ORIGINAL ARTICLE

Survey on Operating Systems for the Applications of the Internet of Things

Kausar Parveen[1], Akber Ali[2], Ghalib Asadullah[3]

Author`s Affiliation

Department of Computer Science, Lahore Garrison University Lahore

Article Info

Received: Nov 07, 2016
Revised: Dec 15, 2016
Accepted: Dec 20, 2016

How to Cite this Manuscript

Fakhar KP, Ali A. (2016).survey on the operating system for the application of the internet of things. JICTRA, Volume 7, 9-16

Funding Source: Nil
Conflict of Interest: Nil

Address of Correspondence
Kausar Parveen
Kausarnawaz6@gmail.com

ABSTRACT

At present, there is a large variety of applications where internet of things plays a vital role, not only supporting all fields of life but also rapidly developing due to progress in smart things technology. Software framework especially operating systems are in great demand to grease the wheels of the internet of things. In this paper main subject is the operating system from the internet of things perspective. In this paper, we have surveyed current operating systems which are basically supporting the applications of the internet of things.

The purpose of this survey is to highlight major concerns pertaining to OS design in IoT by keeping in mind the requirements of emerging IoT applications. Survey parameters are actually the current requirements of the internet of things.

Keywords: Embedded systems, internet of things (IoT), Multithreading, Standalone systems, Machine to Machine.

Introduction

Currently, the most prevailing field of the computer is IoT.It involves all communication fields of computer[1]. The main idea of IoT is to create an association of all things via the internet.

In the IoT, smart things or objects are those things or objects which have sensors and they can participate in all business as well as in social processes, interact with each other, communicate with surrounding environment plus communicate with themselves, react with the real or physical world and run processes with or without human intervention. IoT idea has a high impact on many aspects of life. According to the view of a private user, IoT is part of all fields of life. In the working field, it assists in all fields like learning, living, security, e-health. From business user's view, the main advantage is in field's industrialized engineering, computerization, logistics, a trade organization, and intelligent conveyance of people and things. It is providing new services to the users and product owners. Smartphone's, tablets are now using wireless technology to connect with a wide range of machines for instance vehicles, household appliances. Smart Objects can also handle account security, risks, and privacy issues.[2]

In IoT, smart objects services are linked by the internet to find their query and make decisions on the basis of required data information. The Internet of Things (IoT) is already in use in multiple machines, devices and appliances and all these devices are connected to the Internet through multiple networks. There are different applications of IoT, the smart home is also one of them. The connected home is fast becoming a reality because Mobile Broadband networks have spread as well as the cost of hardware falls. These connected devices are now the components of IoT.

IoT systems are becoming complex due to proliferation number of applications in it. This in return increasing the use of processor-based devices. As IoT is working through networks of the internet, therefore, it is dependent on the interaction of distributed processors, therefore the development of operating systems for each application is focused according to the network and applications requirements.

In this paper section, 1 includes an introduction.A literature review is included in section 2 and that will help to identify the current status of IoT as well as of operating systems (OS)available in the market and judge the requirements of IoT OS. Section 3 is the comparison of different IoT Os feature. In the end, section 4 concludes the whole paper and future work is recommended.

Internet of Things

Now the most rapidly growing field is Internet protocol technology with domains embedded devices like sensors and actuators. In IoT all things are object with unique address and protocol and these objects are connected with each other via internet.[3]There are different similar concepts of IoT like machine to machine communication (M2M).[4]

IoT applications:

IoT is currently part of smart home appliances; wearable's and includes components of the industry. Actually, it has a wider concept and it is supposed that all things can be connected in the world. Today's most prominent applications are Smart home which means that all home devices are smart devices and they are connected with each other.

Wearables are those smart devices which are wearable.

Smart City manages a variety of use cases like traffic management, water distribution, urban areas living problems, etc.

Smart grids give information about the behaviors of different suppliers like electricity to improve the efficiency, reliability, and economics of electricity.

Industrial internet includes applications among others include smart factories or connected industrial equipment.

Connected car facility is used whether in self-driving or assisted driving. It also connects with other cars, provides mapping service.

Connected Health concept is related to the health care systems in which real-time monitoring take care of health issues.

Smart retail means better ways of shopping via intelligent payments solutions.

Smart supply chain tracks the temperature of the product on the road, to get inventory information.

Smart farming means to monitor animals in the farms.

Visions of IoT

An enormous number of objects are tangled in the IoT processes, it is divided into three visions on objects base. There three visions of IoT are things oriented vision, semantically oriented vision, and internet oriented vision.[5]. In fact ''Internet of Things" model is created after merging these different visions.

