ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A STUDY AND REASEARCH ON MONITORING SYSTEM AND EMBEDDED WEB SERVER BASED AUTOMATION

V.Sai Kishore*

* (M.Tech Student , Department Of VLSI And EMBEDDED SYSTEMS , Talla Padmavathi College of Engineering , Kazipet , Warangal).

ABSTRACT

Instead of PC based servers; ARM processor based servers are becoming trend of today's market. Cost reduction is achieved using ARM processor along with Ethernet module as Embedded Web Server. Idea is utilized for monitoring and controlling maximum no. of either home appliances or industry devices. Without using a computer, Ethernet module can communicate to the owner of the overall system, who is able to manage appliances from any location outside. This server provides a powerful networking solution and enables web access for automation and monitoring of different systems. For industry automation, instrumentation and household devices control, this is an optimized solution. System home page can be accessed using web browser. Operational status of the appliances can be observed and changed in case of necessity. This paper proposes development of low cost system for above purpose. Different sensors installed at working place help in sensing real time environmental conditions like temperature, light, humidity etc.

KEYWORDS: Web of Things, Smart Home, Wireless SensorNetworks, Web Caching, IPv6, REST, DACs

INTRODUCTION

Embedded sensors and wireless sensor networks (WSN) are being deployed around the world in a large scale, measuring with high precision environmental conditions such as tempera-ture and illumination or physical events such as vibration and motion. WSN provide a reliable and extensible solution for real-world deployments.Household appliances are being equipped with embedded micro-controllers and wireless transceivers, offering smart behavior and communication capabilities. These augmented appliances become able to form wireless networks, transform-ing residential areas into smart home environments. High-precision sensors, incorporated in smart homes, pro-vide context-awareness of the environmental conditions that exist at each room of the house. Combining context-awareness with administration of the household appliances that ”live” inside the residence, allows advanced home automation. When electrical appliances are also augmented with individual mon-itoring and control, efficient energy management is possible. In the latest years, technologies like sensor networks, RFID, short-range wireless communications and real-time localiza-tion are becoming largely common, bringing embedded In-ternet into commercial use. The Internet of Things (IoT) [6] includes technologies and research disciplines that enable the Internet to reach out into the real world of physical objects. The introduction of IPv6 and the efforts for porting the IP stack on embedded devices [4], [9] facilitate the effective penetration of the Internet in embedded computing. Extending the notion of the IoT, the Web of Things (WoT) [19] builds upon the success of the Web 2.0, and reuses well-accepted and understood Web principles to interconnect the expanding ecosystem of embedded devices, built into everyday appliances. It is about taking the Web as we know it and extending it so that anyone can plug devices into it. In this paper, we address the limitations of our previous work in using Web standards to build the smart home [13]. Our contribution is the use of Web-enabled sensor devices operating in a 6LoWPAN WSN and the improvement of the device discovery procedure considering the IPv6-based wireless network. We compensate the overhead introduced with embedding the IPv6 stack on sensor devices by using HTTP caching to significantly reduce the mean response time for accessing sensor data. Additionally, we investigate how modern Web messaging can be used to incorporate event-based scenarios in the smart home.

Finally, we develop a Web-based graphical interface to abstract home automation procedure for typical home residents. In the following sections we present our work in using solely the Web for interacting with smart devices. In Section II we present related work in our field while Section III provides background information concerning our earlier work in implementing a Web-based smart home. In Section IV we mention the changes we made, to fully transform our smart home into a Web-based smart space. Next, in Section V we describe a Web application we built to facilitate home automation procedure for residents and in Section VI we perform an evaluation of our system. Finally, in Section VII we discuss the implications of our results and propose issues to be addressed in future work before concluding this paper.

