Authors: Rohit, Sukumar, Krishna, Vinod.
--You can reach me @: rohit.gunnala@yahoo.com for further queries.

CHAPTER – 1

INTRODUCTION

CHAPTER – 1

INTRODUCTION

1.1 Power Factor:

The power factor defined by IEEE and IEC is the ratio between the applied active (true) power and the apparent power, and can be expressed as:

$$pf = \frac{P}{S}$$

In general, power factor is loosely defined as the cosine of the angle between voltage and current, which can be expressed mathematically as:

$$pf = \cos \emptyset$$

Where,

pf = power factor

P = active (true or real) power (in Watts)

S = apparent power (in VA, volts amps)

$\emptyset$= angle between voltage and current

- *Active (Real or True) Power* is measured in watts (W) and is the power drawn by the electrical resistance of a system doing useful work.
- *Apparent Power* is measured in volt-amperes (VA) and is the voltage on an AC system multiplied by all the current that flows in it. It is the vector sum of the active and the reactive power.
- *Reactive Power* is measured in volt-amperes reactive (VAR). Reactive Power is power stored in and discharged by inductive motors, transformers and solenoids.

Reactive power is required for the magnetization of a motor but does not perform any action. The reactive power required by inductive loads increases the amounts of apparent power measured in *kilovolt-amps (kVA)* in the distribution system. Increasing of the reactive and apparent power will cause the power factor (pf) to decrease.

A low power factor is the result of inductive loads such as transformers and electric motors. Unlike resistive loads creating heat by consuming kilowatts, inductive loads require a current flow to create magnetic fields to produce the desired work.

Power factor is an important measurement in electrical AC systems because

- An overall power factor less than 1 indicates that the electricity supplier need to provide more generating capacity than actually required.
- The current waveform distortion that contributes to reduced power factor is caused by voltage waveform distortion and overheating in the neutral cables of three-phase systems.

International standards such as IEC 61000-3-2 have been established to control current waveform distortion by introducing limits for the amplitude of current harmonics.

Example – 1.1:

An industrial plant draws *200A* at *400V* and the supply transformer and backup UPS is rated *200A × 400V = 80 kVA.*

If the power factor (pf) of the loads is only *0.7*

Then, active power, P=80 kVA × 0.7

=56 kW

So, only 56kW of real power is consumed by the system. If the power factor is close to 1 (purely resistive circuit) the supply system with transformers, cables, switchgear and UPS could be made considerably smaller.

If the power factor is less than 1, it means that the circuit's wiring has to carry more current than what would be necessary with zero reactance in the circuit to deliver the same amount of active power to the resistive load.

A low power factor is expensive and inefficient and some utility companies may charge additional fees when the power factor is less than *0.95*. A low power factor will reduce the electrical system's distribution capacity by increasing the current flow and causing voltage drops.

1.2. "Leading" or "Lagging" Power Factors:

Power factors are usually stated as "leading" or "lagging" to show the sign of the phase angle.

- With a purely resistive load current and voltage changes polarity in step and the power factor will be 1. Electrical energy flows in a single direction across the network in each cycle.

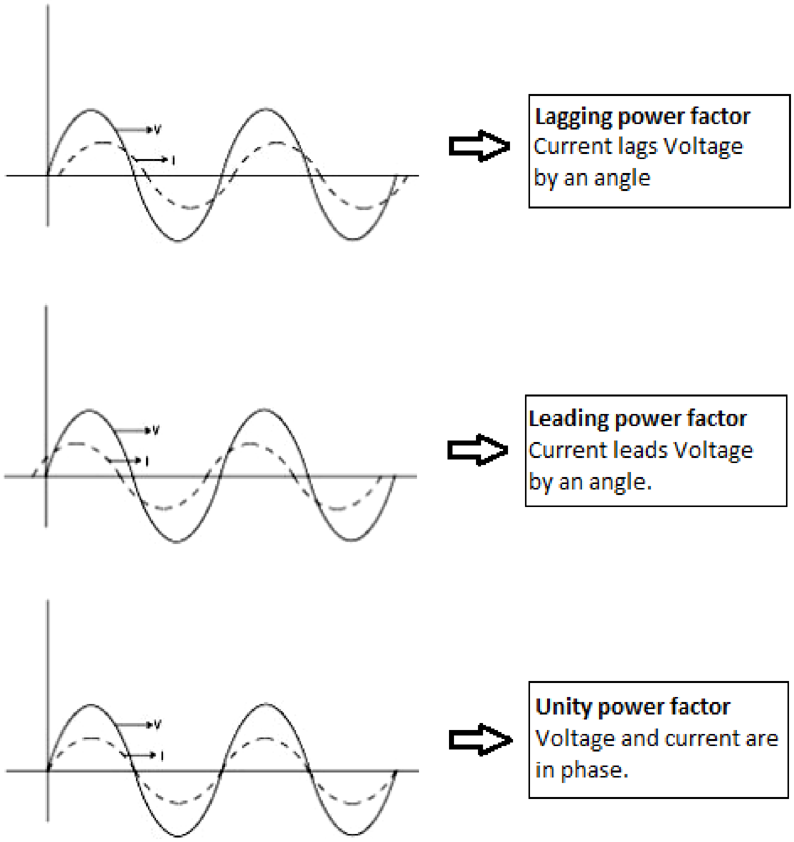


Fig. 1.1

- Inductive loads such as transformers, motors and wound coils consume reactive power with current waveform lagging the voltage.
- Capacitive loads such as capacitor banks or buried cables generate reactive power with current phase leading the voltage.

Inductive and capacitive loads store energy in the form of magnetic or electric fields in the devices during parts of the AC cycles. The energy is returned back to the power source during the rest of the cycles.

1.3. Power Factor for a Three-Phase Motor:

The total power required by an inductive device as a motor or similar consists of

- Active (true or real) power (measured in kilowatts, kW)
- Reactive power - the nonworking power caused by the magnetizing current, required to operate the device (measured in kVARs)

The power factor for a three-phase electric motor can be expressed as:

$$pf = \frac{P}{\sqrt{3}\,VI}$$

Where,

pf = power factor

P = power applied (W, watts)

V = voltage (V)

I = current (A)

1.3.1. Typical Motor Power Factors:

Power (hp) (1 hp = 745.7 W)	Speed (rpm)	Power factor		
		½ load	¾ load	Full load
0 – 5	1800	0.72	0.82	0.84
5 – 20	1800	0.74	0.84	0.86
20 – 100	1800	0.79	0.86	0.89
100 – 300	1800	0.81	0.88	0.91

1.4. Importance of power factor in distribution systems:

The significance of power factor lies in the fact that utility companies supply customers with volt-amperes, but bill them for watts. Power factors below 1.0 require a utility to generate more than the minimum volt-amperes necessary to supply the real power (watts). This increases generation and transmission costs. For example, if the load power factor were as low as 0.7, the apparent power would be 1.4 times the real power used by the load. Line current in the circuit would also be 1.4 times the current required at 1.0 power factor, so the losses in the circuit would be doubled (since they are proportional to the square of the current). Alternatively all components of the system such as generators, conductors, transformers, and switchgear would be increased in size (and cost) to carry the extra current.

Utilities typically charge additional costs to customers who have a power factor below some limit, which is typically 0.9 to 0.95. Engineers are often interested in the power factor of a load as one of the factors that affect the efficiency of power transmission.

1.5. Power factor Correction:

It is often desirable to adjust the power factor of a system to near 1.0. This power factor correction (PFC) is achieved by switching in or out banks of inductors or capacitors. For example the inductive effect of motor loads may be offset by locally connected capacitors. When reactive elements supply or absorb reactive power near the load, the apparent power is reduced.

Power factor correction may be applied by an electrical power transmission utility to improve the stability and efficiency of the transmission network. Correction equipment may be installed by individual electrical customers to reduce the costs charged to them by their electricity supplier. A high power factor is generally desirable in a transmission system to reduce transmission losses and improve voltage regulation at the load.

Power factor correction brings the power factor of an AC power circuit closer to 1 by supplying reactive power of opposite sign, adding capacitors or inductors which act to cancel the inductive or capacitive effects of the load, respectively. For example, the inductive effect of motor loads may be offset by locally connected capacitors. If a load had a capacitive value, inductors (also known as *reactors* in this context) are connected to correct the power factor. In the electricity industry, inductors are said to consume reactive power and capacitors are said to supply it, even though the reactive power is actually just moving back and forth on each AC cycle.

The reactive elements can create voltage fluctuations and harmonic noise when switched on or off. They will supply or sink reactive power regardless of whether there is a corresponding load operating nearby, increasing the system's no-load losses. In a worst case, reactive elements can interact with the system and with each other to create resonant conditions, resulting in system instability and severe overvoltage fluctuations. As such, reactive elements cannot simply be applied at will, and power factor correction is normally subject to engineering analysis.

Authors: Rohit, Sukumar, Krishna, Vinod.
--You can reach me @: rohit.gunnala@yahoo.com for further queries.

CHAPTER – 2

INTELLIGENT POWER FACTOR MANAGER (IPFM)

2.1 INTRODUCTION

2.2 BLOCK DIAGRAM OF IPFM

2.3 FLOW CHART OF IPFM

2.4 WORKING PRINCIPLE

2.5 CIRCUIT DIAGRAM

CHAPTER – 2

INTELLIGENT POWER FACTOR MANAGER

2.1. INTRODUCTION

An Intelligent Power factor manager/ automatic power factor correction unit is used to improve power factor. A power factor correction unit usually consists of a number of capacitors/ capacitor banks that are switched by means of contactors. These contactors are controlled by a microcontroller that measures power factor in an electrical network. To be able to measure power factor, the microcontroller uses a current and potential transformer's output signal on any one of the phase.

Depending on the load and power factor of the network, the power factor controller will switch the necessary blocks of capacitors in steps to make sure the power factor stays above a selected value (usually demanded by the energy supplier), say 0.9.

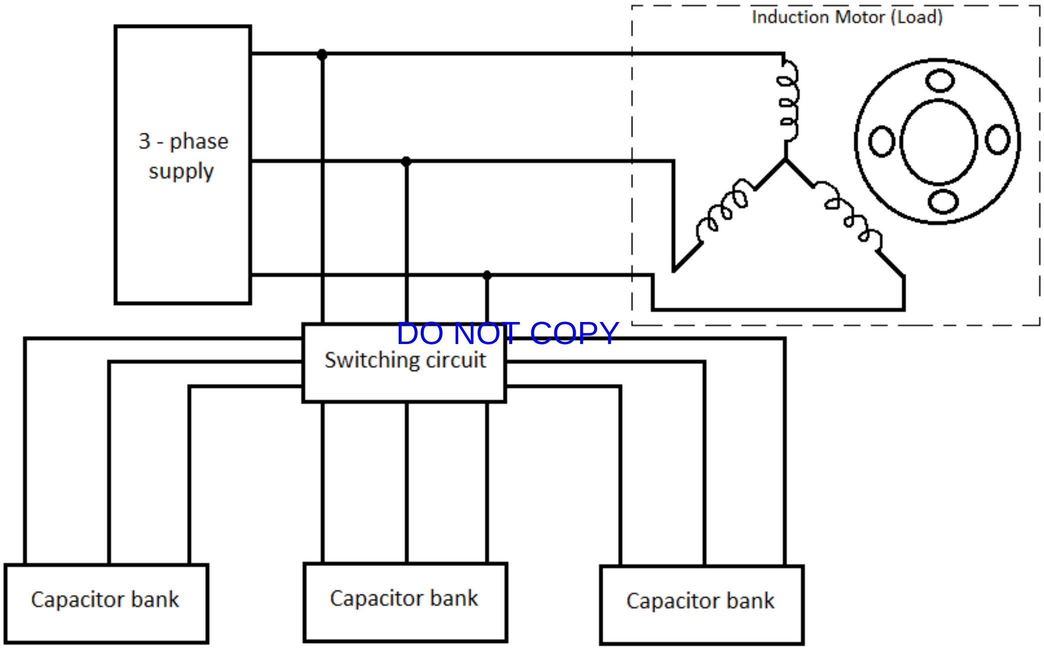


Fig. 2.1

Phasor diagram for the above representation can be drawn as shown below:

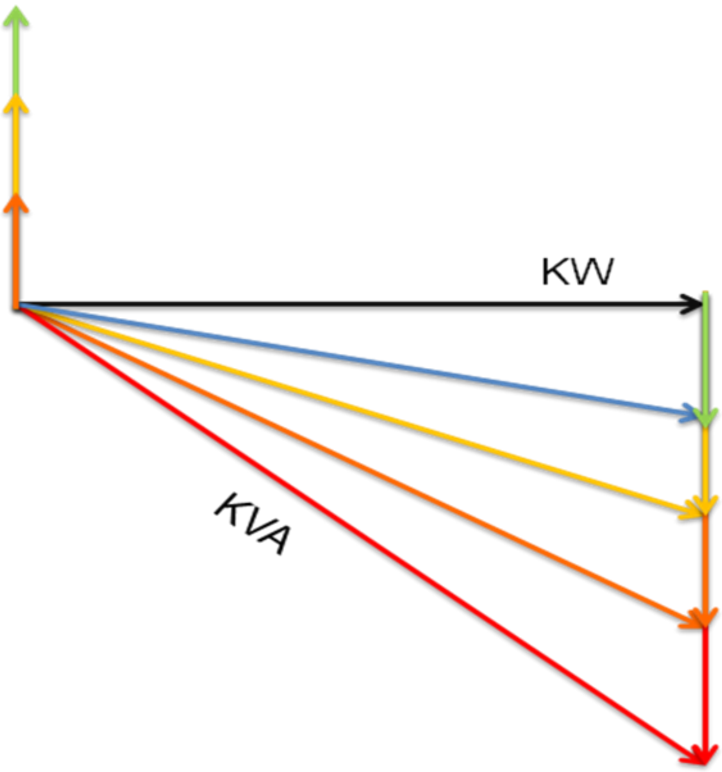


Fig. 2.2

2.2 BLOCK DIAGRAM OF IPFM

As shown in figure 2.3, inputs of CT and PT are taken from a 3-phase supply which is being fed to an inductive load. Output of CT and PT is given to a comparator which consists of op-amps being used as zero crossing detectors. The moment when PT signal crosses zero it triggers a counter in microcontroller. Zero crossing of CT signal stops the counter and the value in the counter is used to calculate the power factor of the load. According to the program written on microcontroller, switching circuit connects required number of capacitor banks so as to get power factor closer to 1.0. Meanwhile, an LCD display is used to display power factor of load intermittently.

