A Cost-Effective Method for Power Factor Metering Systems

Case Study

Ahmed M. T. Ibraheem Al-Naib
Northern Technical University,
Technical Engineering College, Department of Electrical Power Technology Engineering, Mosul, Iraq
ahmed_alnaib2018@ntu.edu.iq

Bashar Abdullah Hamad
Northern Technical University,
Technical Engineering College, Department of Electrical Power Technology Engineering, Mosul, Iraq
bashar.hamad@ntu.edu.iq

Abstract *– The power factor (PF) is an important measurement in an AC electrical system that indicates how much power is utilized to accomplish productive work by a load and how much power is consumed. As a result, it's one of the primary causes of excessive energy costs and power outages. This paper aims to present a simple, cost- effective, and accurate PF metering and monitoring system implemented using an Arduino microcontroller with a novel methodology different from other papers. The proposed method is to design the software code instead of using external components of Zero-Crossing Detectors (Z-CDs) for both voltage (V) and current (I) signals, and instead of using Exclusive-OR (X-OR) gates also. Determining the phase-angle and PF in an efficient manner can be useful in many approaches to electrical systems: 1-for synchronization of parallel connections of alternators; 2-for directional protection systems; 3-for PF correction and load management; and 4-for designing watt/energy meters. Using the Proteus 8 Professional (ISIS program), the proposed designed circuit was simulated for more verification. The simulation and experimental findings are presented to validate the proposed metering system's effectiveness.*

Keywords: *Power factor, phase-angle, Zero-Crossing Detectors, Microcontroller, Wattm eter.*

1. INTRODUCTION

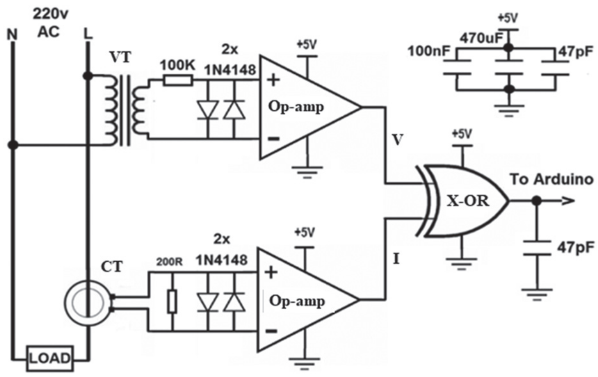
Technology is developing at a fast pace, and so is energy consumption. There is an immediate necessity for us to exceed from electro-dynamic meters to digital meters that calibrate them efficiently. Because of the mechanical components in the meter, the commonly used electro-dynamic meters are subject to variations in temperature and time. They're additionally sensitive to flux variation and its dependency. Accurate measurement of the PF for electrical energy is becoming a developing topic among researchers since it distributes electrical energy at an acceptable cost. The cost of electrical energy rises as a result of power losses in transformers, transmission lines, and loads [1, 2]. Due to its versatile features and rich library functionalities, the Arduino is gaining traction in this field (i.e., measurement devices and monitoring systems). It is robust, fast, and user-friendly all at the same time. Many methods for measuring, monitoring, and detecting defects in electrical systems based on microcontrollers have been employed. [3]. There are many previous studies and research on metering the PF: The authors of [4] created a PF measurement circuit prototype. A Z-CD (based on operational amplifiers of the LM358-type) is employed. The logical signals obtained from the detector are fed into the PIC16F877 microcontrollers. PIC16F877 performs the conversion of the difference between the V and I logical signals of the load into time in seconds and angle in degrees, as well as calculates the PF. On the LCD panel is displayed the PF measurement value. The authors of [5] designed and built a single-phase PF meter using an Arduino Nano-type. Two operational amplifiers (LM358) were used as Z-CDs for both V and I signals. Both logical signals from the Z-CDs are sent to the Exclusive-OR (X-OR) gate (4030). The information gotten by the X-OR gate is sent to Arduino for calculation of the phase-shift (Θ) and the corresponding PF using cos (Θ) and the results are displayed on the 16×2 LCD. The authors of [6] have designed an automotive PF correction system that always measures and displays the instant value of PF. If the PF deviates from the specified limits, then the switching of capacitors is done automatically. If the PF is not set within the limits, the information regarding this will be sent to the industry manager through the Internet of Things (IOT) for necessary actions. The major components that were used in this system are the Potential (PT) and Current Transformer (CT), two UA741 op-amps, a 7486 XOR gate IC, the Arduino Uno, the relay module, power capacitors, a 16 x 2 LCD display, an ESP8266 Wi-Fi module, and various kinds of electrical loads. The authors of [7] present a multi meter to measure electric parameters like real power, reactive power, and PF of different loads where data processing is FPGA based. The meter includes an ADS1220, which is a 24-bit ADC interfacing with an FPGA (Artix 7), which provides V and I digital samples. The main problem with this method is that it requires an external ADC, which adds to the complexity and cost of the whole thing. The author [8] used the Arduino Mega type for calculating the PF and activating the relays that connect the capacitor banks to the load in parallel as an automatic PF correction. A PT and a CT were used to step down and measure the V and I of the load. The V and I waveforms obtained from PT and CT were sent to the Z-CD circuits, which contain LM324 operational amplifiers to change the waveform of the V and I from sinusoidal to square. The outputs of the Z-CD circuits are fed to the XOR (3040) gate, which produces a series of pulses. The amplitude and width of these pulses are then calibrated to obtain the phase angle between V and I. The author [9] measured the PF with Arduino. A ZMPT101B AC voltage sensor and a non-invasive SCT-013-050 split core CT were used. Op-Amps (LM324N) were utilized for Z-CDs on the I and V signals. The XOR gate (IC4030) output will determine whether there is a phase shift between V and I or not, comparing the time delay for each input value. Table 1 lists the advantages and disadvantages of the proposed metering method compared with the traditional method for determining the PF.