Things oriented vision considered things of items of Radio-Frequency Identification (RFID) tags. There are also other components of the path to deploy fully the IoT vision. In fact, IoT has not only one objects that are RFIDs; it is just at the forefront of IoT the technology driving the vision. For the RFID maturity, low cost plus strong supports from the business community are the consequences.

Other components which work with "RFIDs are Near Field Communications (NFC), Wireless Sensor and Actuator Networks (WSAN) they are recognized as ''the atomic components that link the real world with the digital world". One of the tasks is being carried out with the aim of developing platforms, such as the WISP (Wireless Identification and Sensing Platforms) project".

In the same way, other relevant institutes have explained the IoT main focus is primarily on the ''Things" and on the way to fulfill the augmentation in the 'Things' intelligence. "This is explained as spine in IoT, it is an object, which can be tracked via space and time till its lifetime and it be maintainable, enhance able, as well as can be identifiable".[6]

From a theoretical point of view, the spine definition finds real-world implementations that are why known as Smart Items. In table 1 equipment's are present these are sort of sensors, wireless communication, memory, and elaboration capabilities.

Table 1: Visions of Internet of Things

Visions of IoT		
Things Oriented Vision	**Semantic Oriented Vision**	**Internet Oriented Vision**
1.RFID 2.UID 3.Spime 4.Smart Items 5.NFC 6.Everyday Objects 7.Wireless Sensors and Actuators	1.Semantic Technologies 2.Reasoning over data 3.Semantic execution environments	1.Connectivity for anything 2.Communicating Things 3.IPSO (IP for Smart Objects) 4.Internet 0 5.IPSO (IP for smart objects) 6.Web of things

Required capabilities are "autonomous and proactive behavior, context awareness, collaborative communications, and elaboration.Through above definitions one can define vision of the IoT, as: ''from

anytime, anyplace connectivity for anyone, we now has connectivity for anything".[7] Such vision is also available European Commission, it states about IoT ''Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts" .This IoT vision not only include ''RFID-centric" approach. Members of CASAGRAS.[8] Focus on ''the world where things can automatically communicate to computers and each other providing services to the benefit of the human kind". CASAGRAS consortium"

(i) Give an idea of IoT vision as a global infrastructure to connect virtually as well physical generic objects.

(ii) It also enlightened the existing and evolving Internet and in the vision of IoT.

According to above statements, IoT is the natural architecture for independent services plus applications, which describe distinct features by a high degree of self-data possession, phenomenon transfer, connectivity with networks and interoperability.

This leads to the role of trait union which is now known as ''Things oriented" vision and an "internet oriented vision". This category of the IoT vision is known IPSO (IP for Smart Objects) Alliance, it is the platform of 25 founding companies, it was established in September 2008 and its purpose was to nurture the Internet Protocol for connecting Smart Objects throughout the world as the network technology. In "IPSO vision IP stack is a light protocol that has joined a huge amount of interlinked devices and it is used to operate not only tiny but as well as battery operated embedded devices. This leads that IP has all the abilities to make IoT a reality. IPSO whitepapers show that through a clever IP adaptation and by connecting IEEE 802.15.4 into the IP architecture, in the view of 6LoWPAN[9] the full deployment of the IoT be enabled automatically".

The Internet also applies the same approach to reducing the difficulty of the IP stack to attain a protocol scheme to route ''IP over anything".[10] In some conferences this is looked at as the intelligent way to proceed from the Internet of devices to the Internet of Things. As specified by IPSO technique and Internet approaches, the IoT be utilized by means of a kind of simplification of the current IP to redesign it to any smart object and make those objects identifiable and to reach to destination from any location.

So on above discussion, we can say internet of vision is based on "IP for smart objects, the internet, and the web of things. ''Semantic-oriented" IoT visions are available in the literature. [11–13]. The basic thing involves behind is that the products involved in the Future Internet are scheduled to become extremely high. That's why there are issues like to represent, store, interconnect, search, and to recognize. To overcome these issues, semantic technologies can play a key role. Actually, these can use suitable modeling ways for objects description, reasoning for data produce by IoT, architectures which can facilitate IoT demands for semantic execution domain and proper storing and provide the infrastructure for exchange of information.

Another vision related to the IoT is also known as ''Web of Things".[14] according to this Web standards are utilize to connect and to integrate into the Web every-day-life things that contain an embedded device or computer".

