International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P), Volume-6, Issue-3, November 2016

Iot for Self Monitoring Analysis Remote Transducers

T.Ram Babu, Laxmiganesh, A.Gowrisankar

***Abstract*— This paper discusses the vision, the challenges, possible usage scenarios and technological building blocks of the "Internet of Things". in particular, we consider raspberry pi and other important technological developments such as IP stack, wireless networks and web servers, python programming for smart everyday objects. The paper concludes with a discussion of social innovation and governance issues that are likely to arise as the vision of the internet of things.**

***Index Terms*— Internet of Things, Raspberry pi, Raspberry pi OS, IP-Stack, HTML Coding, wireless networks, python programming, cisco packet tracer simulation tool**

I. Introduction

The Internet of Things represents a vision in which the Internet extends into the real world embracing everyday objects. Physical items are no longer disconnected from the virtual world, but can be controlled remotely and can act as physical access points to Internet services. An Internet of Things makes computing truly ubiquitous – a concept initially put forward by Mark Weiser in the early 1990s [29]. This development is opening up huge opportunities for both the economy and individuals. However, it also involves risks and undoubtedly represents an immense technical and Social challenge.

The Internet of Things vision is grounded in the belief that the steady advances in microelectronics, communications and information technology we have witnessed in recent years will continue into the foreseeable future. In fact – due to their diminishing size, constantly falling price and declining energy consumption – processors, Communications modules and other electronic components are being increasingly integrated into everyday objects today.

"Smart" objects play a key role in the Internet of Things vision, since embedded communication and information technology would have the potential to revolutionize the utility of these objects. Using sensors, they are able to perceive their context, and via built-in networking capabilities they would be able to communicate with each other, access Internet services and interact with people. "Digitally upgrading" conventional object in this way enhances their physical function by adding the capabilities of digital objects, thus generating substantial added value. Forerunners of this development are already apparent today – more and more devices such as sewing machines, exercise bikes, electric toothbrushes, washing machines, electricity meters and photocopiers are being "computerized" and equipped with network interfaces.

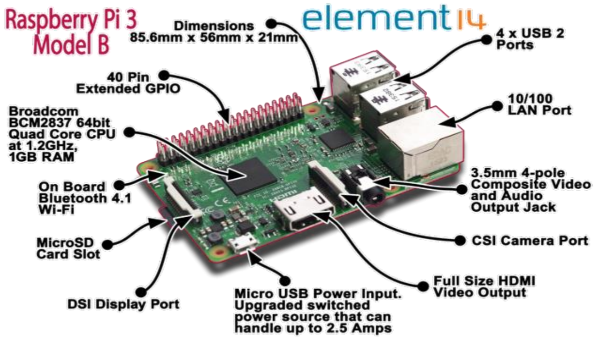
T.Rambabu, is pursuing his master of Technical, Department of computer science engineering, Gokul Institute of Technology and Science,Piridi, BobbiliVizianagaram
B.Ganesh ,Associate professor,Department of Computer Science, Gokul Institute of Technology and Science,Piridi,BobbiliVizianagaram
ALLU Gowrisankar is pursuing his master of Technical in Embedded systems in Miracle education society group of institutions,vizianagaram.

II. Raspberry pi

A. The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

B. A lot of computer companies were named after fruit. There's Tangerine Computer Systems, Apricot Computers, and the old British company Acorn, which is a family of fruit. "Pi is because originally we were going to produce a computer that could only really run Python. So the Pi in there is for Python.

C. Raspbian comes preloaded with Python, the official programming language of the Raspberry Pi and IDLE 3, a Python Integrated Development Environment



III. Raspberry pi OS

Raspbian is the "official" operating system of the Raspberry Pi and because of that, it's the one most people will want to start with.Raspbian is a version of Linux built specifically for the Raspberry Pi. It comes packed with all the software you'll need for every basic task with a computer. You'll get LibreOffice as an office suite, a web browser, email program, and some tools to teach programming to kids and adults alike. Heck, it even includes a special (no longer in development) version of Minecraft. Raspbian is the backbone for pretty much every DIY project out there, so if you're looking to make something, Raspbian is most likely where you want to start. Because it's so widely used, it's also easy to find guides and troubleshooting tips.