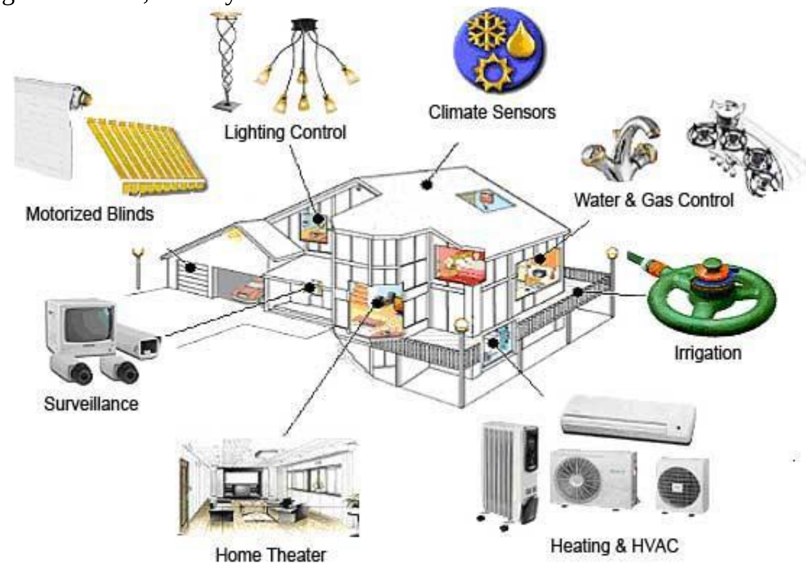

Fig. 1. Different Devices Connected For Automation

DESCRIPTION

Science fiction has advertised an idealized vision of a fully integrated smart home, where all the operations of a house can be efficiently controlled by a central application. However, a major impediment in the realization of this scenario has been the proliferation of incompatible standards and protocols used by various device manufacturers, which makes the smooth integration of appliances from different vendors a time- and money-consuming process at best.To address the heterogeneity of devices, SOAP-based Web services have been proposed as a basis to build an infrastruc-ture for domestic networks [1]. More recently, Priyantha et al. [15] have successfully used Web services to interact with sensor networks. They state that a key challenge in using Web services on embedded sensors is the energy overhead of the structured data formats used in them. Clearly, SOAP/XML are not energy-efficient in such areas. The work in [7] quantifies this statement in terms of time and energy. Recent approaches suggest REST as the architectural style used in constrained environments. Yazar et al. [20] implement an IP-based sensor network system where nodes communicate their information using RESTful Web services. Similarly, Schor et al. [17] developed a 6LoWPAN-enabled WSN, di-rectly integrated into the IPv6-based future Internet. Sensor nodes, in this scenario, offer RESTful APIs and service discovery is based on Apple's mDNS proposal. Current and emerging architectures and technologies that are suitable for wireless home area networks (WHAN) have been surveyed in [14]. According to this survey, IPv6 and 6LoWPAN technologies are preferred because of their low cost, ease of installation, flexibility and reliability. A path towards the integration of things into the Web has been presented in [19], where physical objects are made available through RESTful principles. The concept of physical mashups introduced in [8], builds upon the success of Web 2.0 mashup applications, by proposing a similar approach for integrating real-world devices to the Web. pREST [3] defines a RESTful protocol to bring the simplicity and holistic view of data and services on the Web, to pervasive environments. In previous work, we combined the principles of the WoT in the area of smart homes [13]. We developed an application framework that enables multiple home residents to interact concurrently with home appliances, and our results indicated a reliable

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

and efficient operation. However, devices were not directly Web-enabled, as a gateway was needed to access them.A social perspective for a Web-enabled smart home is discussed in [11]. A Web mashup is developed that involves the Web-enabled household appliances of a smart home, in a social networking application. Web 2.0 technologies are utilized through Facebook to transform the interaction with the home into a shared, social experience. Finally, Web-based smart homes can easily be extended to exploit demand response, which will be a critical functionality in the smart grid of electricity [12]

HOMEWEB APPLICATION FRAMEWORK

HomeWeb is a fully Web-based application framework for smart homes [13], where all interactions with embedded devices are done via standard HTTP requests. In this section we describe how to leverage the ubiquitous and well-known Web standards to integrate by design household appliances into the Web fabric. Our framework supports concurrent access by multiple residents and facilitates the development of ubiquitous applications by the habitants, who may probably have no programming experience.