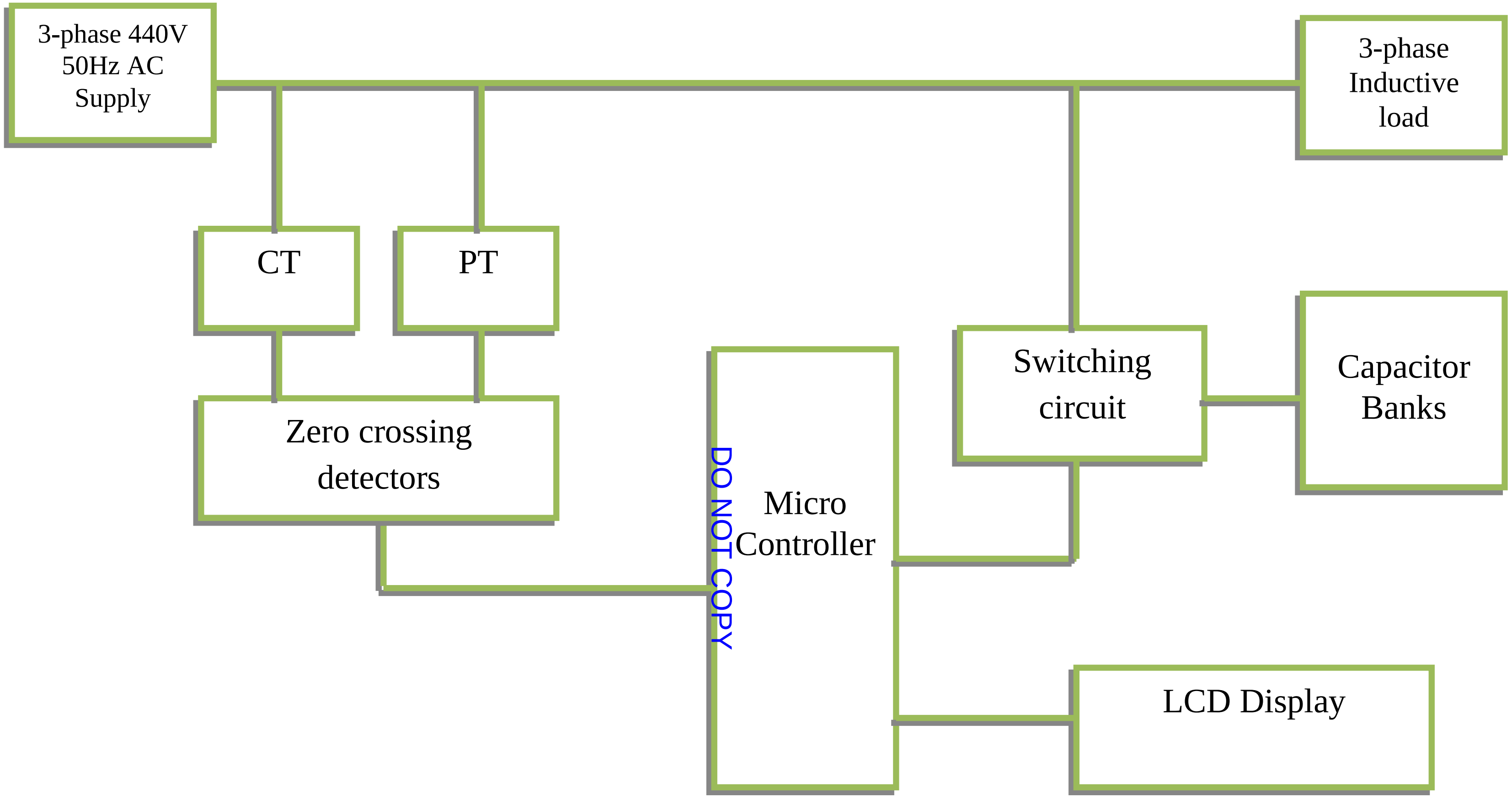


Fig. 2.3 Block diagram of intelligent power factor manager

2.3. FLOW CHART OF IPFM

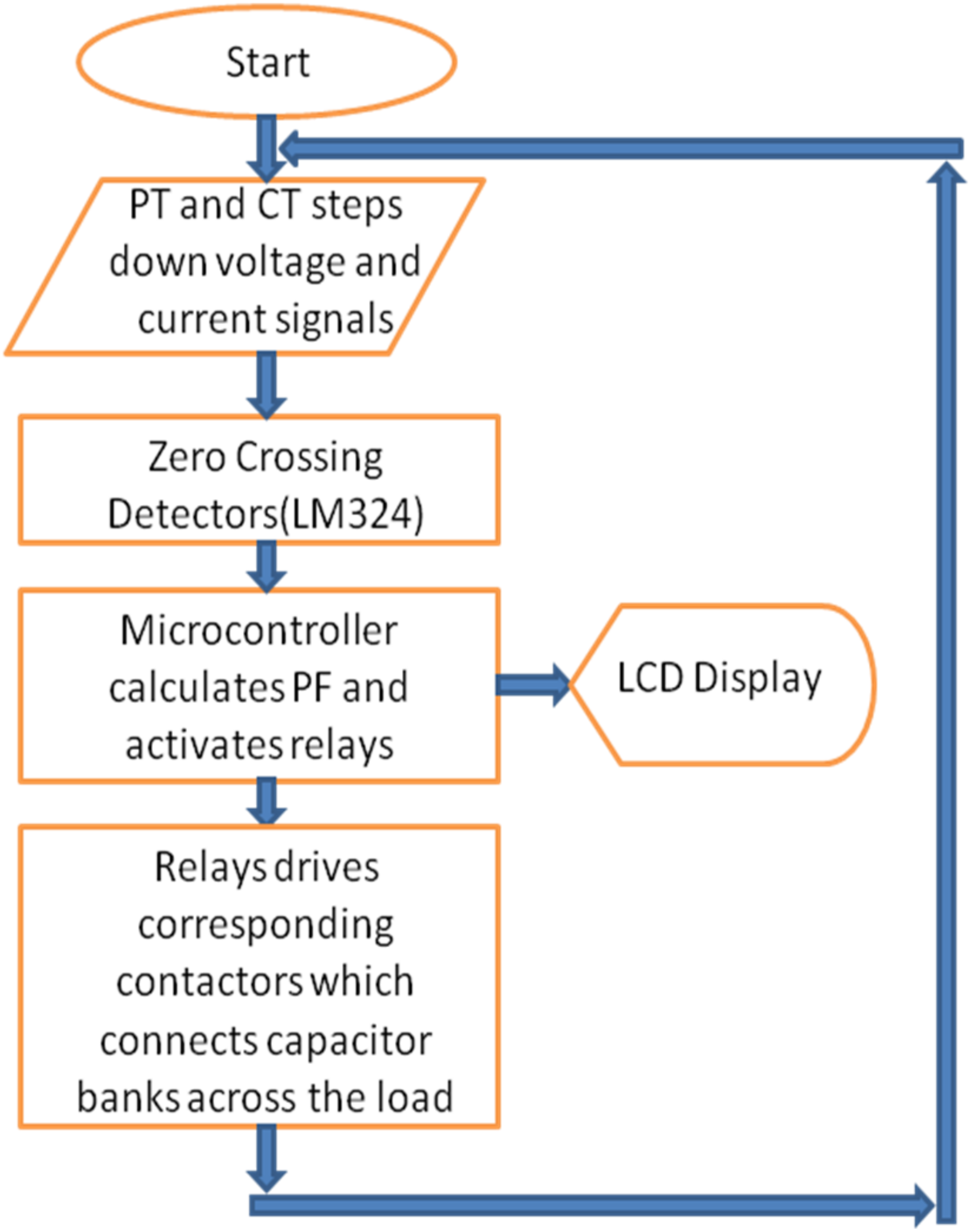


Fig. 2.4

2.4. WORKING PRINCIPLE

A 3-phase, 440V, 50Hz, 5hp induction motor is fed from a balanced 3-phase supply. A CT is placed in any one of the phase and a PT is placed between any two phases as shown in fig. 3. Secondaries of CT and PT are loaded with 50Ω and 100KΩ (potential divider) respectively. As we are using an induction motor as load, the current drawn by the induction motor lags behind the voltage by some angle. These two waveforms (almost sinusoidal) are given to a LM324 comparator which consists of four op-amps where any two of them are used as zero crossing detectors. Output of these zero crossing detectors is fed to Port-C of ATmega 16L microcontroller. This microcontroller is programmed such that when the PT signal crosses zero then a counter variable starts counting and stops as soon as CT signal crosses zero. A crystal oscillator of 8MHz is connected to ATmega 16L to make the calculation of time lag between CT and PT waveforms discernible. Thus, the power factor of the load is calculated by microcontroller using cosine function under math library. This power factor is displayed on an LCD display JHD 162A. At the same time, depending on the displayed power factor, microcontroller decides required number of capacitor banks to be connected across the load. Here, we are using three capacitor banks which are connected across the load by eight different combinations using microcontroller.

2.5. CIRCUIT DIAGRAM

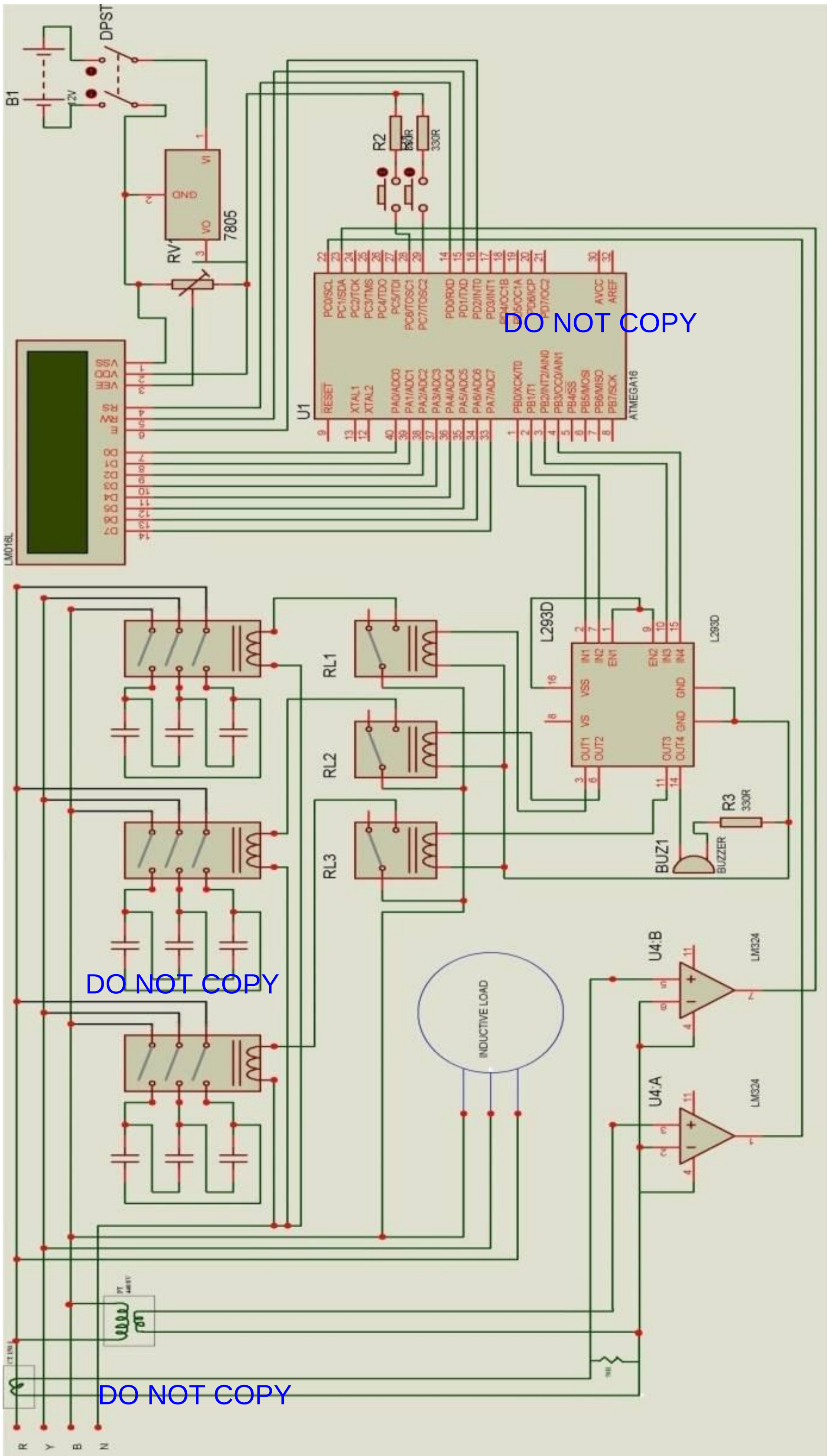


Fig. 2.5 Circuit diagram of intelligent power factor manager

CHAPTER – 3

COMPONENTS USED

3.1 CURRENT TRANSFORMER

3.2 POTENTIAL TRANSFORMER

3.3 MICROCONTROLLER

3.4 COMPARATOR

3.5 LCD DISPLAY

3.6 INDUCTION MOTOR

3.7 GENERAL PURPOSE PCB

3.8 RELAYS AND CONTACTORS

3.9 VOLTAGE REGULATOR

3.10 CAPACITORS, TRIMPOTS AND LEDs

3.11 H – BRIDGE

3.12 PARALLEL PORT

3.13 STUDS, CONNECTORS AND CONNECTING WIRES

CHAPTER – 3

COMPONENTS USED

3. COMPONENTS USED

1. Current Transformer – 150/1
2. Potential Transformer – 440/8V
3. Microcontroller – ATmega 16L
4. Comparator – LM324
5. LCD display – JHD 162A
6. 3-phase, 440V, 50Hz, 5hp induction motor
7. General purpose PCB
8. Relays and Contactors
9. Voltage regulator – L7805
10. Capacitors, Trimpots, LED
11. H – Bridges
12. Parallel port
13. Studs, Connectors and connecting wires

3.1. Current Transformer – 150/1:

Generally, there are different types of current transformers which are segregated on the basis of the ratio of amount of current that flows in primary to that of the secondary. In this project, we are using a current transformer of rating 150/1, which means that when 150A of current flows in the primary winding only 1A flows in the secondary winding of the current transformer at full load.

A current transformer of low current rating mainly consists of a primary winding, core and secondary winding. A burden (load) must be connected across secondary winding. In the absence of burden, a high voltage may result across secondary of current transformer which leads to burning of winding and nearby components. Ideally, secondary of a current transformer is short circuited. Practically we use a burden of low resistance.

Core of the current transformer is made up of finely laminated cold rolled grain oriented (CRGO) Silicon steel. It maintains low core loss and high permeability. All the laminations are of circular in shape and stacked together with insulating material in between them. Primary winding of the current transformer is nothing but a single current carrying conductor which passes through the circular core and physically isolated from the secondary of current transformer. The secondary is the number of turns wound on current transformer to step down the amount of current to be able to measure. The whole set up works like a step-up transformer, where number of turns at secondary is more than that of primary.

Role of current transformer in this project is that, it gives the load current signal to comparator where its zero crossing is checked with respect to that of the potential transformer.

3.2. Potential Transformer – 440/8V:

Potential transformers are also available in different types based on primary to secondary voltage ratings. In this project, we are taking a potential transformer of rating 440/8V, which means that 440V in primary circuit of potential transformer is converted into 8V in secondary circuit.

A potential transformer is nothing but a step-down transformer, where number of turns at primary is more than that of the secondary. Core of the potential transformer is made up of E-I laminated cold rolled grain oriented (CRGO) Silicon steel. This type of core reduces core losses, eddy current losses and provides high permeability.

Ideally, secondary of a potential transformer is kept open circuited. But practically, here we are connecting a high resistance across the secondary from which reduced voltage signal is applied directly to the comparator.

3.3. Microcontroller – ATmega 16L:

The high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 16KB of programmable flash memory, 1KB SRAM, 512B EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. It supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts.

By executing instructions in a single clock cycle, it achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

3.3.1. About ATmega 16:

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

The ATmega16 provides the following features: 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the

AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators and evaluation kits.

3.3.2. Features:

- Advanced RISC Architecture
  - ✓ 131 Powerful Instructions – Most Single-clock Cycle Execution
  - ✓ 32 x 8 General Purpose Working Registers
  - ✓ Fully Static Operation
  - ✓ Up to 16 MIPS Throughput at 16 MHz
  - ✓ On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - ✓ 16 Kbytes of In-System Self-programmable Flash program memory
  - ✓ 512 Bytes EEPROM
  - ✓ 1 Kbyte Internal SRAM
  - ✓ Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - ✓ Data retention: 20 years at 85°C/100 years at 25°C
  - ✓ Optional Boot Code Section with Independent Lock Bits
  - ✓ In-System Programming by On-chip Boot Program
  - ✓ True Read-While-Write Operation
  - ✓ Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - ✓ Boundary-scan Capabilities According to the JTAG Standard
  - ✓ Extensive On-chip Debug Support
  - ✓ Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V for ATmega16L
  - 4.5V - 5.5V for ATmega16
- Speed Grades
  - 0 - 8 MHz for ATmega16L
  - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L

- ✓ Active: 1.1 mA
- ✓ Idle Mode: 0.35 mA
- ✓ Power-down Mode: < 1 µA

3.3.3. Pin diagram:

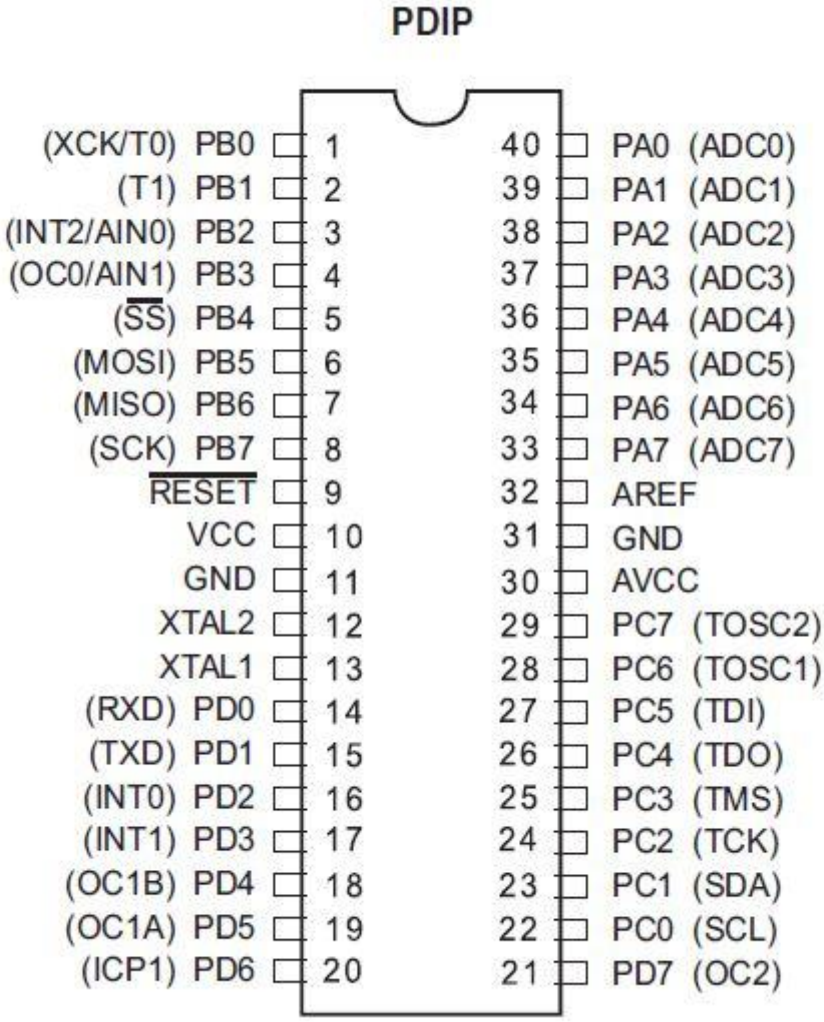


Fig. 3.1

3.3.4. Pin descriptions:

V_{cc}	Digital Voltage Supply
GND	Ground

Port A (PA7...PA0)	Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.
Port B (PB7...PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.
Port C (PC7...PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5 (TDI), PC3 (TMS) and PC2 (TCK) will be activated even if a reset occurs.
Port D (PD7...PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins

	that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.
RESET	Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is 1.5μs. Shorter pulses are not guaranteed to generate a reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.

3.4. Comparator – LM324:

These devices consist of four independent high-gain frequency-compensated operational amplifiers that are designed specifically to operate from a single supply over a wide range of voltages. Operation from split supplies also is possible when the difference between the two supplies is 3 V to 30 V and V_{CC} is at least 1.5 V more positive than the input common-mode voltage. The low supply-current drain is independent of the magnitude of the supply voltage.

3.4.1. Pin Diagram:

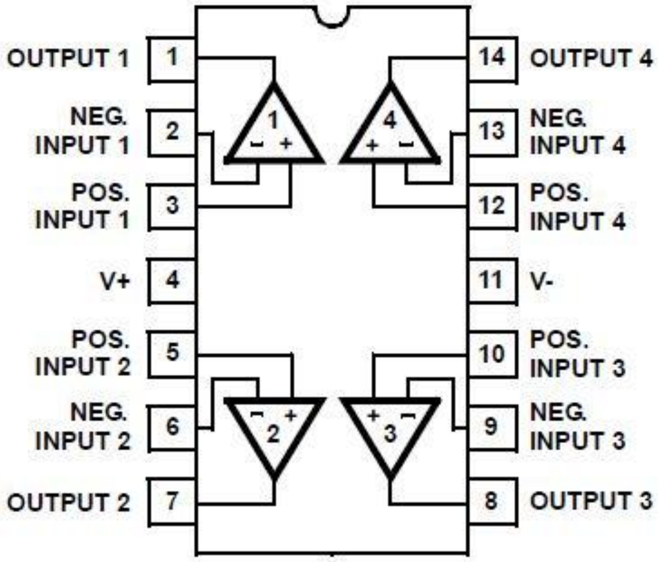


Fig. 3.2

3.4.2. Applications:

Applications include transducer amplifiers, dc amplification blocks, and all the conventional operational-amplifier circuits that now can be more easily implemented in single-supply-voltage systems. For example, the LM124 can be operated directly from the standard 5-V supply that is used in digital systems and easily provides the required interface electronics without requiring additional ±15-V supplies.

3.5. LCD Display – JHD 162A:

The display used here is 16x2 LCD (Liquid Crystal Display); this means 16 display blocks by 2 lines. Each block can be used to display 1 character. So there are total 32 such blocks. One block has 5x7 pixels. Depending on which pixel is ON and which is OFF we can display several Alpha-Numeric characters. This model also has a green backlight, which helps us to see the display even in dark. In reality this module consists of a controller chip, a segment driver chip, LCD display and some passive components.

Fig. 3.3

A very popular standard exists which allows us to communicate with the vast majority of LCDs regardless of their manufacturer. The standard is referred to as HD44780U, which refers to the controller chip which receives data from an external source (in this case, the Atmega16) and communicates directly with the LCD. The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. Here we are using 8-bit mode of LCD, i.e., using 8-bit data bus.