IV. IP stack

If, in a future Internet of Things, everyday objects are to be addressed and controlled via the Internet, then we should ideally not be resorting to special communications protocols as is currently the case with RFID. Instead, things should behave just like normal Internet nodes. In other words, they should have an IP address and use the Internet Protocol (IP) for communicating with other smart objects and network nodes. And due to the large number of addresses required, they should use the new IPv6 version with 128-bit addresses. The benefits of having IP-enabled things are obvious, even if the objects in question are not going to be made globally accessible but instead used in a controlled intranet environment. This approach enables us to build directly on existing functionality such as global interoperability, network-wide data packet delivery (forwarding and routing), data transport across different physical media, naming services (URL,DNS) and network management. The use of IP enables smart objects to use existing Internet services and applications and, conversely, these smart objects can be addressed from anywhere since they are proper Internet participants. Last but not least,it will be easy to use important application layer protocols such as HTTP. IPv6 also provides the interesting capability of automatic address configuration, enabling smart objects to assign their own addresses. Until recently, however, the prospect of full IP support for simple things appeared illusory due to the resources required (such as processor capacity and energy) and thus the costs involved. Instead, it was suggested to connect smart objects to the Internet indirectly via proxies or gateways. But the disadvantage of such non-standardized solutions is that end-to-end functionality is lost because standardized Internet protocols would be converted to proprietary protocols over the last few meters. Gateways would also generate added complexity, making installation, operation and maintenance time-consuming and costly. However, there are now not only 16-bit microcontrollers with sufficient storage that require less than 400 µW/MIPS, but also TCP/IPv6 stacks that can operate with 4 kB RAM and 24 kB flash memory [13]. Equally important are wireless communications standards such as IEEE 802.15.4 that cover the layers below IP and consume relatively little power – ZigBee implementations require approximately 20 to 60 mW (for 1 mW transmission power, a range of 10 to 100 meters and a data transmission rate of 250 Kbit/s). Whenever possible, the wireless unit is being used for short periods of time only in order to save energy. This approach enables AA batteries to provide a modest level of computing power and wireless communication that is nevertheless sufficient for many purposes over many months. The opportunities that this opens up have recently led to companies and standards committees adopting various measures. At the end of 2008, Atmel, Cisco, Intel, SAP, Sun Microsystems and other companies founded the "IP for Smart Objects" (IPSO) corporate alliance to promote the implementation and use of IP for low-powered devices such as radio sensors, consumption meters and other smart objects. More specifically, the "IPv6 over Low Power Wireless Area Networks" (6LoWPAN) working group set up by the Internet Engineering Task Force (IETF) is addressing the problem of supporting IPv6 using the 802.15.4 wireless communication standard [14]. This is a technical challenge because the maximum length of 802.15.4 data frames is only 127 bytes due to lower data rate, higher susceptibility to failure and bit error rate of wireless communications. The IPv6 packet header alone is 40 bytes long (primarily due to the source and target addresses each being 16 bytes long), and fragmented IPv6 packets can be up to 1280 bytes long. To make IPv6 communications function efficiently in wireless networks, a protocol modification layer has been defined that essentially deals with four issues –embedding IPv6 packets in 802.15.4 frames, fragmenting long packets to fit these frames, stateless compressing packet headers (typically to just 6 bytes), and forwarding IPv6 packets via multihop wireless routes. It is possible to compress the IPv6 header so drastically because 802.15.4 nodes communicate mainly within their own wireless network, and therefore most of the information can be reconstructed from the general context or the surrounding 802.15.4 frames and considerably shorter local addresses can be used. The working group's proposal has now been published as proposed Internet standard RFC 4944, and an implementation based on this is described in [13]. In 2009, the ZigBee Alliance announced it would be incorporating this "native IP support" into future ZigBee specifications, "allowing seamless integration of Internet connectivity into each product".

V. WIRELESS NETWORKS

When the term 'wireless network' is used today, it usually refers to a wireless local area network (WLAN). A WLAN connects computers together through radio technology using standard network rules or protocols, but without the use of cabling to connect the computers together. A WLAN can be installed as the sole network in a school or building. However, it can also be used to extend an existing wired network to areas where wiring would be too difficult or too expensive to implement, or to areas located away from the main network or building. The most obvious difference between wireless and wired networks, therefore, is that the latter uses some form of cable to connect computers together. A wireless network does not need cable to form a physical connection between computers. Wireless networks can be configured to provide the same network functionality as wired networks, ranging from simple peer-to-peer configurations to large-scale infrastructures accommodating hundreds of users. paradigm. Sensor nodes may not have global ID because of the large amount of overhead and large number of sensors. WSN based on IoT has received remarkable attention in many areas, such as military, homeland security, healthcare, precision agriculture monitoring, manufacturing, habitat monitoring, forest fire and flood detection and so on [26]. Sensors mounted to a patient's body are monitoring the responses to the medication, so that doctors can measure the effects of the medicines.

VI. PYTHON PROGRAMMING

Python was conceived in the late 1980s,[31] and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL)[33] capable of exception handling and interfacing with the operating system Amoeba.[8] Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, benevolent dictator for life (BDFL).

About the origin of Python, Van Rossum wrote in 1996: Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.[24][25] Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java.[26][27] The language provides constructs intended to enable writing clear programs on both a small and large scale.[28]

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.[29]

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. Using third-party tools, such as Py2exe or Py installer Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, so Python-based software can be distributed to, and used on, those environments with no need to install a Python interpreter.