Visions and Applications of IoT

According to the vision of IoT, every object has property to communicate with each other, understand each other and this procedure involves the communication of worldwide objects.

Embedded Systems and IoT Applications

The majority of these "things" of different visions are embedded systems, many which are running operating systems.[15] There are different operating systems in order to meet the specific challenges and realize the enormous opportunities of IoT.

An embedded system is defined as a hidden system inside an object to perform a special function. In IoT, there are embedded systems in

- Smart home devices like doors, fans, air conditioners, cookers, blinds etc
- Wearable smart devices like watches, shoes, caps etc
- Smart City device of traffic management, water distribution, urban areas living problems, etc.
- Smart grids devices for electricity consumption behavior
- Industrial internet devices of smart factories equipment, connected industrial equipment etc
- Connected car facility devices to connect with other cars, provide mapping service.
- Health monitoring devices
- Smart r shopping devices.
- Smart supply chain track devices

Smart farming devices

Sensors and Embedded systems

All above devices need different functions and different interface. The computers, browser run embedded real-

time applications are often hidden from our view. Embedded systems size is getting smaller too and previously sensors were directly connected to the central computing elements but now, they are becoming embedded systems themselves.

Sensors have a processor and it measures some physical property such as temperature, displacement, pressure, etc. and interpreted this information to central management subsystem with the help of the digital network.

Sensors and IoT Applications

Let consider the example of smart home object as shown in figure 1. In this case, many smart home devices are connected to a room, these room devices are centrally controlled by room controller device and this room controller is further connected with smart mobile via network of mobile system.[16]

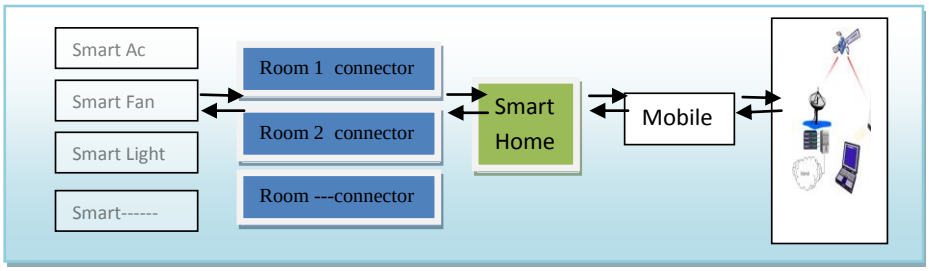

Fig. 1. Smart Ac, fan and light are standalone sensor embed devices which are under control of room objects, these all rooms are part of the smart home system which is control by mobile. The mobile system is part of the mobile network.

Scenario of IoT Application

It is easily understandable from above figure 2 that applications in IoT follow two steps scenario of embedded systems is involved in the application of the IoT.

1. Stand alone sensor objects.
2. Network sensor objects.

In this application, local devices communicate with each other with the help of embedded sensors or actuators, and then they communicate with the help of network embedded systems. Then procedure of all application can be defined generally

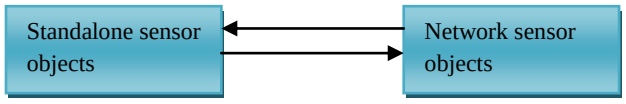

Fig. 2. Smart objects send data towards network sensor objects and vice versa

In IoT many smart devices work in standalone mode, they take input and produce the desired output. These systems have embedded intellectual property.Smartphones and cameras are examples of standalone sensor systems. In most cases, they use powerful processors and are developed by large programming teams.

In IoT application networked embedded system is used, this embedded system is actually collection of distributed embedded nodes which are interconnected by wires or wireless communication infrastructure and protocols, with some sensing and actuation elements.[17]

Network Sensor Systems of IoT are basically composed of a large number of tiny devices which actually monitoring in a specific environment and report back to a central server. IoT field is using sensor networks in all fields and all these examples of networked embedded systems. This wireless sensor technology have wide range of applications in IoT, such as in farming, environmental monitoring, health, industry.[18]

Current OS for IoT

There are many OSes for IoT but here we discuss TinyOS, Contiki, MANTIS, NanoRK and RIOT

TinyOS: TinyOS.[21] is the operating system for sensor networks and it is flexible, open source and component-based operating system. It requires slow memory requirements and has footprint that fits in 400 bytes. Its library consists of network protocols, distributed services, sensor drivers, and data acquisition tools.