3.5.1. Pin Diagram:

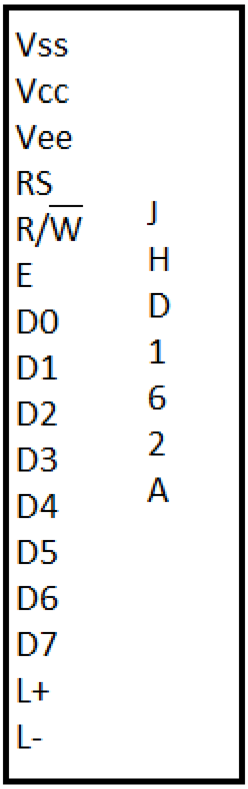


Fig. 3.4

3.5.2. Pin description:

Function	Pin number	Name	Logic status	Description
Ground	1	V_{ss}	-	0V
Power Supply	2	V_{cc}	-	+5V
Contrast	3	V_{ee}	-	0-V_{cc}
Control of operating	4	RS	0	D0-D7 are interpreted as commands
			1	D0-D7 are interpreted as data
	5	R/W	0	Write data (from controller to LCD)
			1	Read data (from LCD to controller)
	6	E	0	Access to LCD disabled
			1	Normally operating
			Falling edge (from 1 to 0)	Data/commands are transferred to LCD
Data/Commands	7	D0	0/1	Bit 0/LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB
Back light power +5V	15	L+	-	0-V_{cc}
Back light power ground	16	L-	-	Ground

The three control lines are referred to as EN, RS, and R/W.

The EN line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The RS line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

The RW line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low. In our case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

3.5.3. LCD Basic Commands:

Sl. No.	Name	Hex	Decimal
1.	Function Set: 8-bit, 1 Line, 5x7 dots	0x30	48
2.	Function Set: 8-bit, 2 Line, 5x7 dots	0x38	56
3.	Function Set: 4-bit, 1 Line, 5x7 dots	0x20	32
4.	Function Set: 4-bit, 2 Line, 5x7 dots	0x28	40
5.	Entry mode	0x06	6
6.	Display OFF, Cursor OFF (clearing display without clearing DDRAM content)	0x08	8
7.	Display ON, Cursor ON	0x0E	14
8.	Display ON, Cursor OFF	0x0C	12
9.	Display ON, Cursor blinking	0x0F	15

10.	Shift entire display left	0x18	24
11.	Shift entire display right	0x1C	30
12.	Move cursor left by one character	0x10	16
13.	Move cursor right by one character	0x14	20
14.	Clear display (also clear DDRAM content)	0x01	1
15.	Set DDRAM address or cursor position on display	0x80+add	128+add
16.	Set CGRAM address or set pointer to CGRAM location	0x40+add	64+add

3.6. Induction motor:

An induction motor or asynchronous motor is a type of alternating current motor where power is supplied to the rotor by means of electromagnetic induction. An electric motor turns because of magnetic force exerted between a stationary electromagnet called the stator and a rotating electromagnet called the rotor. Different types of electric motors are distinguished by how electric current is supplied to the moving rotor. In a DC motor and a slip-ring AC motor, current is provided to the rotor directly through sliding electrical contacts called commutators and slip rings. In an induction motor, by contrast, the current is induced in the rotor without contacts by the magnetic field of the stator, through electromagnetic induction. An induction motor is sometimes called a *rotating transformer* because the stator (stationary part) is essentially the primary side of the transformer and the rotor (rotating part) is the secondary side. Unlike the normal transformer which changes the current by using time varying flux, induction motors use rotating magnetic fields to transform the voltage. The current in the primary side creates an electromagnetic field which interacts with the electromagnetic field of the secondary side to produce a resultant torque, thereby transforming the electrical energy into mechanical energy. Induction motors are widely used, especially polyphase induction motors, which are frequently used in industrial drives.

Induction motors are now the preferred choice for industrial motors due to their rugged construction, absence of brushes (which are required in most DC motors) and—thanks to modern power electronics—the ability to control the speed of the motor.

3.6.1. Principle of Operation:

When the stator or primary winding of 3-phase induction motor is connected to a 3-phase ac supply, a rotating magnetic field is established which rotates at synchronous speed. The direction of revolution of this field will depend upon the phase sequence of the primary currents and, therefore, will depend upon the order of connection of the primary terminals to the supply. The direction of rotation of the field can be reversed by interchanging the connection to the supply of any two leads of a 3-phase induction motor. The number of magnetic poles of the revolving field will be the same as the number of poles for which each phase of the primary or stator winding is wound. The speed at which the field produced by the primary currents will revolve is called the *synchronous speed of* the motor and is given by an expression

$$N_s = \frac{120\,f}{P}$$

Where,

f is supply frequency and

P is the number poles on stator.

The revolving magnetic field produced by the primary currents sweeps across the rotor conductors and thereby induces an emf in these conductors. Since the rotor winding is either directly shorted or closed through some external resistance, the emf induced in the secondary by the revolving field causes a current to flow in the rotor conductors whose direction is such as to oppose the cause which is producing it. Because the cause producing the induced currents is the relative speed between the rotating magnetic field and the stationary rotor conductors, therefore, they circulate in such a way that a torque is produced in the rotor tending to cause it to follow the rotating magnetic field and thus reducing the relative speed.

An induction motor cannot run at synchronous speed. If it were possible, by some means, for the rotor to attain synchronous speed, the rotor would then be standstill with the respect to the rotating flux. Then no emf would be induced in the rotor, no rotor current would flow, and therefore, there would be no torque developed.

Fig. 3.5

An introduction motor running on no-load will have a speed very close to synchronous speed and, therefore, emf induced in the rotor winding will be very small. As the mechanical load is applied to the motor shaft, the motor slows down, the relative motion between the rotating magnetic field and the rotor increases causing increase in rotor emf, rotor current and so in the torque developed. Thus the motor meets the increased load.

3.7. General purpose PCB:

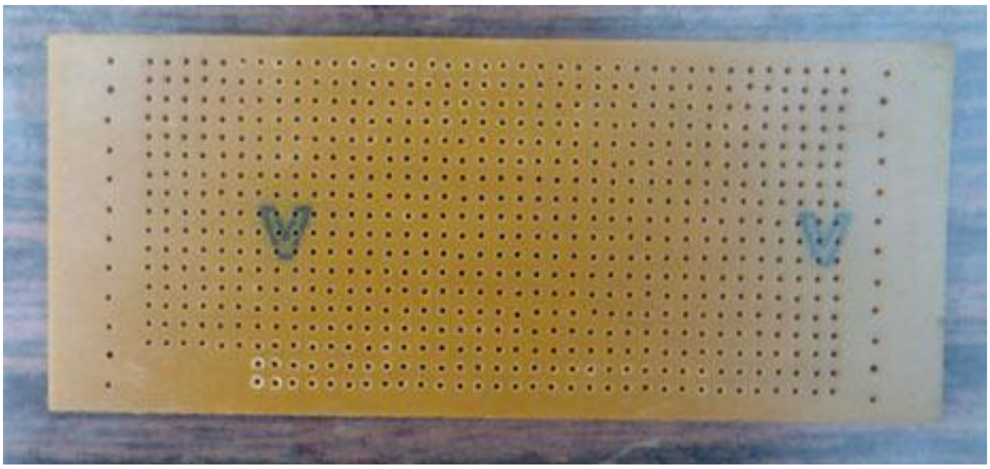


Fig. 3.6

When compared to protoboards, General purpose PCB/ vector board have more control over wire routing, can accommodate different sized parts, all traces are visible, and scope probes can easily and safely clip on to them. A simple General purpose PCB is shown in Fig. 3.6.

3.8. Relays and Contactors:

3.8.1. Relays:

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

3.8.2. Basic design and operation:

A simple electromagnetic relay consists of a coil of wire surrounding a soft iron core, an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.

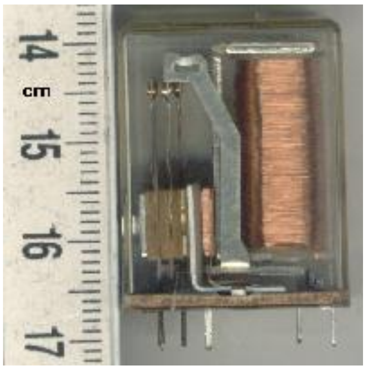


Fig. 3.7

When an electric current is passed through the coil it generates a magnetic field that attracts the armature and the consequent movement of the movable contact either makes or breaks (depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduces arcing.

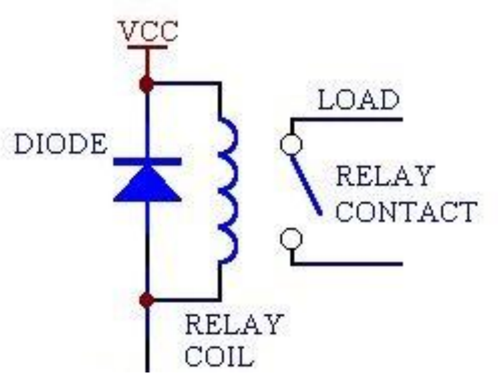


Fig. 3.8 Circuit diagram representation of relay

When the coil is energized with direct current, a diode is often placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to semiconductor circuit components. Some automotive relays include a diode inside the relay case. Alternatively, a contact protection network consisting of a capacitor and resistor in series (snubber circuit) may absorb the surge. If the coil is designed to be energized with alternating current (AC), a small copper "shading ring" can be crimped to the end of the solenoid, creating a small out-of-phase current which increases the minimum pull on the armature during the AC cycle.

A solid-state relay uses a thyristor or other solid-state switching device, activated by the control signal, to switch the controlled load, instead of a solenoid. An optocoupler (a light-emitting diode (LED) coupled with a photo transistor) can be used to isolate control and controlled circuits.

3.8.3. Applications – Relays:

Relays are used to and for:

- ✓ Control a high-voltage circuit with a low-voltage signal, as in some types of modems or audio amplifiers.
- ✓ Control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile.
- ✓ Detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays).
- ✓ Isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch. The latter is often applied to control office lighting as the low voltage wires are easily installed in partitions, which may be often moved as needs change. They may also be controlled by room occupancy detectors in an effort to conserve energy.
- ✓ Logic functions. For example, the Boolean AND function is realised by connecting normally open relay contacts in series, the OR function by connecting normally open contacts in parallel. The change-over or Form C contacts perform the XOR (exclusive or) function. Similar functions for NAND and NOR are accomplished using normally closed contacts. The Ladder programming language is often used for designing relay logic networks.
    - Early computing. Before vacuum tubes and transistors, relays were used as logical elements in digital computers.
    - Safety-critical logic. Because relays are much more resistant than semiconductors to nuclear radiation, they are widely used in safety-critical logic, such as the control panels of radioactive waste-handling machinery.
- ✓ Time delay functions. Relays can be modified to delay opening or delay closing a set of contacts. A very short (a fraction of a second) delay would use a copper disk between the armature and moving blade assembly. Current flowing in the disk maintains magnetic field for a short time, lengthening release time. For a slightly longer (up to a minute) delay,

a dashpot is used. A dashpot is a piston filled with fluid that is allowed to escape slowly. The time period can be varied by increasing or decreasing the flow rate. For longer time periods, a mechanical clockwork timer is installed.

3.8.4. Relay application considerations:

Selection of an appropriate relay for a particular application requires evaluation of many different factors:

- ✓ Number and type of contacts – normally open, normally closed, (double-throw)
- ✓ Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make-before-break so that the connection didn't get dropped while dialling the number.
- ✓ Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current
- ✓ Voltage rating of contacts – typical control relays rated 300 VAC or 600 VAC, automotive types to 50 VDC, special high-voltage relays to about 15 000 V
- ✓ Coil voltage – machine-tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milli amperes
- ✓ Coil current
- ✓ Package/enclosure – open, touch-safe, double-voltage for isolation between circuits, explosion proof, outdoor, oil and splash resistant, washable for printed circuit board assembly
- ✓ Assembly – Some relays feature a sticker that keeps the enclosure sealed to allow PCB post soldering cleaning, which is removed once assembly is complete.
- ✓ Mounting – sockets, plug board, rail mount, panel mount, through-panel mount, enclosure for mounting on walls or equipment
- ✓ Switching time – where high speed is required

- ✓ "Dry" contacts – when switching very low level signals, special contact materials may be needed such as gold-plated contacts
- ✓ Contact protection – suppress arcing in very inductive circuits
- ✓ Coil protection – suppress the surge voltage produced when switching the coil current
- ✓ Isolation between coil circuit and contacts
- ✓ Aerospace or radiation-resistant testing, special quality assurance
- ✓ Expected mechanical loads due to acceleration – some relays used in aerospace applications are designed to function in shock loads of 50*g* or more
- ✓ Accessories such as timers, auxiliary contacts, pilot lamps, test buttons
- ✓ Regulatory approvals
- ✓ Stray magnetic linkage between coils of adjacent relays on a printed circuit board.

3.8.5. Contactors:

A contactor is an electrically controlled switch used for switching a power circuit, similar to a relay except with higher current ratings. A contactor is controlled by a circuit which has a much lower power level than the switched circuit. Contactors come in many forms with varying capacities and features. Unlike a circuit breaker, a contactor is not intended to interrupt a short circuit current.

Contactors range from those having a breaking current of several amps and 24 V DC to thousands of amps and many kilovolts. The physical size of contactors ranges from a device small enough to pick up with one hand, to large devices approximately a meter (yard) on a side.

Contactors are used to control electric motors, lighting, heating, capacitor banks, and other electrical loads.

3.8.6. Construction:

A contactor is composed of three different items. The contacts are the current carrying part of the contactor. This includes power contacts, auxiliary contacts, and contact springs. The electromagnet provides the driving force to close the contacts. The enclosure is a frame housing the contact and the electromagnet. Enclosures are made of insulating materials like Bakelite, Nylon 6, and thermosetting plastics to protect and insulate the contacts and to provide some measure of protection against personnel touching the contacts. Open-frame contactors may have a further enclosure to protect against dust, oil, explosion hazards and weather. High voltage contactors (greater than 1000 volts) may use vacuum or an inert gas around the contacts.

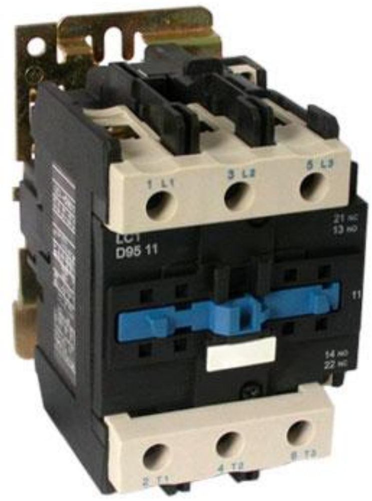


Fig. 3.9

Magnetic blowouts use blowout coils to lengthen and move the electric arc. These are especially useful in DC power circuits. AC arcs have periods of low current, during which the arc can be extinguished with relative ease, but DC arcs have continuous high current, so blowing them out requires the arc to be stretched further than an AC arc of the same current. The magnetic blowouts in the pictured Albright contactor (which is designed for DC currents) more than double the current it can break, increasing it from 600 A to 1,500 A.

Sometimes an economizer circuit is also installed to reduce the power required to keep a contactor closed; an auxiliary contact reduces coil current after the contactor closes. A somewhat greater amount of power is required to initially close a contactor than is required to keep it closed. Such a circuit can save a substantial amount of power and allow the energized coil to stay cooler. Economizer circuits are nearly always applied on direct-current contactor coils and on large alternating current contactor coils.

A basic contactor will have a coil input (which may be driven by either an AC or DC supply depending on the contactor design). The coil may be energized at the same voltage as the motor, or may be separately controlled with a lower coil voltage better suited to control by programmable controllers and lower-voltage pilot devices. Certain contactors have series coils connected in the motor circuit; these are used, for example, for automatic acceleration control, where the next stage of resistance is not cut out until the motor current has dropped

3.8.7. Contactors – Operating principle:

Unlike general-purpose relays, contactors are designed to be directly connected to high-current load devices. Relays tend to be of lower capacity and are usually designed for both normally closed and normally open applications. Devices switching more than 15 amperes or in circuits rated more than a few kilowatts are usually called contactors. Apart from optional auxiliary low current contacts, contactors are almost exclusively fitted with normally open contacts. Unlike relays, contactors are designed with features to control and suppress the arc produced when interrupting heavy motor currents.

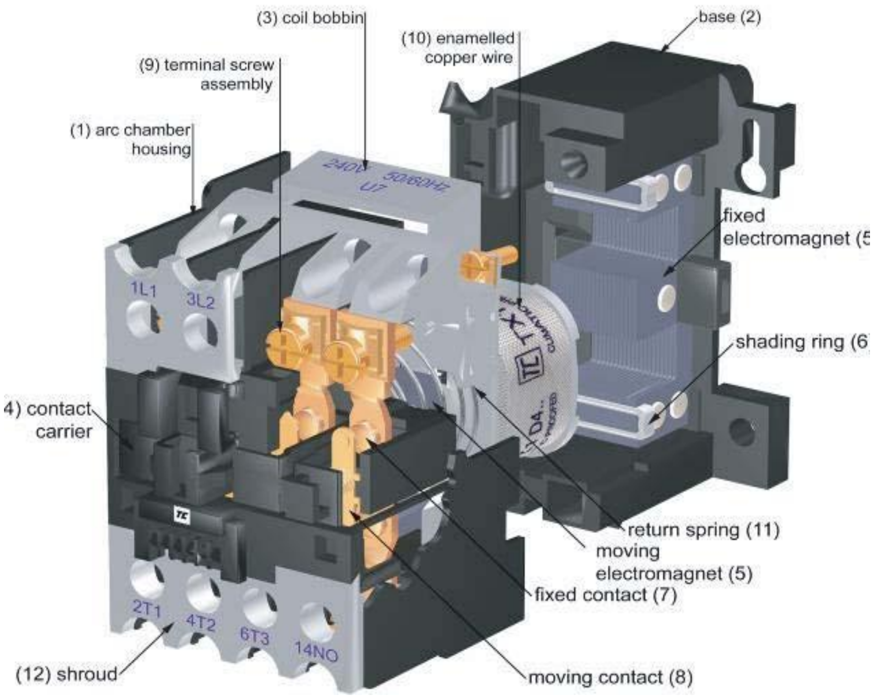


Fig. 3.10

When current passes through the electromagnet, a magnetic field is produced; this attracts the moving core of the contactor. The electromagnet coil draws more current initially, until its inductance increases when the metal core enters the coil. The moving contact is propelled by the moving core; the force developed by the electromagnet holds the moving and fixed contacts together. When the contactor coil is de-energized, gravity or a spring returns the electromagnet core to its initial position and opens the contacts.

For contactors energized with alternating current, a small part of the core is surrounded with a shading coil, which slightly delays the magnetic flux in the core. The effect is to average out the alternating pull of the magnetic field and so prevent the core from buzzing at twice line frequency.

Most motor control contactors at low voltages (600 volts and less) are air break contactors; i.e., ordinary air surrounds the contacts and extinguishes the arc when interrupting the circuit. Modern medium-voltage motor controllers use vacuum contactors.

Motor control contactors can be fitted with short-circuit protection (fuses or circuit breakers), disconnecting means, overload relays and an enclosure to make a combination starter.