VII. CISCO PACKETRACER

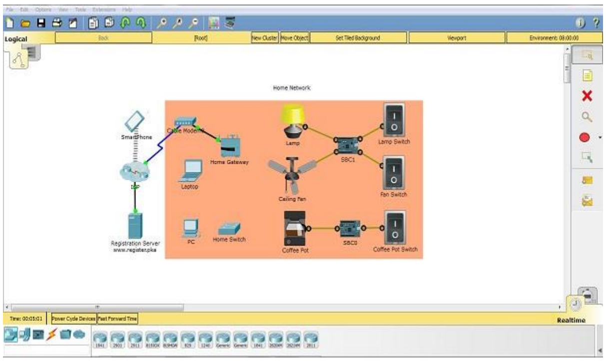
A. Packet Tracer is a cross-platform network simulator designed by Cisco Systems to run on Mac OS, Linux and Microsoft Windows. A similar Android app is also available. Packet Tracer allows users to create simulated network topologies by dragging and dropping routers, switches and various other types of network devices. A physical connection between devices is represented by a "cable" item. Packet Tracer supports an array of simulated Application Layer protocols, as well as basic routing with RIP, OSPF, EIGRP, BDP, to the extents required by the current CCNA curriculum. As of version 5.3, Packet Tracer also supports the Border Gateway Protocol.

Version 6.0 added support for IOS version 15 and Hot Standby Routing Protocol. Version 6.1.1 added support for various DHCP, EIGRP and OSPF commands, improved support for Zone-Based Firewall policies. As of version 6.2, Packet Tracer supports an embedded web server with JavaScript and CSS support. The command line can be used for creating a router-to-pc connection

Packet Tracer allows students to design complex and large networks, which is often not feasible with physical hardware, due to costs.[3] Packet Tracer is commonly used by CCNA Academy students, since it is available to them for free.[1] However, due to functional limitations, it is intended by Cisco to be used only as a learning aid, not a replacement for Cisco routers and switches.[5] The application itself only has a small number of features found

Within the actual hardware running a current Cisco IOS version. Thus, Packet Tracer is unsuitable for modeling production networks. It has a limited command set, meaning it is not possible to practice all of the IOS commands that might be required.[7]

Packet Tracer can be useful for understanding abstract networking concepts, such as the Enhanced Interior Gateway Routing Protocol by animating these elements in a visual form.[3][7] Packet Tracer is also useful in education by providing additional components, including an authoring system, network protocol simulation and an assessment system



VIII. CONCLUSION

In this paper, we analyzed the solutions currently available for the implementation of urban IoTs. The discussed technologies are close to being standardized, and industry players are already active in the production of devices that take advantage of these technologies to enable the applications of interest, such as those described in Section II. In fact, while the range of design options for IoT systems is rather wide, the set of open and standardized protocols is significantly smaller. The enabling technologies, furthermore, have reached a level of maturity that allows for the practical realization of IoT solutions and services, starting from field trials that will hopefully help clear the uncertainty that still Prevents a massive adoption of the IoT paradigm.

REFERENCES

[1]. Adelmann, R., Langheinrich, M., Floerkemeier, C.: A Toolkit for Bar Code Recognitionand Resolving on Camera Phones – Jump-Starting the Internet of Things. Proc. Workshop Mobile and Embedded Interactive Systems. In: Hochberger, C., Liskowsky, R. (eds.)Informatik 2006 – GI Lecture Notes in Informatics (LNI) 94, pp. 366–373 (2006)

[2]. Ashton, K.: That 'Internet of Things' Thing. RFID Journal, www.rfidjournal.com/article/print/4986 (2009)

[3]. Coroama, V.: The Smart Tachograph – Individual Accounting of Traffic Costs and its Implications. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A.J. (eds.) Proc. Pervasive 2006, LNCS 3968, Springer, pp. 135–152 (2006)

[4]. Duquennoy, S., Grimaud, G., Vandewalle, J.-J.: Smews: Smart and Mobile Embedded Web Server. Proc. Int. Conf. on Complex, Intelligent and Software Intensive Systems, pp. 571–576 (2009)

[5]. European Commission: Internet of Things – An action plan for Europe. COM(2009)278,http://eur-lex.europa.eu/LexUriServ/site/en/com/2009/com2009_0278en01.pdf (2009)

[6]. Fleisch, E., Mattern, F.: (eds.) Das Internet der Dinge. Springer (2005)

[7]. Fleisch, E.: What is the Internet of Things? When Things Add Value. Auto-ID Labs White Paper WP-BIZAPP-053, Auto-ID Lab St. Gallen, Switzerland (2010)

