Abstract – Energy conservation in homes has become imperative due to rising energy costs, increasing energy consumption, and a world-wide shift in environmental protection concerns. Thus, there is a growing demand for new technologies that will help to provide energy conservation techniques. Automating the control of energy consumption in common household devices provides a starting point for establishing efficient usage. Through use of existing technologies, consumption statistics and the appropriate algorithms can be combined with the ability to remotely and independently control individual devices for the purpose of energy conservation. The home automation system proposed in this paper aims to provide a method for monitoring and controlling energy consuming devices common to households.

Keywords: home automation, energy conservation, microprocessors

1. Introduction

As the global population continues to increase, the resources required to sustain the energy demand are growing rapidly. As a result, countries and institutions around the world are becoming more aware of the need to conserve energy and are actively seeking new methods of decreasing per capita consumption [1].

Energy conservation in households begins with the homeowner’s understanding of the methods that must be implemented and the sacrifices that must be made. Some general examples are: turning off unused lights, purchasing more energy efficient appliances, and unplugging unused devices to prevent idle energy consumption. However, these methods are often viewed as inconvenient and as having little impact. As a result, the many small contributions required for energy conservation are often not implemented.

Presently, there are few affordable technologies available to aid consumers in achieving a balance between energy conservation and convenience. The key to a solution is the ability to develop and provide the necessary products and methods while also establishing a balance between the many economic, social and environmental concerns.

Environmental concerns lie at the very core of the home automation system discussed in this paper. Optimizing energy conservation is directly correlated to decreasing the use of natural resources and reducing the overall impact on the environment. By combining the automated control of devices with a system capable of monitoring consumption, decreasing energy consumption at the device level can be easily addressed through the use of software.

As is often the case, economic concerns are a determining factor in system design. The costs associated with the process of automating a home currently are much more than the typical household is willing to spend. While automating homes during their construction would alleviate much of the cost, this does not provide a solution for existing homes. A system must be designed that can be easily retrofitted to existing homes.

Historically, standardization has been the key to successful and widespread implementation of new technologies [2, 3]. The proposed design focuses on making use of an existing wireless standard to develop a system that facilitates the automated metering and control of devices in existing households. Through the use of this technology, coupled with an easy to use graphical user interface, the energy consumption of individual devices can be monitored and automatically controlled via customizable algorithms.

The most important constraint of the proposed home automation system is the desire to design a product that is price competitive. In addition, the product will need to be adaptable to varying home designs; updating the entire electrical system of an existing home is simply not an option in most cases.

2. Embedded System Overview

The proposed system is the vision of an energy conservation solution that will provide a user-friendly, reliable, and accessible product that can be adopted and implemented wirelessly on a large scale [4-8]. Such a product will offer end-users the tools necessary to monitor and control the use of energy throughout their homes. By using well-known and well-supported open source hardware and software standards, long term support through existing online communities will be available.

The current version of the home automation system consists of a simple 120Vac, 6.3A design. The device plugs in to a standard 15A receptacle and provides a standard outlet rated at 6.3A. Electrical measurements (RMS voltage, RMS
current, Apparent Power, True Power, Kw/Hr., and Power Factor) are taken internally. An internally housed control relay provides convenient on/off capability at the touch of a button, while the measurements can be recorded for reference, or used to automatically control the device. The user can monitor and control individual devices wirelessly. With the addition of a gateway, an end user has access to their devices from anywhere in the world through an internet connection.

3. Design Objectives
The design objectives for the proposed home automation system encompass hardware and software specifications and project accessibility and are described more in what follows.

The hardware consists of ZigBee compatible hardware components [9-11] that are interoperable with ZigBee devices from multiple vendors. In addition, the hardware requires minimal user configuration and is reliable and safe for the end user. The software also is friendly as well as secure and stable. The project accessibility is ensured through the use open-source hardware/software and standards.

4. Implementation
Development of the proposed home automation system consists of two distinct sections: hardware and software. The seamless integration of the two requires the use of the various tools shown in Figure 1. Hardware level programming is used to allow for a more intricate interaction among the devices. The iDigi gateway is programmed with Digi ESP for Python, an IDE designed specifically for the Python language and the gateway. Finally, the design of a web interface is made possible by using Aptana Studio and the HTML and CSS languages.