3.8.8. Contactors – Ratings:

Contactors are rated by designed load current per contact (pole), maximum fault withstand current, duty cycle, voltage, and coil voltage. A general purpose motor control contactor may be suitable for heavy starting duty on large motors; so-called "definite purpose" contactors are carefully adapted to such applications as air-conditioning compressor motor starting. North American and European ratings for contactors follow different philosophies, with North American general purpose machine tool contactors generally emphasizing simplicity of application while definite purpose and European rating philosophy emphasizes design for the intended life cycle of the application.

Current rating of the contactor depends on utilization category. For example IEC Categories are described as:

- **AC1** - Non-inductive or slightly inductive rows
- **AC2** - Starting of slip-ring motors

- **AC3** - Starting of squirrel-cage motors and switching off only after the motor is up to speed. (Make Locked Rotor Amps (LRA), Break Full Load Amps (FLA))
- **AC4** - Starting of squirrel-cage motors with inching and plugging duty. Rapid Start/Stop. (Make and Break LRA)
- **AC11** - Auxiliary (control) circuits

3.8.9. Applications:

Contactors are often used to provide central control of large lighting installations, such as an office building or retail building. To reduce power consumption in the contactor coils, latching contactors are used, which have two operating coils. One coil, momentarily energized, closes the power circuit contacts, which are then mechanically held closed; the second coil opens the contacts.

A magnetic starter is a contactor designed to provide power to electric motors. The magnetic starter has an overload relay, which will open the control voltage to the starter coil if it detects an overload on a motor. Overload relays may rely on heat produced by the motor current to operate a bimetal contact or release a contact held closed by a low-melting-point alloy. The overload relay opens a set of contacts that are wired in series with the supply to the contactor feeding the motor. The characteristics of the heaters can be matched to the motor so that the motor is protected against overload. Recently, microprocessor-controlled motor digital protective relays offer more comprehensive protection of motors.

3.9. Voltage regulator – L7805:

The 78xx is a family of self-contained fixed linear voltage regulator integrated circuits. The 78xx family is commonly used in electronic circuits requiring a regulated power supply due to their ease-of-use and low cost. For ICs within the family, the *xx* is replaced with two digits, indicating the output voltage (for example, the 7805 has a 5 volt output, while the 7812 produces 12 volts). The 78xx lines are positive voltage regulators: they produce a voltage that is positive relative to a common ground. There is a related line of 79xx devices which are complementary negative voltage regulators. 78xx and 79xx ICs can be used in combination to provide positive and negative supply voltages in the same circuit.

78xx ICs have three terminals and are commonly found in the TO220 form factor, although smaller surface-mount and larger TO3 packages are available. These devices support an input voltage anywhere from a couple of volts over the intended output voltage, up to a maximum of 35 or 40 volts, and typically provide 1 or 1.5 amps of current (though smaller or larger packages may have a lower or higher current rating).

Fig. 3.11

3.9.1. Pin diagram:

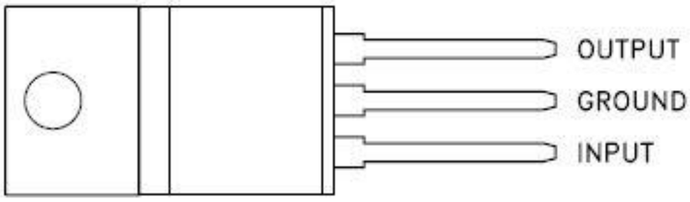


Fig. 3.12

3.9.2. Circuit connection:

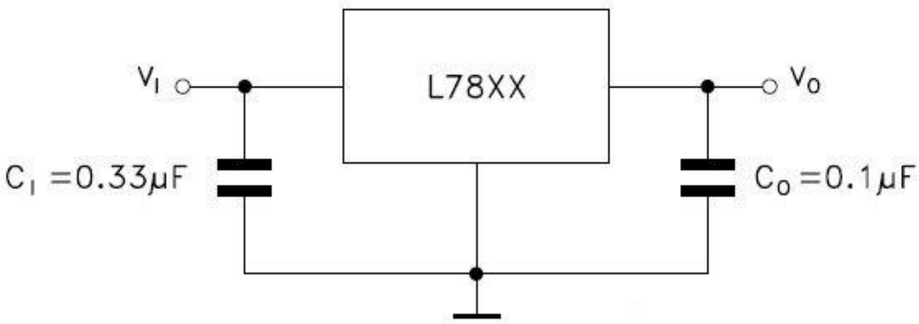


Fig. 3.13

3.9.3. Applications:

- 78xx series ICs do not require additional components to provide a constant, regulated source of power, making them easy to use, as well as economical and efficient uses of space. Other voltage regulators may require additional components to set the output voltage level, or to assist in the regulation process. Some other designs (such as a switching power supply) may need substantial engineering expertise to implement.
- 78xx series ICs have built-in protection against a circuit drawing too much power. They have protection against overheating and short-circuits, making them quite robust in most applications. In some cases, the current-limiting features of the 78xx devices can provide protection not only for the 78xx itself, but also for other parts of the circuit.

3.10. Capacitors, Trimpots, LED:

3.10.1. Capacitors:

A capacitor (formerly known as condenser) is a device for storing electric charge. The forms of practical capacitors vary widely, but all contain at least two conductors separated by a non-conductor. Capacitors used as parts of electrical systems, for example, consist of metal foils separated by a layer of insulating film.

Capacitors are widely used in electronic circuits for blocking direct current while allowing alternating current to pass, in filter networks, for smoothing the output of power supplies, in the resonant circuits that tune radios to particular frequencies and for many other purposes.

A capacitor is a passive electronic component consisting of a pair of conductors separated by a dielectric (insulator). When there is a potential difference (voltage) across the conductors, a static electric field develops in the dielectric that stores energy and produces a mechanical force between the conductors. An ideal capacitor is characterized by a single constant value, capacitance, measured in farads. This is the ratio of the electric charge on each conductor to the potential difference between them.

The capacitance is greatest when there is a narrow separation between large areas of conductor; hence capacitor conductors are often called "plates", referring to an early means of construction. In practice the dielectric between the plates passes a small amount of leakage current and also has an electric field strength limit, resulting in a breakdown voltage, while the conductors and leads introduce an undesired inductance and resistance.

3.10.2. Power conditioning capacitors:

Reservoir capacitors are used in power supplies where they smooth the output of a full or half wave rectifier. They can also be used in charge circuits as the energy storage element in the generation of higher voltages than the input voltage.

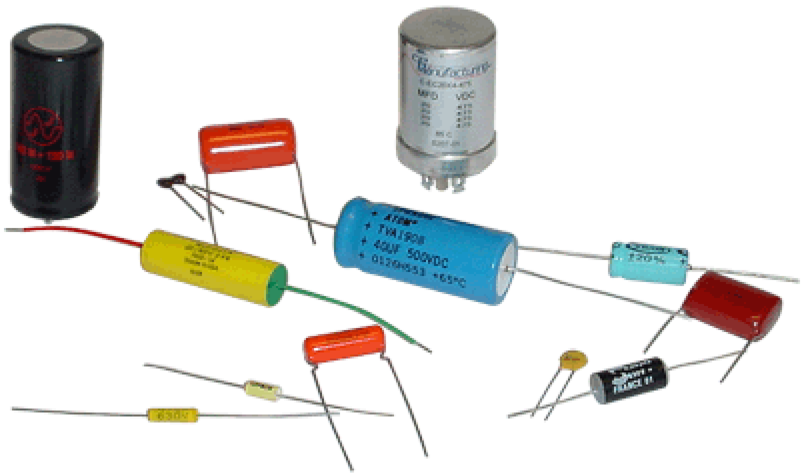


Fig. 3.14

Capacitors are connected in parallel with the power circuits of most electronic devices and larger systems (such as factories) to shunt away and conceal current fluctuations from the primary power source to provide a "clean" power supply for signal or control circuits. Audio equipment, for example, uses several capacitors in this way, to shunt away power line hum before it gets into the signal circuitry. The capacitors act as a local reserve for the DC power source, and bypass AC currents from the power supply. This is used in car audio applications, when a stiffening capacitor compensates for the inductance and resistance of the leads to the lead-acid car battery.

3.10.3. Power factor correction:

In electric power distribution, capacitors are used for power factor correction. Such capacitors often come as three capacitors connected as a three phase load. Usually, the values of these capacitors are given not in farads but rather as a reactive power in volt-amperes reactive (VAR). The purpose is to counteract inductive loading from devices like electric motors and transmission lines to make the load appear to be mostly resistive. Individual motor or lamp loads may have capacitors for power factor correction, or larger sets of capacitors (usually with automatic switching devices) may be installed at a load center within a building or in a large utility substation.

Fig. 3.15

3.10.4. Trimpots:

These are miniature versions of the standard variable resistor. They are designed to be mounted directly onto the circuit board and adjusted only when the circuit is built. For example, to set the frequency of an alarm tone or the sensitivity of a light-sensitive circuit. A small screwdriver or similar tool is required to adjust presets.

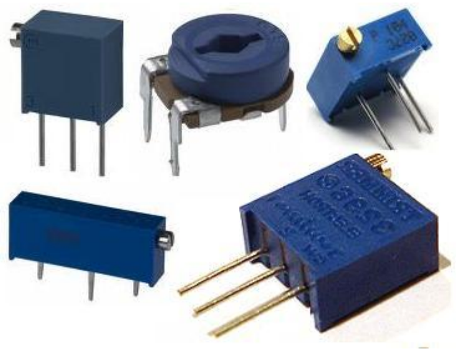


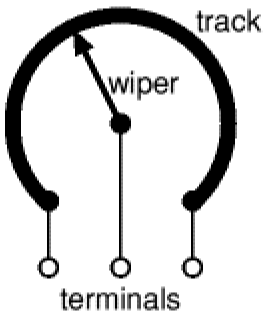
Fig. 3.16

Presets/Trimpots are much cheaper than standard variable resistors so they are sometimes used in projects where a standard variable resistor would normally be used.

Multi-turn presets are used where very precise adjustments must be made. The screw must be turned many times (10+) to move the slider from one end of the track to the other, giving very fine control.

3.10.5. Trimpots – Construction:

Variable resistors consist of a resistance track with connections at both ends and a wiper which moves along the track as you turn the spindle. The track may be made from carbon, cermets (ceramic and metal mixture) or a coil of wire (for low resistances). The track is usually rotary but straight track versions, usually called sliders, are also available.



Variable resistors may be used as a rheostat with two connections (the wiper and just one end of the track) or as a potentiometer with all three connections in use. Miniature versions called presets are made for setting up circuits which will not require further adjustment.

Variable resistors are often called potentiometers in books and catalogues. They are specified by their maximum resistance, linear or logarithmic track, and their physical size. The standard spindle diameter is 6mm.

The resistance and type of track are marked on the body:

4K7 LIN means 4.7 kΩ linear track.

1M LOG means 1 MΩ logarithmic track.

Some variable resistors are designed to be mounted directly on the circuit board, but most are for mounting through a hole drilled in the case containing the circuit with stranded wire connecting their terminals to the circuit board.

3.10.6. Linear and Logarithmic tracks:

Linear (LIN) track means that the resistance changes at a constant rate as you move the wiper. This is the standard arrangement and you should assume this type is required if a project does not specify the type of track. Presets always have linear tracks.

Logarithmic (LOG) track means that the resistance changes slowly at one end of the track and rapidly at the other end, so halfway along the track is not half the total resistance! This arrangement is used for volume (loudness) controls because the human ear has a logarithmic response to loudness so fine control (slow change) is required at low volumes and coarser control (rapid change) at high volumes. It is important to connect the ends of the track the correct way round, if you find that turning the spindle increases the volume rapidly followed by little further change you should swap the connections to the ends of the track.

3.10.7. Light Emitting Diodes (LEDs):

LEDs emit light when an electric current passes through them. LEDs must be connected the correct way round, the diagram may be labeled a or + for anode and k or - for cathode. The cathode is the short lead and there may be a slight flat on the body of round LEDs. If you can see inside the LED the cathode is the larger electrode (but this is not an official identification method).

LEDs can be damaged by heat when soldering, but the risk is small unless you are very slow. No special precautions are needed for soldering most LEDs

We should never connect an LED directly to a battery or power supply. It will be destroyed almost instantly because too much current will pass through and burn it out.

LEDs must have a resistor in series to limit the current to a safe value, for quick testing purposes a 1kΩ resistor is suitable for most LEDs if supply voltage is 12V or less.

LEDs are available in red, orange, amber, yellow, green, blue and white colors. Blue and white LEDs are much more expensive than the other colors. The color of an LED is determined by the semiconductor material, not by the coloring of the 'package' (the plastic body). LEDs of all colors are available in uncolored packages which may be diffused (milky) or clear (often described as 'water clear'). The colored packages are also available as diffused (the standard type) or transparent.

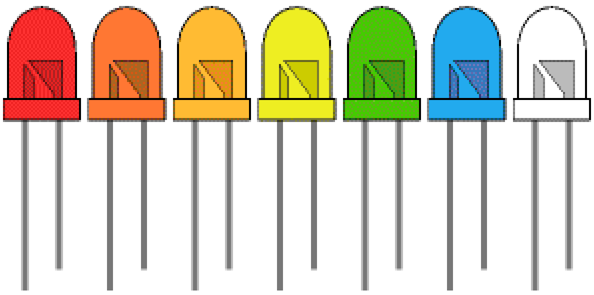


Fig. 3.17

3.10.8. Calculating an LED resistor value:

An LED must have a resistor connected in series to limit the current through the LED; otherwise it will burn out almost instantly.

The resistor value, R is given by:

$$R = \frac{V_S - V_L}{I}$$

V_S = supply voltage

V_L = LED voltage (usually 2V, but 4V for blue and white LEDs)

I = LED current (e.g. 10mA = 0.01A, or 20mA = 0.02A)

Make sure the LED current you choose is less than the maximum permitted and convert the current to amps (A) so the calculation will give the resistor value in ohms (Ω). To convert mA to A divide the current in mA by 1000 because 1mA = 0.001A.

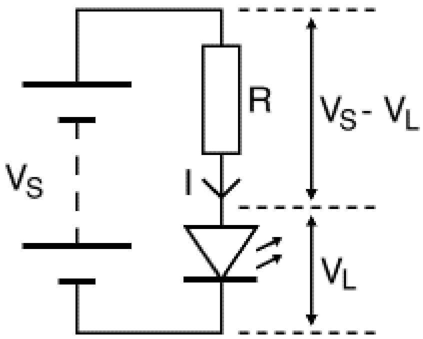


Fig. 3.18 Circuit diagram for calculating an LED resistor value

If the calculated value is not available choose the nearest standard resistor value which is greater, so that the current will be a little less than you chose. In fact you may wish to choose a greater resistor value to reduce the current (to increase battery life for example) but this will make the LED less bright.

3.10.9. Avoid connecting LEDs in parallel:

Connecting several LEDs in parallel with just one resistor shared between them is generally not a good idea.

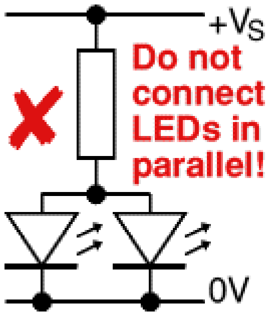


Fig. 3.19

If the LEDs require slightly different voltages, only the lowest voltage LED will light and it may be destroyed by the larger current flowing through it. Although identical LEDs can be successfully connected in parallel with one resistor; this rarely offers any useful benefit because resistors are very cheap and the current used is the same as connecting the LEDs individually. If LEDs are in parallel each one should have its own resistor.

3.11. H – Bridge – L293D:

3.11.1. Description:

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. L293 and L293D are characterized for operation from 0°C to 70°C.

3.11.2. Pin Diagram:

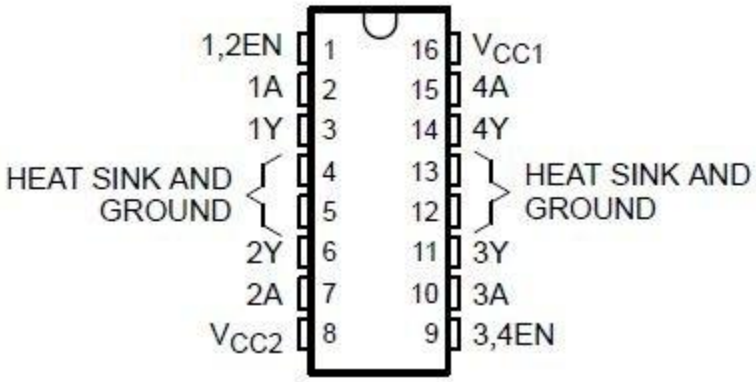


Fig. 3.20

3.11.3. Pin description:

V_{cc1}	Logical voltage supply for a 1. For example, if you connect it to a 5V supply, 5 volts into any of the INPUTs would mean a logical 1. However, if you connect it to a 36V supply, the same 5 volts into any INPUT would mean a logical 0.
V_{cc2}	Actual voltage that needs to be output. This has nothing to do with the logical 0s and 1s.
HEAT SINK AND GROUND	Ground and Heat sink. Heat sink is connected to limit the heat rate when high output current flows.
1,2EN	Enable/disable the corresponding sides. Putting a logical 1 into 1,2EN would enable INPUT1/INPUT2 and OUTPUT1/OUTPUT2. A logical 0 disables the corresponding side.
3,4EN	Enable/disable the corresponding sides. Putting a logical 1 into 3,4EN would enable INPUT3/INPUT4 and OUTPUT3/OUTPUT4. A logical 0 disables the corresponding

	side.

