Cloud based low-cost Home Monitoring and Automation System

Shruthi Raghavan and Girma S. Tewolde
Electrical and Computer Engineering Department
Kettering University
Flint, MI 48504
Email: {ragh1714, gtewolde}@kettering.edu

ABSTRACT

This paper presents a cloud based low cost home automation system implemented using the Digilent chipKIT Uno32 and Arduino Uno. This is a proof of concept for a home that can be monitored and controlled remotely from anywhere in the world via the Internet. A "connected" home like this can make life more convenient and also safer. The remote monitoring aspect of this project demonstrates the ability of being able to know what is going on with different systems at home which can be used for control and safety. For example, we can monitor the temperature for ambient control, the state of some sensors for intruder detection and the state of different devices like fans or lights at home. It is also demonstrated using a few motors, how one can control different systems at home using the cloud service via the Internet. So a virtual "switch" available in the cloud UI can be toggled to turn on/off a fan or open/close a garage door at home. This project has a very large scope and can be integrated with many other systems like smart electronic appliances at home. This paper describes the project implementing the basic framework to achieve such a connected home. It gives a summary of hardware and software in the current implementation of the project, future improvements and scope.

I. INTRODUCTION

The cloud based home automation project is inspired by the necessity to monitor and control the home remotely in an affordable way. Today, technology innovations are being focused more than ever on the Internet of Things (IoT). A world where everything is smart and connected is already work in progress. The Sky Link Home Smart Center AM-100A and advanced home controllers like the ISY994i that operate only with INSTEON devices are few of the available solutions today. The majority of these require compatible appliances to be purchased posing a serious limitation to adapting these solutions. This project was inspired by the fact that integrated home automation systems today are expensive in terms of installation and regular maintenance. Some of them are difficult to operate due to the required interface learning during setup. The presented prototype aims to use as few resources as possible to demonstrate a simple and low cost connected home. It consists of a base node connected to a cloud service (hosted externally) where we can check the status of different appliances and sensors at home. It also extends to a local wireless node connected via an RF transceiver to the base node, which is emulating another room in the house. A PIC32MX320F128H microcontroller was used as the base node and an ATMEGA328 was used to demonstrate a room connected to the base node via RF.

This paper is organized into descriptive sections here on forth. The following section describes some of the work and research in this field. Later the architecture of this proof-of-concept system and its hardware and software components have been discussed in detail. The challenges faced during the implementation of this project and the results achieved including the future scope and direction of work have been described in the paper. An appendix with the Fritzing diagram has been included for future researchers in the field to re-implement, modify and improve upon this idea.

II. RELATED WORKS

Home automation has been around for a few decades now but its adoption today is not widespread[10]. Some home automation technologies like NEST learning thermostat, Lockitron smart lock are gaining wide acceptance. However, providers with automation solutions spanning wider aspects of the home are facing adoption issues. For instance, only 204,000 home automation systems were shipped globally in 2009[10]. The research survey presented in[10] has found that the main challenges in widespread adoption of home automation are high cost, lack of flexibility and integration, poor manageability and security challenges. Home automation research is struggling to overcome these challenges in adoption and some of the home automation commercial and research efforts have been listed here as a reference and comparison with the proposed system.

The system presented in[11] is a GSM based home automation system which is in-home wired communication and does not support wireless nodes in its present form. The wireless home automation system presented in[12], is a ZigBee home automation system that enables voice commands from the user to control the home appliances. ZigBee is targeted specially for personal area networks. It has low power consumption which limits the operating range. ZigBee based systems usually work in a mesh network and use every device in the mesh as repeaters to communicate to other devices over higher range[14]. Bluetooth devices on the other hand, have more range than ZigBee to start with, but cannot increase their range dynamically like ZigBee devices. This is because Bluetooth does not support the use of networked devices as repeaters. However, Bluetooth 4.2, which was released on December 2nd, 2014 is superior and suited to IoT applications than any of its previous versions. It has improved security features and an Internet Protocol Support Profile (IPSP) that adds an IPv6 connection option. This is relatively new and has not been implemented in research projects on home automation systems to date. However, many systems based on previous versions of Bluetooth technology have been implemented. For instance, the system presented[15] works using Blue Bee, a Bluetooth module from Cytron. However, this system can only be controlled from within the house and is targeted towards the elderly. The system does not cater to the needs of the working population today who will benefit more from remote controllability and remote monitoring of the home. In[16], they use the PC running LABVIEW as the main server which receives commands from the mobile device/user interface using Bluetooth. However, communications from the server to device are via parallel port and are wired connections. This makes it less scalable and not user friendly.