Table 1. Comparison between the existing solution method and the proposed method for metering PF

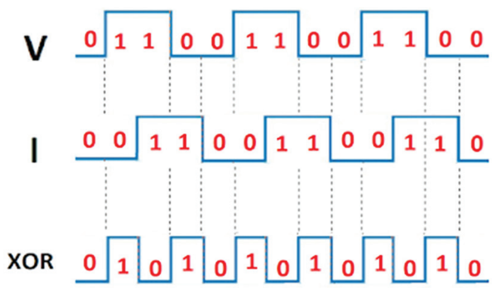
Existing solution in [4 - 9]	Proposed method
Use an external component that works as a Z-CD for both V and I signals	Eliminate using external components such as Z-CDs
Use an external component (X-OR gate) to find out the phase-angle between both the V and I signals.	Eliminate using external X-OR gate
High cost	Low cost
High complexity	Low complexity
Restricted to a sinusoidal waveforms	Restricted to a sinusoidal waveforms

2. DISPLACEMENT PF METERING METHOD

When both voltage and current have a sinusoidal waveform, the PF value equals the cosine of the angle between them and can be known as the displacement PF. The traditional method for determining the displacement PF depends on using external circuits for Z-CDs for both V and I signals and an external X-OR gate (see Fig 1: a). First, the Z-CD circuits, which are based on operational amplifiers, convert the sine wave signals coming from VT and CT with different amplitudes to logical signals (V & I), and then these two logical signals are applied to the X-OR gate to find out the phase-angle between them, as shown in Fig. 1: b. [1, and 10].



(a)



(b)

Fig. 1. Circuit diagram of PF determination

When the load type is resistive, the X-OR gate output is LOW logic because both V and I waveforms begin and end at the same time, but when the load type is inductive or capacitive, the X-OR output is HIGH logic because there is a phase-shift between V and I. By measuring the duration time (dt) difference between the V and I waveforms (XOR output ON-time in seconds) and putting it in equation (1), the Arduino can find the PF.

$$PF = \cos(f * dt * 360). \quad (1)$$

Where: f : frequency (Hz) of the supply voltage.