3.11.4. Schematics of Inputs and Outputs:

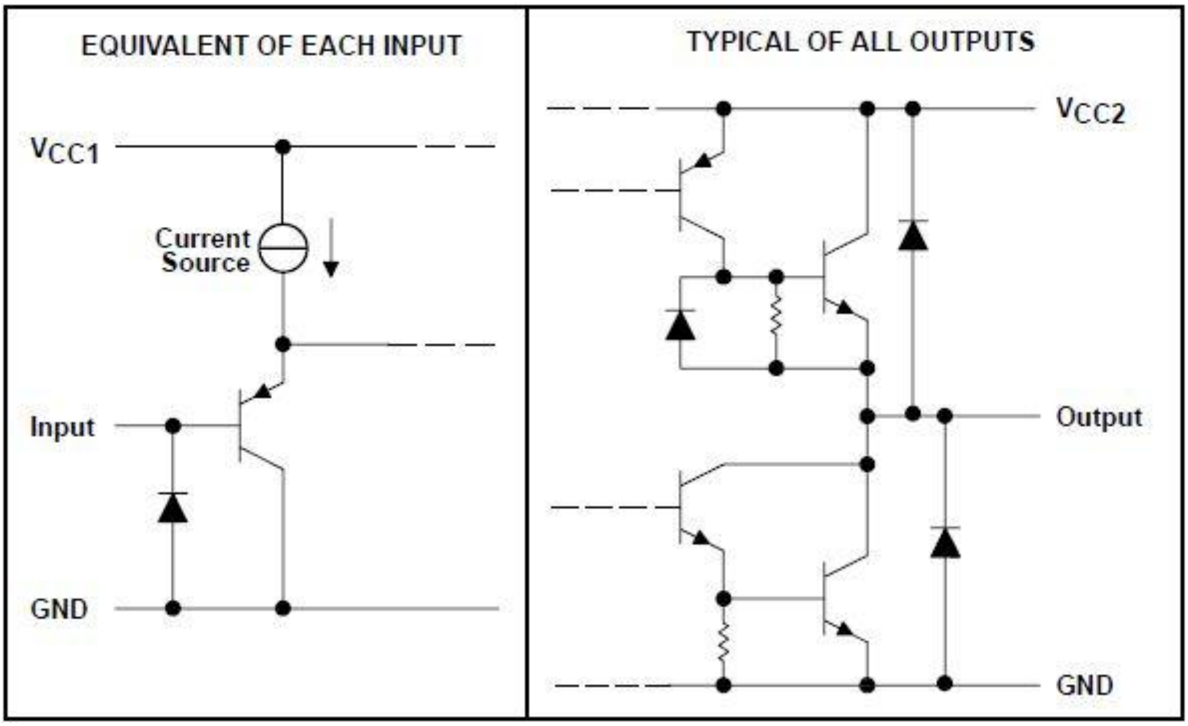


Fig. 3.21

3.11.5. Notice the D..!!

The name of the chip has a D in it… notice that? That indicates the presence of a diode for each OUTPUTx pin.

Whenever the direction of current changes, the device connected across the OUTPUT pins will resist the change. And this results in a back current. The diodes make sure that no back-current damages the circuit inside the chip or before the chip.

One could do without the diodes too (using an L293B in that case). But it is recommended that use the one with diodes.

3.11.6. Features:

- ✓ Wide Supply-Voltage Range: 4.5 V to 36 V
- ✓ Separate Input-Logic Supply
- ✓ Internal ESD Protection
- ✓ Thermal Shutdown
- ✓ High-Noise-Immunity Inputs

- ✓ Functional Replacements for SGS L293 and SGS L293D
- ✓ Output Current 1 A per Channel (600 mA for L293D)
- ✓ Peak Output Current 2 A per Channel (1.2 A for L293D)
- ✓ Output Clamp Diodes for Inductive Transient Suppression (L293D)

3.12. Parallel Port:

A parallel port is a type of interface found on computers (personal and otherwise) for connecting various peripherals. In computing, a parallel port is a parallel communication physical interface. It is also known as a printer port or Centronics port (this name came from a popular printer manufacturing company 'Centronics' who devised some standards for parallel port). The IEEE 1284 standard defines the bi-directional version of the port, which allows the transmission and reception of data bits at the same time.

Fig. 3.22

Before the advent of USB, the parallel interface was adapted to access a number of peripheral devices other than printers. Probably one of the earliest devices to use parallel was dongles used as a hardware key form of software copy protection. Zip drives and scanners were early implementations followed by external modems, sound cards, webcams, gamepads, joysticks and external hard disk drives and CD-ROM drives. Adapters were available to run SCSI devices via parallel. Other devices such as EPROM programmers and hardware controllers could be connected parallel.

3.12.1. Pin diagram:

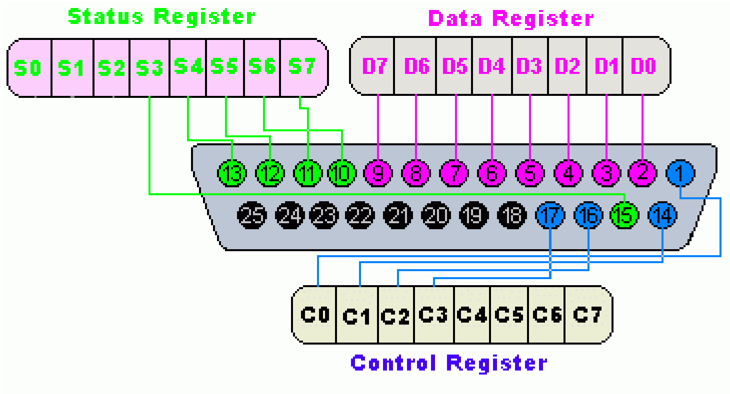


Fig. 3.23

The lines in DB25 connector are divided in to three groups, they are

1) Data lines (data bus)
2) Control lines
3) Status lines

As the name refers, data is transferred over data lines, Control lines are used to control the peripheral and of course, the peripheral returns status signals back computer through Status lines. These lines are connected to Data, Control and Status registers internally. The details of parallel port signal lines are given below

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No

17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

3.12.2. Parallel port Registers:

As we know, the Data, Control and status lines are connected to their corresponding registers inside the computer. So by manipulating these registers in program , one can easily read or write to parallel port with programming languages like 'C' and BASIC.

The registers found in standard parallel port are,

1) Data register
2) Status register
3) Control register

As their names specifies, Data register is connected to Data lines, Control register is connected to control lines and Status register is connected to Status lines. (Here the word connection does not mean that there is some physical connection between data/control/status lines. The registers are virtually connected to the corresponding lines.). So whatever we write to these registers, will appear in corresponding lines as voltages, Of course, we can measure it with a multimeter. And whatever we give to Parallel port as voltages can be read from these registers (with some restrictions). For example, if we write '1' to Data register, the line Data0 will be driven to +5v. Just like this, we can programmatically turn on and off any of the data lines and Control lines.

3.13. Studs, Connectors and connecting wires:

3.13.1. Studs:

Studs are used to join or connect two or more electrical wires safely, strongly at a point to ensure protection from bare exposures of conductors. A stud consists of a screw embedded in a plastic case and nuts to fix it on a wooden plank or metal sheet for stability. Two or more wires to be connected are wound to the screw of stud and closed with a threaded cap on screw.

Studs are available in different sizes depending on the size of the current carrying conductors/ wires. Studs are placed in some parts/ points of the circuit where a temporary change of wiring involves. Studs are widely used in circuits which interface electrical motors.

3.13.2. Connectors:

The standard battery clip fits a 9V PP3 battery and many battery holders such as the 6 × AA cell holder shown. Battery holders are also available with wires attached, with pins for PCB mounting, or as a complete box with lid, switch and wires.

Electrical and electronic components and devices sometimes have plug and socket connectors or terminal blocks, but individual screw terminals and fast-on or quick-disconnect terminals are more common. Small components have bare lead wires for soldering. They are manufactured using casting.

A blade connector is a type of single wire connection using a flat blade which is inserted into a blade receptacle. Usually both blade connector and blade receptacle have wires attached to them either through soldering of the wire to the blade or crimping of the blade to the wire. In some cases the blade is a manufactured part of a component (such as a switch or a speaker unit) and a blade receptacle is pushed onto the blade to form a connection.

A common type of blade connector is the "Faston". While Faston is a trademark of Tyco Electronics, it has come into common usage. Faston connectors come in male and female types. They have been commonly used since 1970s.

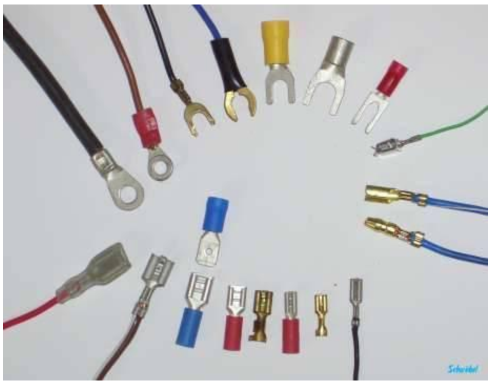


Fig. 3.24 (a)

The connectors in the top row of the image are known as ring terminals and spade terminals (sometimes called split ring terminals). Electrical contact is made by passing a screw or bolt through them. The spade terminal form factor facilitates connections since the screw or bolt can be left partially screwed in as the spade terminal is removed or attached. Their sizes can be determined by the size of the conducting wire AWG and/or the Screw/Bolt diameter size designation.

Fig. 3.24 (b)

3.13.3. Connecting wires:

Often, when building electronics projects, little thought is given thought is given to the connecting wire. While it is possible to "get away with" almost anything for many projects, it is sometimes necessary to connect the various electronics components using the right wire. For example it is often useful to use colored connecting wire to indicate such items as electronics wire used for connecting the supplies, signals, and grounds. In this way it is easier to identify the different signals and lines and this reduces the possibility of errors. In addition to this it is sometimes necessary to have connecting wire of a particular size to ensure the connections are made in the right manner. If the wire is too thick it may not be easy to accommodate in some situations, whereas thicker wire may be needed for higher currents of physical strength or robustness in other situations.

3.13.4. Types of connecting wires:

Electronics wire for connecting is often categorized by the insulation. The type of insulation is important because it often governs the type of use for which it is suitable.

- **Bare copper wire:** Wire that is not insulated can be used in a variety of ways. It may be used to correct problems on a printed circuit board where insulation may not be a problem. It may also be used in areas where sleeving may be slid over the wire to protect it from causing shorts, or it may be used in areas where it is not possible to cause short circuits. Although it is possible to use bare copper wire, usually it is pre-tinned to enable easier soldering.

- **Enameled copper wire:** This type of copper wire has a form of insulation made from enamel. This is effectively like a varnish over the copper wire. Enameled copper wire is used in applications such as coils where insulation is required but the thickness of the insulation may be an issue. The enamel is not as robust as other forms of insulation so it is not used where it may be scratched or knocked. There are some forms of enameled wire where the insulation or enamel will burn of when being soldered. This enables the wire to be used on circuit boards and removes the need for stripping the wire before connections are made. However care must be taken when using this wire as accidental connections may be made if the insulation is abraded.
- **PVC wire:** PVC wire is the most common form of wire today. Although it is usually termed PVC wire, it should be more correctly termed PVC coated wire as the PVC forms the insulation. The advantage of PVC wire is that it is adequate for most situations and the ends are easy to strip to make connections. However it can melt or in severe cases it can burn when it becomes hot. Also when it is cold, the PVC becomes brittle and can crack if bent.
- **PTFE wire:** PTFE wire is far more expensive than PVC wire. Like PVC wire, it should be more correctly termed PTFE coated wire. The PTFE insulation is more robust, and can be used over a much wider range of temperatures. However it is much more expensive and it is also much more difficult to str52ip to expose the bare copper wire for making connections or soldering.

These are the main types of wire that are used, and although there are some other types, these are by far the most widely used.

3.13.5. AWG/ Metric wire conversion table:

Wire "size" can be measured in a variety of ways. There are three main methods that have been used over the years. SWG or Standard Wire Gauge was widely used some years ago, but now two methods are used. One is AWG or American Wire gauge, and the other is the metric system where wire and cable sizes are measured by their cross sectional area in square millimeters. Where multiple strands are used this can be expressed as the number of strands or a certain wire. As an example seven strands of 0.032 mm wire would be expressed as 7 / 0.032.

For the wires and cables defined under the AWG system, the gauge numbers applied to the wire are for the overall wire whether it is a single strand of a bunch of strands with an equivalent size. This means that the gauge size only determines its overall cross sectional area, and not its physical construction.

The table with approximate equivalents for wires and cables is given below:

Cross sectional area (mm^2)	**AWG**	**Approximate resistance (Ω/km)**
0.032	32	580
0.051	30	350
0.081	28	230
0.128	26	150
0.163	25	110
0.25	23	70
0.32	22	55
0.41	21	44
0.52	20	35
0.75	18	23
1.32	16	15
2.08	14	9
2.5	13	7
4.0	11	4.5

CHAPTER – 4

TESTING

4.1 INTRODUCTION

4.2 TESTING OF CT

4.3 TESTING OF PT

4.4 TESTING OF DELAY

CHAPTER – 4

TESTING

4.1 Introduction

Before connecting CT and PT in the circuit, it is essential to test them. Testing of CT and PT assures us that they are carrying permissible amount of currents and voltages in both primary and secondary at any load. Besides, the main purpose of testing CT and PT in this project is to ensure that they are giving almost sinusoidal current signal (with decreased amplitude) and voltage signal (with decreased amplitude) at their secondary windings. The accuracy of these sinusoidal signals effects on the accuracy of power factor being measured by the microcontroller.

4.2 Testing of CT

Testing of CT is carried out in a different circuit. The circuit mainly consists of a variable R-L load. To be elegant, testing is carried out using single phase supply. As shown in figure 4.1, phase line is passed through the CT (considered as primary of CT) whereas; a low resistance (50Ω) is connected across secondary of CT to avoid high voltages (which is to be shorted ideally). Also, secondary of CT is given to zero crossing detector (not shown here).

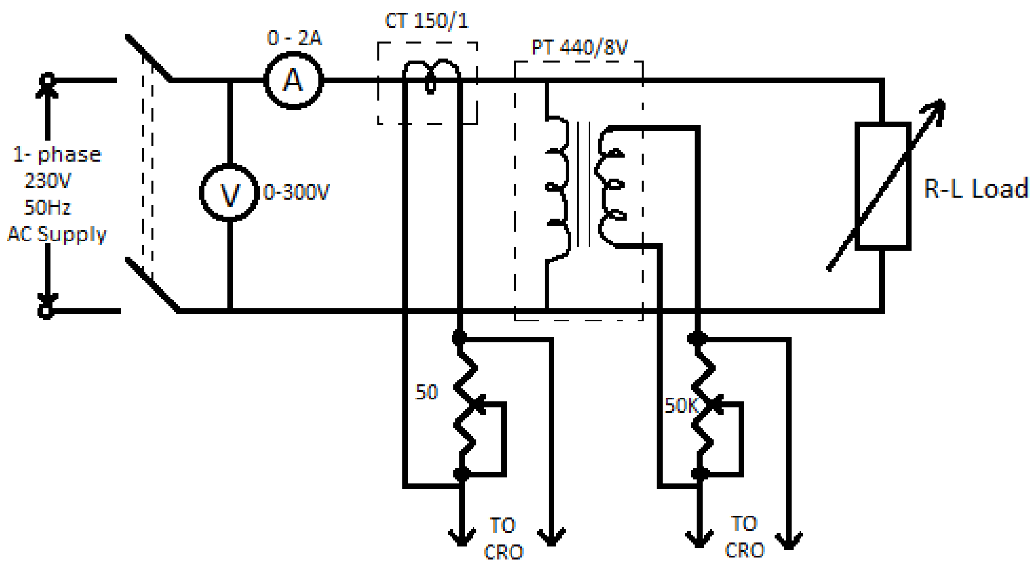


Fig. 4.1

If the waveform in CRO is almost sinusoidal across the secondary of CT, we can ensure that CT is working properly. In the worst case, the waveform may be distorted, which indicates that CT is not designed properly and suffering from harmonics.

Zero crossing of CT is sensed by one of the op-amps in comparator LM324 which acts as zero crossing detector. Secondary of CT is given as input signal to zero crossing detector. Whenever, sinusoidal current signal crosses zero, the output of comparator results in either 0 or 1. This can be clearly observed in digital CRO as shown below.

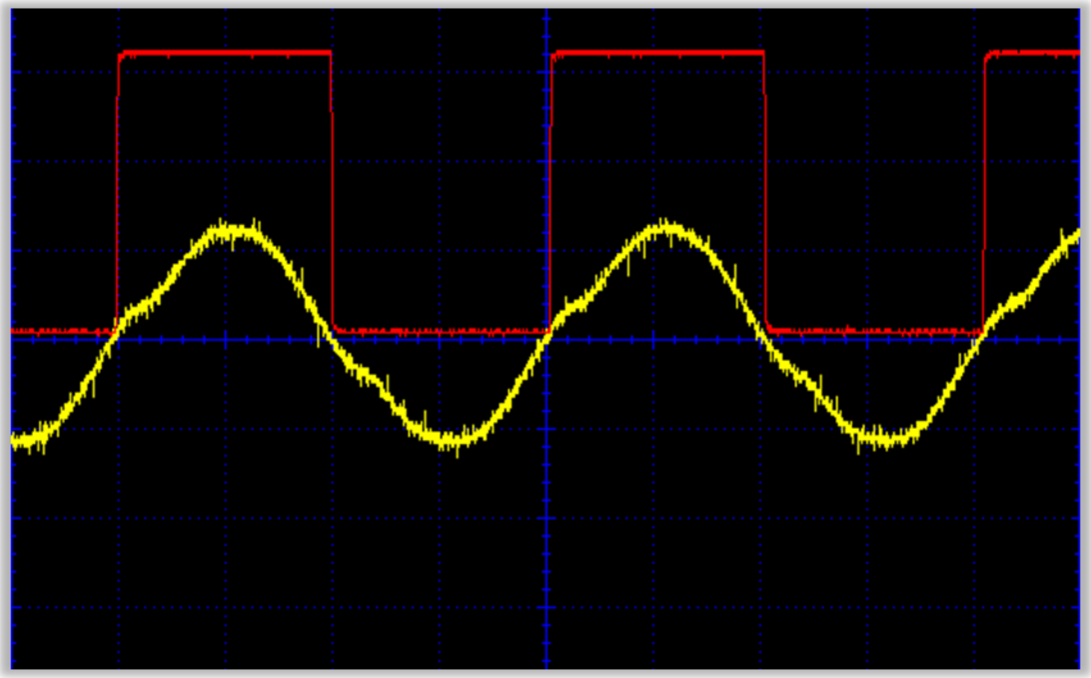


Fig. 4.2

In the above figure, yellow waveform is the current signal and red waveform is the output of zero crossing detector to which secondary of CT is given as input. Output of the comparator switching between 0 and 1 whenever current signal crossing zero level.

4.3 Testing of PT

Testing of PT is carried out in the same circuit as CT tested. The circuit mainly consists of a variable R-L load. To be elegant, testing is carried out using single phase supply. As shown in figure 4.1, primary of PT is connected between phase and neutral of the supply and secondary is connected to the main circuitry. In practical, a high resistance (50KΩ) is connected across the secondary of PT to prevent damages from high currents (which is to be opened ideally). Also, secondary of PT is given as input to another op-amp in comparator LM324 (not shown here).

