Trends in Building Hardware and Software for Smart Things in Internet of Things

Xing Liu
Dept. of Computer Science and Information Technology
Kwantlen Polytechnic University
Surrey, Canada
Email: xing.liu@kpu.ca

Abstract—**Internet of Things (IoT) is considered to be another revolution in information technology. Previous revolutions built a global network of computers for people to communicate digitally. IoT aims to connect the things globally and these things are frequently physical objects. In order to realize the full potential of IoT, the things need to be smart. Hardware and software resources are required to instantiate the smartness. This paper examines the essential smartness attributes for IoT things and the hardware and software for implementing the attributes. Industrial trends in IoT hardware and software design are reviewed.**

Keywords - Internet of Things; IoT; smart things; smartness; implementation; hardware; software.

I. INTRODUCTION

Internet of Things (IoT) is one of the most important revolutions in technology in decades. IoT strives to connect the physical objects (things) globally. In an IoT system, the things communicate with each other and with human beings and act autonomously. In this sense, the things have smartness in addition to their common daily functionalities. For example, the common functionality of a door lock is "lock the door with a latch". The smartness part of an IoT-enabled door lock would be "remotely controllable over the Internet" and "able to recognize the owner and unlock the door" [1].

In order to equip the things with smartness, specific hardware and software are required. Most of the hardware can be built into a single integrated electronic circuit chip called System-on-Chip (SoC). These SoCs are named IoT processors. Special software has to run on IoT processors in order to empower the things with real smartness. In the case it is not feasible for a single IoT processor to carry all the needed hardware, a so-called IoT hardware platform (board) can be developed instead.

Although the attributes that define smartness vary greatly with the types of things, many attributes are commonly shared. Ideally, these attributes should be provided by IoT processors or IoT hardware platforms. However, no up-to-date systematic research has been conducted on what smartness really means for IoT things and what hardware and software are needed to realize such smartness. Results of smartness research can be useful for guiding the design and evaluation of smart products.

This paper gives a brief overview of Internet of Things in Section II. Then the paper identifies the general smartness attributes expected for IoT smart things in Section III. Hardware and software for implementing the smartness attributes are discussed in Section IV. Section V reviews the current industrial trends in hardware and software design for IoT processors and hardware platforms. Conclusions are given in Section VI.

II. INTERNET OF THINGS

A. *Physical Objects and IoT Things*

IoT is essentially a network of connected physical objects or things. The things exist with reasons. For example, a microwave is for heating up food. An air-conditioner is for cooling down a room. Traditionally, most things operate locally and are standalone. People need to be in close proximity in order to operate the objects.

B. *The Internet*

The Internet connects the things together. It provides the communication medium for the things. The things communicate and collaborate with each other via the Internet. Human beings can remotely access and control the things through the Internet. Wired or wireless links can be used to connect to the things. Wired links can be Ethernet, cable or telephone lines. Wireless links can be satellite, cellular, Wi-Fi, Bluetooth, or other new technologies being developed.

C. *The Cloud*

The Cloud is where IoT data is stored, visualized and analyzed. The Cloud also works like a "control center" which relays messages and commands between things and between things and humans. The Cloud is also used to manage things, create control policies, and call upon other Cloud services. The "elasticity" of the cloud makes it a highly suitable candidate for hosting IoT platforms [2].

D. *Smart Homes and Smart Cities*

Smart homes and smart cities [3] are two main application domains of IoT. In a smart home, connected things can be lamps, microwaves, fridges, ovens, heaters, ventilation systems, TVs and motion sensors. A smart city has many more connected things, such as cars, utility meters, parking meters, traffic cameras, power sources, waste collectors, street lights, just to name a few.

E. *Smart Things*

Connected things in an IoT system need to be smart [1][4]. For example, lights should turn on automatically

 ISBN: 978-1-61208-605-7

when people are present and should turn off when people move away. Doors should open automatically when the right person wants to get in. Alerts should be sent to home owners when there is a home invasion. Ventilation fans should change speed based on room occupancy to maintain air quality. Traffic cameras should work together to route street traffic and provide parking availability information. Traffic lights should inform car drivers or even directly interact with cars regarding upcoming color changes to avoid unnecessary braking. All these observations lead to the concept of smartness.

III. Smartness Attributes

Under what condition is an IoT thing smart? What are the attributes that make a thing smart? This section derives these attributes from common sense. Smart things should have some or all of these smartness attributes built-in based on the specific applications. These attributes will determine what hardware and software are required to make things smart and the feasibility of doing so in a practical design.

A. Aware of Environment

In an IoT system, the things should be aware of its surroundings. They should monitor what is going on around them. Parameters to monitor can be temperature, humidity, radiation, presence, proximity and darkness. Other awareness includes what other physical objects are doing and their impacts to the IoT system.