3. PF ANALYSIS UNDER NON-SINUSOIDAL WAVEFORMS

The displacement PF methods may present unacceptable errors under non-sinusoidal voltages and current waveforms. In general, the definition of the PF under non-sinusoidal waveforms is as the ratio between the active power (P) and the apparent power (S), as the current waveform distortion prevents accurately detecting the zero crossing points and the phase displacement of the current with respect to the voltage. So the calculation of P and S requires consideration. These calculations are estimated according to the instantaneous values of the current (Ii) and voltage (Vi) measured on the load, so this method is known as the instantaneous power calculation method. Based on these quantities, it is possible to calculate the rms current (I_{rms}), the rms voltage (I_{rms}), P, S, and then the PF, according to (2)–(6) equations [11-14]:

$$I_{rms} = \sqrt{\frac{\sum_{i=1}^{n} I_i^2}{n}} \qquad (2)$$

$$V_{rms} = \sqrt{\frac{\sum_{i=1}^{n} V_i^2}{n}} \qquad (3)$$

$$P = \frac{1}{n}\sum_{i=1}^{n} V_i * I_i \qquad (4)$$

where n is the number of samples.

$$S = V_{rms} * I_{rms} \ (5) \qquad (5)$$

$$PF = \frac{P(W)}{S(VA)} \ (6) \qquad (6)$$

4. PROPOSED METHODS FOR MEASURING THE PF

The circuit diagram of the proposed system is simply composed of voltage and current transformers, signal conditioning circuits for V and I signals, an Arduino, and a 4 x 20 LCD, as presented in Fig. 2. In this work, there are abridgments in using external electrical elements compared to Fig. 1, which saves installation time and space in the prototype. It also does not increase the cost and complexity of the system.

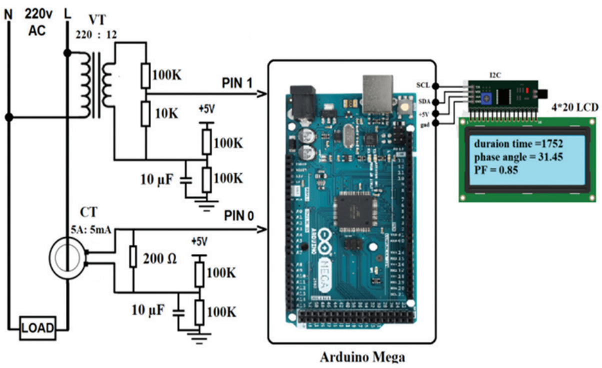
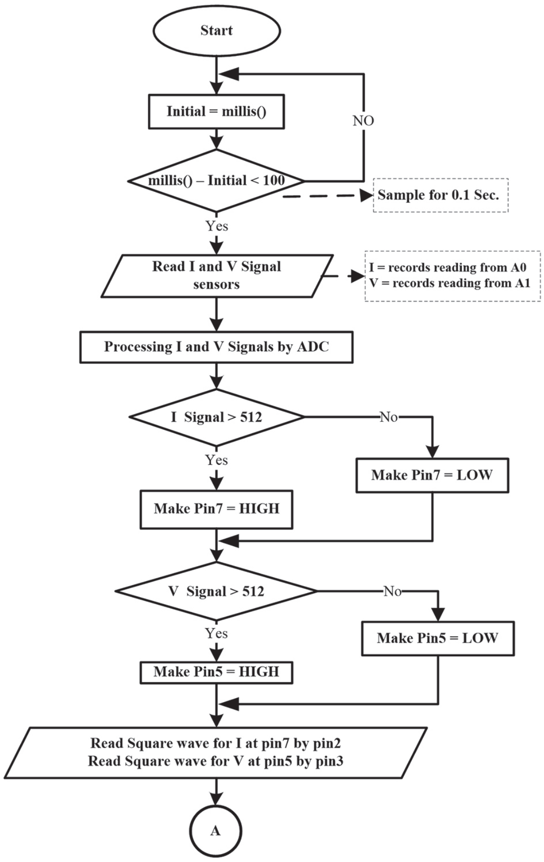