Commercially available solutions today are from Time Warner Intelligent Home[21], Protect America[22], Front point[23], ADT Pulse[24], Protection1[25] and other vendors. As the name suggests, most of these solutions are driven by security issue using cameras and alarms. Additional features might incur additional installation or maintenance costs. The main drawback of these systems is the monthly fees. 40,000 units of Time Warner Intelligent Home solutions were sold as of 31Dec 2014[19], which is less than 0.014% of the population of just the United States at that point in time[18]. In general, lack of budget-friendly prices and universal installation solutions seem to cause the lack of adoption of the aforementioned services[17]. In 2014, only 8%-9% of U.S. broadband households report ownership of any smart home device or controller[20]. Vera is a commercially available solution[13] very similar to the idea behind the proof of concept presented in this paper. The drawback with Vera is that it uses Z-wave technology and requires devices which can talk to it via Z-wave. Today, most devices are integrated with Wi-Fi, but Z-wave devices are specific to home automation and are also individually expensive[26]. Vera Edge, their latest product, has the capability to setup Wi-Fi devices by push button pairing via Wi-Fi protected setup. This might however, require more human intervention during system setup.

The home automation solution proposed in this paper is competent due to requiring one-time installation and the use of Wi-Fi which is commonplace today. Since it is only the installation cost, in the long term it will be less expensive than solutions requiring regular maintenance payments. Also, the direction of this research is intended towards a private cloud setup managed by the user. This will help the user from feeling an invasion of privacy unlike when a third party is monitoring their house for them. It can read inputs from multiple sensors in multiple rooms of the house via RF and has a base node that is able to connect to multiple relays for controlling major appliances of the home. Since it is connected to the cloud, it can be accessed from anywhere in the world as long as internet is available in some form. Further, because this system uses Wi-Fi and not GSM the strength of the signal is not an issue. Wi-Fi routers installed at many homes today should have enough capacity to run to support it. Also, Internet access is very convenient with many countries setting up public Wi-Fi hotspots, which would make the control of home from anywhere in the world possible.

III. HIGH LEVEL SYSTEM ARCHITECTURE

The implemented system consists of a chipKIT Uno32 used with a Wi-Fi Shield. It is connected to a web server in the cloud (hosted by portals.exosite.com). The various sensors connected to the chipKIT Uno32 are communicating values to the cloud and the readings are published to the Exosite server for remote monitoring. An additional node, an Arduino in this project, is locally connected to the chipKIT Uno32 using RF transceiver modules. One is on the base node, a chipKIT Uno32 and another is on the Arduino Uno which is mimicking a node installed in a different room of the house. This publishes sensor data to the base node. The collected data is then published to the cloud. The cloud has control switches to turn off/on the appliances at home, to open/close the garage door etc. A basic block diagram of this system is shown in Figure 1.

IV. HARDWARE COMPONENTS

The different hardware components that were used in the project are detailed in this section with their features.