REpresentational State Transfer (REST) [5] has been pro-posed for Web-based interaction with household appliances as it is a lightweight architectural style that defines how to use HTTP as an application protocol. It advocates in providing Web services modeled as resources, identified by Unique Resource Identifiers (URI). Resources can only be manipulated by the methods specified in the HTTP standard (e.g. GET, PUT, POST, DELETE), under a uniform interface. REST promotes the practice of Resource Oriented Architectures (ROA) [16], in order to provide and connect together services on the Web. REST guarantees interoperability, loose-coupling and a smooth transition from the Web to home environments.Web services tend to fall into one of two camps: Big Web services (WS-*) and RESTful Web services. WS-* [2] are a set of complex standards and specifications for enterprise application integration. We believe that RESTful Web services are more appropriate for resource-constrained, ad hoc environ-ments due to their simplicity and flexibility.. It follows a modular architecture and is composed of three principal layers: Device Layer, which is responsible for the management and control of embedded devices, Control Layer, which is the central processing unit of the system and Presentation Layer, which generates dynamically a representation of the available devices and their corresponding services to the Web, enabling the uniform interaction with them over a RESTful interface.Each time a new device is discovered, a new thread dedi-cated to the device is created. In order to provide multi-user support, we attach a Request Queue to each thread, to enqueue concurrent requests to it. Requests are stored in a FIFO manner and are transmitted sequentially to the device.The uniform RESTful interface facilitates home automation through the development of flexible applications, which can easily be created on top of heterogeneous sensor devices. Our Web-oriented framework supports simple creation of Web mashups and advanced rules in any programming language that supports HTTP. Web mashups are extended into physical mashups by exploiting real-world services offered by physical devices and combining them using the same tools and tech-niques of classic Web mashups [8].Sensor nodes are used to emulate household appliances, as they can be programmed easily and they offer networking and sensing capabilities. In our previous work, we did not directly embed Web support in our sensor devices, but hid the native programming model of TinyOS1 behind a RESTful API using a custom driver inside the application framework. Additionally, our device discovery procedure was simplistic and not based on existing standards. These constraints decreased the overall flexibility of our system and hindered our vision for a truly Web-based smart home.