Fig. 2. Circuit diagram of the proposed system

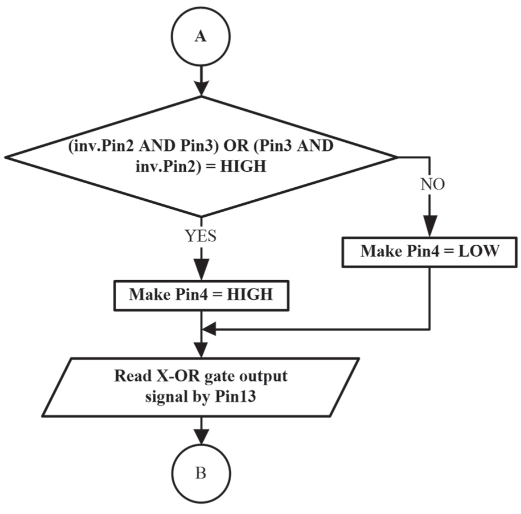
The AC supply voltage of 220 volts is stepped down to 12 volts AC via a VT. This 12 volt signal is divided and stepped down into a 1.5 volt sine wave signal by a voltage divider circuit. The AC voltage (1.5 volts) alternates from positive to negative values with respect to ground (0 volt). However, the analog inputs of the Arduino require positive voltage values. So a 2.5 volt DC offset is applied to the input sine wave signal. As a result, the whole input sine wave can be sensed at the positive threshold (0 – 5 volts). The V signal remains positive because it will now swing above and below 2.5 volts. The Arduino can read the entire sine wave signal through its analog input (pin A1). The I signal flowing through the load is retrieved through a CT, which has a 5 A: 5 mA rating. A shunt resistor (also called a burden resistor) converts the I signal into a voltage form that represents the properties of the current sine wave. As in the V signal, a DC offset voltage of 2.5 volts is applied to the sine wave signal so that the reference point is lifted up and the entire sine wave can be read in analog mode (pin A0) within its operative range (0-5 volts).

5. DESIGN FLOW OF THE SYSTEM

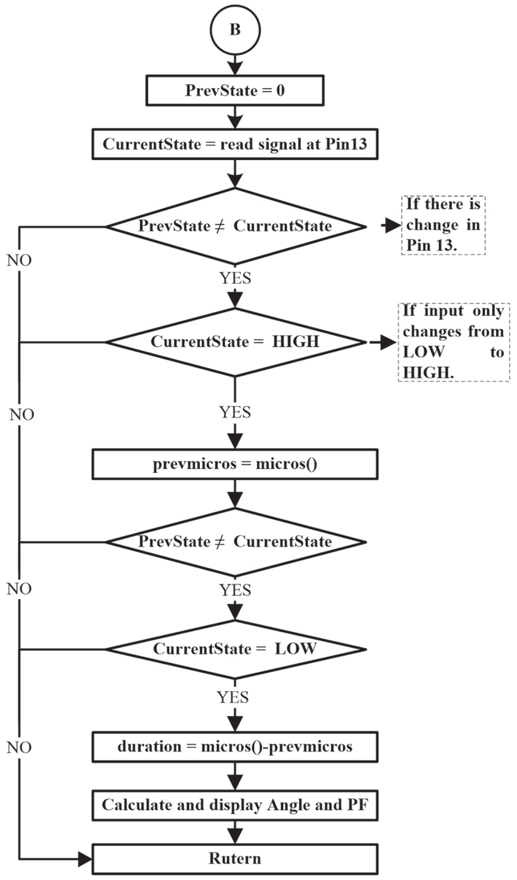
The design flow of the system software implementation is an important part of this system. The microcontroller is programmed to calculate the PF and display it on the LCD. The flowchart for the PF measurement is shown in Fig. 3.



