4;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~4;    //Convert pin to output
    
```

```

#endif

        if( 1 )           //Active high polarity
            portd = portd | 4;
        else             //Active low polarity
            portd = portd & ~4;
    }

void FCD_LED0_LEDOff()
{

    #ifdef MX_10F_TRIS
        tvar = tvar & ~4;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~4; //Convert pin to output
    #endif

        if( 1 )           //Active high polarity
            portd = portd & ~4;
        else             //Active low polarity
            portd = portd | 4;
}

//LED(1): //Macro implementations

void FCD_LED1_LEDOn()
{
    #ifdef MX_10F_TRIS
        tvar = tvar & ~8;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~8; //Convert pin to output
    #endif

        if( 1 )           //Active high polarity
            portd = portd | 8;
        else             //Active low polarity
            portd = portd & ~8;
}

void FCD_LED1_LEDOff()
{
}

```

```

#ifdef MX_10F_TRIS
    tvar = tvar & ~8;
    asm("movf(_tvar),w");
    asm("tris 6");
#else
    trisd = trisd & ~8; //Convert pin to output
#endif

    if( 1 ) //Active high polarity
        portd = portd & ~8;
    else //Active low polarity
        portd = portd | 8;
}

//LED(2): //Macro implementations

void FCD_LED2_LEDOn()
{
    #ifdef MX_10F_TRIS
        tvar = tvar & ~16;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~16; //Convert pin to output
    #endif

        if( 1 ) //Active high polarity
            portd = portd | 16;
        else //Active low polarity
            portd = portd & ~16;
}

void FCD_LED2_LEDOff()
{
    #ifdef MX_10F_TRIS
        tvar = tvar & ~16;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~16; //Convert pin to output
    #endif

        if( 1 ) //Active high polarity
            portd = portd & ~16;
}

```

```

        else                      //Active low polarity
            portd = portd | 16;
    }

//LED(3): //Macro implementations

void FCD_LED3_LEDOn()
{
    #ifdef MX_10F_TRIS
        tvar = tvar & ~32;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~32; //Convert pin to output
    #endif

        if( 1 )                  //Active high polarity
            portd = portd | 32;
        else                      //Active low polarity
            portd = portd & ~32;

    }

void FCD_LED3_LEDOff()
{
    #ifdef MX_10F_TRIS
        tvar = tvar & ~32;
        asm("movf(_tvar),w");
        asm("tris 6");
    #else
        trisd = trisd & ~32; //Convert pin to output
    #endif

        if( 1 )                  //Active high polarity
            portd = portd & ~32;
        else                      //Active low polarity
            portd = portd | 32;

    }

//Macro implementations

void FCM_codecheck()
{
    //Input

```

```

//Input: PORT C -> input
trisc = trisc | 0x0f;
FCV_INPUT = portc & 0x0f;

//Switch
//Switch: input?
switch (FCV_INPUT)
{
    case 1:
    {
        //Decision
        //Decision: flag1=1?
        if (FCV_FLAG1==1)
        {
            //Call Component Macro
            //Call Component Macro: LED(0)::LEDOff
            FCD_LED0_LEDOff();

            //Calculation
            //Calculation:
            //  flag1 = 0
            FCV_FLAG1 = 0;

        } else {
            //Call Component Macro
            //Call Component Macro: LED(0)::LEDOn
            FCD_LED0_LEDOn();

            //Calculation
            //Calculation:
            //  flag1 = 1
            FCV_FLAG1 = 1;
        }
    }

    break;
}

case 2:
{
    //Decision
    //Decision: flag2=2?
    if (FCV_FLAG2==2)
    {
        //Call Component Macro
        //Call Component Macro: LED(1)::LEDOff

```

```
    FCD_LED1_LEDOff();

    //Calculation
    //Calculation:
    //  flag2 = 0
    FCV_FLAG2 = 0;

} else {
    //Call Component Macro
    //Call Component Macro: LED(1)::LEDOn
    FCD_LED1_LEDOn();

    //Calculation
    //Calculation:
    //  flag2 = 2
    FCV_FLAG2 = 2;

}

break;
}

case 3:
{
    //Decision
    //Decision: flag3=3?
    if (FCV_FLAG3==3)
    {
        //Call Component Macro
        //Call Component Macro: LED(2)::LEDOFF
        FCD_LED2_LEDOff();

        //Calculation
        //Calculation:
        //  flag3 = 0
        FCV_FLAG3 = 0;

    } else {
        //Call Component Macro
        //Call Component Macro: LED(2)::LEDOn
        FCD_LED2_LEDOn();

        //Calculation
        //Calculation:
```

```

        // flag3 = 3
FCV_FLAG3 = 3;

}

break;

}

case 4:
{
    //Decision
    //Decision: flag4=4?
if (FCV_FLAG4==4)
{
    //Call Component Macro
    //Call Component Macro: LED(3)::LEDOFF
FCD_LED3_LEDOFF();

    //Calculation
    //Calculation:
    // flag4 = 0
FCV_FLAG4 = 0;

} else {
    //Call Component Macro
    //Call Component Macro: LED(3)::LEDON
FCD_LED3_LEDON();

    //Calculation
    //Calculation:
    // flag4 = 4
FCV_FLAG4 = 4;
}

break;
}

default:
{
    //Calculation
    //Calculation:
    // input = 0
FCV_INPUT = 0;
}

```

```

        }

    }

void main()
{
    //Initialisation
    adcon1 = 0x07;

    //Interrupt initialisation code
    option_reg = 0xC0;

    //Connection Point
    //Connection Point: [A]: AFCC_Main_A:;
    //Interrupt
    //Interrupt: Enable RB0INT
    st_bit(option_reg, INTEDG);
    st_bit(intcon, GIE);
    st_bit(intcon, INTE);

    //Goto Connection Point
    //Goto Connection Point: [A]: A
    goto FCC_Main_A;