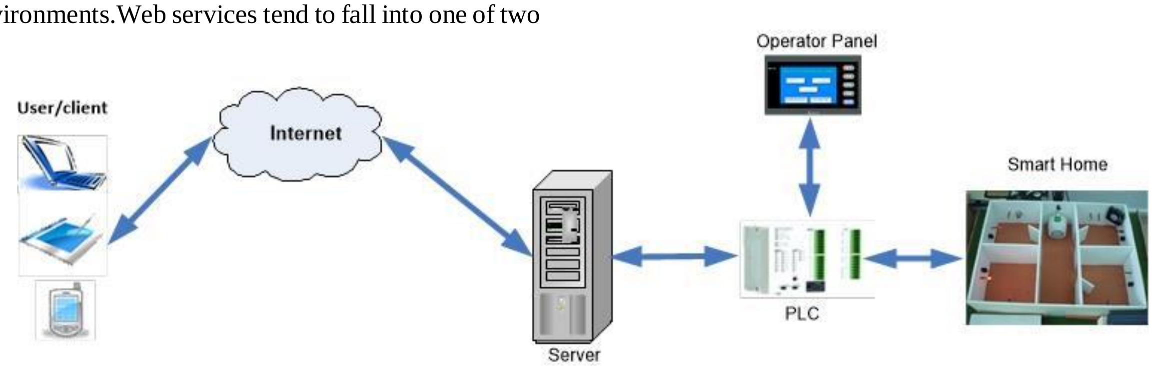

Fig. 2. Home web application framework

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

WEB SERVER CONFIGURATION

Complete implementation of an HTTP Web Server in an ARM7 microcontroller is possible. The website pages are stored inside the program space of the ARM7 microcontroller. Web server includes pages to be delivered to the browser. Using any standard web browser on any PC you can access the web pages displaying information about temperature, light intensity etc. Uploading any changes to present data into flash memory is possible and it allows to change the status of any device connected into the system. The web server implements different protocols such as ARP, IP, IGMP, TCP, UDP, HTTP for overall operation. Initially the server is configured by entering IP address to the Ethernet module and tested for the working using ping command. Now the embedded web server is responding to the clients. Request is made to embedded web server, by typing the IP address of the server in the client's browser. This request is taken by the OS of the client and given to the LAN controller of the client system. The LAN controller sends the request to the router which processes and checks for the system connected to the network with the particular IP address. If the IP address entered is correct and matches to that of the server, a session is established and a TCP/IP connection is established. Web server starts sending the web pages to the client.

Device Discovery

Since we envision Web-enablement of embedded devices, we need to follow Web-based approaches also for discovering these devices. We alter the device discovery procedure in [13] and we extend it to take into account the IPv6 operation environment. Our new device discovery protocol is similar to WS-Discovery3, which is a multicast discovery protocol to locate WS-* services on a local network. We adapted this protocol for RESTful Web services by transmitting a single URL instead of a heavy SOAP/XML payload in the multicast message. This URL points to a Web page, where the services offered by the device are described. As soon as a sensor mote joins the 6LoWPAN network, it announces its availability by means of the multicast message. The application framework listens at the pre-specified multicasting channel and it acknowledges the newly found sensor device immediately. When the sensor receives the acknowledgment, it starts operating normally as an embedded Web server.

Service Description

As we mentioned in the previous subsection, an embedded device needs to transmit a URL, in which the services offered by it are described. We followed a standardized, Web-based way to perform resource description, in order to achieve high interoperability with heterogeneous devices and services. We adopted Web Application Description Language (WADL4). WADL is an XML-based file format that provides a machine-readable description of HTTP-based Web applica-tions, and can be considered as the RESTful equivalent of Web Services Description Language (WSDL5), which is a standard for describing SOAP-based Web services. WADL is intended for applications that are based on the existing architecture of the Web, and it is meant as a platform-and language-independent way of describing services, to pro-mote reuse of applications. We used WADL to model the resources (services) provided by an embedded device and the relationships between them.

Web Messaging for Event-driven Scenarios

Since sensor motes operate as Web servers, interaction with them follows the classical client-server model employing pull technology, in which the request for the transmission of information is initiated by the client, which is the application framework in our case. This model is appropriate for ad hoc interactions with sensors and actuators and for environmental monitoring which happens on demand. However, the client-server model is not efficient in event-based scenarios. These scenarios include events that are sensed sporadically when something important happens e.g. detection of fire or the opening of a door.Push technology describes a style of Internet-based commu-nication where the service request is initiated by the publisher, which is the sensor device in our case. We build upon recent developments in Web push techniques and extend them for

