International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 7, July 2017)

A Survey of Hardware and Software Components Required to Build an IoT Solution

Arunita Kundaliya[1], Hem Dutt Dabral[2]

[1]*M.Tech, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, 110067, India*
[2]*M.S. (Software Systems) Birla Institute of Technology & Science, Pilani - 333031. Rajasthan, India*

***Abstract*— "Internet of Things" or IoT refers to an inter-network of physical "things". IoT "things" collaborate with other such "things", their environment, and cloud based applications to provide useful services to the user. "Things" can be ultra violet light sensors, smart speaker systems, smart washing machines, smart refrigerators, connected cars, surveillance cameras, smart door-bells, point-of-sale terminals, smart televisions etc. IoT paradigm aims to inter-connect these "things" with each other and optionally to cloud-based-applications in a secure manner, so as to provide useful services to the user. Sometimes "things" are also referred as "end-devices", edge-device, or "nodes". In this paper, we will refer to things/end-device/nodes/edge-device as "end-device". IoT end-devices are generally expected to work with-out requiring to be recharged for a long time. So one important constraint for end-devices is power efficiency but this constraint can be relaxed when an IoT system has a power source connected to it e.g. a connected car. Other important constraints for IoT systems are privacy and security of information. Any compromise on user privacy and information security will effectively make IoT system useless.**

In this paper we wish to study various aspects of software and hardware components needed to develop such systems.

***Keywords*— Actuators, Cloud, CoAP, end-device, gateway, Internet of things, IoT, IPSec, Microcontrollers, MQTT, Sensors, Security, things, TLS, WSN**

I. INTRODUCTION

Internet is a global system of interconnected computer networks which uses TCP/IP as their main networking protocol to connect with each other. In the last decade of 21st century internet became very popular worldwide and as of today almost forty percent of world population has an internet connection. This is an overwhelming number. Internet has provided cheap, fast, and reliable infrastructure for "people" to communicate with each other.

Recent advancements in the field of electronics manufacturing has made it possible to manufacture electronic components like sensors, micro-controllers, actuators, communication hardware etc. at very low cost.

Also size of these components has decreased drastically. As a result, it is now possible to manufacture small-sized, application specific electronic devices at very low cost. As shown in Figure 1, an "end-device" typically consists of following parts:

i. Microcontroller
ii. Sensor sub-system
iii. Actuator sub-system
iv. Communication sub-system
v. Battery

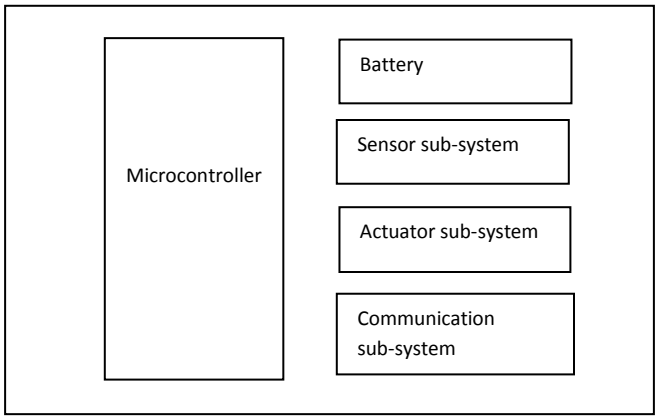

Figure 1 End-Device block diagram

The vision of IoT is to use the massive infrastructure of "internet" for connecting these small electronic devices with each-other, and provide useful services to the end user. Some of these services are Smart cities, Smart buildings, Smart home, smart health, smart transport, Industrial IoT etc. Usually a "Gateway" device is used as an intermediary between end-device and cloud. Figure 2 shows a typical IoT scenario in which an end-user is trying to access the services of the IoT system using PC/Tablet/Smartphone based application or a web-browser.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 7, July 2017)

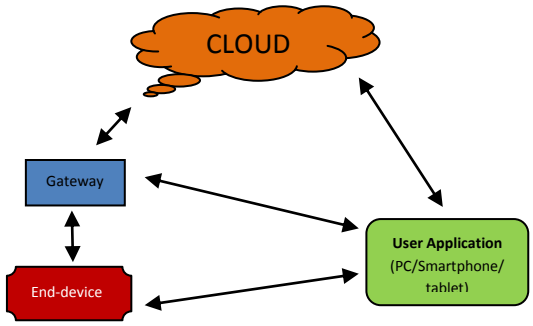

Figure 2 IoT scenario

The main elements of an IoT system are described below:

i. *End-Device:* End devices collect application specific information using sensors. This information is then forwarded to a gateway in secured manner.
ii. *Gateway:* Gateways are devices which provide functions such as commissioning an end-device to the network and protocol translation. Gateway may also process information received from various end-devices, and take some decisions locally. Optionally, in IOT with cloud computing (cloud assisted IOT), this information may be forwarded to cloud based applications.
iii. *Cloud side application:* Cloud side applications combine and process information received from gateways and returns the results to gateway or user application.
iv. *User application:* User application provides an interface to the end-user for accessing services of IoT system.