TinyOS falls have monolithic architecture class. It supports multithreading and “non-preemptive First-In-First-Out (FIFO) scheduling algorithm, due to this does not support the real-time application. In sensor nodes, hardware-based memory protection is not available and the resources are limited and it uses a static memory management approach. TinyOS version 2.1.1 supports 6lowpan andIPv6.[22]

It manages the resources by Virtualization and Completion Events. It does not provide any specific MAC, network, or transport layers protocol implementations instead of it support TDMA, which can

be fine-tuned depending upon the requirements of an application to support multimedia traffic streams". It provides single level file system and supports database, security, simulation, and documentation.

Contiki: "Contiki is an open sourceOS written in C for sensor nodes. It is easily portable OS and builds on the event-driven kernel.[23] It uses preemptive multitasking can be used at the individual process level and require 2 kilobytes of RAM and 40 kilobytes of ROM. The Contiki OS follows the modular architecture.[24] It is an event-driven OS, that why does not have sophisticated scheduling algorithm.

It supports dynamic memory management and memory block management functions.It also provides a memory protection mechanism between different applications.[25] It can only support one network interface, and it supports TCP, UDP, ICMP, and IP protocols(IPv4 and IPv6).It does not allow interrupt handlers and provides serialized access to all resources.

In Contiki, there is no support for real-time applications and its additional features are, Coffee File System, which provides a programming interface for building efficient and portable storage abstractions. Coffee requires 5 Kb ROM for the code and 0.5 Kb RAM at run-time.

Contiki does not provide support for secure communication and supports sensor network simulations through Cooja. It also supports C language and sensing platforms".

MANTIS: "The MANTIS is an acronym of a MultimodAl system for Networks of In-situ wireless Sensors. It provides a new multithreaded operating system for the wireless network system of the IoT. It has two special properties. It is lightweight and energy efficient operating system. Its footprint is 500 bytes, it includes the kernel, scheduler, and network stack. it is easily portable across multiple platforms [26]. It has layered architectural design and it handles the timer interrupt. It supports preemptive multitasking and uses preemptive priority-based scheduling. It has UNIX-like scheduler with multiple priority classes.MANTIS have dynamic memory management, it manages thread tables of different threads and it does not provide any mechanism for memory protection. It provides a facility of custom routing and transport layer protocols on top of the MAC layer. So it helps in implementing real-time transport and routing protocols for multimedia sensor networks and with help of semaphores performs resource sharing".

MANTIS has "little support for real-time applications. Its additional features are wireless sensor network simulation through AVRORA, supports application development in the C language and the Unix-like shell comes with MANTIS that runs on the sensor node.

Nano-RK: Nano-RK is a real-time OS for the wireless system. It is fixed and performs preemptive multitasking in wireless sensor networks.[27] the consumption of memory in Nano-RK is 2 Kb of RAM and 18 Kb of ROM. It provides support for CPU, sensors, network bandwidth reservations and applications of hard as well as soft real-time. The real-time applications are implemented with the help of real-time scheduling algorithms. It has monolithic kernel architecture model.

It provides priority scheduling at the process level and at the network level too. It supports static memory management but does not support dynamic memory management. It uses lightweight networking protocol stack which helps in communication. It also uses Time-Synchronized Link Protocol for Energy Constraint. Nano-RK applies mutexes and semaphores for serialized access of resources like memory. Additional features provided by the Nano-RK OS are C language Support and supports MicaZ and FireFly".

RIOT OS: "RIOT OS is IoT friendly, it makes applications ready for the smaller things on the Internet with common system support of,6LoWPAN, IPv6, RPL, TCP, and UDP.[28] It provides Content-centric networking (CCN-lite).It provides Static and dynamic memory allocation. It has high resolution and long-term timers.

IOT is resource friendly. It has microkernel architecture and a tackles schedule on very lightweight devices. It provides maximum energy-efficiency and real-time capability due to ultra-low interrupt latency (~50 clock cycles) and priority-based scheduling. It provides multi-threading with ultra-low threading overhead (<25 bytes per thread). RIOT is also developer friendly".

IoT Applications and Software Requirements

As IoT is emerging field and it is challenging because of its dynamic nature of functionality. Different kinds of software especially operating systems are requiring for IoT applications, this software can be extremely complex and have a number of requirements.[20] Operating systems of IoT Applications must have some common features such as the following.[19]

Resource constrained: In it processing speed, memory size and energy supply are concerned in all applications of IoT in standalone as well as in network embedded systems.

Architecture: An OS of IoT should have the architecture of "small kernel size; hence small memory footprint.[20] The architecture must be extendable and flexible.