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

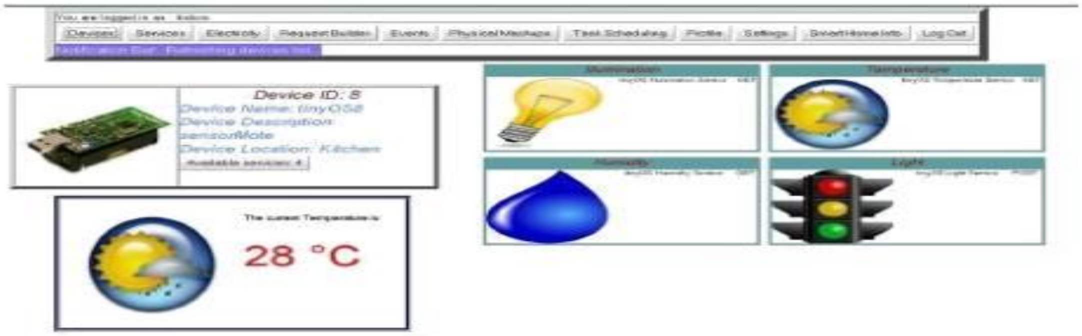

Fig. 3. Snapshot of the Web-based Graphical Interface

embedded devices with a RESTful messaging system, in order to efficiently integrate event-driven services in our system.

In a previous work, we developed the RESTful Message System (RMS) [18], which is a lightweight publish/subscribe messaging suited for sensor devices. We apply RMS to our IPv6-enabled sensors by enhancing them with a simple REST-ful API to use the eventing system.In case the application framework wants to receive notifi-cations about an event from a sensor mote, it creates a new subscription by POSTing an HTTP request to this particular sensor. Whenever a new event is sensed, it will be POSTed back to this subscriber at a callback URL, specified by the subscriber at the subscription message.

HTTP Caching

Adding IPv6 capabilities on sensor motes has apparent ef-fects on their performance and energy consumption. However, we can decrease these disadvantages by exploiting the HTTP caching feature.HTTP has a very thorough and well supported caching mechanism. A Web cache sits between the (origin) Web servers and clients, and monitors incoming requests, saving copies of the responses for itself. Then, if there is another request for the same URL, it can use the cached response, instead of asking the origin server for it again. Web caches reduce latency because the request is satisfied directly from the cache, and also reduce network traffic as the amount of bandwidth used by a client is reduced. Gateway caches are typically deployed by Web masters, to make their sites more scalable, reliable and better performing. For our purposes, origin Web servers are the sensor devices, clients are the residents of the smart home and our application framework includes a gateway cache, to accelerate perfor-mance and save sensors' energy. Our caching mechanism is used only for GET requests and it uses the expiration model, which is a way for the server to say how long a requested resource stays fresh. We will test the performance of the caching mechanism in Section VI-A.

WORKING OF SYSTEM

Sensor inputs are connected through signal conditioning circuits to ARM Controller. It processes on these signals (conversion from analog to digital) and then parameter values are stored in the memory. Status of the working devices or appliances is decided on the basis of stored values. Relays can be operated ON or OFF to change the status of devices as per our requirement. By comparing the standard values of parameters to be tested, further status of devices is decided. Thus purpose of automation works. For controlling the devices using web browser, owner of the system has to access the webpage and change the settings. Software code will be written in Embedded C for all actions. It will be transferred to processor using serial port. Serial port is interfaced using driver/receiver interface. Sensed data is processed by microcontroller and continuously provides feedback to the website. Ethernet module plays an important role in transferring web pages to the client. Web page code is designed in HTML and uploaded on server using Ethernet module and TCP/IP address.

ETHERNET MODULE

The Ethernet specification (IEEE 802.3) has evolved over the last number of years to address higher transmission rates and new functionality. For this project selected Ethernet module is WIZ810MJ. This network module includes W5100 (TCP/IP hardwired chip, include PHY), MAG-JACK (RJ45 with X'FMR) with other glue logics. It can be used as a component and no effort is required to interface W5100 and transformer. The WIZ810MJ is an ideal option for users who want to develop their Internet enabling systems rapidly.WIZ810MJ consists of W5100 and MAG-JACK.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Fig 4: Ethernet Module

Ethernet Specifications

- TCP/IP, MAC protocol layer : W5100
- Physical layer : Included in W5100
- Connector : MAG-JACK(RJ45 with Transformer)
- RJ-45 Connector RDA - 125BAG1A
- Input Voltage : 3.3V Internal Operation and 5V Tolerant I/Os
- Power Consumption : 10/100 base T : Max 185mA (3.3V)