(a) Flowchart of the Z-CD algorithm,



(b) Flowchart of the X-OR algorithm,

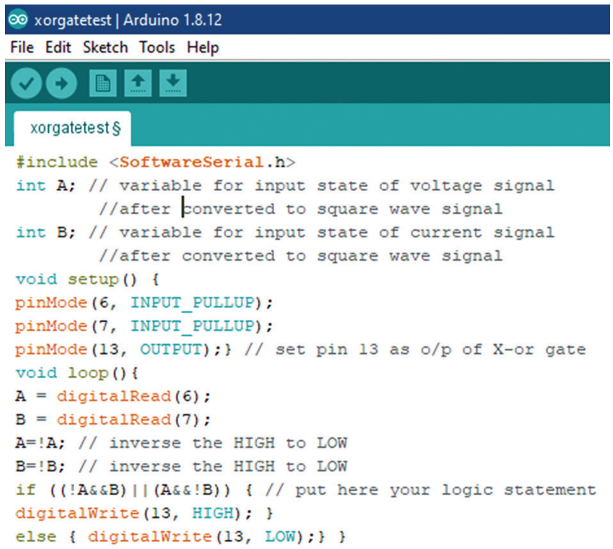


(c) Flowchart for time duration & PF determination

Fig. 3. Flowchart of the proposed PF measuring System

Instead of using an external component of the Z-CDs, the offsetting signals of V and I are converted to square waves via the proposed Arduino code by comparing these signals with defined values (bias level, which represents zero level). Fig. 3 (a): Using the proposed code of the Z-CD algorithm, the Arduino generates a HIGH state signal at pin 7 and pin 5 when the I and V signals exceed this defined value (512 bits, which is equal to the 2.5 volt value at the analog input of the Arduino, which is also equal to the value 0A at the secondary side of CT and 0 volt at the secondary side of VT). Also, the Arduino generates a "low state" signal when the signal is lower than this defined value. In other words, by the code, the Arduino will produce a high-logic square wave signal (5 volts) at the positive half-cycle of both V and I signals, and for the negative half-cycle, the square wave will have low logic (0 volts).

Instead of using an external component of the X-OR gate, this gate is implemented programmatically via Arduino code, as shown in Fig. 4 according to the flowchart of the X-OR gate implementation shown in Fig. 3 (b). The implementation code depends on the equivalent circuit of the X-OR gate function, which is shown in Fig. 5. The output signal of the X-OR gate (ON-time) represents the time difference between the V and I waveforms. Fig. 3 (c) represents the flowchart for determining this duration time. Finally, the mentioned equation 1 was programmed to find the PF.




```
#include <SoftwareSerial.h>
int A; // variable for input state of voltage signal
       //after converted to square wave signal
int B; // variable for input state of current signal
       //after converted to square wave signal
void setup() {
pinMode(6, INPUT_PULLUP);
pinMode(7, INPUT_PULLUP);
pinMode(13, OUTPUT);} // set pin 13 as o/p of X-or gate
void loop(){
A = digitalRead(6);
B = digitalRead(7);
A=!A; // inverse the HIGH to LOW
B=!B; // inverse the HIGH to LOW
if ((!A&&B)||(A&&!B)) { // put here your logic statement
digitalWrite(13, HIGH); }
else { digitalWrite(13, LOW);} }
```

Fig. 4. Software implementation code of the X-OR gate

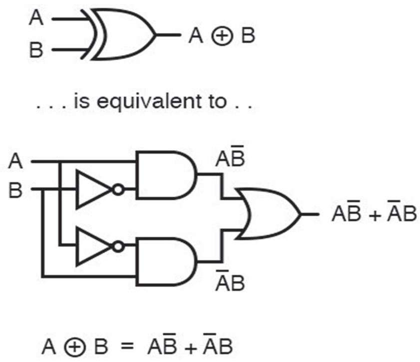


Fig. 5. Equivalent circuit of X-OR gate

6. SIMULATION CIRCUIT OF THE PROPOSED SYSTEM

The proposed measuring system shown in Fig. 2 was simulated as shown in Fig. 6 using the Proteus 7 Professional.

