Accurate Measurement of the Mains Electricity Frequency

Dogan Ibrahim

Near East University, Faculty of Engineering, Lefkosa, TRNC
dogan@neu.edu.tr

Abstract

The frequency of the mains electricity supply is very important in many industrial and commercial applications. This paper describes the design of a microcontroller based system for measuring the mains frequency accurately. The measured frequency is displayed on an LCD, and in addition the measured data is sent to a PC continuously for time-stamped logging and offline analysis.

1. Introduction

There are several mains electricity power systems in use around the world [1]. These different systems are characterised by their:

- Voltage
- Frequency
- Type of plugs and sockets used

In general, the type of plugs and sockets used is not a problem and passive adapters are available to convert between different varieties as long as the voltage and frequency are correct for the electrical device to be used.

In general, we can divide the mains voltage and frequency usage in the world into four groups:

- 100-127V, 50Hz
- 100-127V, 60Hz
- 220-240V, 50Hz
- 220-240V, 60Hz

The voltage quoted is the root mean square, and the peak voltage can be calculated by multiplying the voltage with $\sqrt{2}$, or the peak-to-peak voltage is found by multiplying with $2\sqrt{2}$. The frequency used in mains electricity is either 50Hz (20ms period) sinusoidal or 60Hz (16.66ms period) sinusoidal.

Some of the appliances we use at home may be affected if the mains voltage is not correct. For example, the motor speed of some CD players may be affected even though the motor supply voltage is regulated. The result of this is that the music can play slightly slower or faster. Also, some more sensitive appliances such as televisions may not operate correctly if the mains voltage is lowered.

As the demand on the electricity supply increases the frequency usually drops. The electricity suppliers monitor the mains frequency constantly and aim to keep it within the allowed tolerances. A change in the mains frequency has negative effects in some of the appliances. For example, the speed of AC synchronous motors depend upon the supply frequency, and any appliance such as a turntable using such a motor will run slower or faster depending upon the changes. Also, some home or industrial clocks operate by counting the mains pulses, and such clocks will run slower or faster depending upon the changes in the supply frequency.

By law, the Electricity Board is required to supply 230V + 10% ,-6%. i.e. between 216.2 volts and 253 volts. In addition, the frequency is required to be maintained at 50Hz ± 1%. i.e. between 49Hz and 51Hz. The European standard supply voltage is 220V and back in 2008 it was decided to change the European standard voltage to 230V and also the UK mains voltage to 230V. Although this sounded easy in theory, in practise it was too costly and uneconomic to change all the supply equipment to give 230V. Instead, the voltage limits were changed in UK, and the UK voltage was kept at 240V which is within the allowed limits of 230V + 10% ,-6%. At the same time, the European standard was left at 220V.

2. Measuring the Mains Frequency

The expected variation of the mains frequency (in UK) is between 49Hz and 51Hz. The mains frequency can in practise be measured using a simple frequency counter. But here the problem is that we need to measure very small changes, in the order of less than 0.01% and the cost of frequency counters to measure such small changes are rather high. In addition, we usually want to log the variations of the mains frequency over long periods of time and then to analyse these changes by for example plotting the results. Most low cost frequency counters are not capable of logging the frequency changes.

In this paper, the design of a microcontroller based mains frequency measuring and logging device is given. The device has an LCD display that shows the frequency changes in real-time. In addition, the frequency data is sent to a PC over the RS232 serial port and stored in a file on the PC. The stored data can easily be analysed for example by plotting it or by using a statistical analysis package such as Excel. The stored data is time-stamped by the PC software so that the actual date and time of each data point is available. The data can be stored for very long times, such as for weeks, months, or even years.

3. The Hardware

There are basically two methods used in the literature for frequency measurement. The first method, which is not accurate, involves setting up a time window and calculating the number of cycles within this window. The second method which is more accurate and is the one used in this paper, involves calculating the period of the waveform. Here, basically an accurate timer is used to measure the period and hence calculate the frequency of the waveform.