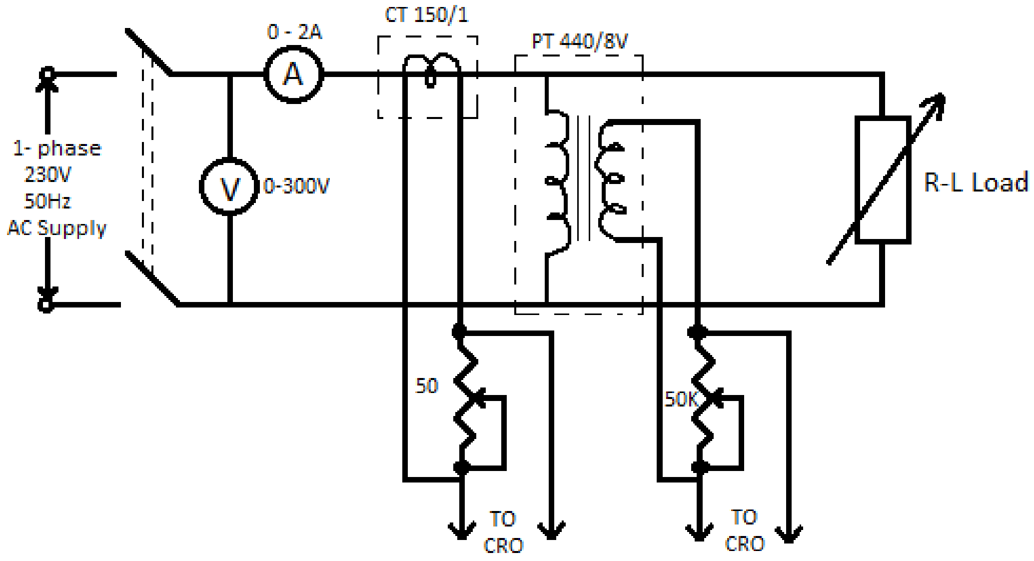


Fig. 4.1

Zero crossing of PT is sensed by another op-amp in comparator LM324 which also acts as zero crossing detector. Secondary of PT is given as input signal to that zero crossing detector. Whenever, sinusoidal voltage signal crosses zero, the output of comparator results in either 0 or 1. This can be clearly observed in digital CRO as shown below.

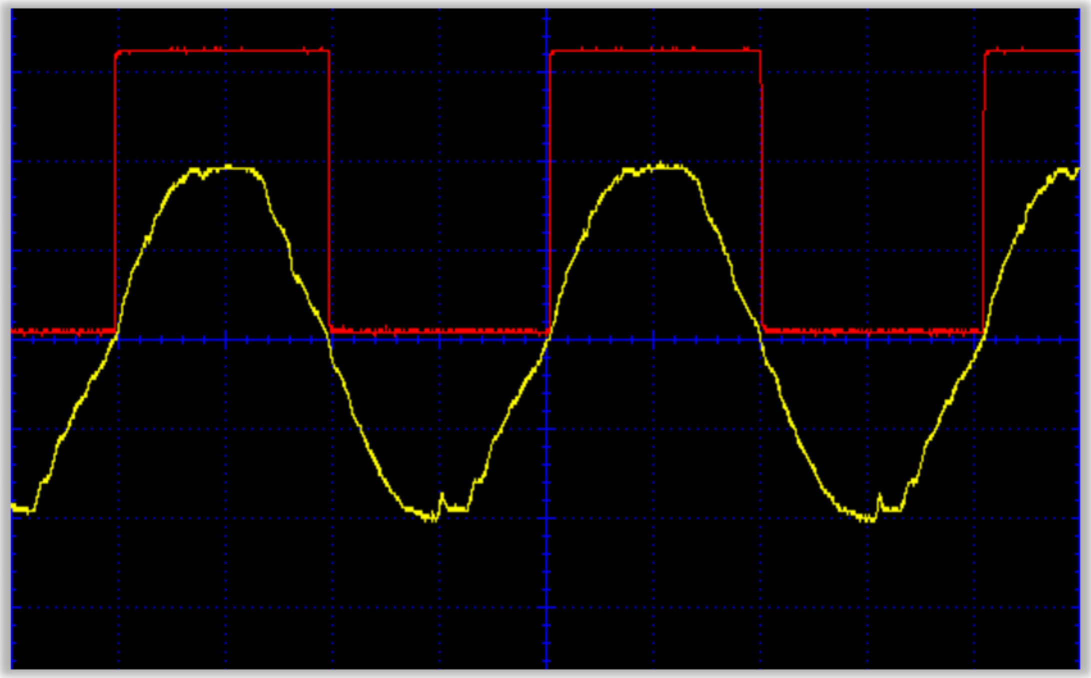


Fig. 4.3

In the above figure, yellow waveform is the voltage signal and red waveform is the output of zero crossing detector to which secondary of PT is given as input. Output of the comparator switching between 0 and 1 whenever voltage signal crossing its zero level.

4.4. Testing of delay

Here, in this context, *delay* is defined as the time difference between the zero crossings of PT and CT at the same instant. To measure power factor of the load, first and foremost thing is to ensure that the comparator is showing a time delay between two zero crossing detectors. Cosine of this delay (in radians) is nothing but power factor of the load.

Testing of delay is also carried out in the same circuit as CT and PT tested. During testing, a digital CRO is kept in dual channel mode to compare two different waveforms of zero crossing detectors as shown below.

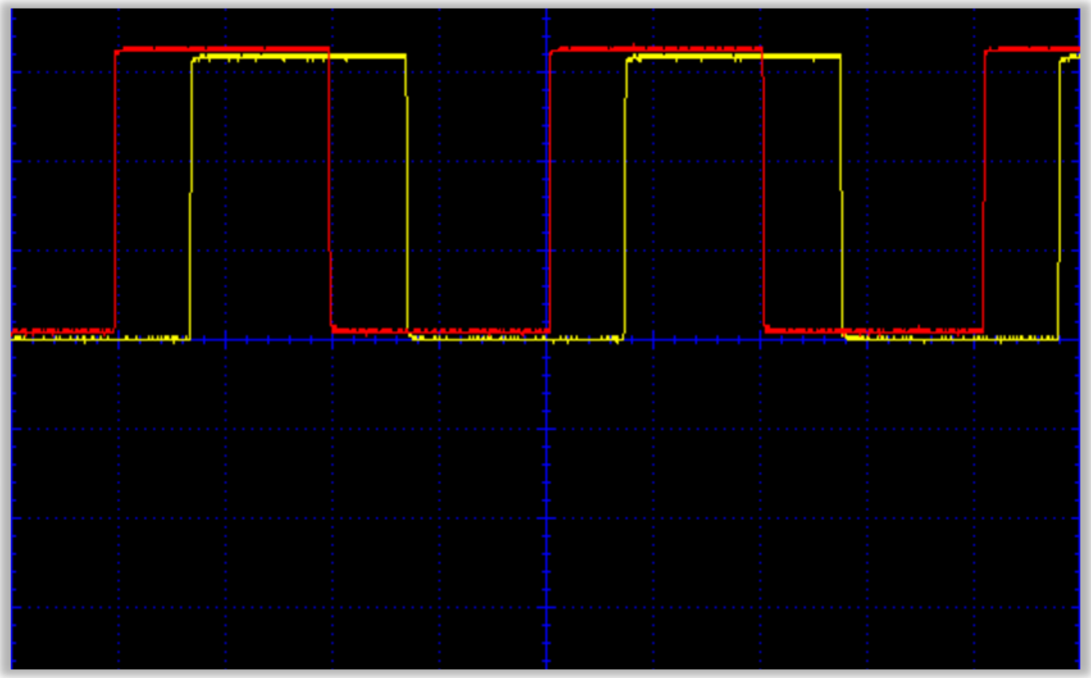


Fig. 4.4

In the above figure, red and yellow waveforms are output of zero crossing detectors of PT and CT respectively. The delay between them is clearly observed in the above figure. This delay can be varied by using variable R-L load. If the resistance of the R-L load is increased, yellow waveform approaches red waveform. Similarly, when resistance of the R-L load is decreased, yellow waveform moves away from the red waveform. In this way, delay between the PT and CT waveforms can be tested.

Authors: Rohit, Sukumar, Krishna, Vinod.
--You can reach me @: rohit.gunnala@yahoo.com for further queries.

CHAPTER – 5

PROGRAMMING

5.1 INTRODUCTION

5.2 FUNDAMENTAL PROGRAM STRUCTURE

5.3 GENERAL ACCESS TO REGISTERS

5.4 ACCESS TO PORTS

5.5 DIGITAL SIGNALS (I/O)

5.6 BURNING A PROGRAM

CHAPTER – 5

PROGRAMMING

5.1 Introduction

In this project, we used WinAVR software to write and burn the program on microcontroller. WinAVR (pronounced "whenever") is a suite of executable, open source software development tools for the Atmel AVR series of RISC microprocessors hosted on the Windows platform. It includes the GNU GCC compiler for C and C++. WinAVR contains all the tools for developing on the AVR. WinAVR comes with Programmers Notepad UI by default. It is very powerful editor and universal open source IDE which supports almost any compiler by using plug-ins.

The use of any high-level language keeps the programmer at a distance from the actual microcontroller hardware. The C compiler takes care of the details of the software like register allocation, memory access, and the like. This allows the programmer to concentrate on the details of the software and not the intricate details of the specific microcontroller. This separation from the hardware also allows the software to be portable (movable to other microcontrollers) with a minimal amount of effort.

The C programming language has several advantages associated with it when compared to assembly or other high level languages. These advantages make the C language well suited for use with microcontrollers. Once the C software has been written, it must go through a process to create the actual software that is loaded into the microcontroller. The creation of software should also follow a set of standards. Standards aid in the readability and maintainability of software.

C for microcontrollers has several unique features not found in standard C implementations, so even experienced C programmers have to spend some time on the learning curve. The major difference between a program written for a PC and a program written for an embedded microcontroller is that the PC program will most likely end at some point and turn control back to an underlying operating system. Embedded systems do not have an operating system and therefore, embedded applications never end. Embedded applications are not allowed to run out of things to do.

5.2 Fundamental program structure

Mainly there are two types of methods in writing a program for a microcontroller. These two methods are referred to as fundamental program structures, they are:

1. Sequential program sequence
2. Interrupt controlled program sequence

5.2.1 Sequential program sequence

With this programming techniques a continuous loop is programmed, which has essentially always the same structure:

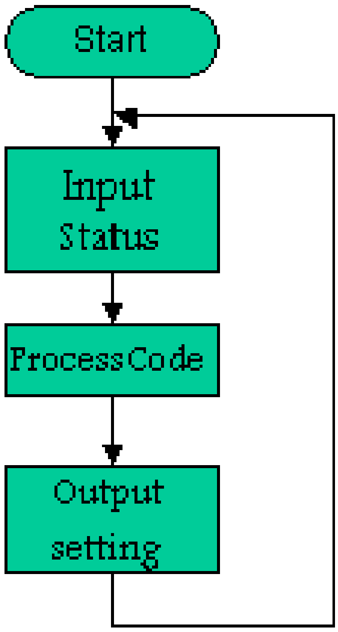


Fig. 5.1

5.2.2 Interrupt controlled program sequence

With this method of programming technique, first the desired sources of interrupt are activated and then enter into a continuous loop, in which things to be settled accordingly, which is not time-critical. If an interrupt is released automatically the assigned interrupt function is in such a way implemented.

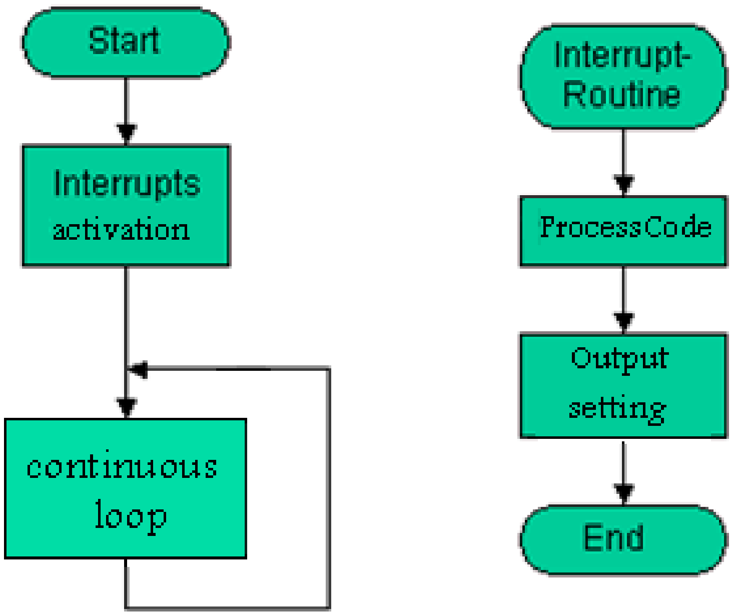


Fig. 5.2

5.3 General access to registers

The AVR microcontrollers have a multiplicity of registers. Most of it is so-called write/read registers. That is, the program can pick out and describe contents of the registers both. Some registers have special functions; others can be used for general purposes (memory of data values). Individual registers are present with all AVRs, others again only with certain types. So are for example the registers, which are necessary for the access to the UART naturally only with those models available, which have integrated hardware a UART and/or a USART.

The names of the registers are defined in the header files to the appropriate AVR types. If in the Makefile the MCU type is defined, by the system to types fitting definition file is used automatically, as soon as one merges the general “io.h” header file in the code.

```
#include <avr/io.h>
```

If the MCU type is defined e.g. with the contents of atmega16 in the Makefile, when reading the io.h file in implicitly (“automatically”) also the iom16.h-file with the register definitions for the ATmega16 is read in.

5.3.1 I/O Registers

The I/O registers have a special value with the AVR control-learn. They serve the access to Port and the interfaces of the microcontroller. In AVR, there are 8 – bit as well as 16 – bit registers. For the time being we treat times the 8-bits registers. Each AVR implements a different quantity of GPIO registers (GPIO - General Purpose Input/ Output). The following examples proceed from an AVR, which possesses both Port A and Port B. Accordingly they must be adapted for other AVRs (for example ATmega8/16/48/88/168).

5.3.2 Read an I/O register

For reading, one can access I/O registers simply as a variable. In source codes, which were developed for older versions of the avr GCC/avr lib.c, the read access is made by the function inp (). Current versions of the compiler enables the access now directly and inp () is no longer necessary.

Example – 5.1:

```
#include <avr/io.h>
#include <stdint.h>
uint8_t foo;
//...
int main(void)
{
/* copies the status of the input pins at PortB
Into the variable foo: */
Foo = PINB;
//...
}
```

5.3.3 Read a bit

The AVR library (avr lib.c) also places functions to the inquiry of an individual bit of a register to the order:

bit_is_set (<Register>, <Bitnumber>)

The function *bit_is_set* examines whether a bit is set. If the bit is set, a value is returned not equal 0.

Exactly taken it is the priority of the queried bit, thus 1 for Bit0, 2 for Bit1, 4 for Bit2 etc.

bit_is_clear (<Register>, <Bitnumber>)

The function *bit_is_clear* examines whether a bit is deleted. If the bit is deleted, thus on 0, a value is not equal 0 is returned.

The functions **bit_is_clear** and/or **bit_is_set** are not *necessary*; one can also use "simple" C-syntax, which is universally used. ***bit_is_set*** e.g. corresponds thereby (register name & (1 << bit number)). The result is set to <>0 ("truly") if the bit and 0 ("wrongly") if it is not set. The denial of the expression, thus**!** (Register name & (1<< bit number)), corresponds ***bit_is_clear*** and returns a value to <>0 ("truly"), if the bit is not set.

5.3.4 Write an I/O register

To Write, one can set I/O registers simply like a variable. In source codes, which were developed for older versions of the avr GCC/avr lib.c, the write access is made by the function outp (). Current versions of the compiler support the access now directly and outp () is no longer necessary.

Example – 5.2:

```
#include <avr/io.h>
...
int main(void)
{
{/* sets the direction register of the PORTA on 0xff
(all pins as output): */
DDRA = 0xFF;
/* PortA sets remaining on 0x03, bit 0 and 1 "high", "low": */
PORTA = 0x03;
...
}
```

5.3.5 Write of bits

Individual bits (standard C conformal) can be set/ reset by means of more logical (bit) operations.

With the expression:

```
X |= (1 << Bitnumber); // a bit in x is set
X &= ~(1 << Bitnumber) ; // a bit is deleted in x
```

The least significant bit (for 1) of a byte has the bit number 0, the "most significant" (for 128) the number 7.

Example – 5.3:

```
#include <avr/io.h>
#define MyBIT 2
PORTA |= (1 << MyBIT); /* sets bit 2 at PortA on 1 * /
PORTA &= ~(1 << MyBIT); /* deletes bit 2 at PortA */
```

With the method shown below, several bits of a register can set/ reset simultaneously.

Example – 5.4:

```
#include <avr/io.h>
…
DDRA &= ~( (1<<PA0) | (1<<PA3) ); /* PA0 and PA3 as inputs */
PORTA |= (1<<PA0) | (1<<PA3); /* internal Pull UP for both switch on */
```

In source codes, which were developed for older version that of the avr GCC/avr lib.c, individual bits are set and/or reset by means of the functions **sbi** and **cbi**. Both functions are obsolete.

5.4 Access to ports

The entire Ports of AVR microcontroller is steered via registers. In total, there are 3 main registers viz. DDR, PIN, PORT

DDRx	Data directions register for PORTx x corresponds to A, B, C, D etc. (dependent on the amount of the Port of the used AVR). Bit in the register set (1) for output, bit deleted (0) for input.
PINx	Input address for PORTx

	Condition of the Port. The bits in PINx correspond to the condition of the Port pins defined as input. Bit 1 if pin "high", bit 0 if Port pin low.
PORTx	Data pointer for PORTx This register is used, in order to head for the outputs of a Port. With pins, which were switched by means of DDRx to input, the internal Pull UP of resistors can be activated or deactivated over PORTx (1 = actively).

5.5 Digital signals (I/O)

It is simplest to seize and/or spend digital signals with the microcontroller.

5.5.1 Outputs

If one wants to set defined pins (appropriate DDRx bits = 1) as output on logic 1, one sets the appropriate bits in the Port register.

With the instruction:

```
#include <avr/io.h>
...
PORTB = 0x04; /* better PORTB= (1<<PB2)*/
```

The output is thus set at pin PB2 (consider that the bits are always counted *from 0*, the least significant bit is thus bit number 0 and not bit number 1).

We must notice that for the assignment by means of "**=**" all pins are always indicated at the same time. So, if only certain outputs are to be switched, first declare the current value of the Port then set/ reset the bit of the desired Port. If we want to set only the third pin (bit No. 2) at Port B on "high" and to leave the status of the other outputs unchanged, we can use this form:

```
#include <avr/io.h>
...
PORTB = PORTB | 0x04; /* better: PORTB = PORTB | (1<<PB2) */
/* simplifies by use |= of the operator:*/
PORTB |= (1<<PB2);
/* also several "at the same time":/
PORTB |= (1<<PB4) | (1<<PB5); /* pin PB4 and PB5 "high" */
```

A "switching off", thus outputs on "low" set, affected similarly:

```
#include <avr/io.h>
```

```
...
PORTB &= ~(1<<PB2); /* bit resets 2 in PORTB and sets thereby for pin PB2 on low */
PORTB &= ~( (1<<PB4) | (1<<PB5) ); /* pin PB4 and pin PB5 "low" */
```

In source codes, which were developed for older version that of the avr GCC/avr lib.c, individual bits are set and/or reset by means of the functions **sbi** and **cbi** which are obsolete.

If the initial condition of outputs is critical, the sequence must be considered, with which the data direction (DDRx) is stopped and the expenditure value (PORTx) is set. For output pins, which are initialized with Initial value "high", first the bits in the PORTx register set and afterwards the data direction on output place. From it the succession input results - > set PORTx: internal Pull UP actively - > set DDRx: Output "high". With the order only DDRx and PORTx, can come it to a short "low pulse", the also exterene of Pull UP resistors "outvoted" (unfavorable). The succession: Input - > set DDRx: Output (on "low", there PORTx after RESET 0) - > set PORTx: Output on high.