4.1 Hardware
The hardware portion of the proposed home automation system directly interfaces with a variety of devices found in residences. The ZigBee RF wireless standard enables communication among individual devices. The system allows for the voltage, current, power, and energy to be measured at individual devices in addition to providing the means to control the device operations. The data obtained from the devices is transmitted across the ZigBee wireless mesh network to a gateway, which enables the use of a web-based user interface. A block representation of the system is shown in Figure 2.

The current system revision is rated for 120Vac and 6.3A, but can be easily modified to accommodate other voltages and larger currents. Through the use of voltage and current sensing transformers, the wireless metering and control circuit is interfaced with the line voltage as shown in Figure 3. The two low voltage power supplies (3.3 and 5 Vdc) are derived from the 120Vac line. By using a center-tapped, 120/30Vac transformer, and by connecting the center-tap to ground, two 15Vac waveforms (which are 180° out of phase) are half-wave rectified to produce a DC voltage. A large capacitor is incorporated to remove the DC ripple voltage. The two transformed AC phases can be viewed in Figure 4, while the DC output is shown in Figure 5. In order to provide constant DC voltages to the various circuit components, the rectified DC output was connected to 3.3 and 5Vdc voltage regulators.

Figure 1: Overview of System Design Aspects
The two regulated DC voltages are used to power the PIC microcontroller, the CS5460A energy metering IC, and an XBee RF module. The Cirrus Logic CS5460A is an integrated circuit, designed to measure and calculate energy, instantaneous power, and RMS voltage/current values for single-phase applications [12]. These measurements are obtained from the voltage and current sensing transformers. The $V_{\text{IN+}}$, $V_{\text{IN-}}$, and $I_{\text{IN+}}$ signals, shown in Figure 3, are conditioned before being input to the associated input terminals of the CS5460A. The resistor/capacitor networks used to condition these inputs are shown in Figure 6.

Data is written to and read from the CS5460A via an on-chip serial peripheral interface (SPI). Through the use of Microchip Technology’s PIC16F882 microcontroller [13, 14], data is read from the CS5460A, interpreted, and formatted. The PIC16F882 is also interfaced with an XBee RF module via the PIC16F882 universal synchronous/asynchronous receiver transmitter module. The XBee RF module [15] allows data to be transmitted and received by the system, thus allowing the ability to monitor and control devices at the individual level. Figure 7 and Figure 8 show the SPI and USART connections.
4.2 Software

The various software portions of the home automation system enable the full energy conserving potential of the system. By using assembly language to integrate the aforementioned hardware components with the ZigBee RF standard, the home automation system becomes a wireless device. Use of an iDigi ConnectPort X4 ZigBee to Ethernet gateway and the proper Python programming makes the system internet enabled. Finally, in order to provide a user friendly interface, the HTML and CSS programming languages are used to develop a web-based control interface.

Figure 7: SPI and Data Connections between the CS5460A and the PIC16F882

Figure 8: USART Connections between the XBee and PIC16F882

4.2.1 Assembly

The PIC16F882 microprocessor used in the system was programmed with the Microchip Technology assembly language, using the PICkit in-circuit debugger and programmer. In addition, MPLAB IDE and MPASM, Microchip Technology’s integrated development environment and assembler, are used to facilitate the development and testing of the source code. The use of assembly language to program the microcontroller allowed for a finer control of the device settings and operations.

The main task of the microcontroller, and thus the assembly code, is to coordinate the operations of the CS5460A and the XBee module. In addition, the processing power of the microcontroller is used to calculate full scale data values, from the scaled versions recorded by the CS5460A.

When powered on, the PIC microcontroller loads the internally stored variables and definitions. The next task of the assembly source code is the initialization of the CS5460A, the XBee, and the built-in peripheral modules. Once initialized, the CS5460A performs continuous calculations, and the XBee module is put to sleep. In future project implementations, the sleep functions of the CS5460A and the PIC16F882 will be incorporated.