    mainendloop: goto mainendloop;
}

void MX_INTERRUPT_MACRO(void)
{
    //Handler code for [RB0INT]
#ifndef MX_INTHANDLER_intcon_INTF
#define MX_INTHANDLER_intcon_INTF
    if (ts_bit(intcon, INTF) && ts_bit(intcon, INTE))
    {
        FCM_codecheck();
        cr_bit(intcon, INTF);
    }
    #warning "This interrupt has previously been enabled,
so the macro <codecheck> may never get called."
#endif
}

```



---

# PIC16F87XA

## Data Sheet

28/40/44-Pin Enhanced Flash  
Microcontrollers



**MICROCHIP**

# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input  
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM),  
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin  
PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

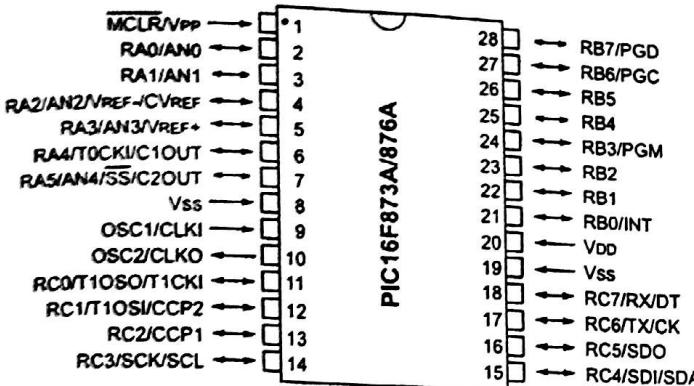
- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I²C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

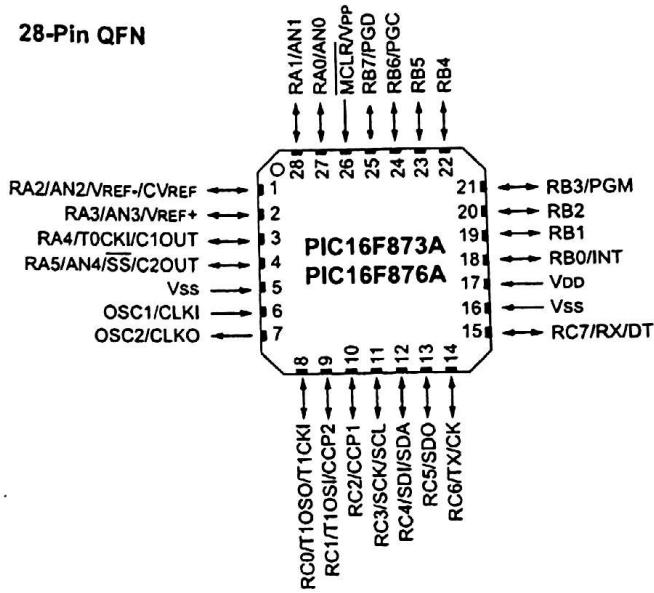
# PIC16F87XA

## Pin Diagrams

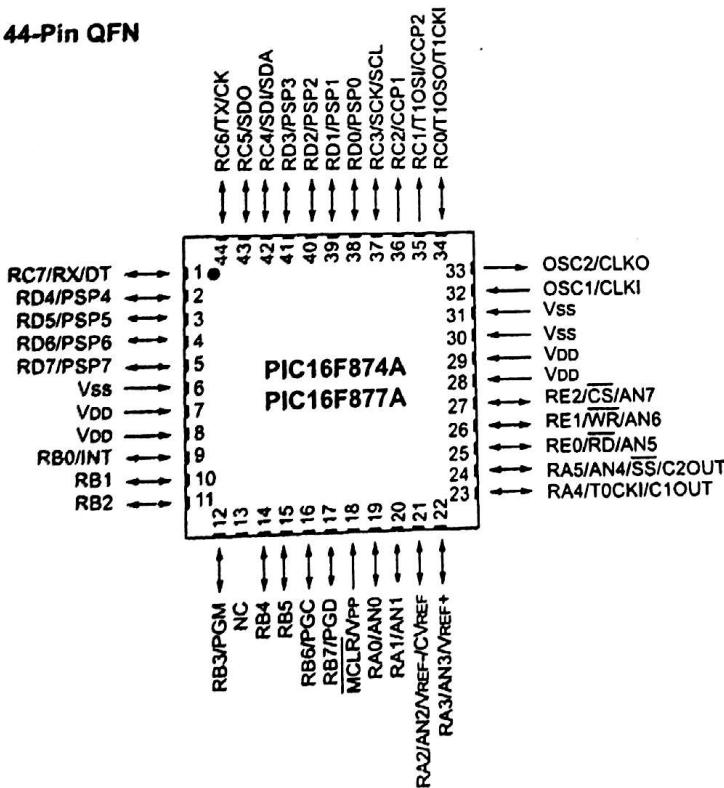
### 28-Pin PDIP, SOIC, SSOP



### 28-Pin QFN



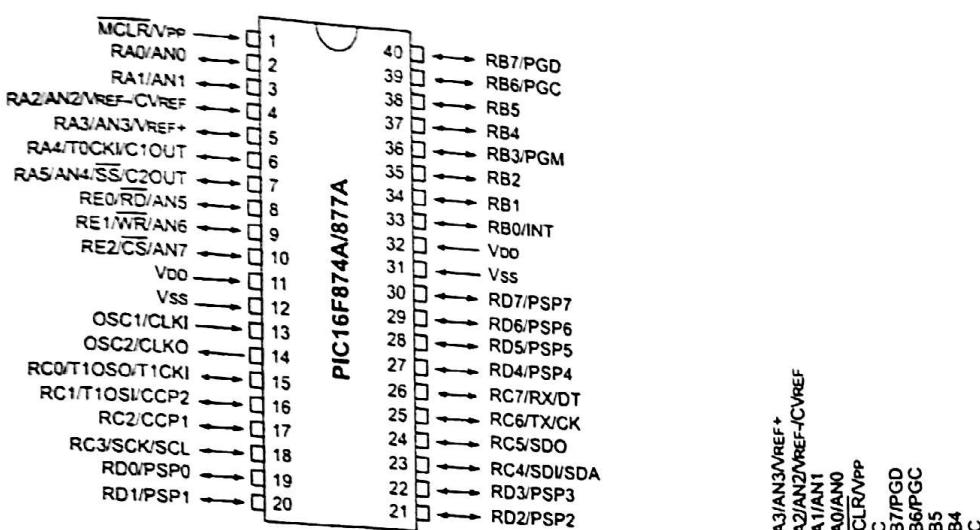
### 44-Pin QFN



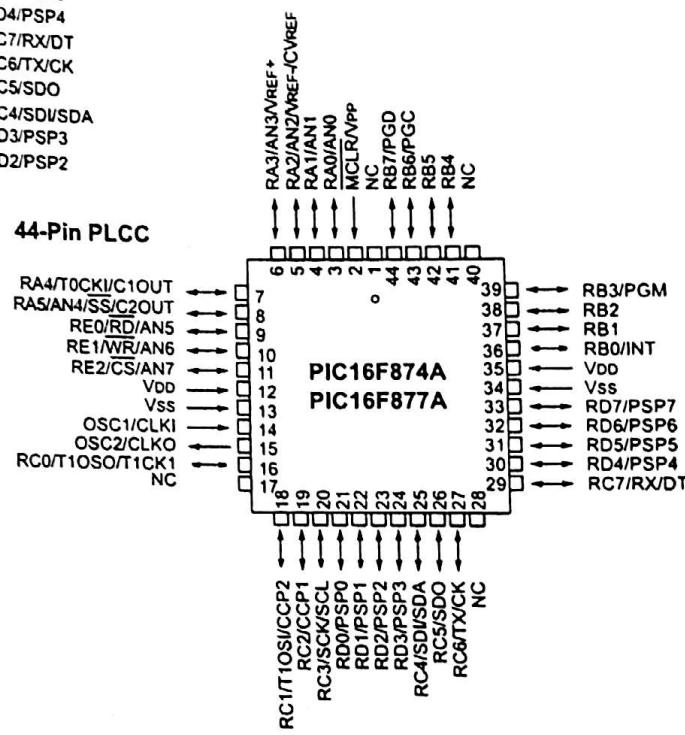
# PIC16F87XA

## Pin Diagrams (Continued)

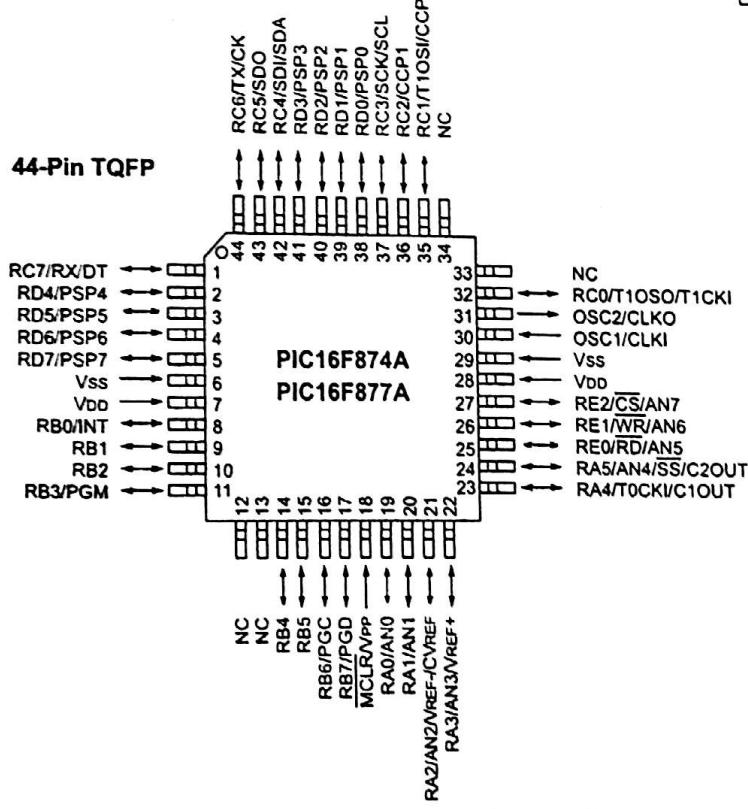
### 40-Pin PDIP



### 44-Pin PLCC



### 44-Pin TQFP



## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

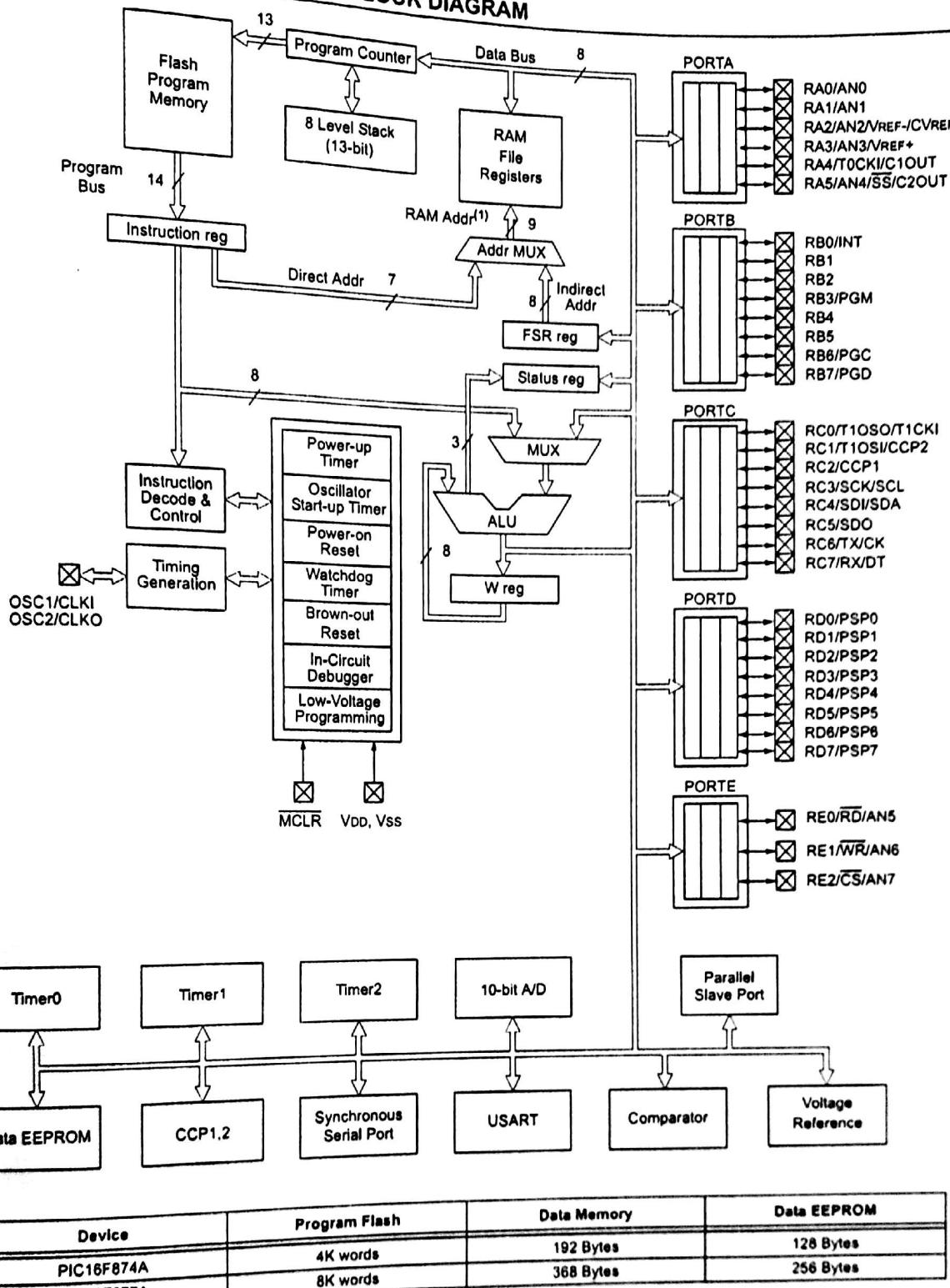
TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz			
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

# PIC16F87XA

FIGURE 1-2:

## PIC16F874A/877A BLOCK DIAGRAM



Note 1: Higher order bits are from the Status register.

# PIC16F87XA

**TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION**

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1	9	6	I	ST/CMOS <sup>(3)</sup>	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI			I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2	10	7	O	—	Oscillator crystal or clock output.
CLKO			O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP MCLR	1	26	I	ST	Master Clear (input) or programming voltage (output).
VPP			P		Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
RA0/ANO RA0	2	27	I/O	TTL	PORTA is a bidirectional I/O port.
ANO			I		Digital I/O.
RA1/AN1 RA1	3	28	I/O	TTL	Analog input 0.
AN1			I		Digital I/O.
RA2/AN2/VREF-/CVREF RA2	4	1	I/O	TTL	Analog input 1.
AN2			I		Digital I/O.
VREF- CVREF			O		A/D reference voltage (Low) input. Comparator VREF output.
RA3/AN3/VREF+ RA3	5	2	I/O	TTL	Digital I/O.
AN3			I		Analog input 3.
VREF+			I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT RA4	6	3	I/O	ST	Digital I/O – Open-drain when configured as output.
T0CKI			I		Timer0 external clock input.
C1OUT			O		Comparator 1 output.
RA5/AN4/SS/C2OUT RA5	7	4	I/O	TTL	Digital I/O.
AN4			I		Analog input 4.
SS			I		SPI slave select input.
C2OUT			O		Comparator 2 output.

**Legend:** I = input      O = output      I/O = input/output      P = power

— = Not used      TTL = TTL input      ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

**TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	21	18	I/O I	TTL/ST <sup>(1)</sup>	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB1	22	19	I/O	TTL	Digital I/O. External interrupt.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	24	21	I/O I	TTL	Digital I/O.
RB4	25	22	I/O	TTL	Digital I/O. Low-voltage (single-supply) ICSP programming enable pin.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	27	24	I/O I	TTL/ST <sup>(2)</sup>	Digital I/O. Digital I/O.
RB7/PGD RB7 PGD	28	25	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-circuit debugger and ICSP programming clock.  Digital I/O. In-circuit debugger and ICSP programming data.
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST	PORTC is a bidirectional I/O port.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	12	9	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Timer1 external clock input.
RC2/CCP1 RC2 CCP1	13	10	I/O I/O	ST	Digital I/O. Capture2 input, Compare2 output, PWM2 output.
RC3/SCK/SCL RC3 SCK SCL	14	11	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I <sup>2</sup> C mode.
RC4/SDI/SDA RC4 SDI SDA	15	12	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	16	13	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	17	14	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	18	15	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.
V <sub>SS</sub>	8, 19	5, 6	P	—	Ground reference for logic and I/O pins.
V <sub>DD</sub>	20	17	P	—	Positive supply for logic and I/O pins.

**Legend:** I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# PIC16F87XA

TABLE 1-3:

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1	13	14	30	32	I	ST/CMOS <sup>(4)</sup>	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI					I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2	14	15	31	33	O	—	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP MCLR	1	2	18	18	I	ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
VPP					P		
RA0/AN0 RA0 AN0	2	3	19	19	I/O	TTL	PORTA is a bidirectional I/O port.
RA1/AN1 RA1 AN1	3	4	20	20	I/O	TTL	Digital I/O. Analog input 0.
RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF	4	5	21	21	I/O	TTL	Digital I/O. Analog input 1.
RA3/AN3/VREF+ RA3 AN3 VREF+	5	6	22	22	I/O	TTL	Digital I/O. Analog input 2. A/D reference voltage (Low) input. Comparator VREF output.
RA4/T0CKI/C1OUT RA4	6	7	23	23	I/O	ST	Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output.
T0CKI C1OUT					I		
RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	7	8	24	24	I/O	TTL	Digital I/O. Analog input 4. SPI slave select input. Comparator 2 output.