The block diagram of the designed mains frequency monitoring device is shown in Fig. 1. The operation of the device is based on a near-zero-cross-detector circuit [2]. Mains supply is reduced to 9V using a wall mains adapter. As shown in Fig. 2, the near-zero-cross-detector circuit is made up of a bridge rectifier and a transistor. Full-wave rectified mains signal is applied to the base of the transistor. The transistor is normally on and its output is low when the signal is high. As the signal drops to 0.7V, the transistor turns off and the collector voltage rises to the supply voltage (+5V), generating a pulse. Fig 3 shows the rectifier output and the transistor output. As shown in the figure, three such pulses are obtained during a full period of the mains frequency. These pulses are then fed to one of the inputs of a PIC microcontroller. The microcontroller starts an accurate timer when a pulse arrives. The timer is stopped at the arrival of the third pulse. Thus, the timer count is proportional to the period and hence to the frequency of the waveform. This timer count is converted into real frequency and is displayed on an LCD display. In addition, the data is sent to a PC using the serial RS232 port. A Visual Basic [3] program on the PC receives the frequency data, time stamps the data and then stores it in a file. The frequency data can be displayed by the Visual Basic program, or for more flexibility and statistical analysis it can easily be imported into Excel [4] and analysed or displayed.

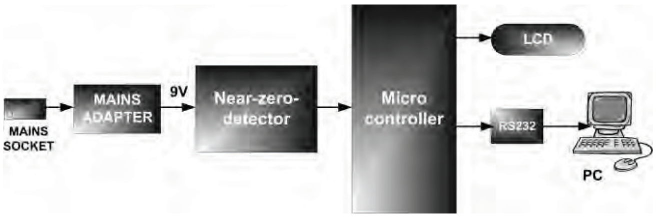


Fig. 1. Block diagram of the frequency monitoring device

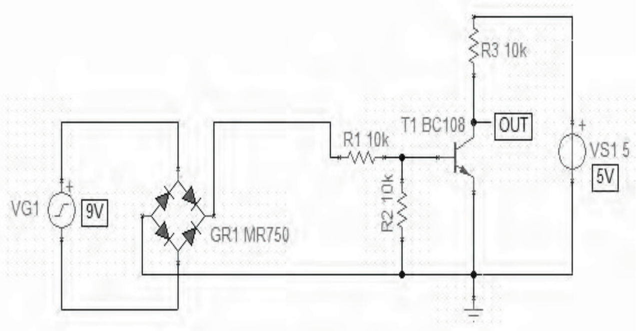


Fig. 2. The near-zero-cross-detector circuit

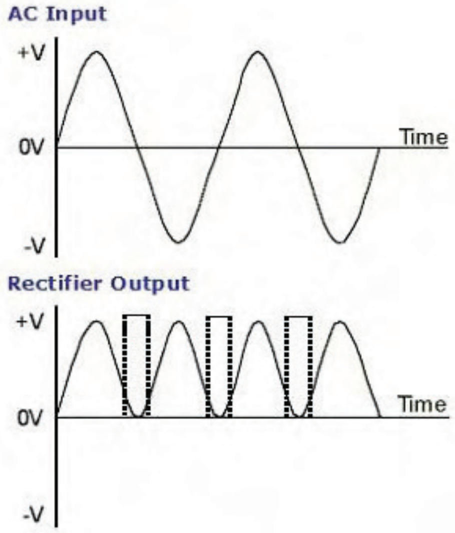
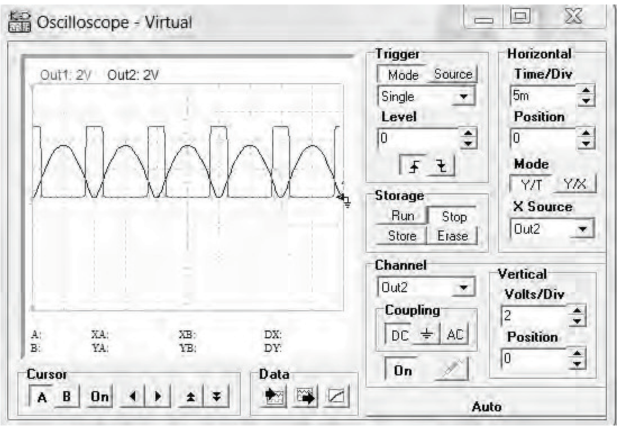


