**Microcontroller-Based Automatic Power Factor Correction System Using Arduino**

**1. Introduction**

Electrical power systems are designed to supply energy efficiently to various loads. However, many industrial and domestic loads are inductive in nature. Inductive loads such as induction motors, transformers, fluorescent lighting ballasts, air conditioners, refrigerators, and welding machines draw not only active power (measured in watts), but also reactive power (measured in volt-ampere reactive). This reactive component causes the power factor of the system to fall below the desirable value.

A low power factor leads to several major drawbacks, including increased line losses, reduced system capacity, higher electricity bills, overloading of transformers, and poor voltage regulation. To address these issues, power factor correction techniques are widely implemented in electrical systems.

Traditionally, power factor improvement is achieved by introducing capacitors into the electrical network. Capacitors supply leading reactive power and cancel out the lagging reactive power consumed by inductive loads. Manual capacitor banks, however, are inefficient and prone to human error. For dynamic loads, manual correction becomes impractical.

Automatic Power Factor Correction (APFC) systems solve this problem by automatically monitoring the power factor and switching the required number of capacitors based on load conditions. This ensures that the system operates with optimum power factor at all times.

In this project, an Arduino-based microcontroller system has been implemented to simulate the behavior of an APFC panel. The system measures input conditions through switching logic, applies capacitor banks through relays, and displays power factor values on a liquid crystal display. The project uses multi-stage capacitors and relay switching logic to demonstrate how industrial APFC panels operate. This simulation provides valuable learning for electrical engineering students about the concepts of power factor, reactive power compensation, and the working principle of APFC equipment.

**2. Project Title**

**Microcontroller-Based Automatic Power Factor Correction System Using Arduino**

**3. Objective of the Project**

The main objectives of this project are:

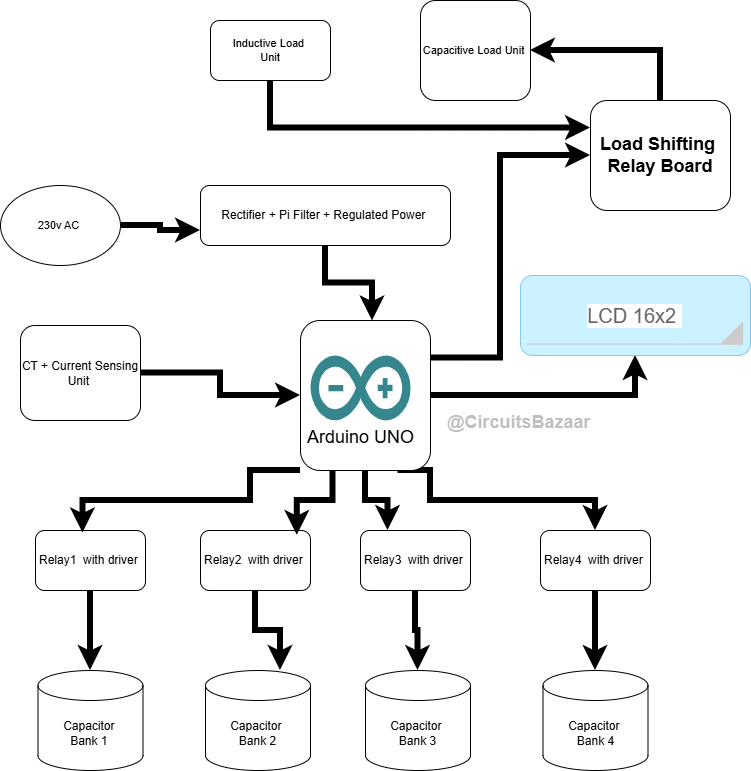
1. To design and develop an Arduino-based system for automatic correction of low power factor.
2. To simulate real-time behavior of inductive and resistive loads with varying power factor values.
3. To demonstrate relay-based switching of capacitor banks for improving power factor.
4. To display updated power factor readings on an LCD after each correction stage.
5. To help students understand the importance and working of power factor correction techniques used in industries.
6. To provide an economical, simple, and effective approach for developing APFC concepts in a laboratory environment.

**4. Problem Statement**

Industries and commercial establishments often operate devices that draw inductive load. These loads cause a lagging power factor, which in turn increases current, copper losses, and reduces the efficiency of the entire electrical system. Power distribution companies often impose penalties when the power factor falls below a specified range. Manual correction using capacitor banks is inefficient and cannot respond to dynamic load variations.

Therefore, a compact, automated, microcontroller-based APFC system is required to continuously monitor and correct power factor without human intervention. This project provides a simplified model of such a system using Arduino and relays.

**5. Components Used**



(With detailed explanation)

This section provides an in-depth explanation of every hardware component used in the project.

**5.1 Arduino Uno Microcontroller Board**

Arduino Uno is the central processing unit of this project. It is based on the ATmega328P microcontroller and operates at 16 MHz. The board contains digital input/output pins, analog input pins, UART interface, I2C interface, timers, and PWM support. It is easy to program using the Arduino IDE, making it ideal for academic projects.

Key features:

* Operating Voltage: 5V
* Digital I/O Pins: 14
* Analog Inputs: 6
* Flash Memory: 32 KB
* SRAM: 2 KB
* EEPROM: 1 KB

Role in the project:

* Controls all relays based on switching logic
* Displays messages and PF values on the LCD
* Generates random power factor values to simulate load conditions
* Performs timed control sequences and capacitor switching

**5.2 LCD Display (16x2, 4-bit Interface)**

A 16x2 Liquid Crystal Display is used to show messages like:

* Power Factor
* Calculating
* Corrected power factor values
* Real-time capacitor switching

It uses a 4-bit communication mode to save digital pins on the Arduino. The LCD is used to display both intermediate messages (Calculating...) and final PF values after applying capacitor correction.