					O		

Legend: I = input      O = output      I/O = input/output      P = power

— = Not used      TTL = TTL input      ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

Note 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	9	I/O I	TTL/ST <sup>(1)</sup>	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	External interrupt.
RB3/PGM RB3 PGM	36	39	11	12	I/O	TTL	Digital I/O.
RB4	37	41	14	14	I/O I	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Low-voltage ICSP programming enable pin.
RB6/PGC RB6 PGC	39	43	16	16	I/O	TTL/ST <sup>(2)</sup>	Digital I/O.
RB7/PGD RB7 PGD	40	44	17	17	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-circuit debugger and ICSP programming clock.
<b>Legend:</b> I = input      O = output      I/O = input/output      P = power — = Not used      TTL = TTL input      ST = Schmitt Trigger input							

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.

**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.

**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	34	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	36	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK  SCL	18	20	37	37	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I <sup>2</sup> C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	24	26	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	44	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	1	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.

Legend: I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-3:

PIC16F874A/877A

Pin Name	PINOUT DESCRIPTION (CONTINUED)						
	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	38			PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.
RD1/PSP1 RD1 PSP1	20	22	39	39	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	23	40	40	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	24	41	41	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	30	2	2	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	31	3	3	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	32	4	4	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	33	5	5	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
					I/O		Digital I/O. Parallel Slave Port data.
RE0/RD/AN5 RE0 RD AN5	8	9	25	25		ST/TTL <sup>(3)</sup>	PORTE is a bidirectional I/O port.
RE1/WR/AN6 RE1 WR AN6	9	10	26	26	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE2/CS/AN7 RE2 CS AN7	10	11	27	27	I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Write control for Parallel Slave Port. Analog input 6.
VSS	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12, 13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

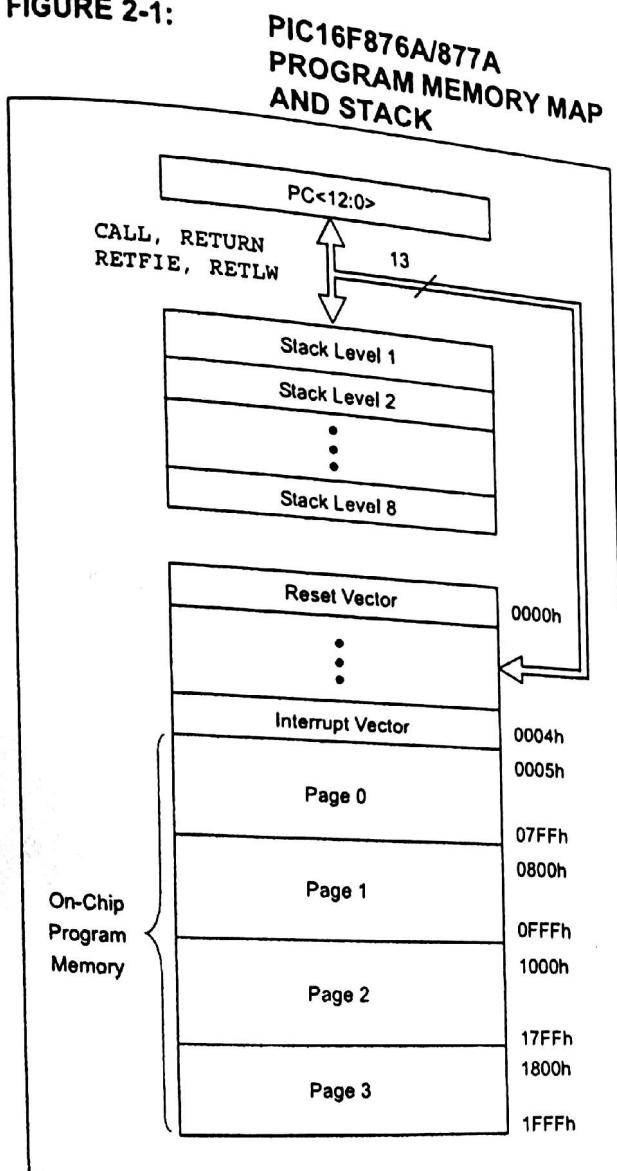
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

## 2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87XA devices. The program memory and data memory have separate buses so that concurrent access can occur and is detailed in this section. The EEPROM data memory block is detailed in Section 3.0 "Data EEPROM and Flash Program Memory". Additional information on device memory may be found in the PICmicro® Mid-Range MCU Family Reference Manual (DS33023).

**FIGURE 2-1:**

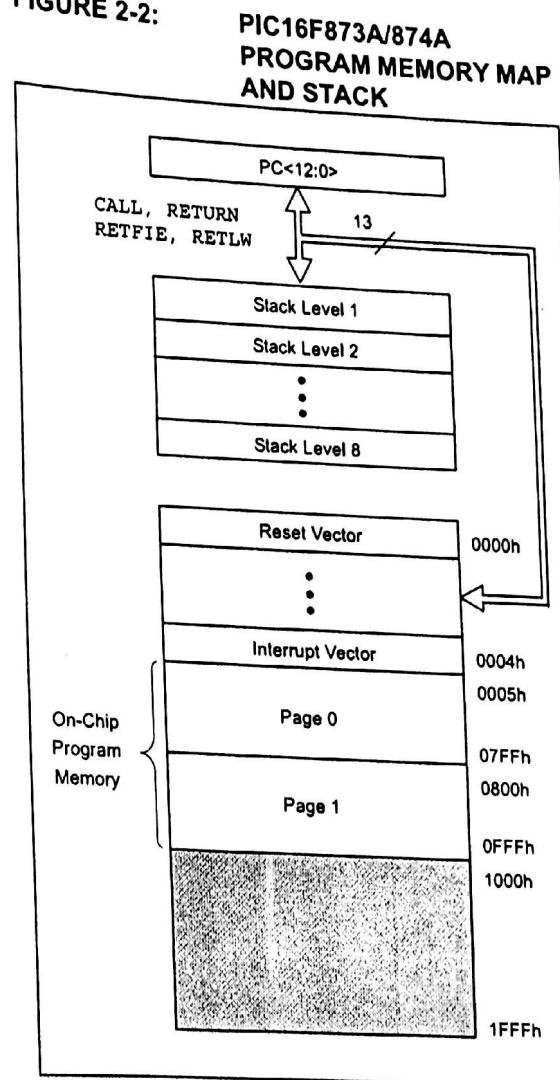


## 2.1 Program Memory Organization

The PIC16F87XA devices have a 13-bit program counter capable of addressing an 8K word x 14 bit program memory space. The PIC16F876A/877A devices have 8K words x 14 bits of Flash program memory, while PIC16F873A/874A devices have 4K words x 14 bits. Accessing a location above the physically implemented address will cause a wraparound.

The Reset vector is at 0000h and the interrupt vector is at 0004h.

**FIGURE 2-2:**



# PIC16F87XA

## 2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (Status<6>) and RP0 (Status<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

**Note:** The EEPROM data memory description can be found in Section 3.0 "Data EEPROM and Flash Program Memory" of this data sheet.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly, through the File Select Register (FSR).

PIC16F87XA

**FIGURE 2-3:** PIC16F876A/877A REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr.(*)	Indirect addr.(*)	Indirect addr.(*)	Indirect addr.(*)
TMR0	OPTION_REG	TMR0	OPTION_REG
PCL	PCL	PCL	PCL
STATUS	STATUS	STATUS	STATUS
FSR	FSR	FSR	FSR
PORTA	TRISA	PORTB	TRISB
PORTB	TRISB	PORTB	TRISB
PORTC	TRISC	PORTB	TRISB
PORTD <sup>(1)</sup>	TRISD <sup>(1)</sup>	PORTB	TRISB
PORTE <sup>(1)</sup>	TRISE <sup>(1)</sup>	PORTB	TRISB
PCLATH	PCLATH	PORTB	TRISB
INTCON	INTCON	PORTB	TRISB
PIR1	PIE1	PORTB	TRISB
PIR2	PIE2	PORTB	TRISB
TMR1L	PCON	PORTB	TRISB
TMR1H		PORTB	TRISB
T1CON		PORTB	TRISB
TMR2		PORTB	TRISB
T2CON	SSPCON2	General Purpose Register	General Purpose Register
SSPBUF	PR2	16 Bytes	16 Bytes
SSPCON	SSPADD	General Purpose Register	General Purpose Register
CCPR1L	SSPSTAT	16 Bytes	16 Bytes
CCPR1H		General Purpose Register	General Purpose Register
CCP1CON		16 Bytes	16 Bytes
RCSTA	TXSTA	General Purpose Register	General Purpose Register
TXREG	SPBRG	16 Bytes	16 Bytes
RCREG		General Purpose Register	General Purpose Register
CCPR2L		16 Bytes	16 Bytes
CCPR2H		General Purpose Register	General Purpose Register
CCP2CON	CMCON	16 Bytes	16 Bytes
ADRESH	CVRCON	General Purpose Register	General Purpose Register
ADCON0	ADRESL	16 Bytes	16 Bytes
	ADCON1	General Purpose Register	General Purpose Register
General Purpose Register		80 Bytes	80 Bytes
96 Bytes		General Purpose Register	General Purpose Register
		80 Bytes	80 Bytes
		accesses 70h-7Fh	accesses 70h-7Fh