Fig. 3. Three pulses in one cycle of the waveform

Although the near-zero-cross-detector circuit does not detect the exact zero crossing point of the signal, it generates pulses at the same near zero points of the waveform, thus making it possible to accurately measure the signal period. Fig. 4 shows simulation of the near-zero-cross detection circuit. The circuit was simulated using the popular TINA circuit simulation suite [5], developed by DesignSoft. Fig. 4 shows the rectified full-wave mains waveform together with the output pulses of the near-zero-detector circuit on a virtual oscilloscope of TINA. TINA is a general purpose, low-cost Spice based simulation program that can be used to simulate complex analog and digital circuits. In addition, microcontroller systems can also be simulated using TINA. The circuit to be simulated is drawn using the built-in graphical editor of TINA. A large library of standard components are provided by the package. In addition, models of other components can be extracted from the manufacturers' data sheets over the Internet and they can be loaded into TINA. The package provides virtual instruments such as oscilloscopes and spectrum analyzers that enable the user to observe the waveforms. A compatible hardware kit can also be purchased and used with TINA for real-time simulation work.

Fig. 5 shows full circuit diagram of the device. A PIC18F4520 microcontroller [6] is used in the design with the timing provided with an 8MHz crystal. PORT B of the microcontroller is connected to a 2x16 character LCD display. UART output pin (RC6) is connected to a MAX232 type RS232-TTL voltage level converter chip and then to the PC serial port via a 9-pin D-type connector. Output pulses of the near-zero-detector circuit are fed to port pin RC2 of the microcontroller.



Out1: 2V Out2: 2V

Trigger Mode: Single; Level: 0

Horizontal Time/Div: 5m; Position: 0

X Source: Out2

Channel: Out2

Vertical Volts/Div: 2; Position: 0

Fig. 4. Simulation of the near-zero-detector circuit

The project was built and tested using the EasyPIC6 microcontroller development board [7], shown in Fig. 6, manufactured by mikroElektronika. A small breadboard was used to construct the near-zero-detector circuit and then wire connections were made to the main development board.

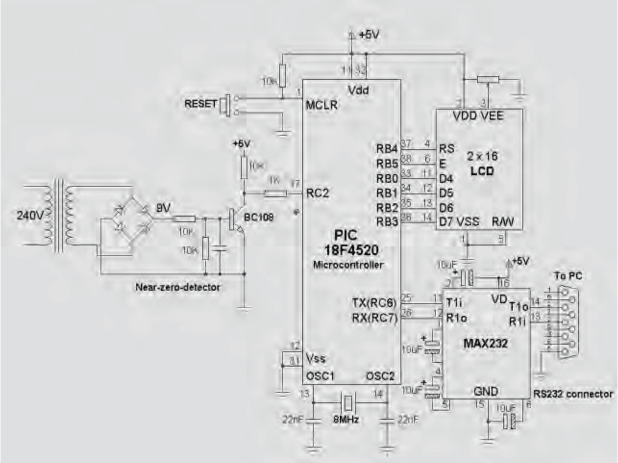


Fig. 5. Circuit diagram of the monitoring device

4. Operation of the Circuit

The output pulses of the near-zero-detector circuit are counted using 16 bit timer/counter TMR1 of the microcontroller. With an 8MHz crystal, the counting period is 0.5μs and maximum count is 65535. In a perfect 50Hz signal, with 20ms period, the maximum count will be 40,000. Table 1 shows the counter values at different frequencies of the mains supply. The frequency (f) of the waveform is then given in Hz by:

$$f = \frac{2x10^6}{count} \tag{1}$$

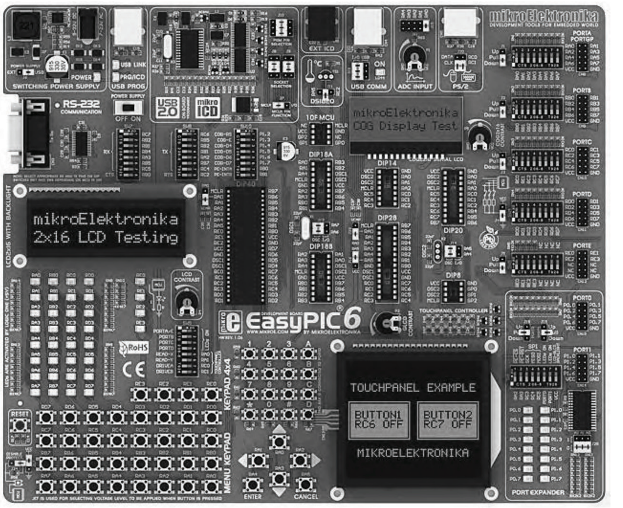


Fig. 6 The EasyPIC6 Development Board

By considering that a difference of one count can be measured, the accuracy of the frequency measurement is then given by approximately 0.001Hz, or 0.002%.

Table 1. Frequency and counter values

Frequency (Hz)	Counter value
49.0	40816
49.2	40650
49.4	40485
49.6	40322
49.8	40160
50.0	**40000**
50.2	39840
50.4	39682
50.6	39525
50.8	39370
51.0	39215

5. The Software

The software consists of the microcontroller software (or the measuring software), and the PC software (or the data logging software).

5.1 The Microcontroller Software

Fig. 7 shows operation of the microcontroller software in the form of a Program Description Language (PDL). Counter TMR1 is cleared and internal counting starts on the high-to-low transition of the first input pulse on pin RC2 (see Fig. 5). The counting continues until the third pulse is detected on pin RC2, and stops on the high-to-low transition of the third pulse (i.e. after a complete cycle of the waveform is received). The frequency is then calculated using the equation given in (1) above. Floating point calculations are used in the program for high accuracy.

The calculated frequency is displayed on the LCD in real-time, as well as it is sent to the PC over the serial link. The PC receives the frequency, adds date and time data to each record and then stores each record in a file for offline analysis. This process is repeated forever with a 5 second delay between each measurement (this time can easily be changed if required). Time-critical parts of the program are written in Assembly language so that the pulse edges can be captured quickly and accurately. Fig. 8 shows a typical display of the measured frequency on the LCD.

```
BEGIN
        Initialise global program variables
        Configure LCD
        Configure UART
        DO FOREVER
                Wait for low-to-high transition of pulse
                Clear timer/counter TMR1
                Start timer/counter TMR1
                Wait for low-to-high transition of pulse
                Wait for low-to-high transition of pulse
                Get timer/counter value
                Calculate the frequency
                Display frequency on LCD
                Send frequency to RS232 port
                Wait 5 seconds
        ENDDO
END
```

Fig. 7. Operation of the microcontroller software

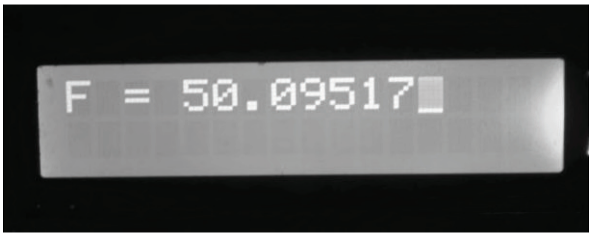


Fig. 8. Typical display of the measured frequency

The microcontroller software is based on the popular mikroC [8] language, which is a standard C language with additional support for libraries and features for PIC microcontroller programming. mikroC offers a Graphical User Interface (GUI) type design, enabling the user to program using the built-in editor. In addition, the compiler offers simulation and in-circuit-debugging (ICD) features, thus enabling the user to develop and test complex programs easily. mikroC is fully compatible with the EasyPIC6 development board so that a compiled program can very easily be downloaded to the program memory of the target microcontroller chip via the in-circuit-serial programming (ISP) tools provided on the development board.