It is important to note that end-user may not be aware about the existence of various elements of the IoT system. Remaining part of this paper will explain the characteristics of hardware and software components, required to build these IoT elements in more details.

II. Hardware Components

IoT end-devices are resource-constrained devices with limited amount of processing capability and battery power. They are supposed to work without needing to recharge their batteries for long periods of time. At their core, they contain miniature computer systems suitable for the target application. Main hardware components required to build such a device are described in this section.

A. Microcontroller

A Microcontroller is an Integrated circuit containing all components of a computer system namely processor(s), Memories (e.g. RAM, ROM, EEPROM, Flash), and IO devices (e.g. UART, I2C, SPI, GPIOs, ADC, DAC). There are many electronic hardware manufacturing companies like Arduino, ARM, Atmel, Maxim Integrated, NXP, STMicroelectronics etc. which are providing a range of inexpensive microcontroller boards which can be used for creating applications. Following diagram shows image of Arduino 101 microcontroller board which is a popular microcontroller board for creating IoT applications.

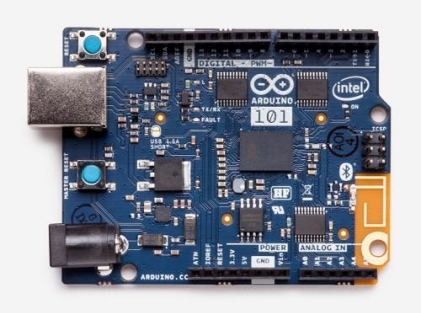

Figure 3 Arduino 101 microcontroller board

Microcontroller vendors provide support for either online or offline Integrated Development Environment (IDE) for creating end-device firmware application. *IAR Embedded Workbench* and *KEIL uVision* are popular offline IDEs. On the other hand Arduino and mbed.org provide support for online web-based IDE. The middleware software components like device drivers and Board Support Packages (BSP) are provided by their respective hardware vendor. Developer can start working on their target IoT application using these resources. Choosing appropriate microcontroller for the target application is first important decision that must be addressed while designing the end-device. Some of the important constraints for microcontroller selection are cost, computational requirement, battery requirements, and possibility to integrate sensors and actuators required for the application. It is important to note that, unlike personal computers, in an embedded system it is not possible to upgrade computational capabilities of the system once it has been manufactured. So, a designer must take care of all future system requirements while selecting a microcontroller.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 7, July 2017)

B. Sensors

An IoT end-device may need to sense its surroundings in order to function properly e.g. a thermostat may need temperature and humidity sensors for performing its intended task. Other commonly used sensors are gyroscope, magnetometer, people-in-range, optical, and pressure sensors. These sensors are usually available as separate electronic boards for easy prototyping. To communicate with the microcontroller a sensor can use standard protocols like Inter-Integrated Circuit (I2C), Universal Asynchronous Receiver Transmitter (UART), Serial Peripheral Interface (SPI), Universal Serial Bus (USB) etc. The hardware manufactures provide middleware software for using their respective sensors, but hardware components of different electronic manufactures may not be compatible with each other.

C. Actuators

Actuators are used to control electrical or mechanical characteristics of a system. An end-device can provide such functionality by using suitable hardware component to provide control signals to the actuator sub-system. E.g. a combination of General Purpose Input Output (GPIO) and relay boards can be used to support on-off functionality for a high voltage electric device (e.g. an air-conditioner).

D. Communication Hardware

An end-device has to exchange information and control signals with other elements of the IoT system. This is achieved by using a dedicated communication sub-system. This communication can be based on any of following technologies:

i. Bluetooth Low Energy (BLE)
ii. Near Field communication (NFC)
iii. Radio Frequency Identification (RFID)
iv. Wi-Fi
v. Zigbee
vi. Cellular connectivity
vii. LoRaWAN
viii. Sigfox
ix. Thread

A system designer can select any of these protocols depending on the requirement of the system being designed. In some cases, more than one communication methodologies may be needed to satisfy a usage scenario e.g. a designer can using NFC for device configuration while some other technology for inter device communication between various devices. In some cases microcontroller board integrates hardware for communication (e.g. Arduino 101 includes BLE support).

But if communication support is not present developer may use microcontroller GPIOs for using an external communication related hardware component.