		EFh	16Fh
		F0h	170h
		FFh	17Fh
Bank 0	Bank 1	Bank 2	Bank 3

- Unimplemented data memory locations, read as '0'.
- Not a physical register.

**Note 1:** These registers are not implemented on the PIC16F876A.  
**2:** These registers are reserved; maintain these registers clear.

# PIC16F87XA

FIGURE 2-4: PIC16F873A/874A REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr. <sup>(1)</sup>	Indirect addr. <sup>(1)</sup>	Indirect addr. <sup>(1)</sup>	Indirect addr. <sup>(1)</sup>
TMR0	OPTION_REG	TMR0	OPTION_REG
PCL	PCL	PCL	PCL
STATUS	STATUS	STATUS	STATUS
FSR	FSR	FSR	FSR
PORTA	TRISA	PORTB	TRISB
PORTB	TRISB	PORTC	TRISC
PORTC	TRISC	PORTD <sup>(1)</sup>	TRISD <sup>(1)</sup>
PORTD <sup>(1)</sup>	TRISD <sup>(1)</sup>	PORTE <sup>(1)</sup>	TRISE <sup>(1)</sup>
PORTE <sup>(1)</sup>	TRISE <sup>(1)</sup>	PCLATH	PCLATH
PCLATH	PCLATH	INTCON	INTCON
INTCON	INTCON	PIE1	EEDATA
PIR1	PIE1	PIE2	EEADR
PIR2	PIE2	PCON	EEDATH
TMR1L	PCON		EEADRH
TMR1H			
T1CON			
TMR2	SSPCON2		
T2CON	PR2		
SSPBUF	SSPADD		
SSPCON	SSPSTAT		
CCPR1L			
CCPR1H			
CCP1CON			
RCSTA			
TXREG	TXSTA		
RCREG	SPBRG		
CCPR2L			
CCPR2H			
CCP2CON			
ADRESH			
ADCON0			
General Purpose Register 96 Bytes	General Purpose Register 96 Bytes	accesses 20h-7Fh	accesses A0h - FFh
Bank 0	Bank 1	Bank 2	Bank 3
	FFh	A0h	
		120h	1A0h
		16Fh	1EFh
		170h	1F0h
		17Fh	1FFh

■ Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.

Note 1: These registers are not implemented on the PIC16F873A.  
 2: These registers are reserved; maintain these registers clear.

## 2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

**TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
<b>Bank 0</b>											
00h <sup>(3)</sup>	INDF										
01h	TMR0										
02h <sup>(3)</sup>	PCL										
03h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	22, 150
04h <sup>(3)</sup>	FSR									0000 0000	30, 150
05h	PORTA	—	—	—	—	—	—	—	—	xxxx xxxx	31, 150
06h	PORTB									—0x 0000	43, 150
07h	PORTC									xxxx xxxx	45, 150
08h <sup>(4)</sup>	PORTD									xxxx xxxx	47, 150
09h <sup>(4)</sup>	PORTE	—	—	—	—	—	—	—	—	xxxx xxxx	48, 150
0Ah <sup>(1,3)</sup>	PCLATH	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	49, 150
0Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIIF	---0 0000	30, 150
0Ch	PIR1	PSPIF <sup>(3)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 000x	24, 150
0Dh	PIR2	—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF	-0-0 0--0	28, 150
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	—	TMR1CS	TMR1ON	--00 0000	57, 150
11h	TMR2	Timer2 Module Register								0000 0000	62, 150
12h	T2CON	—	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	-000 0000	61, 150
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	79, 150
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	82, 82, 150
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	63, 150
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	63, 150
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	64, 150
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	112, 150
19h	TXREG	USART Transmit Data Register								0000 0000	118, 150
1Ah	RCREG	USART Receive Data Register								0000 0000	118, 150
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	63, 150
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	63, 150
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	64, 150
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	133, 150
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	127, 150

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.

Shaded locations are unimplemented, read as '0'.

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.

**3:** These registers can be addressed from any bank.

**4:** PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.

**5:** Bit 4 of EEADRH implemented only on the PIC16F876A/877A devices.

# PIC16F87XA

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:				
<b>Bank 1</b>															
80h <sup>(3)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)													
81h	OPTION_REG	RBU	INTE0G	TOCS	TOSE	PSA	PS2	PS1	PS0	0000 0000	31, 150				
82h <sup>(3)</sup>	PCL	Program Counter (PC) Least Significant Byte													
83h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx	23, 150				
84h <sup>(3)</sup>	FSR	Indirect Data Memory Address Pointer													
85h	TRISA	—	—	PORTA Data Direction Register											
86h	TRISB	PORTB Data Direction Register										xxxx xxxx	31, 150		
87h	TRISC	PORTC Data Direction Register										--11 1111	43, 150		
88h <sup>(4)</sup>	TRISD	PORTD Data Direction Register										1111 1111	45, 150		
89h <sup>(4)</sup>	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits					1111 1111	48, 151		
8Ah <sup>(1,3)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter							0000 -111	50, 151		
8Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TMROIE	INTE	RBIE	TMROIF	INTF	RBIF	0000 000x		30, 150			
8Ch	PIE1	PSPIE <sup>(2)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000		24, 150			
8Dh	PIE2	—	CMIE	—	EEIE	BCLIE	—	—	CCP2IE	0000 0000		25, 151			
8Eh	PCON	—	—	—	—	—	—	—	POR	BOR	---- --qq	27, 151			
8Fh	—	Unimplemented										—	—		
90h	—	Unimplemented										—	—		
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000		83, 151			
92h	PR2	Timer2 Period Register										1111 1111	62, 151		
93h	SSPADD	Synchronous Serial Port (I <sup>2</sup> C mode) Address Register										0000 0000	79, 151		
94h	SSPSTAT	SMP	CKE	D <sup>—</sup> A	P	S	R <sup>—</sup> W	UA	BF	0000 0000		79, 151			
95h	—	Unimplemented										—	—		
96h	—	Unimplemented										—	—		