5.2 The PC Software

The PC software reads measured frequencies from the microcontroller, inserts the current date and time, and then stores the time-stamped data in a file on the PC for offline processing. This program is based on Visual Basic 6 [3]. The user starts and stops data logging by clicking the appropriate buttons on the main form of the program. Data is stored with the fields being separated with a comma so that it can easily be imported to spreadsheet programs (e.g. Excel) for offline analysis. For example, the file can be opened in Excel, and the data can be read into separate columns by specifying that the data fields are separated by commas.

Fig. 9 shows a typical plot of the collected data after it is imported into Excel. Here, for demonstration purposes the data collection time was about 15 minutes. Notice that the absolute date and time information of each record is shown on the horizontal axis.

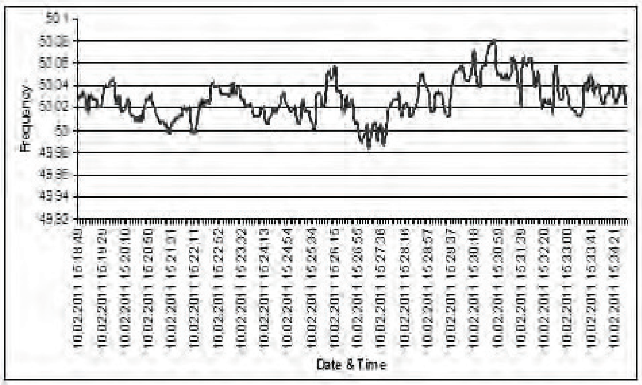


Fig. 9. Plotting the collected data using Excel

6. Conclusion

This paper has described the design of a microcontroller based system for measuring the mains electricity frequency accurately. The system is based on using an accurate counter module inside the microcontroller. The designed system provides measurement accuracy in excess of 0.001Hz. One of the nice things about this design is that the measured frequency data can easily be stored on a PC and then it can be analyzed offline using spreadsheet programs such as Excel.

The developed system can be enhanced further by including hardware and software to measure value of the mains voltage in addition to the frequency. Thus, both the frequency and the voltage variations can be displayed in real-time and also they can be stored on the PC for offline analysis.

Another possible improvement is in the physical PC interface. In the design presented here, the connection to the PC is via the RS232 serial port. Most computers nowadays offer USB ports as standard, and they do not have RS232 ports. It is possible to use a USB-RS232 adapter for connecting the development board to the PC. Alternatively, a USB hardware and software interface can be designed on the development board for direct connection to the PC.

7. References

[1] "Mains Electricity by Country", Web Site: http://en.wikipedia.org/wiki/Mains_electricity_by_country [Accessed on: 23 February, 2011].

[2] R. Elliot, "Zero crossing Detector and Comparators", Web Site: http://sound.westhost.com/appnotes/an005.htm, [Accessed on: 12 January, 2011].

[3] "Visual Basic User Guide", Microsoft Inc, Web site: http://www.microsoft.com,[Accessed on: 2 March, 2011].

[4] "Excel User Guide", Microsoft Inc., Web site: http://www.microsoft.com, [Accessed on: 3 March, 2011]

[5] "TINA User Manual", Web Site: http://www.tina.com, [Accessed on: 20 February, 2011].

[6] "PIC18F4520 Data Sheet", Web Site: http://microchip.com, [Accessed on: 10 January 2011].

[7] "EasyPIC6 Data Manual", Web Site: http://www.mikroe.com, [Accessed on: 12 March, 2011]

[8] "mikroC User Manual", Web Site: http://mikroe.com, [Accessed on: 10 December, 2010].