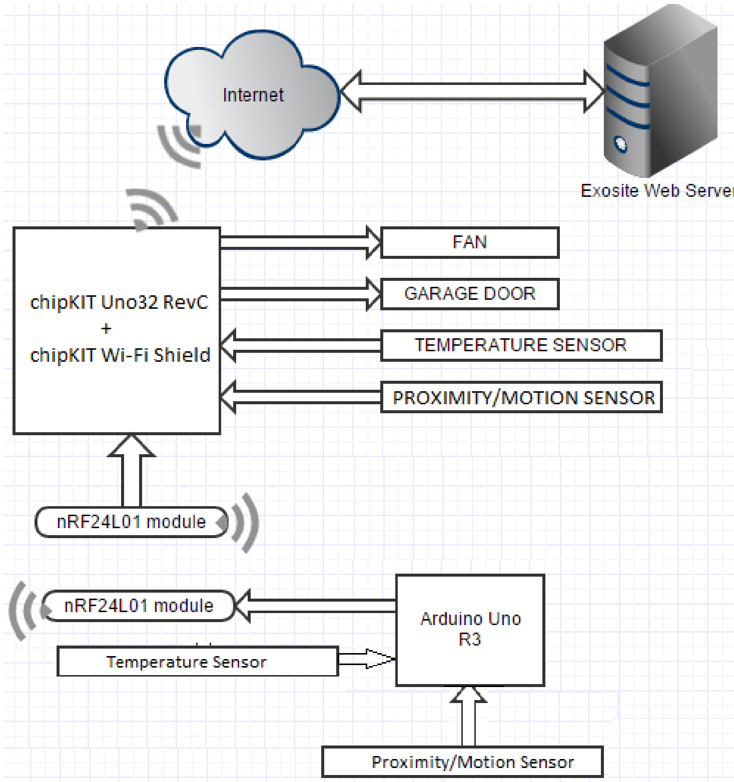

Fig. 1. System Block Diagram

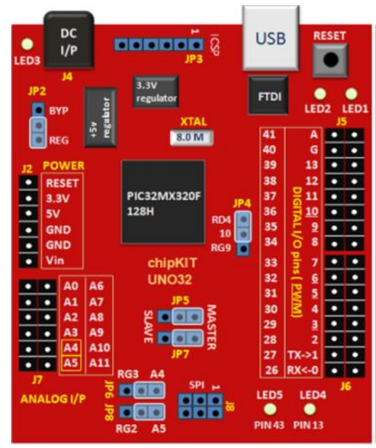

Fig. 2. chipKIT UNO32

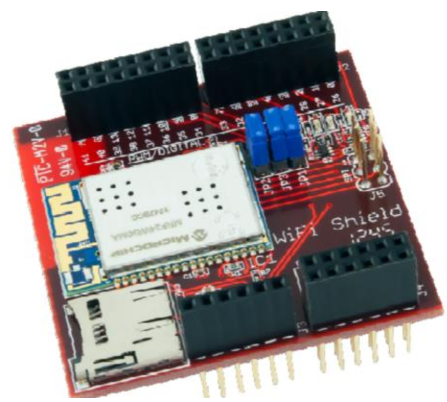

Fig. 3. Wi-Fi Shield MRF24WG0MA[4]

A. chipKIT Uno32

The chipKIT Uno32 development board, shown in Figure2 is based on the PIC32MX320F128 microcontroller, which consists of a 32-bit MIPS processor core running at 80MHz, 128K of flash program memory, and 16K of SRAM data memory. The board can be powered via USB or an external power supply[2]. It also has an FTDI chip to convert USB to UART serial (using the UART1 module on pins 0 and 1 in digital pin numbers) since the PIC32 on board does not have USB controller peripheral. It can be used with a programmer/debugger to be compatible with MPLABx or earlier versions of MPLAB. To be used directly, it needs to be loaded with a boot loader first. Sketches can then be uploaded via the mini USB cable. It has a 16-channel 10-bit ADC that can sample at rates up to 1Msps out of which 12-channels (A0 through A11) have been brought out on to the board. One Hardware SPI interface and UART2 multiplexed digital pins are also available on this board[2]. Given the wide number of peripherals, it seemed to be a promising platform for a home automation environment.

B. chipKIT Wi-Fi Shield

The chipKIT Wi-Fi shield has a built-in antenna module for wireless communication to the internet. This helps chipKIT Uno32 connect to the cloud via HTTP (Exosite's Rest API cloud service) running on top of TCP/IP. Since the MRF24WG0MA chip in the shield used has 802.11 compatible firmware, it is possible to connect it to the TCP/IP library provided by Digilent. It comes with an SD card slot that has not been used in this project. The Wi-Fi shield uses some of the digital pins and also the Hardware SPI that is brought out on the Uno. Using multiple slave capability of SPI interface (via slave select) we can re-use the SPI pins. It is to be noted that using the shield imposes a very hard restriction on the number of pins remaining to connect other devices.