97h	—	Unimplemented										—	—		
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010		111, 151			
99h	SPBRG	Baud Rate Generator Register										0000 0000	113, 151		
9Ah	—	Unimplemented										—	—		
9Bh	—	Unimplemented										—	—		
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111		135, 151			
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000 - 0000		141, 151			
9Eh	ADRESL	A/D Result Register Low Byte										xxxx xxxx	133, 151		
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000		128, 151			

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved.  
Shaded locations are unimplemented, read as '0'.

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8>, whose contents are transferred to the upper byte of the program counter.
- 2:** Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.
- 3:** These registers can be addressed from any bank.
- 4:** PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.
- 5:** Bit 4 of EEADRH implemented only on the PIC16F876A/877A devices.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
<b>Bank 2</b>											
100h <sup>(3)</sup>	INDF										
101h	TMR0										
102h <sup>(3)</sup>	PCL										
103h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0000 0000	31, 150
104h <sup>(3)</sup>	FSR									xxxx xxxx	55, 150
105h	—									0000 0000	30, 150
106h	PORTB									0001 1xxx	22, 150
107h	—									xxxx xxxx	31, 150
108h	—									xxxx xxxx	45, 150
109h	—									—	—
10Ah <sup>(1,3)</sup>	PCLATH	—	—	—						—	—
10Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TMROIE						---0 0000	30, 150
10Ch	EEDATA				INTE	RBIE	TMROIF	INTF	RBIF	0000 000x	24, 150
10Dh	EEADDR									xxxx xxxx	39, 151
10Eh	EEDATH	—	—							xxxx xxxx	39, 151
10Fh	EEADDRH	—	—	—	(5)					--xx xxxx	39, 151
<b>Bank 3</b>											
180h <sup>(3)</sup>	INDF										
181h	OPTION_REG	RBPU	INTEDG	T0CS	TOSE	PSA	PS2	PS1	PS0	0000 0000	31, 150
182h <sup>(3)</sup>	PCL									1111 1111	23, 150
183h <sup>(3)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0000 0000	30, 150
184h <sup>(3)</sup>	FSR									0001 1xxx	22, 150
185h	—									xxxx xxxx	31, 150
186h	TRISB									1111 1111	45, 150
187h	—									—	—
188h	—									—	—
189h	—									—	—
18Ah <sup>(1,3)</sup>	PCLATH	—	—	—						---0 0000	30, 150
18Bh <sup>(3)</sup>	INTCON	GIE	PEIE	TMROIE	INTE	RBIE	TMROIF	INTF	RBIF	0000 000x	24, 150
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	34, 151
18Dh	EECON2									---- ----	39, 151
18Eh	—									0000 0000	—
18Fh	—									0000 0000	—

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0'. z = reserved.

Shaded locations are unimplemented, read as '0'.

**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.

**2:** Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.

**3:** These registers can be addressed from any bank.

**4:** PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.

**5:** Bit 4 of EEADDRH implemented only on the PIC16F876A/877A devices.

## Functional Description

M-8870 operating functions (see block diagram on page 1) include a bandsplit filter that separates the high and low tones of the received pair, and a digital decoder that verifies both the frequency and duration of the received tones before passing the resulting 4-bit code to the output bus.

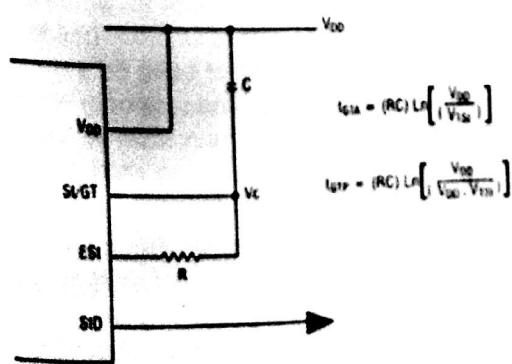
### Filter

The low and high group tones are separated by applying the dual-tone signal to the inputs of two 6th order switched capacitor bandpass filters with bandwidths that correspond to the bands enclosing the low and high group tones. The filter also incorporates notches at 350 and 440 Hz, providing excellent dial tone rejection. Each filter output is followed by a single-order switched capacitor section that smooths the signals prior to limiting. Signal limiting is performed by high-gain comparators provided with hysteresis to prevent detection of unwanted low-level signals and noise. The comparator outputs provide full-rail logic swings at the frequencies of the incoming tones.

### Decoder

The M-8870 decoder uses a digital counting technique to determine the frequencies of the limited tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm is used to protect against tone simulation by extraneous signals (such as voice) while tolerating small frequency variations. The algorithm ensures an optimum combination of immunity to talkoff and tolerance to interfering signals (third tones) and noise. When the detector recognizes the simultaneous presence of two valid tones (known as signal condition), it raises the Early Steering flag (EST). Any subsequent loss of signal condition will cause EST to fall.

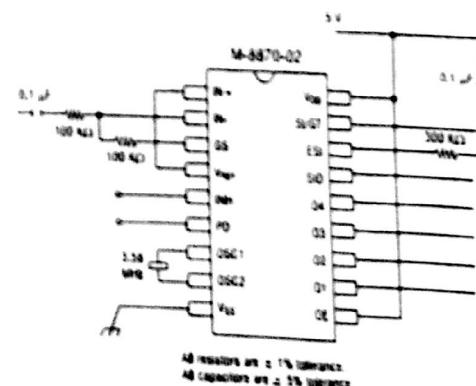
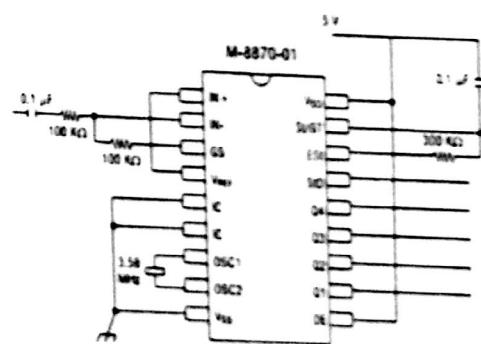
### Basic Steering Circuit



### Steering Circuit

Before a decoded tone pair is registered, the receiver checks for a valid signal duration (referred to as character-recognition-condition). This check is performed by an external RC time constant driven by EST. A logic high on EST causes VC (see block diagram on page 1) to rise as the capacitor discharges. Provided that signal condition is maintained (EST remains high) for the validation period ( $t_{GTF}$ ),  $V_C$  reaches the threshold ( $V_{TSI}$ ) of the steering logic to register the tone pair, thus latching its corresponding 4-bit code (see DC Characteristics on page 2) into the output latch. At this point, the GT output is activated and drives  $V_C$  to  $V_{DD}$ . GT continues to drive high as long as EST remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (S/D) goes high, signaling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three-state control input (OE) to a logic high. The steering circuit works in reverse to validate the interdigit pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropouts) too short to be considered a valid pause. This capability, together with the ability to select the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