E. Battery

A battery is needed for running all the operations in the end-device. An IoT device may use lithium (typically used in smartphones), nickel, or alkaline (A, AA, AAA etc.) batteries. Lithium and nickel based batteries are rechargeable while alkaline batteries are generally not rechargeable. Each type of battery is available in varied capacities which are measured in Ampere-hour. Some applications like forest fire detection systems may require long periods between battery replacements. In such cases a high capacity battery will be needed and software-hardware design should consider using strategies to ensure low power consumption. For designing a practical end-device, analyzing its battery requirement and equipping it with a battery of suitable capacity is very important.

III. Software Components

All three elements of an IoT solution (end-device, gateway, and cloud) need suitable software components to perform their task. Besides this, user may also need a smartphone/PC/tablet based application for accessing and configuring vital parameters of these three components. In this section we will provide an overview of various software components used in IoT systems.

A. End-Device Software

An end-device can be perceived as a resource constrained computer. So its software design is influenced by the motivation to use hardware resources efficiently. Minimizing latency is an important criterion for end-devices, so *Real Time Operating System* (RTOS) based approach is preferred. In some cases the operating system can be nothing but amalgamation of device drivers, Board Support Packages (BSP), and initialization routines followed by a never ending while-loop, which includes main application logic for end-device. In most cases even dynamic memory management is not needed. To process events, the microprocessor uses interrupts. Some of the constraints on end-device software are:

i. *CPU clock Speed:* Minimize CPU clock speed to save on battery and avoid device heating.
ii. *Main memory requirement:* Memory footprint should be low so as to reduce the requirement of RAM.
iii. *ROM size:* To reduce ROM requirement, a small binary size of application is desired.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 7, July 2017)

iv. *Battery Usage:* Communication sub-system should be turned ON only while sending or receiving data. Otherwise, it should be turned OFF, or put in a low power consuming sleep state.
v. *Power:* Only those peripheral clocks, which are used in the application should be turned ON. All other peripheral clocks should remain in OFF state.

B. Gateway Software

A gateway is very critical part of the IoT infrastructure. It performs following functions:

i. Commissioning or on boarding new end-devices to the network.
ii. Connect end-devices to internet.
iii. Protocol translation between various end-devices in the network and between end-device and internet.
iv. Process information received from end-devices and perform local decision making.
v. Network security.
vi. Manage firmware updates for end-devices.

A gateway is required, as it is not always possible to connect all end-devices directly to internet. This can happen in any of the following situations:

i. An end-device is a fairly simple piece of hardware with no support of Wi-Fi or cellular connectivity.
ii. An end-device is capable of supporting some type of internet connectivity but still it may not be desirable to connect all end-devices directly to internet e.g. When internet connectivity infrastructure cost is dependent on the number of connections being used at a given time.

The Gateways can use application layer protocols like Message Queue Telemetry Transport (MQTT), Constrained Application Protocol (CoAP) etc. for communicating with cloud applications. MQTT uses publish-subscribe mechanism for transferring data between various nodes in the network. CoAP is designed for easy translation to HTTP which facilitates easy integration of end-device communication with the web. In some situations, an end-device can also perform the task of a gateway and hence a separate hardware entity is not needed. The trade-off of using or not using a dedicated gateway device should be analyzed carefully by the system designer. Having just one gateway in the system introduces single point of failure for cloud dependent applications. In such scenarios more than one devices having gateway capabilities can be used, if one of them fails others can take over the responsibilities of failed device.

C. Cloud Side Application

Cloud side application is usually the most sophisticated piece of software used in the IoT systems. End-devices and gateways have limited processing capabilities. So, it may not be feasible to perform complex analysis using them. Sometimes meaningful processing of information received can only be done if an aggregation of information from more-than-one end devices is taken into account, necessitating the use of a cloud based application. There are many cloud service providers providing different types of services like AWS, IBM Watson, and Microsoft Azure etc

D. User Application

Usability of an IoT system is highly dependent on the ease with which user can access various elements of the IoT system like end-devices, gateways, and cloud side objects. A user application is used for:

i. Accessing information for which IoT system is implemented.
ii. Commissioning new devices to the network.
iii. Configuring end-devices and gateways.
iv. Knowing the state of various end-devices, gateways etc.

User application is usually a Smartphone/PC/tablet based application, which provides an interface to the end-user for accessing services of the IoT system. The ease with which a user is able to access an IoT application is the single most important factor in deciding the usability of the IoT system. Sometimes HTML browser based application can also be used for user interfacing with the IoT system.

IV. Privacy And Security

In IoT scenario entities like people, end-devices, software, and hardware, exchange information with each other using public-untrusted IP networks. In such scenario any compromise on user privacy and information security can make the IoT system unreliable (and thus useless). One particular type of threat model is suggested by Dolev-Yao (DY) [6, 7]. DY intruder can intercept, overhear, and synthesize any message exchanged over the network. IoT network infrastructure should be DY attack resilient.