C. Arduino Uno

The Arduino Uno platform is based on an Atmel ATMega328 microcontroller which is an 8-bit core that can run at up to 20MHz. Arduino operates at a 5V level. Recent versions of this board, like the Arduino Uno R3 used in this project, have replaced the FTDI chip with an ATMEGA16U2 programmed as a USB to serial converter. 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB port, a power jack, an ICSP header, and a reset button are available on this board as shown in Figure 4.

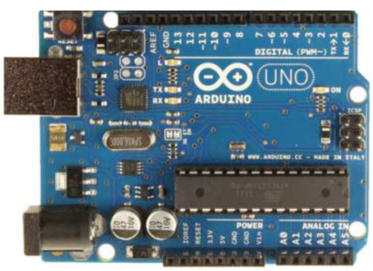

Fig. 4. Arduino Uno[5]


Fig. 5. RF transceiver module[6]

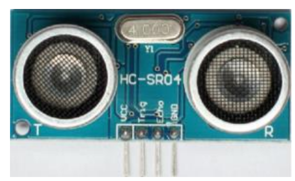

Fig. 6. Ultrasonic Range Sensor[7]

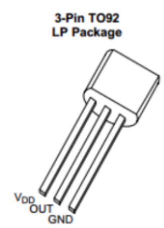

Fig. 7. LMT84 Temperature Sensor[8]

D. nRF24L01 RF Transceiver Module

It is a radio transceiver that operates universally in the 2.4GHZ-2.5GHz range with programmable data rates as 250kbps/1Mbps/2Mbps. By default the library code uses the transceiver at 1Mbps. In this project, since 1Mbps was more than sufficient for demonstration, this wasn't changed. This module is shown in Figure 5. The transceiver module used has 250m transmission range in open area[6]. It can also receive from 6 transmitting devices simultaneously, since it has 6 parallel pipes for radio communication with unique addresses[9]. However, to connect more devices, we can use same addresses with markers in the payload or we can schedule to receive from different addresses in a round robin manner. The frequency of receiving from each node in this case should be acceptable. Using the PIC32 micro-controller on the chipKIT Uno32, we can achieve very high sampling rates for each node.

E. Ultrasonic Range Sensor

The HC-SR04 range sensor requires a 10us trigger on its trigger input and it will send 8-pulses at 40 KHz and listen for a reflected signal. If obstacle is detected, an output pulse with width proportional to distance is sent out on the echo pin. Using 340m/s as the speed of ultrasound waves in air we can calculate the actual distance of the object. HC-SR04 can be installed in front doors and windows to notice if there is a change in state indicating intrusion. This sensor can be used to detect motion and proximity to secure the house.

F. Temperature Sensor

LMT84LP temperature sensor it used for this project. It can measure temperatures from -55 to +150 in degree Celsius[8]. Its output is linear and inversely proportional to the temperature. The analogRead() function defaults to 100Ksps sampling like in the Arduino Uno. The LP package of this sensor was used.

V. SOFTWARE DESIGN

A. Tools Used

In this project, UECIDE (Universal Embedded Computing IDE) was used which is a forked version of MPIDE distributed by Digilent in support for chipKIT. This decision was based on the fact that UECIDE can support more platforms and would be helpful in this project, which has a wide scope for future enhancements.

B. Libraries

The libraries are all implemented in C++ by using the underlying boot loader settings along with the PIC32 specific API libraries for wiring etc. that come installed with the UECIDE tool by default. Some externally downloaded libraries were installed in UECIDE libraries folder for using with new modules like nRF24L01. They needed some modification for using with the chipKIT Uno32, but not re-writing of the complete library in the modules used in this project.