Real-time requirements: These are related to the process control, multimedia processing, instrumentation etc to act with real world constraints.

Programming Model: Programming models provided in IoT must be attention on developing a light-weight multithreading programming model.

Scheduling: As IoT OS must have used in real-time and as well in non-real-time environments, therefore it must be provided with scheduling algorithms that can accommodate the application requirements and have suitable scheduling algorithm" which increase the efficiency of the memory.

Memory Management and Protection: IoTconcepts have revealed that there is need of support for multiple threads execution, so memory management is required IoT OS.

Communication Protocol Support*:* "As there are heterogeneous sensor nodes in networks of IoT, therefore heterogeneity protocol mechanism property must be in OS plus it has transport, network, and MAC layer protocol implementations.

Resource Sharing*:* The majority network need some sort of multithreading, so require a resource sharing mechanism. This can be performed in time e.g., scheduling of a process/thread on the CPU and in space e.g., writing data to system memory. In some cases, we need to serialize access to resources and this is done through the use of synchronization primitives".

Portability: Many different types of CPUs, peripheral chips, and memory architectures may be used in IoT system. OS should be portable to custom hardware platforms.

Failure handling: In IoT system, nodes may fail or experience problems because of different reasons. The failure of individual nodes should not affect the overall task of the IoT network embedded system so there must be mechanisms to ensure fault tolerance to the applications.

Safety: This property is most essential part of IoT because it is concerned with the prevention of the loss of life and/or serious damage to people, property or the environment. This is straightforward for medical devices, aircraft, a car, etc.

Security: IoT applications must save data from unauthorized users. So a secure system is required.

Privacy: The main concern in IoT is the ability of an individual or group to isolate them.

Scalability: This property indicates the system ability to either handle growing amounts of work in a graceful manner or to be readily enlarged.

Upgradability: This property is concerned with the degree to which a computer may have its specifications improved by the addition or replacement of components. Most system designers will expect the operating system to make the task easier and to handle some difficult problems like upgrade policy, verification, and security.

Current OS Compatibility for the IoT applications

We have discussed the operating systems for the IoT, now we summarize these operating systems for IoT domain according to the above mention properties for operating system software. Here 'C' is used for complete support,'P' for partial support and 'N for no support

Table 2. Requirements provided by OSes for IoT Applications

OS features/Current OS	TinyOS	Contiki	Mantis	NanoRK	RIOT
Resource constrained Computing	C	C	C	C	C
Real-time requirements	P	P	P	C	C
Portability	C	C	C	C	C
Failure handling	P	P	P	P	C
Safety	N	N	N	N	P
Security	N	N	N	N	P
Privacy	N	N	N	N	P
Scalability	C	C	C	P	P
Upgradability	N	P	C	P	C
Architecture(Micro Kernel)	N	P	N	N	C
Programming model(Light Weight)	P	p	N	N	P
Scheduling(Real time and non Real time use)	N	P	P	P	P
Memory Management and Protection	p	p	p	p	p

Communication Protocol Support	p	p	p	p	P
Resource Sharing	P	C	P.	C	P
Support for Real-time Applications	N	N	P	C	C

According to the table 2 few OSs only support for real-time application for IoT and some OSs provide support for priority scheduling. Nano-RK and RIOT OSs support for real-time applications .It is also observed that early wireless networks OSs stress on event driven programming paradigm.

Conclusion

In this paper, we have examined the IoT applications and the most widely used operating systems for IoT. In it IoT OS features requirements are first analyzed and secondly compared with current IoT OS. This survey helps to understand the OS requirements for IoT and which features should be emphasized in the OS for the applications of IoT

Future Work

IoT is under developing field; its applications are not yet mature so there is plenty of research work has to be done specially for new sensor types. Real time applications, memory management, security, and communication protocol should be addressed in modern OS of IoT in future research.

