An Arduino-Based System for Monitoring and Protecting Overvoltage and Undervoltage

Doan Duc Tung
Faculty of Engineering and Technology
Quy Nhon University
Quy Nhon City, Binh Dinh, Vietnam
doanductung@qnu.edu.vn

Ngo Minh Khoa
Faculty of Engineering and Technology
Quy Nhon University
Quy Nhon City, Binh Dinh, Vietnam
ngominhkhoa@qnu.edu.vn

Abstract—**In this paper, an experimental system based on Arduino Uno microcontroller board was developed for measuring electrical quantities and protecting overvoltage and undervoltage conditions in a single-phase power supply. The main components are described in detail, including the hardware and software to build the system in which an Arduino Uno platform is implemented as a vital microcontroller to read voltage and current measurement from a voltage sensor (ZMPT101B) and a current sensor (ACS712) respectively. In addition, the Arduino is also used to send all measurement to a PC through a serial port for monitoring the measured data graphically. The proposed method for designing the software uses the root mean square (RMS) method for measuring electrical quantities and then the RMS voltage measurement is compared with the minimum voltage and the maximum voltage to switch on or switch off the load. The monitoring GUI is designed with the use of the free and open-source Telemetry Viewer v0.5 software to monitor RMS voltage, RMS current, active power, and trip signal of the experimental system. The experimental results in this paper demonstrate that the system operated correctly.**

Keywords-experiment system; overvoltage; undervoltage; real-time measurement; voltage relay

I. INTRODUCTION

In modern power systems, voltage quality is becoming increasingly important with their growth in power electronics and their high sensitivity of electronic equipment. Voltage quality covers a wide range of voltage disturbances and deviations in voltage magnitude or waveform from the optimum values. Disturbances related to voltage quality could occur as a consequence of the operation of the power grid and/or of units connected to the grid. On the other hand, voltage irregularities are the major issues the industry and home users are facing and are often responsible for damages in sensitive equipment [1-2]. Overvoltage or undervoltage often result in damaging equipment. Electronic loads increase day by day and they are very sensitive to voltage variations [3].

In [4], authors proposed a method for monitoring voltage disturbances based on discrete wavelet transform and adaptive linear network. The types of voltage disturbances studied included voltage sag, voltage swell and voltage interruption. After applying the method, the voltage disturbances were classified according to IEEE Std. 1159-2009. Authors in [5] proposed an algorithm based on a deep belief network for online voltage sags recognition. The causes and waveform characters of voltage sags were analyzed. Then according to the characters of different sag waveforms, ten voltage sag characteristic parameters were proposed and proven to be effective, and finally a deep belief network model was built using the parameters to complete automatic recognition of the sag event types. An adaptive approach for voltage sag automatic segmentation was presented in [6]. An adaptive threshold for different waveforms was addressed, in which sag depth, mean square error, and entropy of the sag waveform were considered. Because of the effects of voltage sags on power system operation, the dynamic voltage restorer, one of the most effective solutions for mitigating voltage sag in power system, was proposed in [7-8]. Authors in [7] presented a configuration and control strategy for dynamic voltage restorer which used a closed-loop PI control law in the d-q reference frame and a linear Kalman filter was employed to estimate three-phase voltage sag. In [8], a new control method of multifunctional dynamic voltage restorer for voltage quality correction was presented. The control method was designed in the stationary frame by combining proportional resonant controllers and sequence-decouple resonant controllers. The method can mitigate many different conditions such as voltage swell, voltage sag due to symmetrical and unsymmetrical short circuit, starting of motors, and voltage distortion. The aim of this work was to design and implement an experimental system based on an Arduino Uno microcontroller board that can monitor electrical quantities and protect over/under voltage in a power supply. The voltage sensor (ZMPT101B) and the current sensor (ACS712) are embedded as the sensors which transmit continuously real-time data of a single-phase power supply to the Arduino Uno board, and then the data are processed to measure all necessary measurements.