### Single-Ended Input Configuration



## Pin Functions

Pin #	Name	Description
1	IN+	Non-inverting input
2	IN-	Inverting input
3	GS	Gain select. Gives access to output of front-end amplifier for connection of feedback resistor.
4	V <sub>REF</sub>	Reference voltage output (nominally V <sub>DD</sub> /2). May be used to bias the inputs at mid-rail.
5	INH*	Inhibits detection of tones representing keys A, B, C, and D.
6	PD*	Power down. Logic high powers down the device and inhibits the oscillator. Internal pulldown.
7	OSC1	Clock input
8	OSC2	Clock output 3.579545 MHz crystal connected between these pins completes the internal oscillator.
9	V <sub>SS</sub>	Negative power supply (normally connected to 0 V).
10	OE	Tri-stable output enable (input). Logic high enables the outputs Q1 - Q4. Internal pullup.
11-14	Q1, Q2, Q3, Q4	Tri-stable data outputs. When enabled by OE, provides the code corresponding to the last valid tone pair received (see Tone Decoding table on page 5).
15	StD	Delayed steering output. Presents a logic high when a received tone pair has been registered and the output latch is updated. Returns to logic low when the voltage on St/GT falls below VT <sub>St</sub> .
16	EST	Early steering output. Presents a logic high immediately when the digital algorithm detects a recognizable tone pair (signal condition). Any momentary loss of signal condition will cause EST to return to a logic low.
17	St/GT	Steering input/guard time output (bidirectional). A voltage greater than VT <sub>St</sub> detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than VT <sub>St</sub> frees the device to accept a new tone pair. The GT output acts to reset the external steering time constant, and its state is a function of EST and the voltage on St. (See Common Crystal Connection on page 5).
18	V <sub>DD</sub>	Positive power supply. (Normally connected to +5V.)

\*-02 only. Connect to V<sub>SS</sub> for -01 version

## Guard Time Adjustment

Where independent selection of signal duration and interdigit pause are not required, the simple steering circuit of Basic Steering Circuit is applicable. Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$

$$t_{GTP} @ 0.67 RC$$

The value of t<sub>DP</sub> is a parameter of the device and t<sub>REC</sub> is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μF is recommended for most applications, leaving R to be selected by the designer. For example, a suitable value of R for a t<sub>REC</sub> of 40 ms would be 300 kΩ. A typical circuit using this steering configuration is shown in the Single-Ended Input Configuration on page 4. The timing requirements for most telecommunication applications are satisfied with this circuit. Different steering arrangements may be used to select independently the guard times for tone-present (t<sub>GTP</sub>) and tone-absent (t<sub>GTA</sub>). This may be necessary to meet system specifications that place both accept and reject limits on both tone duration and interdigit pause.

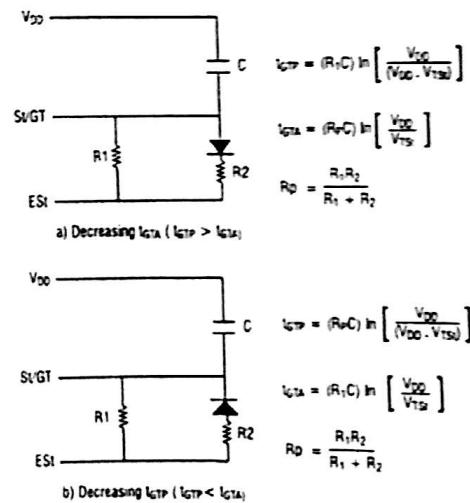
Guard time adjustment also allows the designer to tailor system parameters such as talkoff and noise immunity. Increasing t<sub>REC</sub> improves talkoff performance, since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be

registered. On the other hand, a relatively short t<sub>REC</sub> with a long t<sub>DP</sub> would be appropriate for extremely noisy environments where fast acquisition time and immunity to dropouts would be required. Design information for guard time adjustment is shown in the Guard Time Adjustment below.

## Power-down and Inhibit Mode (-02 only)

A logic high applied to pin 6 (PD) will place the device into standby mode to minimize power consumption. It

Figure 5 Guard Time Adjustment



stops the oscillator and the functioning of the filters. On the M-8870-01 models, this pin is tied to ground (logic low).

Inhibit mode is enabled by a logic high input to pin 5 (INH). It inhibits the detection of 1633 Hz. The output code will remain the same as the previous detected code (see Pin functions table on page 4). On the M-8870-01 models, this pin is tied to ground (logic low).

### Input Configuration

The input arrangement of the M-8870 provides a differential input operational amplifier as well as a bias source ( $V_{REF}$ ) to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for gain adjustment.

In a single-ended configuration, the input pins are connected as shown in the Single - Ended Input Configuration on page 3 with the op-amp connected for unity gain and  $V_{REF}$  biasing the input at  $1/2V_{DD}$ . The Differential Input Configuration below permits gain adjustment with the feedback resistor  $R_S$ .

### DTMF Clock Circuit

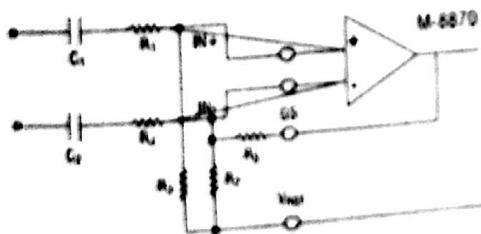
The internal clock circuit is completed with the addition of a standard 3.579545 MHz television color burst crystal. The crystal can be connected to a single M-8870 as shown in the Single - Ended Input Configuration on page 3, or to a series of M-8870s. As illustrated in the Common Crystal Connection below, a single crystal can be used to connect a series of M-8870s by coupling the oscillator output of each M-8870 through a 30 pF capacitor to the oscillator input of the next M-8870.