[8]. Floerkemeier, C., Mattern, F.: Smart Playing Cards – Enhancing the Gaming Experience with RFID. In: Magerkurth, C., Chalmers, M., Björk, S., Schäfer, L. (eds.) Proc. 3rd Int.Workshop on Pervasive Gaming Applications – PerGames 2006, pp. 27–36 (2006)

[9]. Floerkemeier, C., Langheinrich, M., Fleisch, E., Mattern, F., Sarma, S.E.: (eds.) the Internet of Things. First International Conference, IOT 2008, LNCS 4952, Springer (2008)

[10]. Frank, C., Bolliger, P., Mattern, F., Kellerer, W.: The Sensor Internet at Work: Locating Everyday Items Using Mobile Phones. Pervasive and Mobile Computing 4(3):421–447(2008)

[11]. Gershenfeld, N.: When Things Start to Think. Henry Holt and Company (1999)12. Guinard, D., Trifa, V., Wilde, E.: Architecting a Mashable Open World Wide Web of Things. TR CS-663 ETH Zürich, www.vs.inf.ethz.ch/publ/papers/WoT.pdf (2010)

[13]. Hui, J., Culler, D.: IP is Dead, Long Live IP for Wireless Sensor Networks. Proc. 6th Int. Conf. on Embedded Networked Sensor Systems (SenSys), pp. 15–28 (2008)
[14]. Hui, J., Culler, D., Chakrabarti, S.: 6LoWPAN – Incorporating IEEE 802.15.4 into the IP architecture. Internet Protocol for Smart Objects Alliance, white paper # 3 (2009)
[15]. International Telecommunication Union: The Internet of Things. ITU (2005)
[16]. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., Spasojevic, M.: People, Places, Things: Web Presence for the Real World. Mobile Networks and Applications 7(5):365–376 (2002)
[17]. Kollmann, K.: Das „Internet of Things" – Der kurze Weg zur kollektiven Zwangsentmündigung. Telepolis, www.heise.de/tp/r4/artikel/30/30805/1.html (2009)
[18]. Kramarz, D., Loeber, A.: Visualisierung von Transponder-Daten mittels Mashup. Diplomarbeit, Zürcher Hochschule für Angewandte Wissenschaften (2007)
[19]. Mattern, F., Floerkemeier, C.: Vom Internet der Computer zum Internet der Dinge. Informatik-Spectrum 33(2):107–121 (2010)
[20]. Mattern, F., Staake, T., Weiss, M.: ICT for Green – How Computers Can Help Us to Conserve Energy. Proc. e-Energy 2010, ACM, pp. 1–10 (2010)
[21]. National Intelligence Council Global Trends 2025: A Transformed World. www.dni.gov/Nic/NIC_2025_project.html (2008)
[22]. Sarma, S., Brock, D.L., Ashton, K.: The Networked Physical World. TR MIT-AUTOIDWH- 001, MIT Auto-ID Center (2000)
[23]. Schoenberger, C.R.: The internet of things. Forbes Magazine, March 18 (2002)
[24]. Spiekermann, S., Pallas, F.: Technology paternalism – wider implications of ubiquitous computing. Poiesis & Praxis 4(1):6–18 (2006)
[25]. SRI Consulting Business Intelligence: Disruptive Civil Technologies – Six Technologies with Potential Impacts on US Interests out to 2025. www.fas.org//nic/.pdf (2008)
[26]. SRI Consulting Business Intelligence: Disruptive Civil Technologies, Appendix F: The Internet of Things (Background). www.dni.gov/nic/PDF_GIF_confreports//_F.pdf (2008)
[27]. Thiesse, F.: RFID, Privacy and the Perception of Risk: A Strategic Framework. The Journal of Strategic Information Systems 16(2):214–232 (2007)
[28]. Thiesse, F., Floerkemeier, C., Harrison, M., Michahelles, F., Roduner, C.: Technology, Standards, and Real-World Deployments of the EPC Network. IEEE Internet Computing 13(2):36–43 (2009)
[29]. Weiser, M.: The Computer for the 21st Century. Scientific American 265(9):66–75 (1991)

T.Ram babu, is pursuing his master of Technical, Department of computer science engineering,GokulInstitute of Technology and Science,Piridi, BobbiliVizianagaram**,** He has completed his B.Tech in Electronics and communication engineering from lendi college of engineering and technology.His area of interest in research in computer networks

B.Ganesh ,Associate professor,Department of Computer Science, Gokul Institute of Technology and Science,Piridi,BobbiliVizianagaram

ALLU Gowrisankar is pursuing his master of Technical in Embedded systems in Miracle education society group of institutions,vizianagaram. He has completed his B.Tech in Electronics and communication engineering from prasiddha college of engineering and technology.His area of interest in research in Embeddedsystems and robotics