II. HARDWARE DESIGN

The hardware of the experimental system is designed to measure electrical quantities and then to protect overvoltage and undervoltage conditions in single-phase power supply as shown in Figure 1. It includes the following main components: The voltage sensor ZMPT101B used in this work is a voltage transformer ideal to measure the single-phase AC voltage. It

Corresponding author: Ngo Minh Khoa

has high accuracy, good consistency for voltage and power measurement and it can measure up to 250V AC. It is simple to use and comes with a multi turn trim potentiometer for adjusting the analog-digital converter output. Its features consist of measurement within 250V AC, onboard micro-precision voltage transformer, analog output corresponding quantity can be adjusted, good consistency, for voltage and power measurement and high efficiency and accuracy. The current sensor ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, automotive, commercial, and communications systems. The device package allows easy implementation by the customer. Its typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection. In this paper it is used to measure the AC current of the single-phase power supply. The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller, it just simply needs to be connected to a computer (or appropriate wall power adapter) with a USB cable or with a AC-to-DC adapter or battery to get started. In addition, other components are used to build the experimental system such as an LCD to display all data measurements, a relay to switch the circuit when overvoltage or undervoltage occurs, push buttons for automatic/manual mode, trip and close command of the system. The completed experimental system for protecting overvoltage and undervoltage in this work built from the aforementioned components is shown in Figure 2.

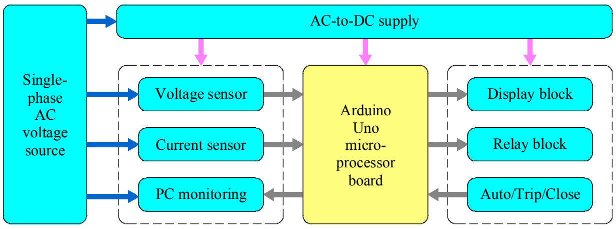
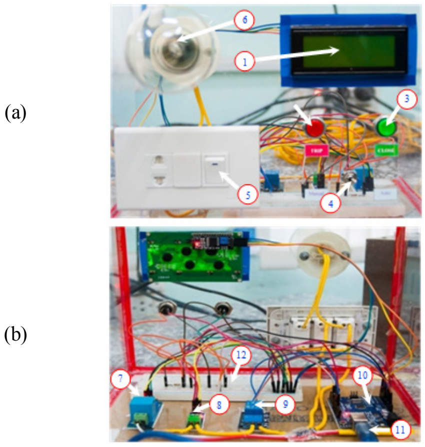


Fig. 1. The single-phase overvoltage and undervoltage protection system based on Arduino Uno.

III. SOFTWARE DESIGN

A. Real-Time Measurement of Electrical Quantities

In general, the output signals of the voltage sensor ZMPT101B and the current sensor ACS712 are sinusoidal voltage waveforms which have their amplitude between 0V and 5V in order to coincide with the analog input of the Arduino Uno. The signals are continuously sampled with a specific sampling time which depends on the processing speed of the microcontroller board. For the Arduino Uno microcontroller board, both voltage and current signals from the voltage and current sensors have sampling time T_s=4ms. The AC voltage source used to supply the experimental system in this work has a fundamental frequency f=50Hz so the cycle duration is 20ms. This means there are 5 samples per cycle of the fundamental frequency.



1. LCD, 2. Trip button, 3. Close button, 4. Auto/Manual switch, 5. Switch of the bulb, 6. 220VAC 60W bulb, 7. Voltage sensor ZMPT101B, 8. Current sensor ACS712, 9. Relay, 10. Arduino Uno, 11. USB cable, 12. Breadboard

Fig. 2. The overvoltage and undervoltage protection system: (a) front (b) back

It is assumed that the equations representing the value of the k^{th} sample of voltage and current from the analog pin A0 and A1 of the Arduino are: The voltage value at the kth sample is below:

$$v(k) = V_m \sin(k\omega T_s) \tag{1}$$

where V_m is the voltage amplitude, ω=2πf (f is the fundamental frequency), T_s is the sampling time (s), and k is the k^{th} sample.

The RMS voltage value can be measured as follows:

$$V_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} v_k^2} \tag{2}$$

where v_k is the voltage at the k^{th} sample and N is the total number of samples.

The current value at the k^{th} sample is:

$$i(k) = I_m \sin(k\omega T_s - \varphi) \tag{3}$$

where I_m is the current amplitude, and φ is the initial phase angle of the current.

The RMS current value can be measured as:

$$I_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} i_k^2} \tag{4}$$

where i_k is the current at the k^{th} sample.