C. Wi-Fi Connection

The Wi-Fi connection is established via the MRF24WG0MA Wi-Fi module on the shield used. It provides various authentication options like WEP, WPA2 with passphrase or key. It needs a TCP client provided by the Digilent library to be fed into the Exosite cloud server library. This gives us the ability to use built-in send and read commands available in the Exosite library. However, it is important to note that these commands are blocking, causing the performance to depend on the efficiency and speed of the cloud service.

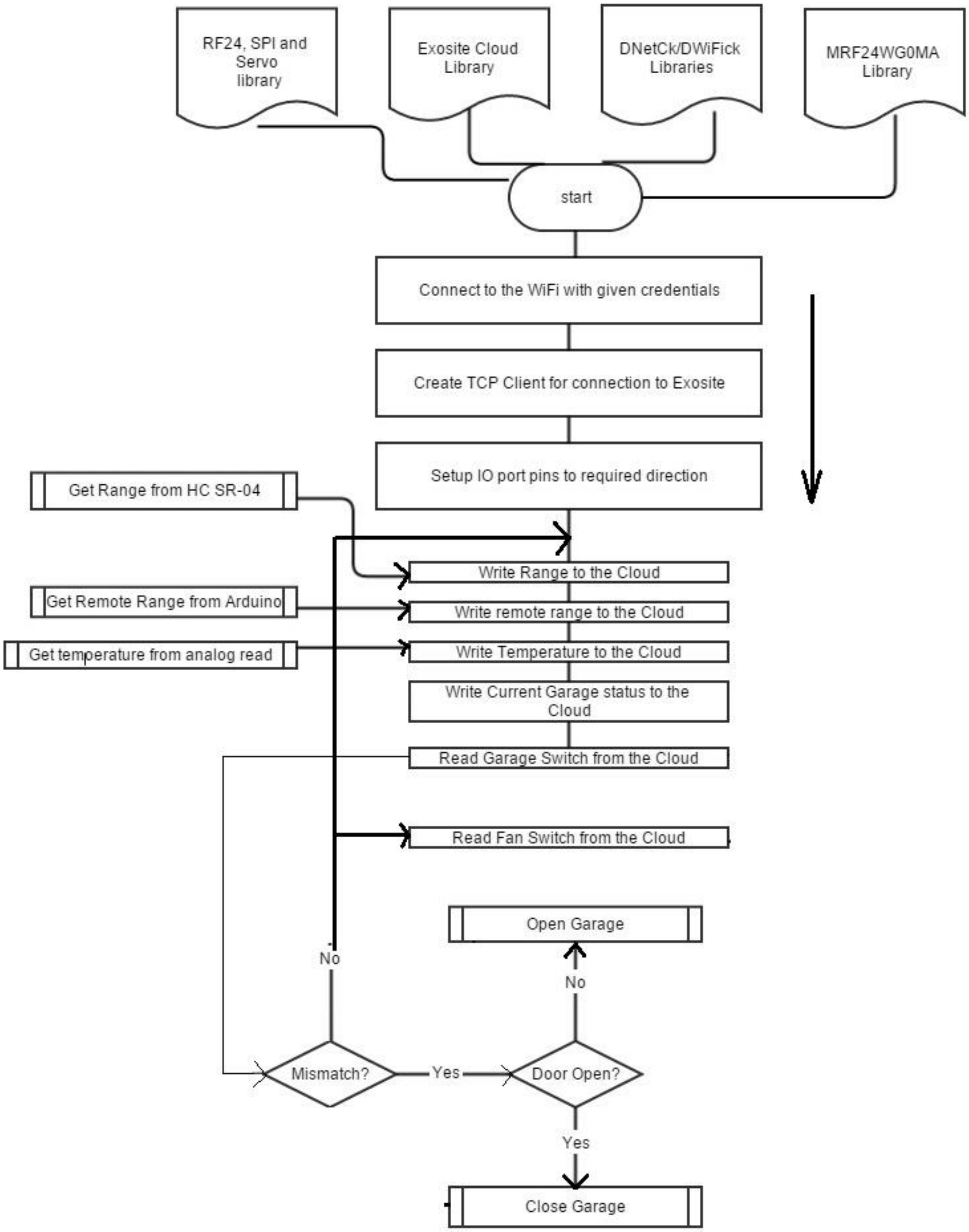

Fig. 8. Software Design Flow

D. Setup

Software setup deals with the following functions:

- Controllable elements are connected to output pins and signals to monitor are fed to the analog input pins on the board. Directions to the various pins are assigned.
- Setting up the TCP client can be established using the Digilent provided TCP/IP libraries. The appropriate options are set here.
- Exosite Client functions can be initialized by providing the TCP client object as input to this HTTP API library. It provides functions to send and receive messages to and from the cloud respectively.

E. Loop

In this project, the loop has been implemented as a state machine. The frequency of execution of a particular state depends on the frequency the processor is running at that point in time. It also depends on the strength of the wireless signal obtained which affects the latency of each state. The latter is variable due to network parameters.

F. Predefined Functions

Each control and monitoring element is accessed via specific predefined functions. This ensures modularity and ease of adding new devices to monitor and/or control when expanding the project scope. These are shown in Figure 8.

VI. CHALLENGES FACED

- Initially, sketches were not getting uploaded to the chipKIT Uno32. It turns out the boot loader had been overwritten due to previous experiments with MPLABx. Upon installing a boot loader from GitHub, it worked but crashed again. So finally the original boot loader from diligent website proved to be stable although not recent.
- Most of the support for chipKIT Uno32 is only with the Arduino IDE-type development software, like MPIDE/UECIDE. Libraries are in C++ and although MPLAB X supports few of these libraries. Most of them however need significant work to integrate. The problem with MPIDE/UECIDE environments is the lack of complete control due to a highly optimized API layer and simplified user interface functions.
- Many of the device and platform supporting libraries are based on Arduino AVR micro specific code and have to be ported to PIC32. For example, Exosite and Ubidots were the only cloud services that were found to support the chipKIT for libraries. For the nRF24L01 module, the library had to be slightly modified for use with the chipKIT since it was originally designed for Arduino's AVR chip.
- The Wi-Fi shield uses the only Hardware SPI interface being brought out on the chipKIT. So additional digital IO pin is used as slave select to support 2 devices on single SPI port.

VII. CONCLUSION AND FUTURE WORK

A cloud based home automation system was setup as proof of concept for a powerful and simple low cost alternative to commercial solutions available today. It achieved the purpose of designing a basic framework to improve and build on. The cost of the prototype is approximately 145$, which includes the chipKit Uno32 prototyping platform for 26.95$[27], chipKIT PGM Programmer/Debugger for 26.99$[28], Arduino Uno for 21.27$[29] , Ultrasonic range sensor for 5$, two breadboard Kits for 10$ each, a dc motor for 5$, a continuous servo for approximately 20$, two RF communication modules for 2$ each and a,temperature sensor for 5$. Miscellaneous electronic components like resistors have been accounted for in 10$.

This prototype is the starting point for development of a plug and play system, wherein the user need not buy devices that are compatible with home automation. They need to only buy an adapter for every component they wish to integrate into the automated system. Also, the installation of such a system would easily integrate into the home Wi-Fi network and not pose a range problem for controlling the appliances. Using development boards with more resources can even allow addition of alternative interfaces like an LCD module. The chipKIT Wi-Fire board could be used because of its integrated Wi-Fi module, more digital IO pins and more hardware SPI interfaces available on board. The additional hardware SPI will help in keeping the Wi-Fi communication uninterrupted while local nodes are communicating to the base node via RF. Designing a custom board tailored to project requirements can help in reducing the area and cost. The devices used in this project can also be replaced with more efficient ones. Relays can be used to control electrical appliances. Monitoring systems can use cameras that record on motion detection and IR sensors. However, these would add to the cost of the overall system.

That said, the presented concept has a wide scope for enhancement. This project involved many components, each of which were not specifically evaluated in terms of power consumption. Although the design was made to operate at speeds for maximum performance, it can be optimized to run at lower speeds to reduce the power consumption. Many variants of the PIC32 family of devices offering low power can be explored. Some of TI’s low power microcontrollers can also be explored to replace Arduino for remote nodes.