5.5.2 Inputs (as signals come into microcontroller)

The digital input signals can arrive at different kinds at our logic.

Signal Coupling:

It is simplest, if the signals can be taken over directly from another digital circuit. If we knew the output of digital circuit TTL level, then we can directly connect it to the input pin of microcontroller. On the other hand, if the output of the digital circuit TTL level does not match with microcontroller, then we must use appropriate hardware (e.g. optocoupler, voltage divider, "Level shifter" aka level transducers) to adapt. The mass of the two circuits must be interconnected naturally.

Logical status at any pin of the microcontroller can be identified as logic 1, if the voltage at that pin is approximately more than 70% of V_{cc} and as logic 0, if it is less than 20% of V_{cc}. Detailed information about it, starting from which tension an input as 0 ("low") and/or 1 ("high") is recognized, is supplied by the DC characteristics table in the datasheet of the microcontroller. The inquiry of the conditions of the Port pins is made directly by the register name. About is important to use for the inquiry of the inputs **not** Port register **PORTx but to** input register **PINx** The inquiry of the pin conditions over PORTx instead of PINx is a frequent error with the AVR "first contact". (Otherwise one reads not the condition of the inputs, but the status of the internal Pull UP resistances.) If one wants to query signal statuses of Port D and into a variable named bPortD store current, one writes the following command lines:

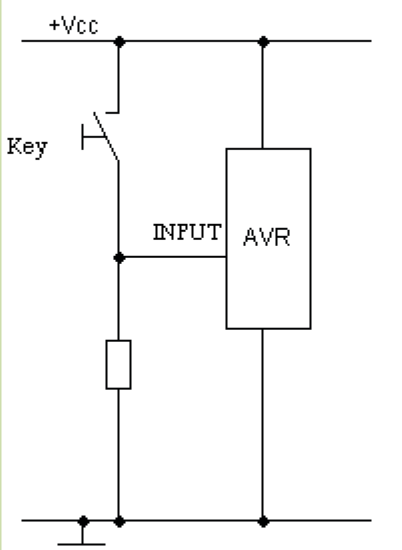
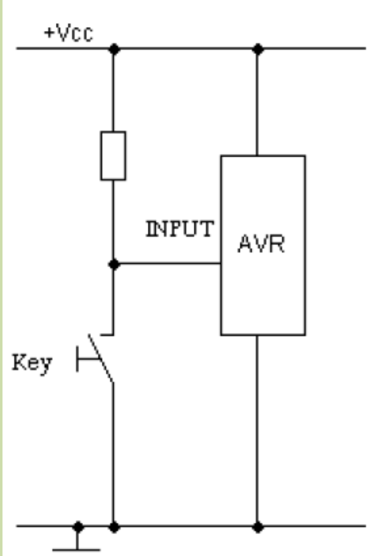
```
#include <avr/io.h>
#include <stdint.h>
...
uint8_t bPortD;
...
bPortD = PIND;
...
```

With the C-Bit operation one can query the status of the bits.

```
#include <avr/io.h>
...
/* implements action, if bit No. 1 (the "second" bit) is set in PINC (1) */
if ( PINC & (1<<PINC1) ) {
/* action */
}
/* implements action, if bit No. 2 (the "third" bit) is deleted in PINB (0) */
if ( !(PINB & (1<<PINB2)) ) {
/* action */
}
```

Keys and Switches:

The connection of mechanical contacts at the microcontrollers becomes likewise completely simple, whereby we must differentiate between two different methods (*Active Low* and *Active High*):

Active High	Active Low
	
Here the contact is switched between supply voltage and the input pin. So that during open switching position no undefined signal at the microcontroller is applied, between the input pin and the mass a Pull down resistance is switched. This serves to hold the level during opened switching position logic 0.	With this method the contact is switched between the input pin of the microcontroller and mass. Thus with open switch the microcontroller an undefined signal does not get between supply voltage and the input pin so-called Pull UP a resistance is switched. This serves to draw the level with opened switch on logic 1. The resistance value of the Pull UP of resistance is not critical actually. If it is however too highly selected, the effect is possibly not given. As usual value 10 Kilo ohms were in- practice. The AVRs has even at most pins by software insert able internal Pull UP of resistances, which we can naturally use.

Pull UP resistors activation:

The internal Pull UP resistors from V_{cc} to the individual Port pins can be activated and/or deactivated via register **PORTx**, if those pins are switched as **inputs**.

If the value of the appropriate Port pin is set to 1, then the Pull UP resistance is activated. At a value of 0 the Pull UP resistance is not active. One should use in each case either the internal or external Pull UP a resistance.

In the example, all pins of the Port D are switched as inputs and all Pull UP of resistors are activated. Further pin PC7 is switched as input and its internal Pull UP resistance is activated, without changing the parameters for the other Port pins (PC0-PC6).

```
#include <avr/io.h>
...
DDRD = 0x00; /* all pins von Port D as input */
PORTD = 0xFF; /* internal Pull Ups at all Port pins activated */
...
DDRC &= ~(1<<DDC7); /* Pin PC7 als Eingang */
PORTC |= (1<<PC7); /* internal Pull UP at PC7 activated */
```

Debouncing inputs

Now all mechanical contacts have, are it from switches, buttons or also from relays to bounce the unpleasant characteristic. This means that when closing the contact the same manufactures not directly contact, but several times switches on and off up to final manufacturing of the contact.

It is to be counted now with a fast microcontroller how often such a contact is switched, and then we have a problem, because bouncing is counted as repeated impulses. And this phenomenon must be absolutely considered during the writing of the program.

```
#include <avr/io.h>
#include <inttypes.h>
#ifndef F_CPU
#define F_CPU 3686400UL /* quartz with 3,6864 MHz */
#endif
#include <avr/delay.h> /* defines _delay_ms () off avr libc version 1.2.0 */
/* simple function to denounce a button */
inline uint8_t debounce(volatile uint8_t *port, uint8_t pin)
{
if ( ! (*port & (1 << pin)) )
{
/* pin was pulled on mass, 100ms waits */
```

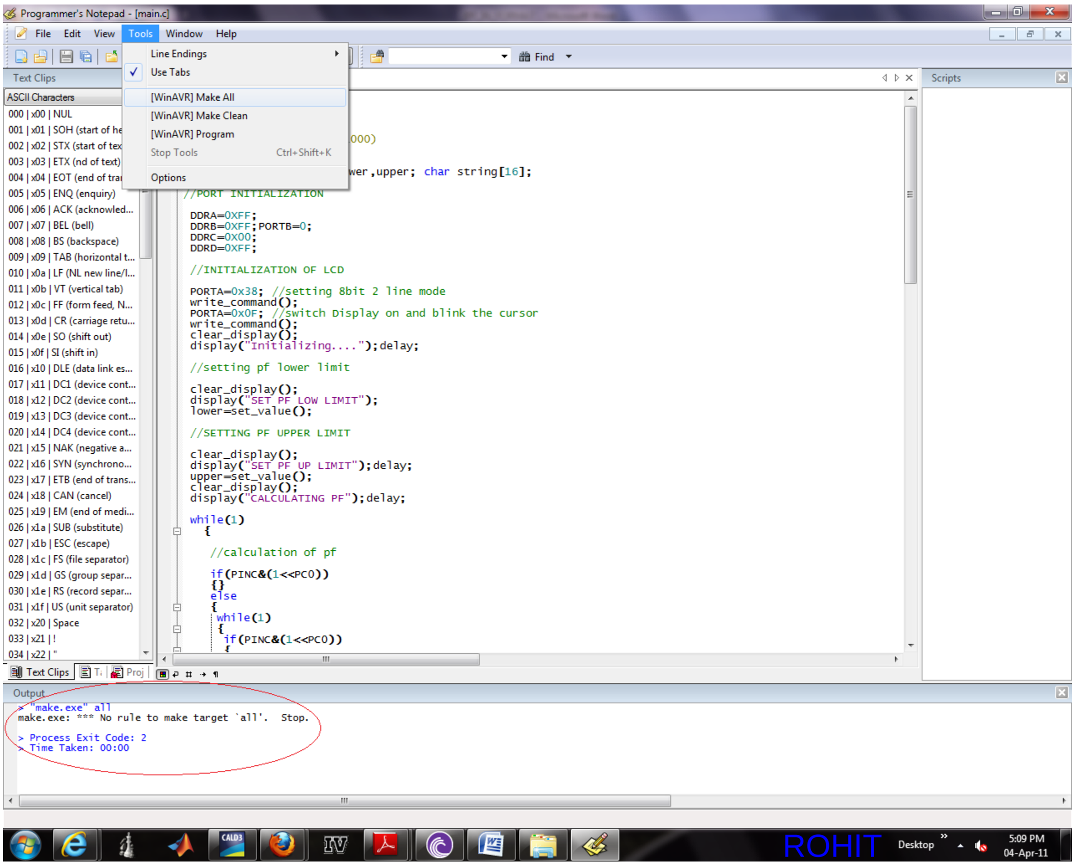
```
_delay_ms(100);
if ( *port & (1 << pin) )
{
/* give user time for releasing the button */
_delay_ms(100);
return 1;
}
}
return 0;
}
int main(void)
{
DDRB &= ~( 1 << PB0 ); /* pin PB0 on input (button)*/
PORTB |= ( 1 << PB0 ); /* Pull-up resistance activate */
...
if (denounce(&PINB, PB0)) /* case button pressed to pin PB0..*/
PORTD = PIND ^ ( 1 << PD7 ); /* .. LED at Port PD7 on and/or switch off */
}
```

With this example it is to be considered that the AVR waits in case of a depressing the key 200ms, therefore fallow lies. Time-critical applications should select another procedure.

5.6 Burning a program

To burn/ write a program on an Atmel microcontroller, we have to be ready with the microcontroller circuit, parallel port, WinAVR software and power supply (to microcontroller). As discussed in chapter 3, parallel port is nothing but a cord which is used in interfacing PC with older printers (dot matrix). Also, WinAVR software is available at free of cost on internet. Before burning a program, we must ensure that the microcontroller is given power supply in its limits as prescribed in datasheet.

To proceed further, first open the program to be burnt on microcontroller using WinAVR Programmer’s Notepad. Save the file (with some name) in a new folder in any drive using an extension as *.c*, which is to indicate that the program is written in C – language. After saving successfully, program must be compiled using WinAVR to debug errors (if any) found during programming. To compile a program, go to *Tools* and then click on *"Make all"* as shown below:



A message is displayed in the output block of the Programmer's Notepad (encircled above) as:

>"make.exe" all

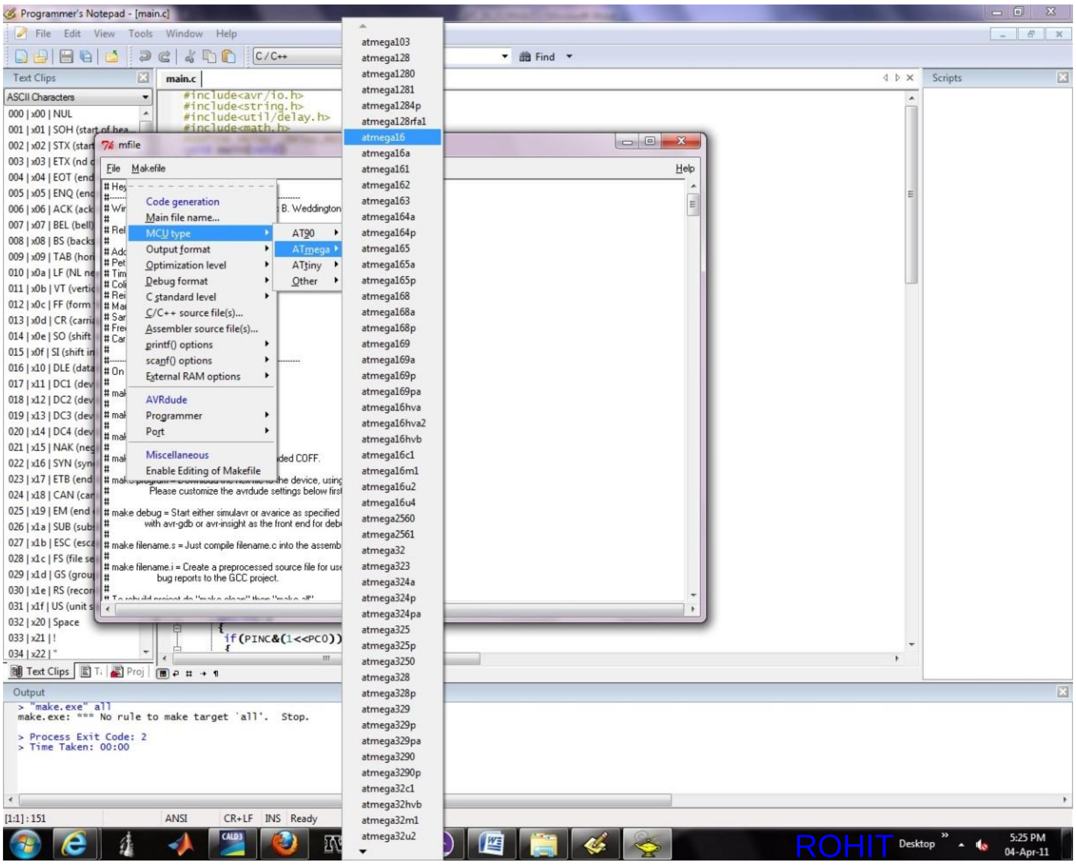
make.exe: *** No rule to make target 'all'. Stop

>Process Exit Code: 2

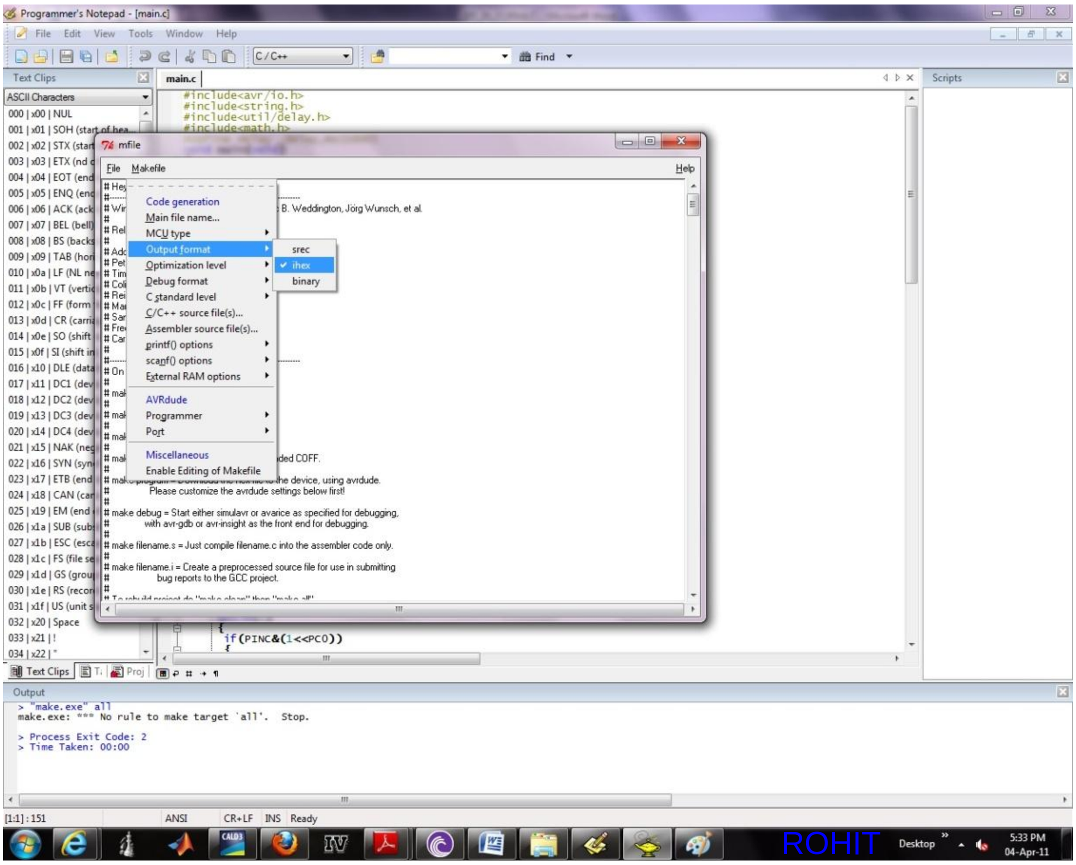
> Time Taken: 00:00

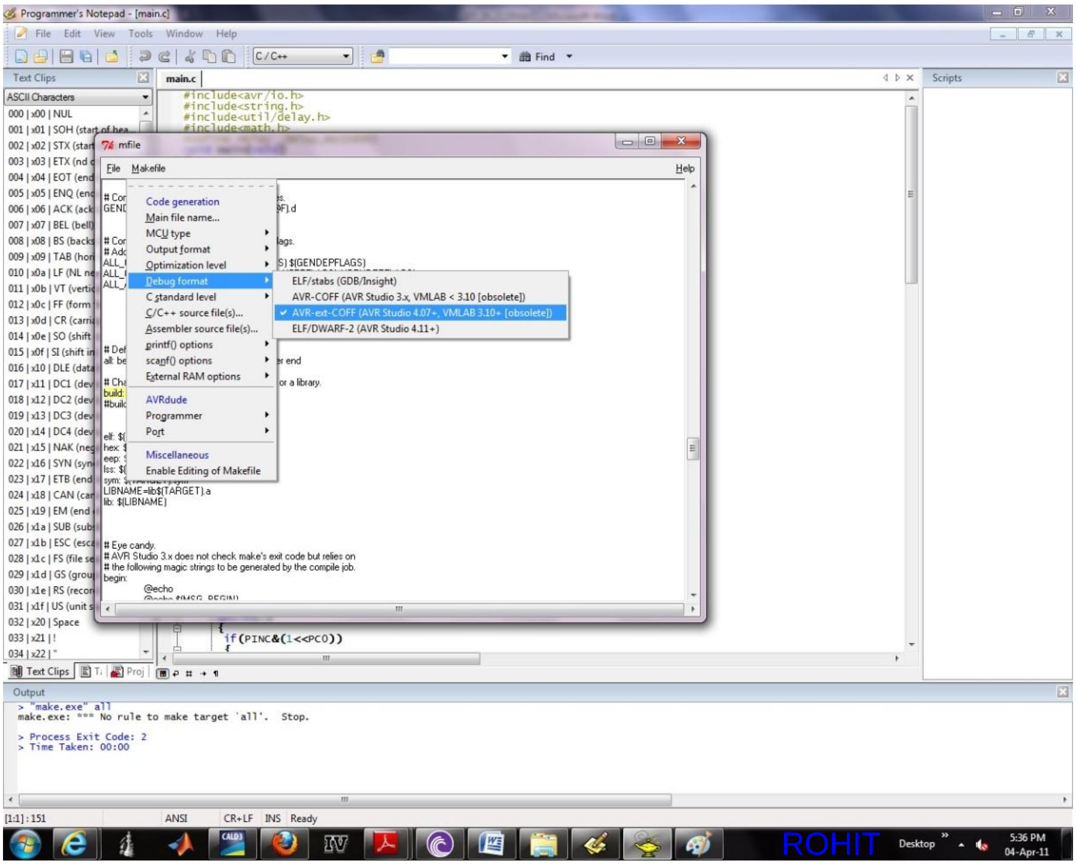
Now an Mfile has to be created to compile the C – program. For that, go to *Start\ All programs\ WinAVR\ Mfile[WinAVR]* and follow the procedure given below:

1. Select MCU type as ATmega 16 (microcontroller being used).

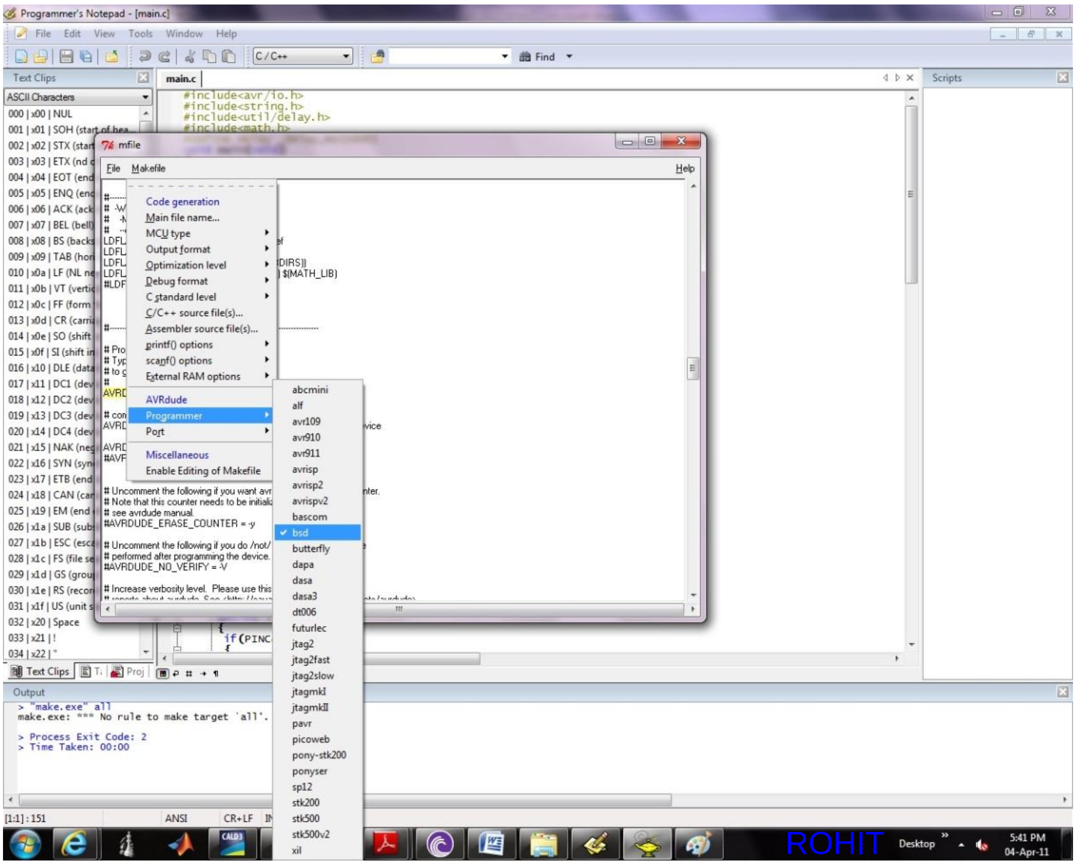


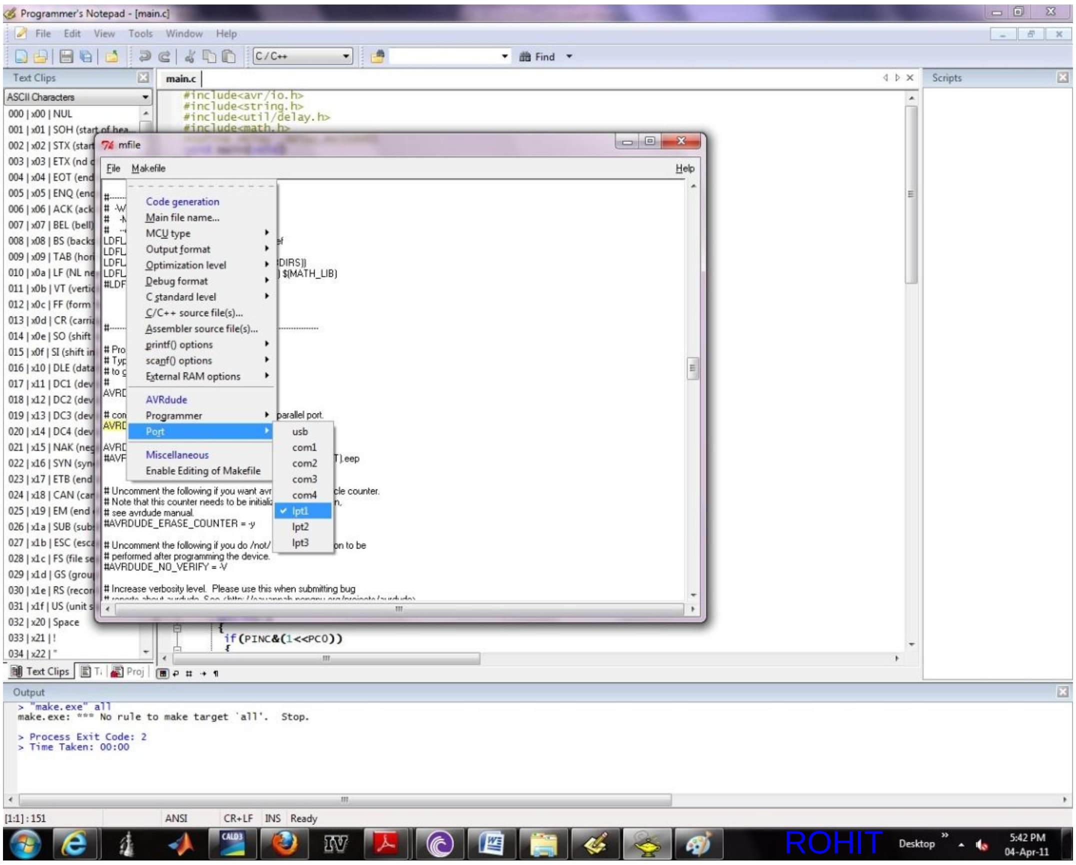
2. Select output format as *"ihex"* and debug format as *"AVR-ext-COFF"*





3. Set the type of programmer and assign the port using.

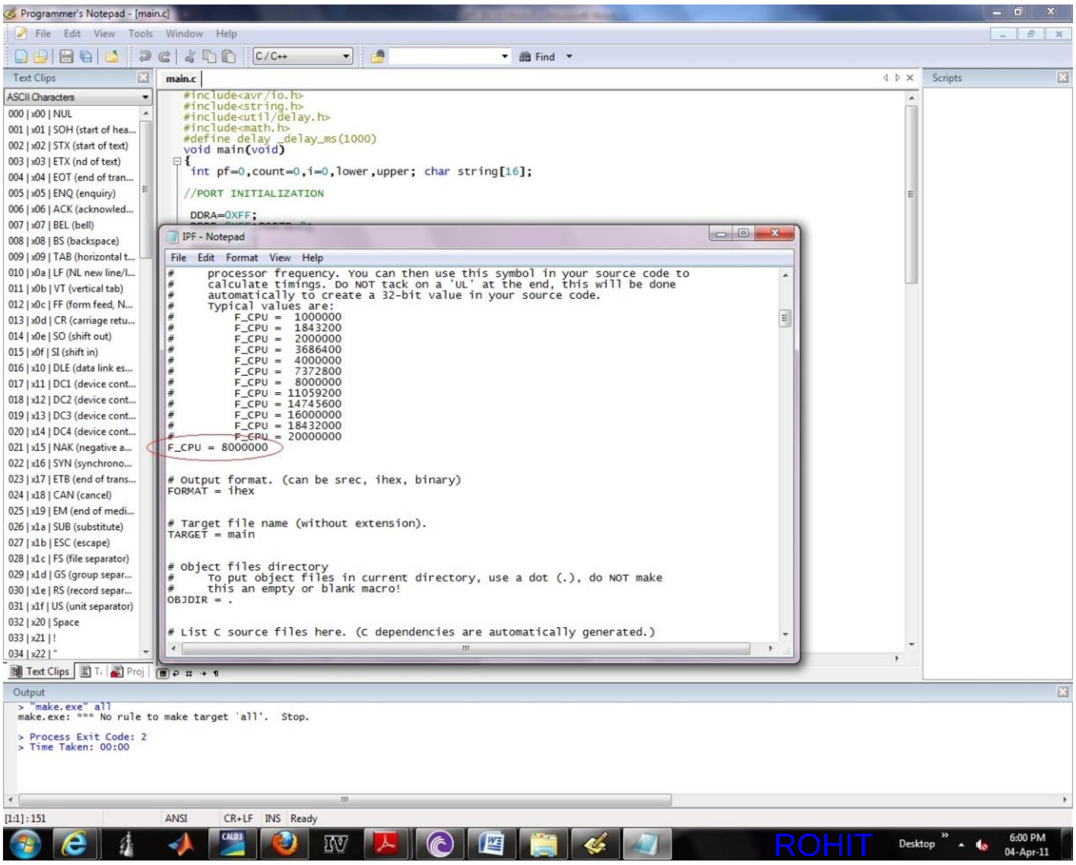




4. Now Mfile has to be saved in a folder where our main program has been saved already (with .c extension) and Mfile has to be opened with Notepad to append the details of crystal frequency (if any used) as shown below.

The value of “F_CPU” is to be set according to the crystal used. For instance, if a crystal oscillator of 1MHz is used then it is replaced by “F_CPU = 1000000”.

Note: If no crystal oscillator is used, then default clock frequency of microcontroller (1MHz) is to be set.



5. Finally, compile the program as directed above (go to *Tools* then click on *"make all"*).

Now, prior to the burning of program on microcontroller, one side of the parallel port must be connected to lpt1 port of PC. On the other side, only 5 pins are used to connect microcontroller. They are:

Parallel port pins	Microcontroller pins
Pin 7	RESET
Pin 8	SCK
Pin 9	MOSI
Pin 10	MISO
Pin 19 -25	Ground

After making all these connections, switch on the power supply to microcontroller (in the limits as prescribed in datasheet) and go to *Tools* then click on *Program* to burn/ write on microcontroller.

CHAPTER – 6

CONCLUSION AND FUTURE SCOPE

CHAPTER – 6

CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusion

In summary, power factor of huge inductive loads, like in industries which operates at different loads can be easily whipped near to unity. So that there is no need for industries to pay fine against power factor tariffs to utility companies. This method not only ameliorates industrial consumers power factor but also alleviates the situation of utility company in supplying more active power.

6.2 Future Scope

- Power factor meter used here can be integrated with consumer energy meter so that power factor can be displayed along with number of units consumed.
- Speed control of motors can be implemented by using triacs in the line whose gate triggering circuits can be controlled using microcontroller.
- Over – current protection and over – voltage protection can be employed using present CTs and PTs.

REFERENCES

[1] V.K. Mehta, Rohit Mehta: "Principles of Power Systems". 4th Revised Edition 2008.

[2] J.B. Gupta: "Switchgear & Protection"

[3] C.L. Wadhwa: "Electrical Power Systems". Sixth Edition.

[4] http://www.engineeringtoolbox.com/power-factor-electrical-motor-d_654.html

[5] http://en.wikipedia.org/wiki/Power_factor

[6] ATmega 16 datasheet

[7] http://www.embedautomation.com/content/tutorial-lcd-interfacing-atmega16

[8] http://iamsuhasm.wordpress.com/tutsproj/using-lcds/

[9] http://en.wikipedia.org/wiki/Induction_motor

[10] http://www.transtutors.com/homework-help/Electrical+Engineering/Induction+Motors/3-phase-induction-motors.aspx

[11] http://en.wikipedia.org/wiki/Relay

[12] http://en.wikipedia.org/wiki/Contactors

[13] http://en.wikipedia.org/wiki/78xx

[14] http://en.wikipedia.org/wiki/Capacitor

[15] http://www.aishack.in/2010/07/l293d-an-h-bridge/

[16] http://www.kpsec.freeuk.com/components/vres.htm#presets

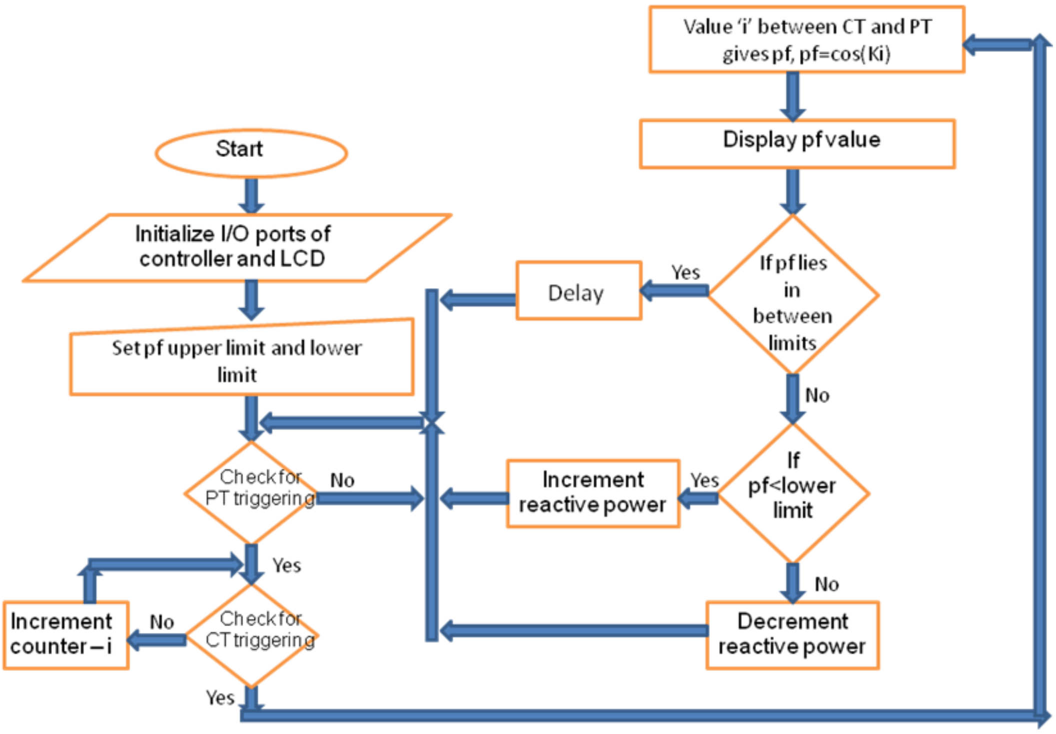
[17] http://en.wikipedia.org/wiki/Electrical_connector

[18] http://robozeal.blogspot.com/2009/06/parallel-port-programmer-for-atmega.html

[19] "AVR GCC Tutorial (WinAVR)" Early before alpha version 1.00 – By Takashi.

APPENDIX

A. Flow chart of program written on microcontroller



B. Program written on microcontroller

```
#include<avr/io.h>

#include<string.h>

#include<util/delay.h>

#include<math.h>

#define delay _delay_ms(1000)

void main(void)

{

 int pf=0,count=0,i=0,lower,upper; char string[16];

//PORT INITIALIZATION

 DDRA=0XFF;

 DDRB=0XFF;PORTB=0;

 DDRC=0X00;

 DDRD=0XFF;

 //INITIALIZATION OF LCD
```

```
PORTA=0x38; //setting 8bit 2 line mode
write_command();
PORTA=0x0F; //switch Display on and blink the cursor
write_command();
clear_display();
display("Initializing....");delay;
//setting pf lower limit
clear_display();
display("SET PF LOW LIMIT");
lower=set_value();
//SETTING PF UPPER LIMIT
clear_display();
display("SET PF UP LIMIT");delay;
upper=set_value();
clear_display();
display("CALCULATING PF");delay;
while(1)
  {
        //calculation of pf
        if(PINC&(1<<PC0))
        {}
        else
  {
         while(1)
         {
          if(PINC&(1<<PC0))
          {
    i=0;
           while(1)
           {
```

```
                    if(PINC&(1<<PC1))
                     break;
                    else
                     i++;
            }
            break;
           }
          }
         pf=1000*cos(0.0019*i);
         count=pf/10;
        }
        clear_display();
        sprintf(string,"PF=0.%d lag",pf);
        display(string);delay;
        //correction of pf
        i=PORTB;
        if(count<lower)
        {
         i++;
         if(i>7)
         {PORTB=15;delay;clear_display;new_line();display("Need more
KVAR");PORTB=7;}
         else
         {PORTB=i;}delay;
        }
        else if(count>upper)
        {
         i--;
         PORTB=i;delay;
        }
```

```
            else
            for(i=0;i<5;i++)
             delay;
   }
}
//FUNCTIONS TO DISPLAY STRINGS IN LCD
display(char y[])
{ int i,j=strlen(y);
  for(i=0;i<j;i++)
   { PORTA=y[i];
   write_data();
   _delay_ms(1);
   }
}
write_command()
{
 PORTD&=~(1<<PD0); //Setting RS = 0, selecting command register
 PORTD&=~(1<<PD1); //Setting RW = 0, writing mode
 PORTD|=(1<<PD2); //EN = 1
 _delay_ms(1);
 PORTD&=~(1<<PD2); //EN = 0, thus giving high to low pulse on Enable pin
 _delay_ms(1); //1ms delay
}
write_data()
{
 PORTD|=(1<<PD0); //Setting RS = 1, selecting data register
 PORTD&=~(1<<PD1); //Setting RW = 0
 PORTD|=(1<<PD2); //EN = 1
 _delay_ms(1);
 PORTD&=~(1<<PD2); //EN = 0, thus giving high to low pulse on Enable pin
```

```
 _delay_ms(1) ; //1ms delay
}
clear_display()
{
 PORTA=0x01;
 write_command();
}
new_line()
{
 PORTA=0xC0;
 write_command();
}
set_value()
{
 int i=0; char small[6];
 while(1)
  {
  if(PINC&(1<<PC7))
  i++;
  sprintf(small,"%d",i);
  new_line();
  display(small);
  if(PINC&(1<<PC6))
   break;
  _delay_ms(150);
  }
 return i;
}
```

Authors: Rohit, Sukumar, Krishna, Vinod.
--You can reach me @: rohit.gunnala@yahoo.com for further queries.

C. Intelligent power factor manager