The instantaneous power at the k^{th} sample is:

$$p(k) = v(k) \times i(k) \tag{5}$$

Therefore, the active and apparent power are defined as:

$$P = \frac{1}{N}\sum_{k=1}^{N} p(k) = \frac{1}{N}\sum_{k=1}^{N} v(k)\,i(k) \qquad (6)$$

$$S = V_{rms} \times I_{rms} \qquad (7)$$

B. The Proposed Algorithm

In this research work, the Arduino software is applied to write code and the code is then uploaded to the Arduino Uno board. The code is written according to the algorithm shown in Figure 3 to measure electrical quantities such as voltage, current, and active power and to protect overvoltage and undervoltage. The proposed algorithm includes three main stages. Stage 1 is to initialize parameters which represent the selection of auto/manual operation mode (if *Auto*=0 the manual mode is selected and if *Auto*=1 the automatic mode is selected), the trip signal to the relay (if *Relay*=0 the circuit is closed and if *Relay*=1 the circuit is open), and the position of the trip button and close button (if *Trip*=1 the trip button is pressed and if *Close*=1 the close button is pressed). Then, the electrical quantities including voltage, current, and power are measured in Stage 2. In this stage, voltage and current samples from the voltage sensor and the current sensor are continuously read from pins A0 and A1 of the board. Then the samples are used to calculate RMS voltage, RMS current, apparent power, and active power based on the method mentioned in the previous section. After the RMS voltage is measured at Stage 2, it is used as an input of Stage 3 to detect and protect overvoltage and undervoltage. In Figure 3, the protection can be accomplished by automatic or manual mode. For automatic mode, the RMS voltage will be compared with the minimum voltage (V_{min}) and the maximum voltage (V_{max}) to detect and protect overvoltage and undervoltage automatically. The proposed algorithm's steps are:

Step 1. Start.

Step 2. Initialize the parameters including the selection of auto/manual operation mode (*Auto*), the trip signal to the relay (*Relay*), and the position of the trip button and close button (*Trip* and *Close*):

$$Auto = 0; Relay = 0; Trip = 0; Close = 0 \qquad (8)$$

Step 3. Set the variables including voltage square (v_{square}), current square (i_{square}), instantaneous power (p_{inst}), and sample (k):

$$v_{square} = 0; i_{square} = 0; p_{inst} = 0; k = 1 \qquad (9)$$

Step 4. Read the voltage sample $v(k)$ and the current sample $i(k)$ from pins A0 and A1 of the Arduino Uno board where the voltage sensor ZMPT101B and the current sensor ACS712 are connected.

Step 5. Update v_{square}, i_{square}, and p_{inst} using the values of voltage and current samples in Step 3 as follows:

$$\begin{cases} v_{square} = v_{square} + v(k) \times v(k) \\ i_{square} = i_{square} + i(k) \times i(k) \\ p_{inst} = p_{inst} + v(k) \times i(k) \end{cases} \qquad (10)$$

Step 6. If k=N, go to Step 7. Otherwise, set k=k+1 and go back to Step 5.

Step 7. Calculate RMS voltage (V_{rms}), RMS current (I_{rms}), active power (P), and apparent power (S) using (11).

$$V_{rms} = \sqrt{\frac{v_{square}}{N}}; I_{rms} = \sqrt{\frac{i_{square}}{N}}; P = \frac{p_{inst}}{N} \qquad (11)$$

Step 8. If Auto=1, go to Step 9. Otherwise, go to Step 12.

Step 9. If V_{rms}<V_{min} the relay will be tripped (*Relay*=1). Otherwise, go to Step 10.

Step 10. If V_{rms}>V_{max}, the relay will be tripped (*Relay*=1). Otherwise, go to Step 13.

Step 11. If Trip=1, the relay will be tripped (*Relay*=1). Otherwise, go to Step 12.

Step 12. If Close=1, the relay will be closed (*Relay*=0). Otherwise, go to Step 13.

Step 13. If we need to stop, go to Step 14. Otherwise, back to Step 3.

Step 14. End.