**5.3 Relays (5V Electromagnetic Relays)**

Five relays are used in the project.

Relay functions:

* One relay is treated as the main load relay (simulating load control).
* Two relays are capacitor bank stages (relay1 and relay2).
* Additional relays are kept for future extension or multi-stage APFC demonstration.

Purpose:

* Relays provide electrical isolation.
* They simulate switching of capacitor banks similar to industrial APFC panels.
* They help demonstrate multi-step capacitor correction.

**5.4 Toggle Switch**

The toggle switch is used as the load-type selector input.

* When the switch is ON, the system simulates resistive load with high PF.
* When the switch is OFF, the system simulates inductive load with low PF.

The switch effectively changes the state of the system to demonstrate dynamic load behavior.

**5.5 LEDs (Red and Green)**

Two LEDs are used for visual indication:

* Red LED: Indicates low power factor condition.
* Green LED: Indicates good or corrected power factor.

This helps in visually understanding the system performance.

**5.6 Connecting Wires**

Used to make the electrical connections between Arduino, LCD, relays, LEDs, and switch.

**5.7 Power Supply (5V DC)**

A USB supply or regulated 5V DC supply is used to power the Arduino and associated circuitry.

**6. Software Used**

**6.1 Arduino IDE**

The Arduino Integrated Development Environment is used to write, compile, and upload the code to the Arduino Uno board. Key features include:

* Supports C/C++ language
* Provides built-in libraries for LCD, timing, and I/O control
* Serial monitor for debugging
* Easy to use even for beginners

**6.2 Embedded C Programming**

The logic of the project is written in embedded C language. Major tasks performed in the code include:

* Display control using the LiquidCrystal library
* Conditional switching of relays
* Random generation of PF values
* Delay management
* Looping and control statements

**7. Working Principle**

(Explained in detail)

The system operates as follows:

**7.1 Load Selection**

A toggle switch is used to simulate the type of electrical load:

* When switch is high: resistive load simulated
* When switch is low: inductive load simulated

**7.2 Initial Power Factor Measurement**

When load type changes, the system simulates power factor calculation by displaying:

Power Factor =  
Calculating ….

This mimics the real behavior of industrial APFC systems where the controller measures voltage, current, and phase difference.

**7.3 Power Factor Value Generation**

Since this is a simulation project, random PF values are used:

For resistive load:  
Random between 0.95 and 0.96

For inductive load:  
Random between 0.53 and 0.59

This variation makes the system more realistic.

**7.4 Relay Switching and Capacitor Addition**

Depending on the power factor, the system switches relays to add capacitor banks:

* For resistive load: One capacitor stage (relay1)
* For inductive load: Two capacitor stages (relay1 and relay2)

Each activation of a relay is immediately followed by a “Calculating…” display for approximately 6.5 seconds. This represents the time required in a real APFC system for load stabilization and measurement.

**7.5 Corrected Power Factor Display**

After the capacitor bank is added and delay elapsed, the updated power factor is displayed:

* For resistive load: PF corrected to 1.00
* For inductive load: PF corrected to 0.99

This demonstrates the working of an automatic multi-stage power factor correction system.

**7.6 LED Indications**

Based on the status:

* Red LED indicates low PF
* Green LED indicates corrected PF

This visual feedback assists in understanding the correction process.

**8. Circuit Diagram Overview**

(Explanation)

Although wiring diagrams vary, the typical connections are:

* LCD connected in 4-bit mode to pins 8–13
* Main relay at pin 2
* Capacitor stage relays at pins 3 and 4
* LEDs connected to A1, A2
* Switch connected to pin 7

**9. Advantages**

1. Helps students understand the concept of power factor and reactive power compensation.
2. Demonstrates multi-stage APFC operation with realistic timing delays.
3. Low-cost hardware accessible for educational institutions.
4. Easy to modify and expand with additional stages.
5. Simple coding structure also helps in learning embedded C programming.
6. Enhances knowledge of relay interfacing and capacitor switching.
7. Provides a clear visual understanding through LCD and LED indicators.

**10. Disadvantages**

1. This project simulates power factor using random values and does not include actual voltage/current measurement.
2. Real industrial APFC requires CT sensors, PT sensors, and zero-cross detectors which are absent in this simulation.
3. Relays used here cannot switch high-power capacitors; this is only demonstration-level hardware.
4. Has limited resolution compared to digital power analyzers.

**11. Applications**

1. Educational demonstration in engineering colleges.
2. Training labs for electrical engineering students.
3. Demonstration of APFC principles in diploma and technical institutes.
4. Can be used as a base model to develop full APFC panels.
5. Useful in workshops and seminars explaining power factor concepts.

**12. Future Scope**

1. Integration of CT and PT sensors to measure actual voltage and current.
2. Use of zero-crossing detection and triac-based control instead of relays.
3. Incorporation of multiple capacitor stages similar to commercial APFC panels.
4. Display of real-time graphs using IoT platforms.
5. Online monitoring system via Wi-Fi or GSM communication.
6. Machine learning algorithms for predictive PF correction depending on past load data.
7. Integration with SCADA or industrial automation systems.
8. Adding harmonic filters for dealing with non-linear loads.

**13. Conclusion**

This project successfully demonstrates the working concept of an Automatic Power Factor Correction system using Arduino. Through switching of capacitor banks, timed delays, and simulated power factor readings, it effectively reproduces the behavior of industrial APFC panels at a small scale. The project helps beginners and engineering students understand the underlying principles of reactive power compensation, power factor improvement, and digital control of electrical loads. With further enhancements, this prototype can evolve into a fully functional APFC controller suitable for real-world applications.