Features

- Supports 10/100 Base TX
- Supports half/full duplex operation
- Supports auto-negotiation and auto crossover
- detection
- IEEE 802.3/802.3u Complaints
- Operates 3.3V with 5V I/O signal tolerance
- Supports network status indicator LEDs
- Includes Hardware Internet protocols: TCP, IP Ver.4, UDP, ICMP, ARP, PPPoE, IGMP
- Includes Hardware Ethernet protocols: DLC, MAC
- Supports 4 independent connections simultaneously
- Supports MCU bus Interface and SPI Interface
- Supports Direct/Indirect mode bus access
- Supports Socket API for easy application programming
- Interfaces with Two 2.0mm pitch 2 * 14 header pin

CONCLUSION

In this paper, we have extended our Web-oriented applica-tion framework into a fully-functional, Web-based smart home. We exploited recent advancements in IPv6 and 6LoWPAN technologies to enhance the functionality of our framework with directly Web-enabled sensor devices. Our technical eval-uation indicates that the application of Web technologies such as HTTP caching and push techniques in the WSN domain can improve performance while wide-scale connectivity and interoperability are guaranteed. As future work, we plan to incorporate more advanced technology in the HomeWeb, such as residential smart meters, lighting, HVAC control etc., enabling all these technologies as first-class citizens of the Web. Our efforts will be dedicated towards energy-efficient, sustainable, flexible and secure smart homes that would operate in a true Web environment. Smart homes have an important role to play in future urban environments. The increasing urbanism implies new challenges for the future citizens. The WoT constitutes a promising practice towards the vision of a real-time digital city [10]. Web-based smart homes can be considered as a real-time platform, for engaging people to sense and shape their cities.

REFERENCES

[1] S.A.N.Sandeep, P.Malyadri "Embedded Web Server Based on DAC System Using ARM" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 , Vol. 2, Issue 4, July-August 2012.

[2] Deepak C.Karia, Vispi Adajania, Manisha Agrawal and Swapnil Dandekar "Embedded Web Server Application Based Automation and Monitoring System" Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011).

[3] Yanni Zhai, Xiaodong Cheng "Design of Smart Home Remote Monitoring System Based on Embedded System" 978-1-4244-9600-6/11/ /$26.00 ©2011 IEEE.

[4] http://www.ijeit.com/vol%202/Issue%209/IJEIT1412201303_51.pdf

[5] Mo Guan Minghai Gu "Design and Implementation of an Embedded Web Server Based on ARM" School of Information Science & Engineering Shenyang University of Technology Shenyang , China 978-1-4244-6055-7/10/$26.00 ©2010 IEEE.

[6] V.Billy Rakesh Roy1, Sanket Dessai1, S. G.Shiva Prasad Yadav "Design and Development of ARM Processor Based Web Server " International Journal of Recent Trends in Engineering, Vol. 1, No. 4, May 2009.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

[7] A. R. Al-Ali, Member, IEEE & M. AL-Rousan "Java-Based Home Automation System" IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, MAY 2004.

[8] Abhilash Ramakrishnan "16 - Bit Embedded Web Server" SIcon/04 - Sensors for Industry Conference New Orleans, Louisiana USA, 27-29 January 2004.

[9] D. Guinard and V. Trifa. Towards the Web of Things: Web Mashups for Embedded Devices. In Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web, in Proc. of WWW Conference, Madrid, Spain, 2009.

[10] http://www.cs.ucy.ac.cy/~akamil01/papers/Kamilaris_ICT2011.pdf

[11] http://www.thaieei.com/embedded/pdf/Automation/20022.pdf

[12] http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp220-224.pdf

[13] http://www.ijcsit.com/docs/Volume%205/vol5issue01/ijcsit20140501196.pdf