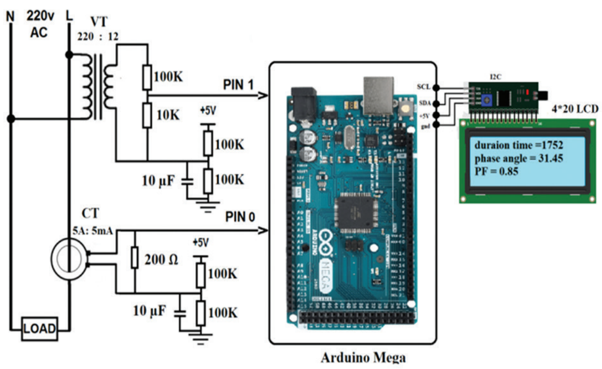
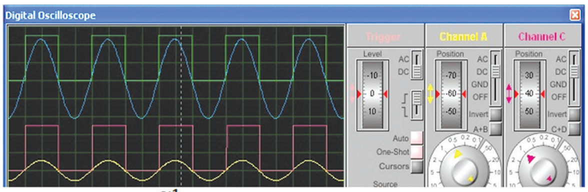


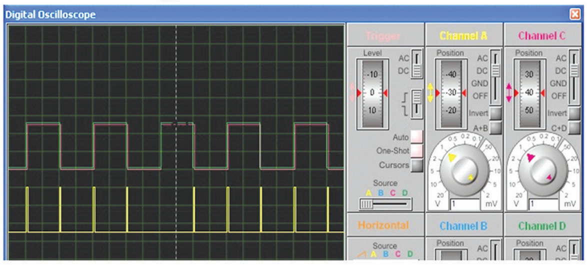
Fig. 6. Simulation circuit diagram of the proposed PF metering system

7. SIMULATION RESULTS

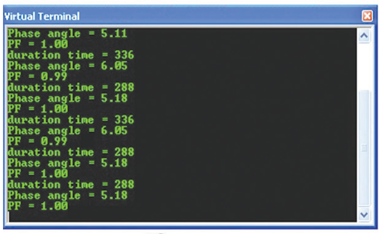
The proposed design is tested at various sorts of loads (resistor and inductor). Figure 7 shows the waveforms of the V and I signals (a:1 and b:1 blue and yellow, respectively) of a resistive and inductive load and their square waveforms (green and red, respectively), which are considered as input signals to the X-OR gate. The second figure (a:2 and b:2) shows the output signal from the X-OR gate, which represents the phase difference between the V and I signals. Figures (a:3 and b:3) (serial monitor), which show the instantaneous values of the duration time, phase-angle, and PF.



a:1

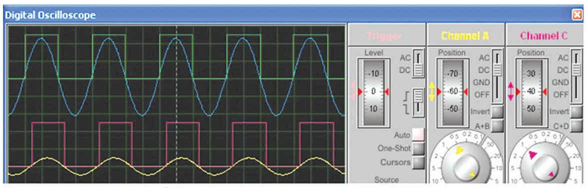


a:2

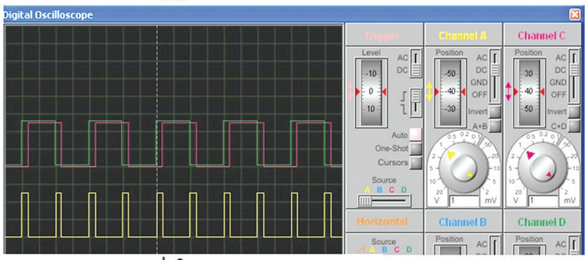


a:3

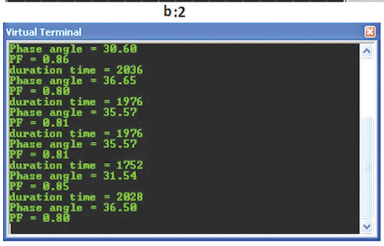
(a) Resistive load



b:1



b:2



b:3

(b) Inductive load

Fig. 7. Waveform of the proposed method

It's observed from Fig. 7 (a) that at resistive load, the time difference between the V and I waveform is very small, approximately zero, and so the phase-angle is very small, so the PF is unity. While in the inductive load in Fig. 7 (b), there is a clear time difference between the V and I signals which equal to 1.988 milliseconds (phase-shift equal to 34.35°) and so the PF is equal to 0.81.