Various types of threats present IoT scenario are:

i. *Physical attack:* Due to distributed nature of IoT some devices may operate in un-secured, outside environment. So, a hacker may physically temper or replace a node. Tempering may include hardware changes (e.g. installing a new motherboard), or software modifications (e.g. changing the firmware).

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 7, July 2017)

For avoiding such attacks end-devices and gateways should be made either physically temper-proof or temper revealing. Also, only trusted and signed software should be able to run on an end-device.

ii. *Man in the middle attack (MITM attack):* An adversary may secretly relay or alter the communication between any two entities (end-device, gateway, people, cloud) in the IoT system. He may use such information to his advantage.
iii. *Attack on privacy:* As bulk of the information travels to the cloud. An adversary may access information and data through remote access mechanisms.
iv. *Denial of Service attack (DOS):* Denial of service attacks occur when hacker floods IoT system with overwhelming number of requests which can make the system services unavailable to the actual user. Such attack can be devised by consuming any of the system resource like computational power, memory, storage, network bandwidth etc

Few guidelines to be kept in mind for protecting information security are enumerated as follows:

i. Resource constrained devices like end-devices should use symmetric key algorithms for security purpose.
ii. All user data stored on end-device should be properly signed and encrypted using suitable cryptographic techniques.
iii. Protocols like Transport Level Security (TLS), Internet Protocol Security (IPSec) etc. should be used for dynamically negotiating session keys for authenticating various entities.
iv. End device firmware should use mitigation strategies for avoiding Denial of Service (DOS) and flood attacks etc.

Privacy aspect is defined as the privacy of the user while

i. Querying data or pulling data value from the system.
ii. Setting IoT system parameters i.e. pushing parameter values to the system.

This necessitates the formation of privacy policies for each usage scenario which can be enforced by either an individual IoT application or by the IoT infrastructure. IoT applications must be able to identify user requests and formulate policies for granting or denying system access. It may be noted that enforcing proper security checks is important for end-user acceptance of IoT enabled systems. If user privacy and security is compromised at any level in these systems, hacker attacks and system-malfunctions will outweigh their benefits.

V. CONCLUSION

IoT paradigm is modifying the way we use various gadgets and physical things in our daily life. For example, a light bulb is an old innovation which can be turned on-off using an electric switch. Some IoT application may attempt to control various characteristics of the light bulb using new fancy ways like hand gestures, voice based control, via smartphones, home gateways etc. Enthusiastic people and technology companies are creating many new services to take advantage of the IoT ecosystem. The vision of IoT is to make such novel services available to the user economically, securely, and with good user experience. In this paper we tried to present an overview of various software-hardware components, and security aspects related to the design and development of an IoT system. Each aspect of creating a fully functional IoT system will require deeper analysis of various trade-offs and constraints e.g. a decision on selection of a microcontroller for an IoT application will need detailed analysis of energy requirement of the application to be developed processing power, RAM requirement, battery source, software ecosystem etc. User privacy and information security aspects should be integrated at all levels while designing various elements of IoT system.

REFERENCES

[1] Thomas Zachariah, Noah Klugman, Bradford Campbell, Joshua Adkins, Neal Jackson, and Prabal Dutta. The Internet of Things Has a Gateway Problem.HotMobile '15 Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications

[2] S. M. Babu, A. J. Lakshmi and B. T. Rao, "A study on cloud based Internet of Things: CloudIoT," 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, 2015

[3] Wentao Shang and Yingdi Yu and Ralph Droms and Lixia Zhang. "Challenges in IoT Networking via TCP/IP Architecture", 2016

[4] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswamia. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", 2013

[5] Mohamed Abomhara, Geir M. Køien "Security and Privacy in the internet of Things: Current Status and Open Issues", 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)

[6] Dolev and A. C. Yao, "On the security of public key protocols," Information Theory, IEEE Transactions on, vol. 29, no. 2, pp. 198–208, 1983.

[7] Cervesato, "The dolev-yao intruder is the most powerful attacker," in 16th Annual Symposium on Logic in Computer ScienceLICS, vol. 1. Citeseer, 2001.

[8] P. Mahalle, S. Babar, N. R. Prasad, and R. Prasad, "Identity management framework towards internet of things (iot): Roadmap and key challenges," in Recent Trends in Network Security and Applications.Springer, 2010, pp. 430–439.

[9] O. Vermesan and P. Friess, "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems." River Publishers, 2013.

[10] Source:http://nfc-forum.org/nfc-and-the-internet-of-things/