Exosite cloud service utilized by this project uses HTTP. Cloud services using other protocols can be explored like MQTT which will be less taxing on the microcontroller and serve the purpose more efficiently. PubSubclient library is readily available for the Arduino but not yet for the chipKIT. An optimum solution would be to host a self-run small private server in place of the cloud. This would be faster and better in terms of latency as it would avoid commercial load balancing and other obstructions to real time accesses. A major part of this enhancement will require a very user-friendly GUI and a modular structure that seamlessly supports integration or removal of nodes during use. Ideally, the end solution provided should be customizable based on the cost and the features a consumer wants.

REFERENCES

[1] www.embedded-lab.com
[2] chipKIT Uno32™ Board Reference Manual (for RevC), Digilent, Inc.July 17,2012
[3] Microchip Technology Inc, PIC32MX3XX/4XX Family Data Sheet,DS61143E
[4] chipKIT™ WiFi Shield Reference Manual, Digilent, Inc.February 2013
[5] MINDS-i, Arduino Resource Guide, https://mindsieducation.com/files/pdf/Arduino Resource Guide.pdf"
[6] http://www.seeedstudio.com/depot/nRF24L01Module-p-1394.html
[7] Fritzing HC-SR04 Project, fritzing.org
[8] LMT84, LMT84-Q1 datasheet, Texas Instruments, SNIS167B MARCH 2013, REVISED MAY 2014
[9] NORDIC SEMICONDUCTOR, nRF24L01+ Single Chip 2.4GHz Transceiver, Preliminary Product Spec v1.0
[10] A.J. Bernheim Brush., Bongshin Lee.,Ratul Mahajan., Sharad Agarwal., Stefan Saroiu., Colin Dixon., Home Automation in the Wild: Challenges and Opportunities,Microsoft Research, University of Washington, Copyright 2011 ACM
[11] Mahmoud shaker Nasr.,Cell Phone and J2ME Based Home Automation System with Feedback Instant Voice Messages Support, Journal Of Telecommunications, Volume 15, Issue 1, July 2012
[12] S.Benjamin Arul ., Wireless Home Automation System Using Zigbee , International Journal of Scientific and Engineering Research, Volume 5, Issue 12, December-2014
[13] http://getvera.com/
[14] http://en.wikipedia.org/wiki/ZigBee
[15] R.A.Ramlee., M.H.Leong., R.S.S.Singh., M.M.Ismail., M.A.Othman., H.A.Sulaiman., M.H.Misran., M.A.Meor Said., Bluetooth Remote Home Automation System Using Android Application, The International Journal of Engineering And Science (IJES), ISSN: 2319 1813 ISBN: 2319-1805
[16] SyedHussainRazaNaqvi., SyedMuhammadTahirRaza., Muhammad- UmerAhmedKhan., TameemAthar., Home automation via Bluetooth using android platform http://mysteriousmaniacs.weebly.com
[17] Weili Lin.,Smart Home in the US:Easier Access Tougher Game-CEDIA Report Part1, www.asmag.com/SMAhome/detail m.aspx?id=3001
[18] U.S. and World Population Clock, http://www.census.gov/popclock/
[19] 2013 Time Warner Cable Annual Report, http://ir.timewarnercable.com
[20] http://www.securitysales.com/article/connecting for a home run/P4
[21] http://www.timewarnercable.com/en/intelligenthome/equipment.html
[22] http://www.protectamerica.com/security-equipment
[23] http://www.frontpointsecurity.com/equipment-overview/controls
[24] http://www.adtpulse.com/home/plans
[25] http://www.protection1.com/home-security-packages/
[26] http://www.zwaveproducts.com/
[27] http://www.digilentinc.com/Products/Detail.cfm?Prod=CHIPKIT-UNO32
[28] https://www.digilentinc.com/Products/Detail.cfm?Prod=chipKIT%20PGM
[29] http://store.arduino.cc/index.php?main_page=product_info&cPath=11&products_id=195