8. EXPERIMENTAL RESULTS

Fig. 8 presents the prototype of the proposed circuit. Practically, the proposed design circuit is tested at various types of loads (resistive load using a tungsten filament bulb, and inductive load using an induction motor).

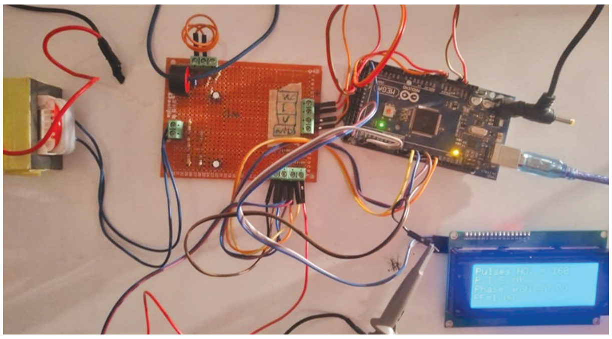
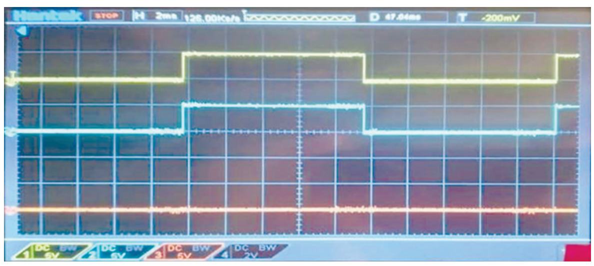
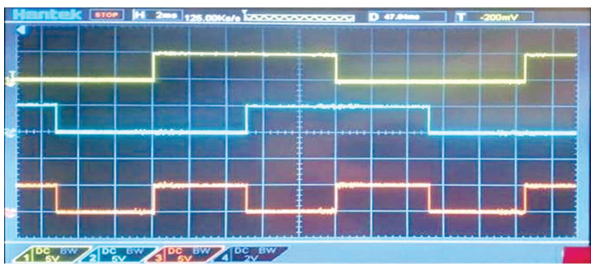


Fig. 8. Experimental setup of the proposed circuit

Fig. 9 shows the signals that are taken from the Arduino for both resistive (Fig. 9: a) and inductive load (Fig. 9: b). These waveforms are the square waveform of the V signal (yellow waveform), the square waveform of the load I signal (blue waveform), and the output of the X-OR (red waveform). The X-OR gate output will be HIGH logic when the inputs have a phase-shift between V and I signals and the time of this period represents the duration time.



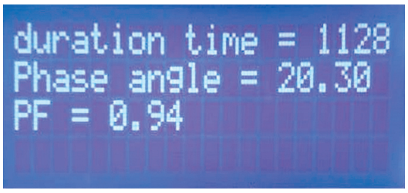
(a)



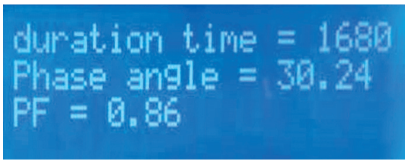
(b)

Fig. 9. Experimental Arduino waveforms of the proposed circuit

Fig. 10 presents the real-time measurement of the time duration, phase-shift, and PF on the LCD for both resistive and inductive loads.