### Tone Decoding

697	1209	1	H	0	0	0	0	1	0
697	1336	2	H	0	0	1	1	1	1
697	1477	3	H	0	0	1	0	0	0
770	1209	4	H	0	1	0	1	1	1
770	1336	5	H	0	1	0	0	0	0
770	1477	6	H	0	1	1	1	1	1
852	1209	7	H	0	1	0	0	0	0
852	1336	8	H	1	0	0	0	1	1
852	1477	9	H	1	0	0	0	0	0
941	1336	0	H	1	0	1	1	1	1
941	1209	S	H	1	0	1	0	0	0
941	1477	#	H	1	1	1	0	0	1
697	1633	A	H	1	1	1	1	0	0
770	1633	B	H	1	1	1	1	1	1
852	1633	C	H	1	1	1	0	0	0
941	1633	D	H	0	0	0	0	0	0
ANY	ANY	ANY	L	Z	Z	Z	Z	Z	Z

L = logic low, H = logic high, Z = high impedance

### Differential Input Configuration



#### Differential Input Amplifier

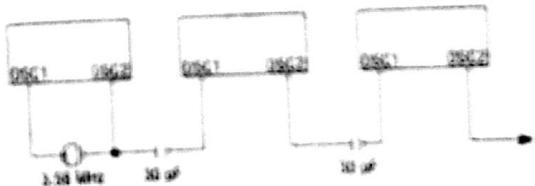
$$\begin{aligned} C_1 &= C_2 = 10 \text{ nF} \\ R_1 &= R_2 = R_3 = 100 \text{ k}\Omega \\ R_4 &= 60 \text{ k}\Omega, R_S = 37.5 \text{ k}\Omega \\ R_S &= \frac{R_4 R_S}{R_4 + R_S} \\ \text{Voltage Gain (Av diff)} &= \frac{R_4}{R_1 + R_2} \end{aligned}$$

All resistors are 1% tolerance  
All capacitors are 1% tolerance

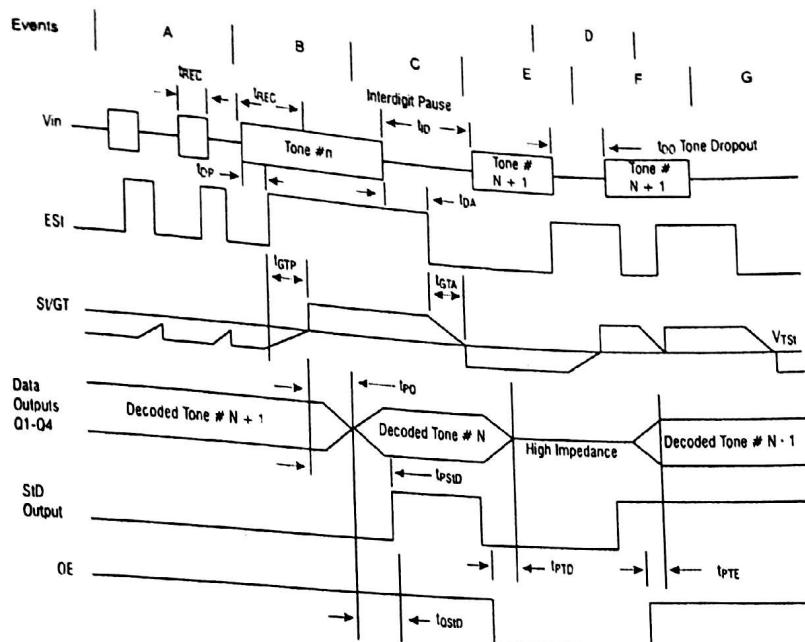
Input Impedance

$$(R_1 R_2) + R_S \parallel \left( \frac{1}{C_1} \right)^2$$

### Common Crystal Connection



## ARE Timing Diagram



## Explanation of Events

- (A) Tone bursts detected, tone duration invalid, outputs not updated.
- (B) Tone #n detected, tone duration valid, tone decoded and latched in outputs.
- (C) End of tone #n detected, tone absent duration valid, outputs remain latched until next valid tone.
- (D) Outputs switched to high impedance state.
- (E) Tone #n + 1 detected, tone duration valid, tone decoded and latched in outputs (currently high impedance).
- (F) Acceptable dropout of tone #n + 1, tone absent duration invalid, outputs remain latched.
- (G) End of tone #n + 1 detected, tone absent duration valid, outputs remain latched until next valid tone.

## Explanation of Symbols

VIN	DTMF composite input signal.
ESI	Early steering output. Indicates detection of valid tone frequencies.
SVGT	Steering input/guard time output. Drives external RC timing circuit.
Q1 - Q4	4-bit decoded tone output.
SID	Delayed steering output. Indicates that valid frequencies have been present/absent for the required guardtime, thus constituting a valid signal.
OE	Output enable (input). A low level shifts Q1 - Q4 to its high impedance state.
$t_{REC}$	Maximum DTMF signal duration not detected as valid.
$t_{REC}$	Minimum DTMF signal duration required for valid recognition.
$t_{ID}$	Minimum time between valid DTMF signals.
$t_{DO}$	Maximum allowable dropout during valid DTMF signal.
$t_{DP}$	Time to detect the presence of valid DTMF signals.
$t_{DA}$	Time to detect the absence of valid DTMF signals.
$T_{GTP}$	Guard time, tone present.
$T_{GTA}$	Guard time, tone absent.

## AC Characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Note
Valid input signal levels (each tone of composite signal)	-	-29	-	+1	dBM	1,2,3,4,5,8
Positive twist accept	-	27.5	-	869	mVRMS	
Negative twist accept	-	-	-	10	dB	2,3,4,8
Frequency deviation accept limit	-	-	-	10	dB	
Frequency deviation reject limit	-	-	-	$\pm 1.5\% + 2 \text{ Hz}$	Nom.	2,3,5,8,10
Third tone tolerance	-	$\pm 3.5\%$	-	-	Nom.	2,3,5
Noise tolerance	-	-25	-16	-	dB	2,3,4,5,8,9,13,14
Dial tone tolerance	-	-	-12	-	dB	2,3,4,5,6,8,9
Tone present detection time	$t_{DP}$	5	8	14	ms	2,3,4,5,7,8,9
Tone absent detection time	$t_{DA}$	0.5	3	8.5	ms	See Timing Diagram on page 7 User adjustable (see Basic Steering Circuit and Guard Time Adjustment on pages 3 and 4.)
Minimum tone duration accept	$t_{REC}$	-	-	40	ms	
Maximum tone duration reject	$t_{REC}$	20	-	-	ms	
Minimum interdigit pause accept	$t_{ID}$	-	-	40	ms	
Maximum interdigit pause reject	$t_{DO}$	20	-	-	ms	
Propagation delay (St to Q)	$t_{PQ}$	-	6	11	$\mu\text{s}$	
Propagation delay (St to StD)	$t_{PSID}$	-	9	16	$\mu\text{s}$	
Output data setup (Q to StD)	$t_{QSID}$	-	4.0	-	$\mu\text{s}$	
Propagation delay (OE to Q), enable	$t_{PTF}$	-	50	60	ns	$R_L = 10 \text{ k}\Omega, C_L = 50 \text{ pF}$
Propagation delay (OE to Q), disable	$t_{PTD}$	-	300	-	ns	
Crystal clock frequency	$f_{CLK}$	3.5759	3.5795	3.5831	MHz	-
Clock output (OSC2), capacitive load	$C_{LO}$	-	-	30	pF	-

All voltages referenced to  $V_{SS}$  unless otherwise noted. For typical values  $V_{DD} = 5.0 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $TA = 25^\circ\text{C}$ ,  $f_{CLK} = 3.579545 \text{ MHz}$ .

\*Typical figures are at  $25^\circ\text{C}$  and are for design aid only; not guaranteed and not subject to production testing.

## Notes:

1. dBm = decibels above or below a reference power of 1 mW into a  $600\Omega$  load.
2. Digit sequence consists of all 16 DTMF tones.
3. Tone duration = 40 ms. Tone pause = 40 ms.
4. Nominal DTMF frequencies are used, measured at GS.
5. Both tones in the composite signal have an equal amplitude.
6. Bandwidth limited (0 to 3 kHz) Gaussian noise.
7. The precise dial tone frequencies are  $(350 \text{ and } 440 \text{ Hz}) \pm 2\%$ .
8. For an error rate of better than 1 in 10,000.
9. Referenced to lowest level frequency component in DTMF signal.
10. Minimum signal acceptance level is measured with specified maximum frequency deviation.
11. Input pins defined as IN+, IN-, and OE.
12. External voltage source used to bias  $V_{REF}$ .
13. This parameter also applies to a third tone injected onto the power supply.
14. Referenced to Single - Ended Input Configuration on page 3. Input DTMF tone level at -28 dBm.

August 2012

## LM78XX/LM78XXA

### 3-Terminal 1A Positive Voltage Regulator

#### Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