References

1. D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of Things, 1661 Springer, 2010. ISBN: 978-1-4419-1673-0
2. Harald Sundmaeker,Patrick Guillemin,Peter Friess Sylvie Woelffl,Vision and Challenges for Realising the Internet of Things, European Union 2010,ISBN 9789279150883
3. INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems, in: Co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future, Version 1.1, 27 May 2008
4. Knud Lasse Lueth,"IoT basics: Getting started with the Internet of Things", White paper 2015 March.
5. L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A Survey," Computer Networks, Vol. 54, No. 15, May 2010, pp. 2787-2805.
6. B. Sterling, Shaping Things – Media work Pamphlets, the MIT Press, 2005.
7. ITU Internet Reports, the Internet of Things, November 2005.
8. A. Dunkels, J.P. Vasseur, IP for Smart Objects, Internet Protocol for Smart Objects (IPSO) Alliance, White Paper #1, September 2008,<http://www.ipso-alliance.org>.
9. J. Hui, D. Culler, S. Chakrabarti, 6LoWPAN: Incorporating IEEE 802.15.4 Into the IP Architecture – Internet Protocol for Smart Objects (IPSO) Alliance, White Paper #3, January 2009, <http:// www.ipso-alliance.org>. [13] N. Gershenfeld, R. Krikorian, D. Cohen, The internet of things, Scientific American 291 (4) (2004) 76–81.
10. N. Gershenfeld, R. Krikorian, D. Cohen, The internet of things, Scientific American 291 (4) (2004) 76–81.
11. A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, V. Terziyan, Smart semantic middleware for the internet of things, in Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics, Funchal, Madeira, Portugal, May 2008.
12. W. Wahlster, Web 3.0: Semantic Technologies for the Internet of Services and of Things, Lecture at the 2008 Dresden Future Forum, June 2008.
13. I. Vázquez, Social Devices: Semantic Technology for the Internet of Things, Week@ESI, Zamudio, Spain, June 2009.
14. D. Guinard, T. Vlad, towards the web of things: web mashups for embedded devices, in Proceedings of the International World Wide Web Conference 2009 (WWW 2009), Madrid, Spain, April 2009.
15. Ovidiu Vermesan & Peter Friess, SINTEF, NO EU, BE, "Internet of Thing Applications from Research and Innovation to Market Deployment", The River Publishers Series in Communications, June 2014
16. Rischan Mafrur, Priagung Khusumanegara, Gi Hyun Bang, Do Kyeong Lee, I Gde Dharma Nugraha and Deokjai Choi , Developing and Evaluating Mobile Sensing for Smart Home Control International Journal of Smart Home, ISSN: 1975-4094 IJSH Vol. 9, No. 3 (2015), pp. 215-230.
17. Pakala, Vignana Bharathi, Ghatkesar, Khan, Raju,, "Sensors integration in embedded systems", ISBN:978-1-4244-8543-7, IEEE international Conference on, Power, Control and Embedded Systems (ICPCES), Dec.2010.
18. Industry Agenda, "Industrial Internet of Things: Unleashing the Potential of Connected Products and Services", World Economic Forum, Jan.2015.
19. T Liu, CM Sadler, "P Zhang, M Martonos," Implementing software on resource- constrained mobile sensors: Experiences with Impala and ZebraNet", 2nd international conference on Mobile systems, 2004.
20. Muhammad Omer Farooq and Thomas Kunz, "Operating Systems for Wireless Sensor Networks: A Survey"Sensors2011, 11(6), 5900-5930, 31 May 2011
21. Levis, P; Madden, S; Polastre, J; Szewczyk, R; Whitehouse, K; Woo, A; Gay, D; Hill, J; Welsh, M; Brewer, E; Culler, D. Tinyos: An Operating System for Sensor Networks, Available online: http://dx.doi.org/10.1007/3-540-27139-2_7 (accessed May 2015).
22. Isam Ishaq *, David Carels, Girum K. Teklemariam, Jeroen Hoebeke, Floris Van de Abeele, "IETF Standardization in the Field of the Internet of Things (IoT):

A Survey", Journal of Sensor and Actuator Networks ISSN 2224-2708, 2013.

23. Dunkels, A; Gronvall, B; Voigt, T. Contiki a Lightweight and Flexible Operating System for Tiny Networked Sensors. Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks, Washington, DC, USA, October 2004; pp. 455–462.
24. http://www.contiki-os.org/ (accessed on May 2015)
25. https://github.com/contiki-os/contiki (accessed on May 2015)
26. Bhatti, S; Carlson, J; Dai, H; Deng, J; Rose, J; Sheth, A; Shucker, B; Gruenwald, C; Torgerson, HR. Mantis OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. Mob. Netw. Appl2005, 10, 563–579. [Google Scholar]
27. Eswaran, A; Rowe, A; Rajkumar, R. Nano-RK: An Energy-Aware Resource-Centric RTOS for Sensor Networks. Proceedings of the 26th IEEE Real-Time Systems Symposium, Miami, FL, USA, 5–8 December 2005.
28 .http://www.riot-os.org/(accessed 30 May 2015)