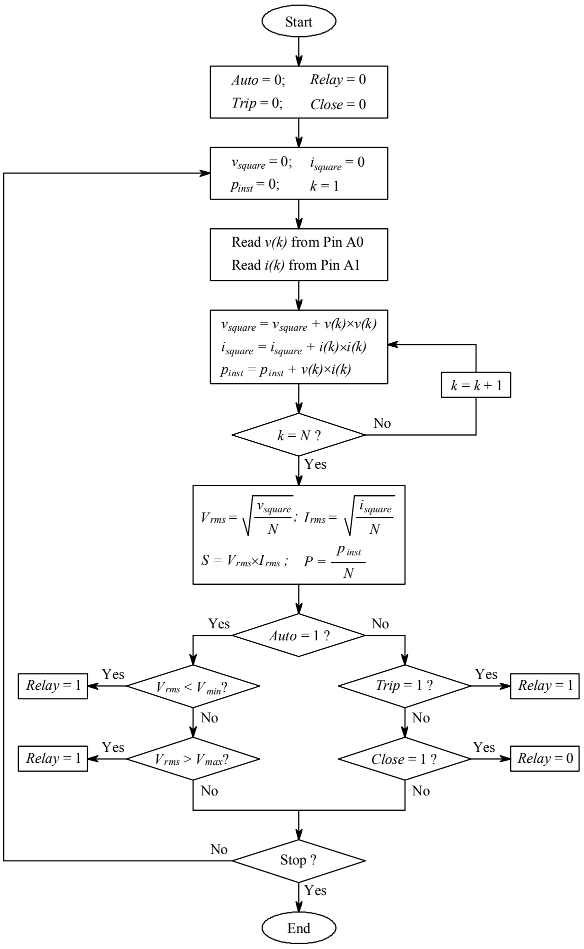


Fig. 3. The proposed algorithm of the experimental system for measuring electrical quantities and protecting overvoltage and undervoltage.

IV. RESULTS AND DISCUSSION

To test the overvoltage and undervoltage protection function of the experimental system, an experimental setup was established as shown in Figure 4. In Figure 4, three additional parts, namely a variable transformer, a PC monitoring, and a voltmeter are added to the setup. The key task of the variable transformer is to generate a variable voltage source. Therefore, various voltage conditions including normal voltage, overvoltage, and undervoltage can be created by tuning the knob on the front of the variable transformer. Then, its output voltage is used to supply the experimental system. All electrical quantities are measured and displayed locally on the LCD on the front of the system. Moreover, the status of the electrical circuit which is closed or open is also monitored and displayed on the LCD. In order to validate the voltage measurement of the experimental system, the voltmeter is added and connected to the output of the variable transformer. The measurement results from the experimental system and the voltmeter can be compared. Moreover, the monitoring software on a PC connected to the experimental system through a USB cable is designed to monitor all electrical quantities remotely. All measurements are continuously transmitted from the Arduino Uno board to the PC and are displayed on the PC's monitor graphically. From the monitoring software, we can observe and save all measurement data to a csv file, therefore, we can use the data for any further analysis.

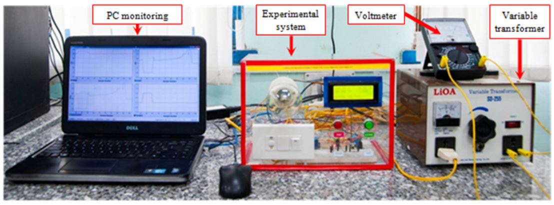


Fig. 4. The set of the experimental system for testing overvoltage and undervoltage protection.

In order to measure RMS voltage and current of a circuit, the voltage and current signals from the voltage sensor and current sensor are sampled in the Arduino Uno. With the processing speed of the Arduino Uno, the sampling time for both voltage and current signals equals to 4ms. This means there are 5 samples of voltage or current per cycle. The AC power supply generated by a variable transformer is nominally 230V AC 50Hz. Using the variable transformer to generate different voltage conditions, in this paper, one voltage condition is established by reducing the voltage magnitude in a period of about 10s. The voltage waveform is recorded in the csv format file by the monitoring software and then it is plotted in Matlab as shown in Figure 5(b). Moreover, we can obviously see the voltage waveform using its magnification from 5.2s to 5.6s as shown in Figure 5(a). To consider the current through the circuit, a bulb is assumed as a fixed load of the experimental system. In addition, other loads can be connected to the system through the outlet on the front of the system. When the load changes, the current waveform through the circuit recorded by the monitoring software also changes. Figure 5(d) shows a current waveform plotted in Matlab from the data of the csv file. Similarly, we can obviously see the current waveform using its magnification from 5.2s to 5.6s as shown in Figure 5(c). The time in Figure 5(c) is 0.4s corresponding to 20 cycles, therefore, there are 100 samples in the Figure. From these samples of voltage and current, the proposed algorithm is applied to measure the aforementioned electrical quantities RMS voltage, RMS current, active power, etc. and then the RMS voltage is used for detecting and protecting overvoltage and undervoltage.