(a)



(b)

Fig. 10. Real-Time monitoring of the PF

9. CONCLUSIONS

The proposed work provides one of the simplest ways to measure and monitor the PF. This method eliminates the use of some external components such as Z-CD circuits and X-OR gates. Thus, the proposed design costs less due to the savings from the eliminated components, which also leads to a smaller size and volume. So, the market approach is possible, and a commercial version would be suitable and could easily integrate. The PF metering circuit has been designed and tested using the Proteus simulator before practical implementation. The proposed method of metering the PF is limited to linear loads only with a sinusoidal terminal voltage that draws a sinusoidal load current. So, for harmonics-rich non-linear loads, the instantaneous power calculation method can be adopted in future work. Also, practical implementation of this work can be done, and the measuring data can be monitored based on IOT devices such as Wi-Fi. In addition to designing and developing a PF correction via compensation for the lagging PF by adding relay and capacitor banks to the designed circuit, wattmeter, energy meter, digital phase sequence meter, and directional protection relay based on the proposed system.

10. REFERENCES

[1] S. V. Nemani, D. Shahi, I. K. Vibhav, "Design and Implementation of Digital Energy Meters with Power Factor Measurement and Load Indication Feature", Proceedings of the IEEE 8th Power India International Conference, Kurukshetra, India, 10-12 December 2018, pp. 1-6.

[2] S. R. Fahim, S. S. Avro, S. K. Sarker, "A novel fractional order power factor measurement", SN Applied Sciences, Vol. 1, 2019, p. 1611.

[3] J. A. Ahmed, M. T. Ahmed, A. Ibraheem, O. T. Mahmood, "Design of a Smart Control and Protection System for Three-Phase Generator Using Arduino", Proceedings of the The Fourth Scientific Conference for Engineering and Postgraduate Research, Baghdad, Iraq, 16-17 December 2019.

[4] S. Rustemli, M. Ates, "Measurement and Simulation of Power Factor using PIC16F877", PRZEGLĄD ELEKTROTECHNICZNY Journal, Vol. 2012, No. 6, 2012, pp. 290-294.

[5] A. A. Mukaila, I. Olugbemi, E. E. Sule, "Design and Construction of A Single-Phase Power Factor Meter", Electrical Science & Engineering, Vol. 2, No. 2, 2020, pp. 11-14.

[6] M. Pampalle S. Srikanth, V. C. J. Mohan, T. R. Babu, "IOT-Based Automatic Power Factor Correction Unit for Industries to Curtail the Retribution of Power Utility Companies", Proceedings of Smart Systems: Innovations in Computing, Vol. 235, 2021.

[7] K. Parimala, K. C. R. Nisha. "FPGA Based Power Quality Monitoring Using FFT Method For Single Phase Power Metering", Proceedings of the International Conference on Emerging Technological Trends, Kollam, India, 21-22 October 2016.

[8] B. Rija, M. K. Hussain, A. M. Vural. "Microcontroller Based Automatic Power Factor Correction for Single-Phase Lagging and Leading Loads", Engineering, Technology & Applied Science Research, Vol. 10, No. 6, 2020, pp. 6515-6520.

[9] O. Andrei et al. "IoT Power Monitoring Device Using Wi-fi and Arduino", Proceedings of the 9th International Conference on Modern Power Systems, Cluj-Napoca, Romania, 16-17 June 2021.

[10] C. M. Diniș et al. "Power Factor Measuring Device Using Microcontroller for Single-Phase Consumers", Proceedings of the International Conference on Applied Science, Vol. 1781, 2021, pp. 1-13.

[11] F. P. Marafao, S. M. Deckmann, J. A. G. Marafao. "Power Factor Analysis Under Non-sinusoidal and Unbalanced Systems", Proceedings of the 10th International Conference on Harmonics and Quality of Power, Vol. 1, 2002.

[12] D. Slomovitz, "Behavior of Power-factor Meters Under Non-Sinusoidal Current and Voltage", In-

ternational Journal of Electronics Theoretical and Experimental, Vol. 70, No. 4, 1991, pp. 827-838.

[13] P. P. Machado et al. "Power Factor Metering System Using Arduino", Proceedings of the IEEE Workshop on Power Electronics and Power Quality Applications, Bogota, Colombia, 31 May - 2 June 2017.

[14] T. S. Gunawan et al. "Development of Power Factor Meter Using Arduino", Proceedings of the IEEE 5th International Conference on Smart Instrumentation, Measurement and Application, Songkhla, Thailand, 28-30 November 2018