Further operations of the system are performed by request. Using the web interface, the system can be asked to cycle the circuit on and off, or to report the current data values. When a request is made, the XBee is awakened from its sleep mode to relay the appropriate messages to and from the PIC16F882. A data request causes the PIC16F882 to read from the CS5460A, perform the necessary calculations, and return the data in a three byte packages. The first byte of each package is an identifier relating to the data being read (e.g. 0x16 refers to an RMS current reading). The remaining two bytes are the full scale values requested. A block diagram of the assembly code operation is shown in Figure 9.

4.2.2 Python and HTML/CSS

In order to enable internet connectivity to the system, the iDigi gateway must be programmed to recognize the devices with which it communicates. In addition, the gateway must be properly configured to accept data and relay it to the web interface.

The frontend of the home automation system is a user friendly GUI, that can be accessed from any internet connection. This web interface can be hosted by the Google App Engine and iDigi’s Client Web Service. By using a hosted web service, the need for server infrastructure within the home automation system is removed. Customers can use their devices from afar and check their applications reliably through Google servers.

Some of the items necessary for the development of the web interface are Digi’s Python Development Environment (DigiESP), the Python 2.5 programming language, and Google’s App Engine Software Development Kit.

The iDigi DIA projects include the drivers for the XBee module. These projects are created with Digi ESP and
uploaded to the ConnectPortX4 Gateway. Creating the project also enables the remote call interface handler presentation to enable the gateway to talk to the iDigi platform.

4.3 Final Design

The first step in completing the final design is implementing the Cirrus Logic CS5460A IC and associated signal conditioning hardware as shown in Figure 12. Next, the PIC16F882 is interfaced with the CS5460A. A crystal oscillator is used for the CS5460A while the PIC16F882 uses its internal oscillator. This allows the PIC16F882 to also provide the serial signal needed to establish SPI communications. The CS5460A serial timing diagram is included as Figure 13.

Figure 9: Assembly Program Flow Chart

A Google Appspot account must be established to use the servers. After the account is set up, the web interface is deployed to Google’s servers. When the webpage is enabled, it is then pointed to the iDigi developer URL, in order to retrieve the serial data from the ConnectPortX4 Gateway. Figures 10 and 11 are helpful in understanding how the languages and development environments are associated.

Figure 10: Python Program Flow Chart

Figure 11: HTML/CSS Program Flow Chart

With the ability to send and receive serial data to and from the CS5460A, the next design step is to access the various data registers and to format the data that is transmitted across the network. The registers of the CS5460A store hexadecimal data as scaled values, ranging from 0 to 1 and from -1 to 1. The PIC16F882 processes this data by multiplying the scaled value by the full scale value. This data is then placed into temporary variables, in order to be transmitted across the XBee network.
Before data can be transmitted via XBee, the USART module of the PIC16F882 needs to be properly initialized. This involves setting the proper control bits and establishing a baud rate to control the flow of data. Once the PIC16F882 is configured to communicate with the XBee via USART, the system is able to communicate with a local XBee enabled computer. The next step establishes communications between the XBee radios and the iDigi gateway. This is a matter of identifying the various XBee radios by their serial number, and by uploading a driver that receives and transmits serial data. The final hardware design is shown in Figure 14.

The final aspect of the home automation system is the user-friendly frontend. The iDigi development kit that accompanies the gateway provides access to the Digi Cloud, which allows the use of Google Apps. A complete web interface using these resources allows for data logging and plotting as well as a friendly environment for the user to interact with their smart devices.

5. Conclusions

The proposed home automation system is a new approach to home energy conservation. The system enables the consumer to conveniently reduce energy consumption. The easily incorporated design will allow consumers to retrofit their homes. As the technologies grow and as the standards are implemented, it is envisioned that appliances will be available with the proposed home automation system. Homeowners will simply log on to their computers and add their new devices to their home automation networks.

This system will have a significant impact on the home automation industry in the realm of energy conservation and environmentalism. The proposed system will merge the luxury of home automation controls with the necessity of reducing energy consumption.

6. References


[15] "XBee/XBee-Pro ZB RF Modules data sheet," Digi International Inc, Minnetonka, MN, USA