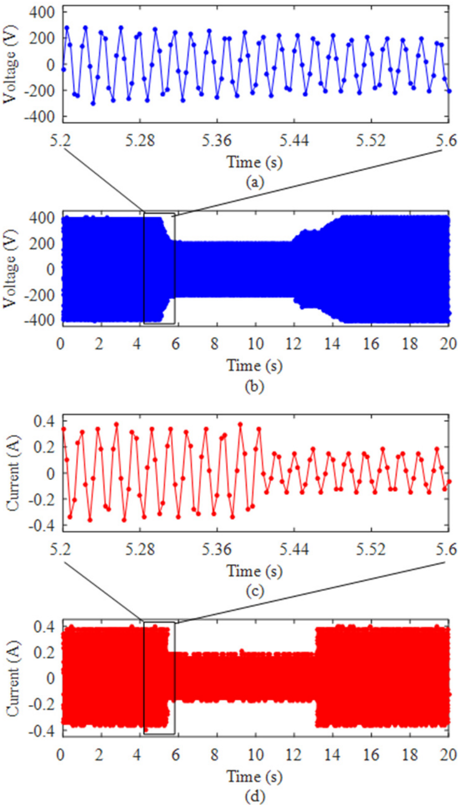


Fig. 5. Voltage and current waveforms. (a-b) Voltage waveform from the pin A0; (c-d) current waveform from the pin A1.

The proposed algorithm is uploaded to the Arduino Uno microcontroller board and is used to measure electrical quantities. The measurement results of the RMS voltage and current are recorded and are presented in Figures 6(a) and 6(b), respectively. With 4ms sampling time of the experimental system, the RMS voltage is calculated and is shown as the blue line in Figure 6(a). However, to display the RMS voltage value on the LCD on the front of the experimental system, the RMS voltage value is updated every 1.0s. In this case, the source voltage is reduced from 5s to 14s as shown in Figure 6(a), therefore, the RMS voltage measured from the proposed method is also reduced. Consequently, the experimental system can use the RMS voltage value to detect and protect the load from an overvoltage or undervoltage condition in the power supply, if the voltage is over the maximum value (V_{max}) or under the minimum value (V_{min}). Depending on the nominal voltage of the grid, V_{max} and V_{min} can be set at different values and in this study they are set at 220V and 180V respectively. Moreover, the current through the circuit is also measured in order to calculate the power consumed by the load of the system. The current waveform shown in Figure 5(d) is applied to test the method for calculating the RMS current and this result is shown in Figure 6(b). Similarly, the current is also updated to display on the LCD of the system per 1.0s shown by the black dots in Figure 6(b).

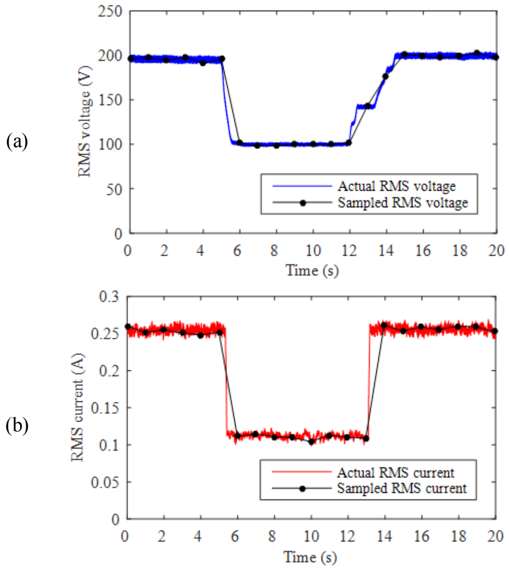


Fig. 6. Voltage and current measurements using the proposed algorithm on Arduino Uno.