#### General Description

The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

#### Ordering Information

Product Number	Output Voltage Tolerance	Package	Operating Temperature
LM7805CT	±4%	TO-220 (Single Gauge)	-40°C to +125°C
LM7806CT			
LM7808CT			
LM7809CT			
LM7810CT			
LM7812CT			
LM7815CT			
LM7818CT			
LM7824CT			
LM7805ACT	±2%		0°C to +125°C
LM7806ACT			
LM7808ACT			
LM7809ACT			
LM7810ACT			
LM7812ACT			
LM7815ACT			
LM7818ACT			
LM7824ACT			

## Block Diagram

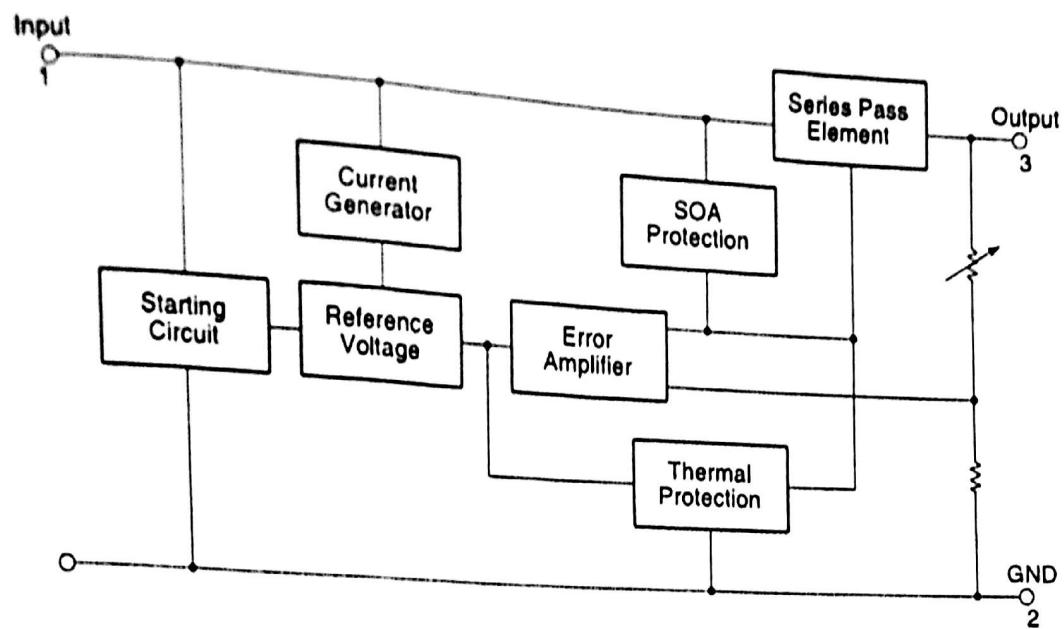


Figure 1.

## Pin Assignment

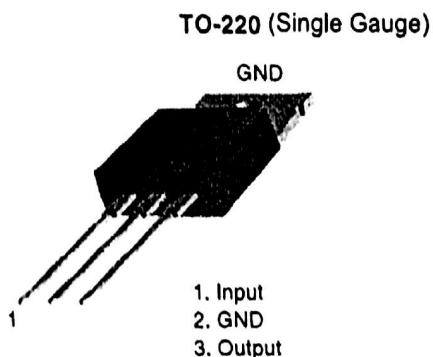


Figure 2.

## Absolute Maximum Ratings

Absolute maximum ratings are those values beyond which damage to the device may occur. The datasheet specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. Fairchild does not recommend operation outside datasheet specifications.

Symbol	Parameter		Value	Unit
$V_I$	Input Voltage	$V_O = 5V \text{ to } 18V$	35	V
		$V_O = 24V$	40	V
$R_{\theta JC}$	Thermal Resistance Junction-Cases (TO-220)		5	°C/W
$R_{\theta JA}$	Thermal Resistance Junction-Air (TO-220)		65	°C/W
$T_{OPR}$	Operating Temperature Range	LM78xx	-40 to +125	°C
		LM78xxA	0 to +125	
$T_{STG}$	Storage Temperature Range		-65 to +150	°C

**Electrical Characteristics (LM7805)**

Refer to the test circuits.  $-40^{\circ}\text{C} < T_J < 125^{\circ}\text{C}$ ,  $I_Q = 500\text{mA}$ ,  $V_I = 10\text{V}$ ,  $C_1 = 0.1\mu\text{F}$ , unless otherwise specified.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$V_O$	Output Voltage	$T_J = +25^{\circ}\text{C}$ $3\text{mA} \leq I_Q \leq 1\text{A}$ , $P_Q \leq 15\text{W}$ , $V_I = 7\text{V}$ to $20\text{V}$	4.8	5.0	5.2	V
Regline	Line Regulation <sup>(1)</sup>	$T_J = +25^{\circ}\text{C}$ $V_O = 7\text{V}$ to $25\text{V}$ $V_I = 9\text{V}$ to $12\text{V}$	4.75	5.0	5.25	mV
Reload	Load Regulation <sup>(1)</sup>	$T_J = +25^{\circ}\text{C}$ $I_Q = 3\text{mA}$ to $1.5\text{A}$	-	4.0	100	mV
$I_Q$	Quiescent Current	$T_J = +25^{\circ}\text{C}$ $I_Q = 5\text{mA}$ to $1\text{A}$	-	9.0	100	mA
$\Delta I_Q$	Quiescent Current Change	$T_J = +25^{\circ}\text{C}$ $I_Q = 5\text{mA}$ to $1\text{A}$	-	0.03	0.5	mA
$\Delta V_O/\Delta T$	Output Voltage Drift <sup>(2)</sup>	$V_I = 7\text{V}$ to $25\text{V}$	-	0.3	1.3	mV/°C
$V_N$	Output Noise Voltage	$I_Q = 5\text{mA}$	-	-	-	µV/√Hz
RR	Ripple Rejection <sup>(2)</sup>	$f = 10\text{kHz}$ to $100\text{kHz}$ , $T_A = +25^{\circ}\text{C}$	-	-	-	dB
$V_{DROP}$	Dropout Voltage	$f = 120\text{Hz}$ , $V_I = 9\text{V}$ to $18\text{V}$	62.0	73.0	-	mA
$r_Q$	Output Resistance <sup>(2)</sup>	$I_Q = 1\text{A}$ , $T_J = +25^{\circ}\text{C}$	-	2.0	-	V
$I_{SC}$	Short Circuit Current	$f = 1\text{kHz}$	-	15.0	-	mA
$I_{PK}$	Peak Current <sup>(2)</sup>	$V_I = 35\text{V}$ , $T_A = +25^{\circ}\text{C}$ $T_J = +25^{\circ}\text{C}$	-	230	-	mA
			-	2.2	-	A

**Notes:**

1. Load and line regulation are specified at constant junction temperature. Changes in  $V_O$  due to heating effects must be taken into account separately. Pulse testing with low duty is used.
2. These parameters, although guaranteed, are not 100% tested in production.

## LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator

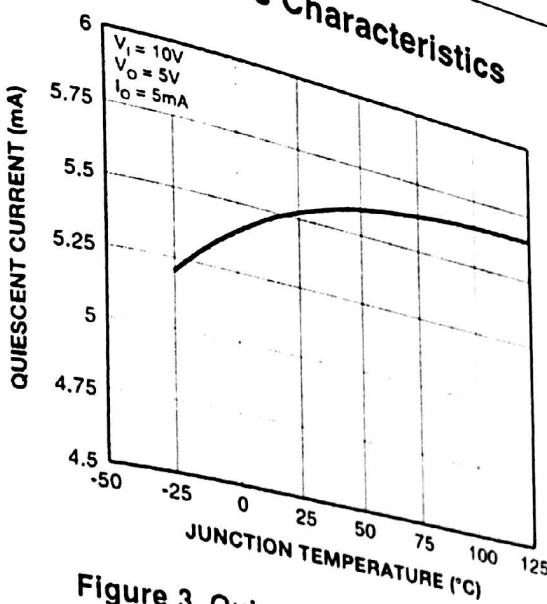


Figure 3. Quiescent Current

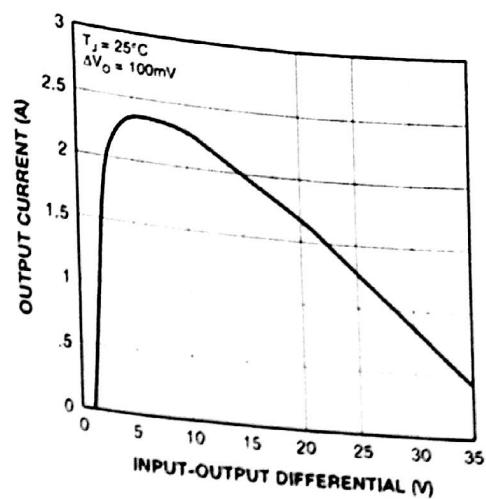


Figure 4. Peak Output Current

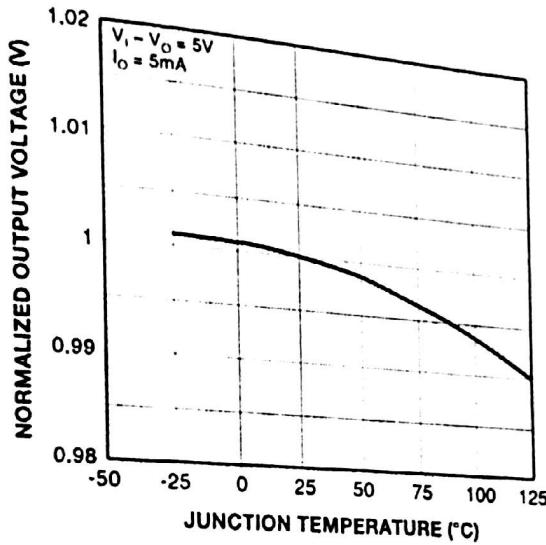


Figure 5. Output Voltage

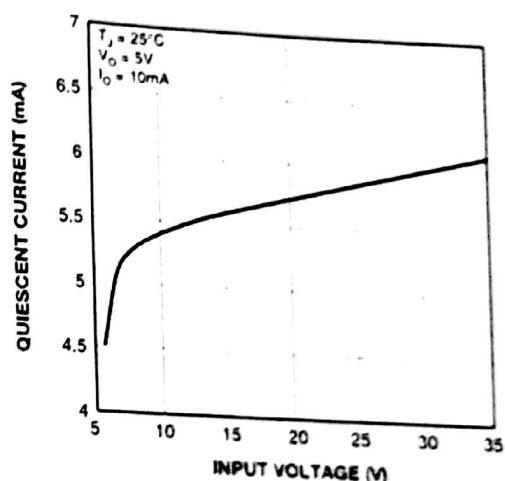


Figure 6. Quiescent Current

## LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator

### Typical Applications

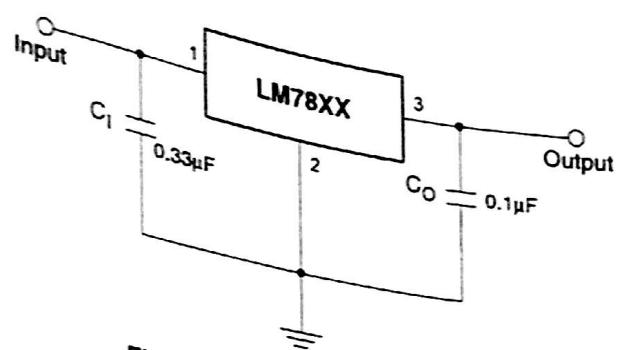


Figure 7. DC Parameters

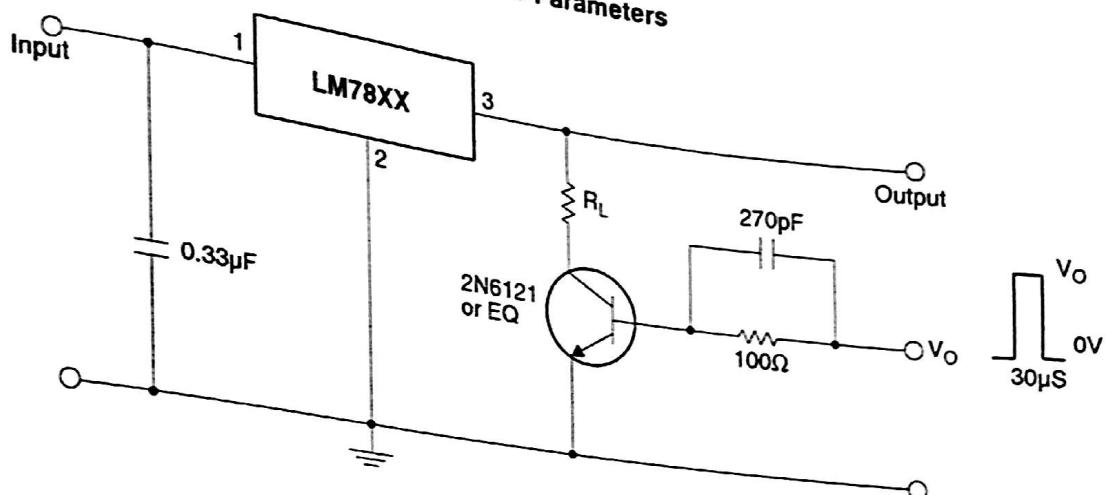


Figure 8. Load Regulation

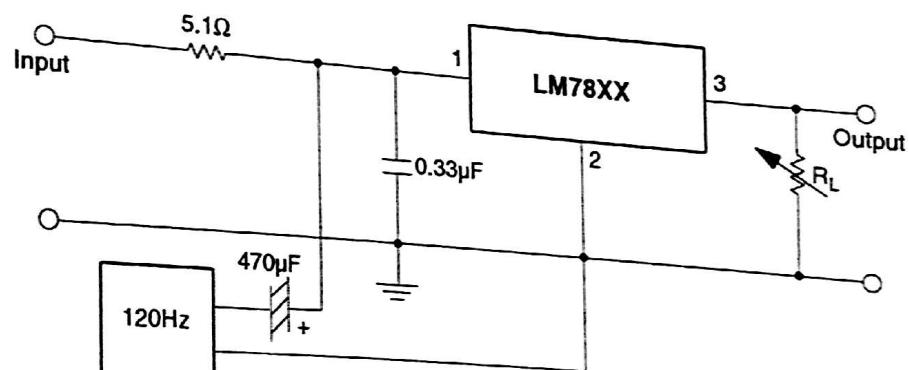


Figure 9. Ripple Rejection

# LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator

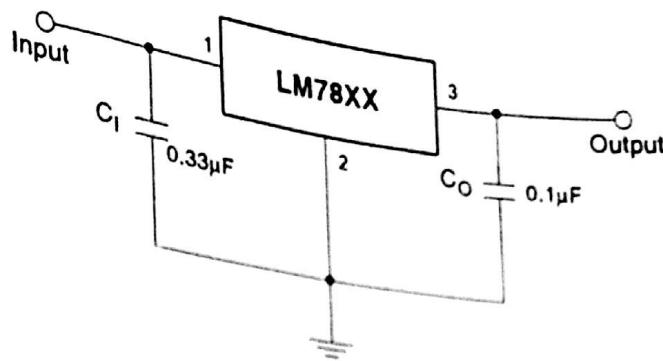
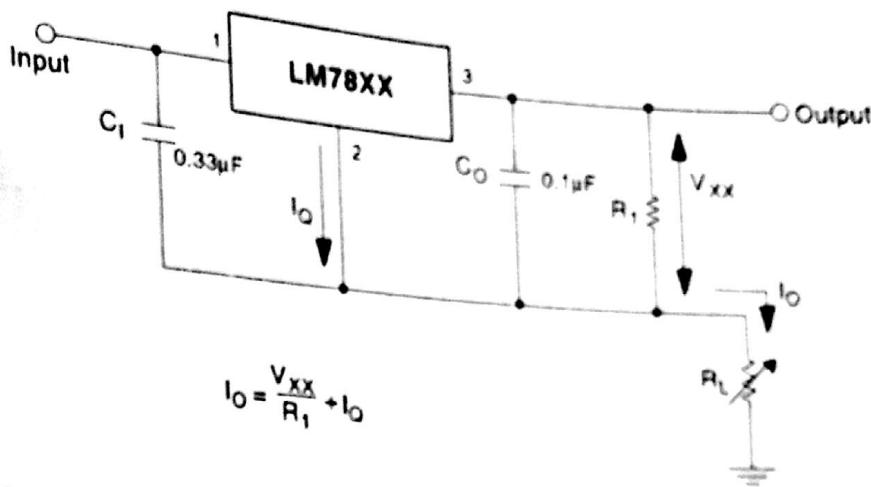


Figure 10. Fixed Output Regulator



**Notes:**

1. To specify an output voltage, substitute voltage value for "XX." A common ground is required between the input and the ripple voltage.
2. C<sub>1</sub> is required if regulator is located an appreciable distance from power supply filter.
3. C<sub>O</sub> improves stability and transient response.

Figure 11.

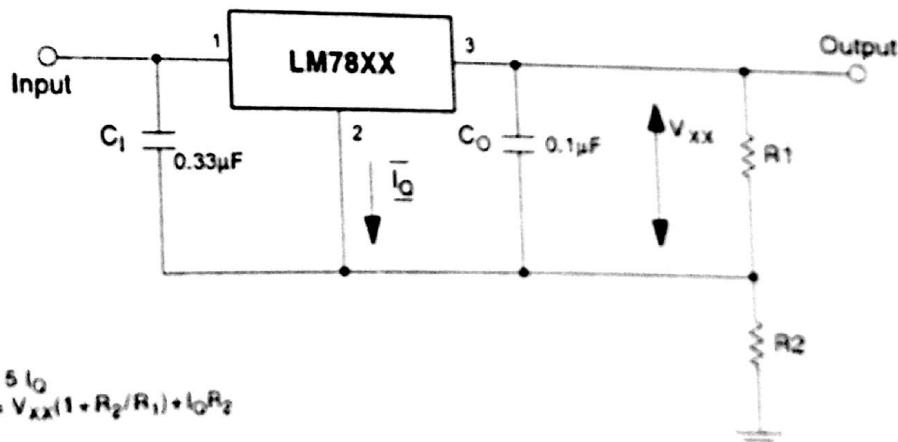
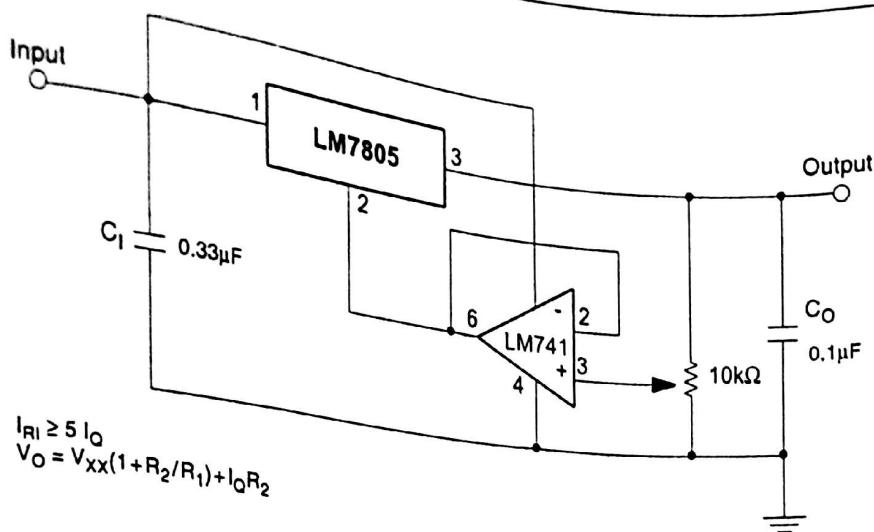
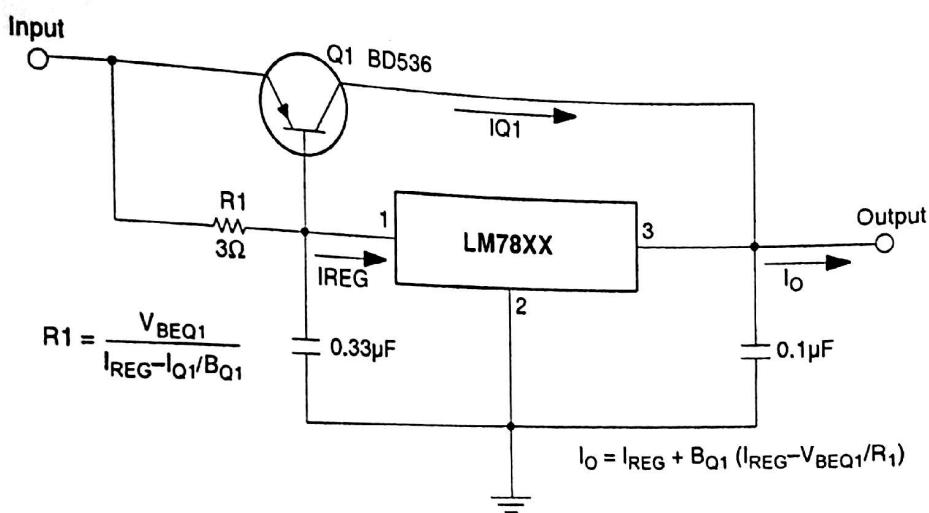


Figure 12. Circuit for Increasing Output Voltage

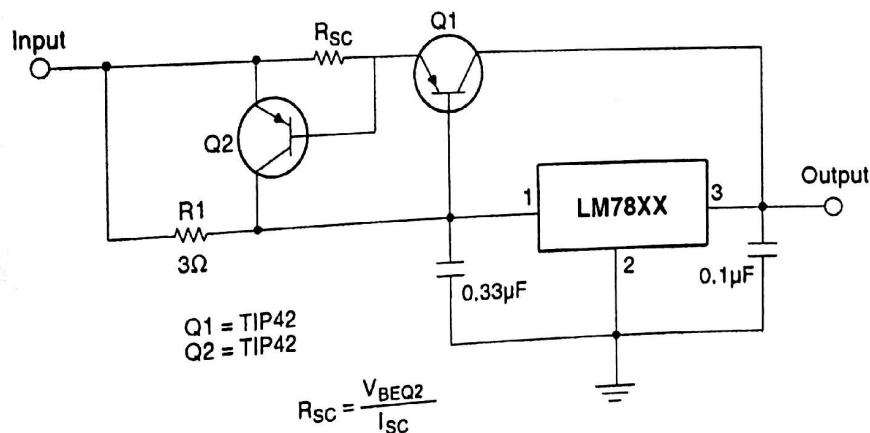
## LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator



**Figure 13. Adjustable Output Regulator (7V to 30V)**



**Figure 14. High Current Voltage Regulator**



**Figure 15. High Output Current with Short Circuit Protection**

**LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator**

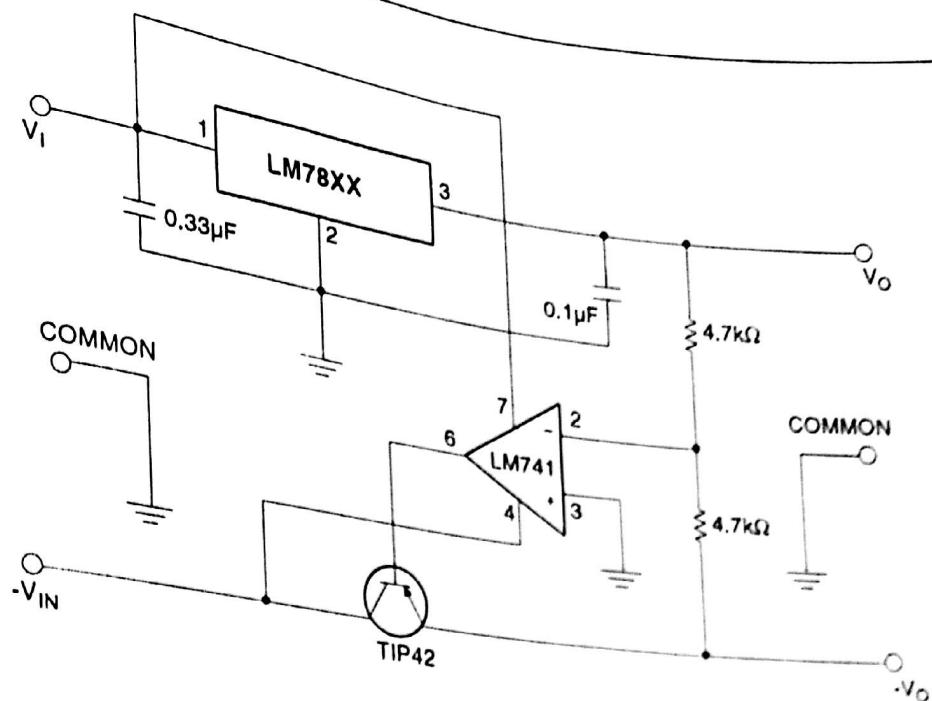


Figure 16. Tracking Voltage Regulator

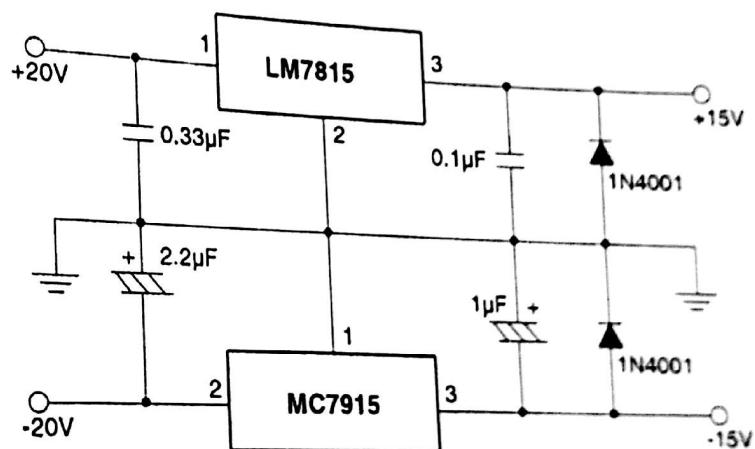


Figure 17. Split Power Supply ( $\pm 15V - 1A$ )

**LM78XX/LM78XXA-3-Terminal 1A Positive Voltage Regulator**

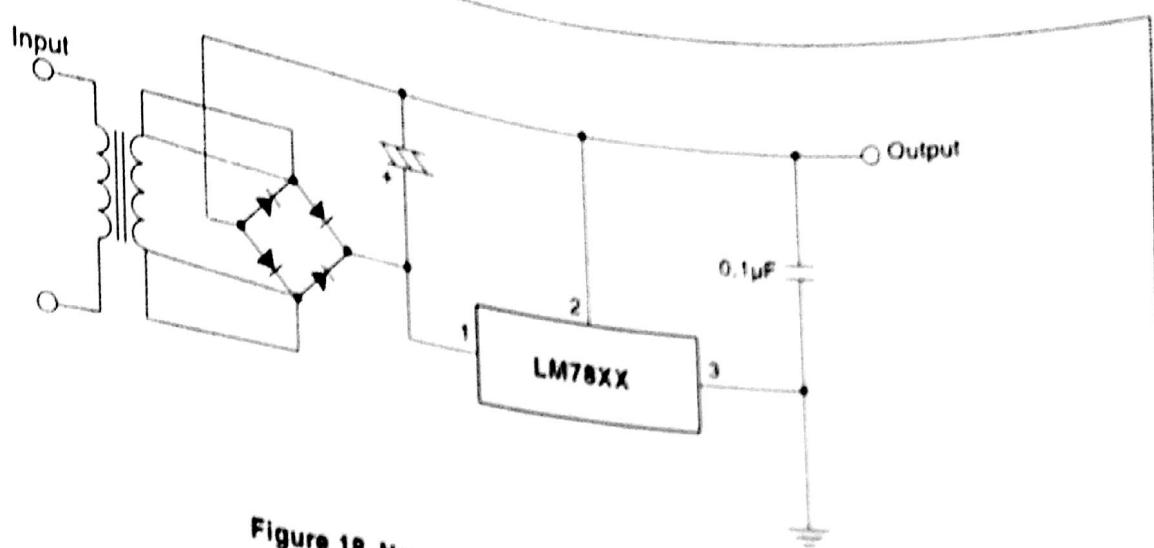


Figure 18. Negative Output Voltage Circuit

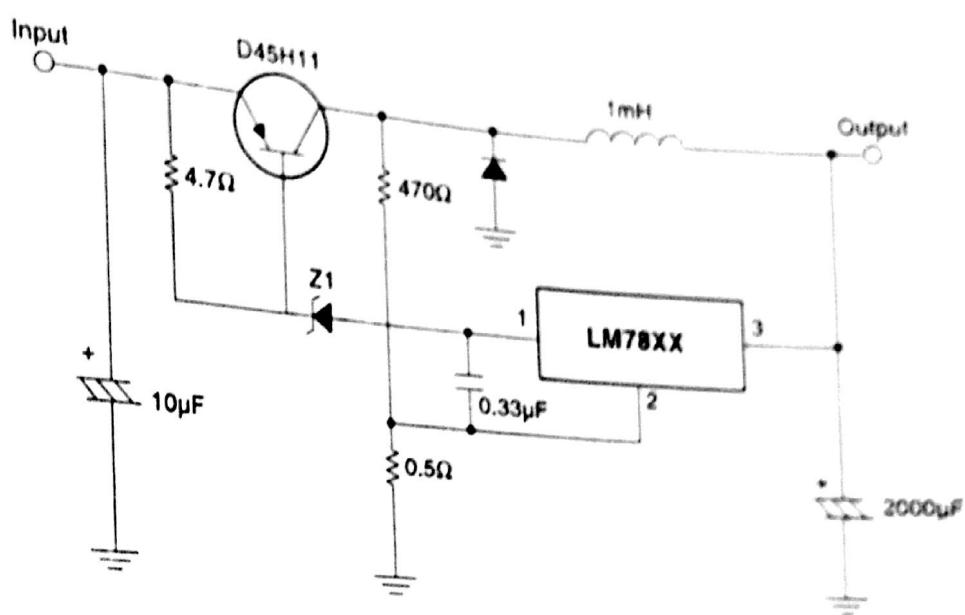


Figure 19. Switching Regulator

## REFERENCES

### **BOOKS :**

- J C Gupta Electronic Devices And Circuits

### **WEBSITES:**

- All datasheets from [www.datasheetcatalog.com](http://www.datasheetcatalog.com)
- [www.triindia.co.in](http://www.triindia.co.in)
- [www.google.com](http://www.google.com)
- [www.microchip.co.in](http://www.microchip.co.in)