According to the design of the experimental system presented above, there are two operation modes, the automatic mode and the manual mode. In the manual mode, the two buttons, namely the trip and closed button in the front of the system are used to switch the relay manually. In this design, the trip signal generated by pressing the trip button will be processed in the Arduino Uno microcontroller board and then transmitted to the relay to switch off the circuit. On the other hand, the closed signal generated by pressing the closed button will be also processed in the Arduino Uno microcontroller and then transmitted to the relay to switch on the circuit. Therefore, if there is an overvoltage or undervoltage condition in the power supply, the trip button can be used manually to protect the load. After that, if the voltage of the power supply is remained within V_{min} and V_{max}, the closed button can be also used manually to switch on to continue supplying power. To evaluate the manual mode of the experimental system, a sequence of the trip-closed-trip-closed order is established while the voltage of the power supply will remain within acceptable limits as shown in Figure 7(a). Because the trip and closed buttons are pressed according to the order shown in Figure 7(d), the relay will switch off and switch on the circuit, therefore, the current and the active power of the circuit will be changed as shown in Figures 7(b) and 7(c) respectively. In the automatic mode, the system will trip overvoltage and undervoltage conditions to isolate the load out of the power supply. Then the load will be supplied again if the voltage of power supply is between V_{min} and V_{max}.

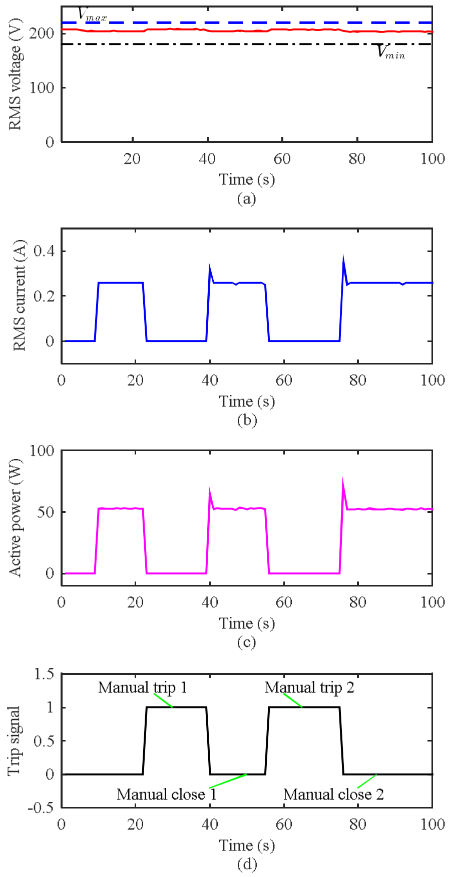


Fig. 7. The experimental results of the manual mode: (a) RMS voltage, (b) RMS current, (c) active power, and (d) trip signal.

In order to evaluate the automatic mode, the overvoltage and undervoltage conditions generated by the variable transformer are recorded in a period of 20s and are displayed on the PC monitoring software. The experimental results are saved in a csv file and Matlab is used to plot all the results as shown in Figure 8 which exhibits the experimental results including RMS voltage, RMS current, active power, and the trip signal as shown in Figures 8(a)-(d) respectively. In this normal condition, the load of the system is a 220V AC 60W bulb, therefore, the RMS current and the active power shown in Figures 8(b)-(c) are consumed by the bulb. However, the RMS current and the active power are equal to zero when there is an undervoltage or overvoltage condition in the power supply. As shown in Figure 8(a), an undervoltage occurs between 20s and 40s and an overvoltage occurs between 55s and 75s. The overvoltage and undervoltage protection of the experimental system is shown obviously in Figure 8(d) which demonstrates the trip signal transmitted to the relay in the system to protect the bulb. In addition, when the voltage of the power supply remains within V_{min} and V_{max}, the closed signal is transmitted to the relay to continue supplying the bulb with power. This is also demonstrated as the period from 40s to 55s in Figure 8(a).

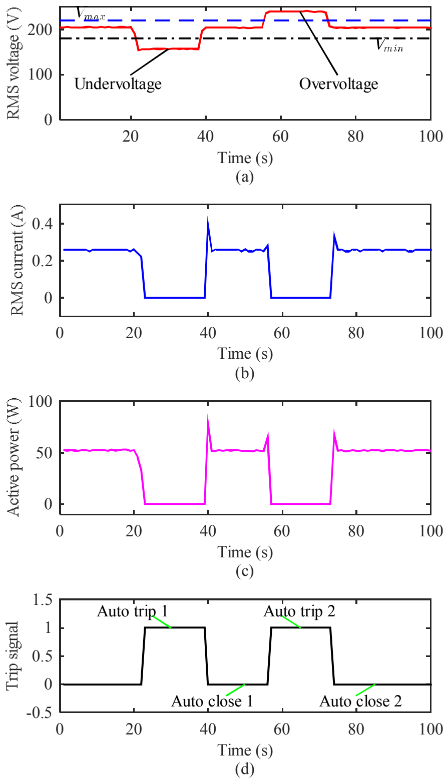


Fig. 8. Experimental results of the automatic mode: (a) RMS voltage, (b) RMS current, (c) active power, and (d) trip signal.

V. Conclusions

In this research work, an experimental system based on the Arduino Uno microcontroller board was developed for measuring electrical quantities and protecting onvervoltage and undervoltage of a single-phase power supply. The hardware and software of the system were designed and tested in a laboratory at our University. In order to evaluate the effectiveness of the system, we generated many different scenarios including automatic and manual mode to demonstrate and test the operation of the system. The real-time measurement results are locally monitored at an LCD on the front of the experimental system. Moreover, they can be graphically monitored through the monitoring GUI software on a PC and they can be also saved in a csv file for further analysis. The main contribution of this work is the propose of an experimental method to build a system based on an Arduino Uno microcontroller board for protecting overvoltage and undervoltage in single-phase power supplies and to implement a simple method which is uploaded to the board for measuring electrical quantities.

References

[1] C. Sankaran, Power Quality, CRC Press, 2001

[2] A. G. Siraki, P. Pillay, “Comparison of two methods for full-load in situ induction motor efficiency estimation from field testing in the presence of over/undervoltages and unbalanced supplies”, IEEE Transactions on Industry Applications, Vol. 48, No. 6, pp. 1911-1921, 2012

[3] M. H. J. Bollen, I. Y. H. Gu, Signal Processing of Power Quality Disturbances, Wiley-IEEE Press, 2006

[4] D. T. Viet, N. H. Hieu, N. M. Khoa, “A method for monitoring voltage disturbances based on discrete wavelet transform and adaptive linear neural network”, International Review of Electrical Engineering, Vol. 11, No. 3, pp. 314-322, 2016

[5] F. Mei, Y. Ren, Q. Wu, C. Zhang, Y. Pan, H. Sha, J. Zheng, “Online recognition method for voltage sags based on a deep belief network”, Energies, Vol. 12, No. 1, pp. 1-16, 2018

[6] X. Xiao, W. Hu, H. Zhang, J. Ai, Z. Zheng, “An Adaptive Approach for Voltage Sag Automatic Segmentation”, Energies, Vol. 11, No. 12, ArticleId 3519, 2018

[7] D. T. Viet, N. H. Hieu, N. L. Hoa, N. M. Khoa, “A Control Strategy for Dynamic Voltage Restorer”, 11th IEEE International Conference on Power Electronics and Drive Systems, Sydney, Australia, June 9-12, 2015

[8] D. V. Tien, R. Gono, Z. Leonowicz, “A Multifunctional Dynamic Voltage Restorer for Power Quality Improvement”, Energies, Vol. 11, No. 6, ArticleId 1351, 2018

[9] C. Moron, D. Ferrandez, P. Saiz, G. Vega, J. P. Diaz, “New Prototype of Photovoltaic Solar Tracker Based on Arduino”, Energies, Vol. 10, No. 9, ArticleID 1298, 2017

[10] P. P. Machado, T. P. Abud, M. Z. Fortes, B. S. M. C. Borba, “Power Factor Metering System Using Arduino”, IEEE Workshop on Power Electronics and Power Quality Applications, Bogota, Colombia, May 31-June 2, 2017

[11] E. Viciana, A. Alcayde, F. G. Montoya, R. Banos, F. M. Arrabal-Campos, F. Manzano-Agugliaro, “An Open Hardware Design for Internet of Things Power Quality and Energy Saving Solutions”, Sensors, Vol. 19, No. 3, ArticleId 627, 2019

[12] T. J. Ho, C. H. Chang, “Robust Speed Tracking of Induction Motors: An Arduino-Implemented Intelligent Control Approach”, Applied Sciences, Vol. 8, No. 2, ArticleID 159, 2018