B. Able to Memorize

The things should have memory. They should memorize what has happened to themselves and to other related things. Information should be saved somewhere and can be recalled when needed. The things should decide what to memorize, how much to memorize and for how long. For example, should temperature be stored? How frequently should temperature be sampled? How many data points should be stored in memory?

C. Able to Communicate

The things should be able to communicate. They should be able to interact with each other and with human beings. They should be able to report their status when requested and accept commands whenever required and respond in proper “language” and “manners”. For example, a thing should be able to tell other things if it is switched on or off when asked, and start taking an action when it is asked to.

D. Able to Make Decisions

The things should be able to decide what they should do and how to do it. For example, should a door lock grant access to someone? If in doubt for any reason, the door lock should initiate an inquiry to the right party; and based on the responses received, it then decides the correct operations to take.

E. Able to Act

The things should be able to take actions. Examples may be start moving, turning off a switch, opening a door, playing music, turning on a heater, starting a fan, sending a message, or sounding an alarm.

F. Able to Perform Many Tasks at the Same Time

The things should be able to perform many tasks at the same time. These tasks may run concurrently. They should know which tasks are more important and have priority, and prioritize the tasks given to them. They should be able to switch between tasks based on operating conditions without losing information.

G. Able to Work Autonomously

The things should be able to work without supervision. They should know when to start, to interact, to adjust, to pause, or to stop. They should know how to protect themselves and protect others in the system.

H. Able to Learn from Experience

The things should be able to improve themselves based on what happened to them in the past, perhaps based on the data stored in local memory. They should learn from mistakes. They should be able to optimize and negotiate. An example of this is a smart robotic vacuum controller. Based on the impact it had against the wall in the past when it returned to the base charging station, it will work out the best moment to start slowing down. The robotic vacuum will find the new location of the charging station if the owner has moved it and use the new location next time. Another example is the smart heating controller which will work with the motion sensor to learn what times the owner usually goes to work. It will develop a flexible schedule to turn off the heating in advance (e.g. 30 minutes) to reduce energy consumption, instead of a pre-set fixed time.

I. Able to Predict into the Future

The things should be able to foresee what is going to happen to themselves and assist in predicting what will happen in the overall IoT system. They should also help human beings in predicting what will happen in the environment. An example of this case is predicting the breakdown of a smart device. If a smart device learns what happened inside a sister device before breaking down, for example, the sister device temperature pattern, then the smart device can predict its own breakdown time and notify maintenance personnel to replace it ahead of time.

J. Protection of Self and the Whole System

The things should know how to fight against physical or virtual attacks to themselves. They should not create security holes and propagate viruses. They should not do anything to damage the credibility of the overall IoT system.

K. Energy Consciousness

The things should be able to manage power consumption themselves. They should be power conscious. They should know when to do the work, when to sleep and do nothing. They should try to harvest energy from the surroundings whenever and wherever possible.

 ISBN: 978-1-61208-605-7

L. Self-Imaging and Twinning

A thing should establish and store a model of itself. This includes storing necessary parameters that can fully describe its own static and dynamic status and behavior. This may include mathematical models and statistical tools which assists in establishing the models. The model together with status information works as an image (twin) of the real thing and can be used to conduct simulations locally or can be uploaded into the Cloud for further simulation, visualization and examination. The twin image can be sent to whoever needs it. An example of this is a smart engine which stores its design model and running states. These states and the model will enable users to conduct simulations and potentially help create maintenance schedules and diagnose the causes of problems.

There can be more smartness attributes than listed above depending on the applications in the real world. The list can grow as time goes on as well.

IV. IMPLEMENTING THE SMARTNESS ATTRIBUTES

In order to equip the IoT things with the above smartness attributes, electronic components, computing hardware and software are required [5].

A. Sensors

Sensors power the things with awareness. Typically, sensors measure temperature, humidity, speed, motion, acceleration, height, position, distance, PH values, flow rate, brightness, color, radiation, sound, images, or a combination of these variables. Sensors can be chemical, mechanical, electrical, optical, or some other forms. They can be standalone components or can be integrated into electronic chips or into circuit boards. Sensors can be cameras when inputs are images or videos. Sensors are microphones when inputs are audio signals.

B. Local Memory

Memorization is achieved using digital storage such as Read Only Memory (ROM), Random Access Memory (RAM), flash memory, Secure Digital (SD) cards, or magnetic tapes. Memorization also includes proper organization of data for easy storage and retrieval, such as appropriate file systems and databases.

C. Communication Interfaces

Interfaces for communications among IoT things can be "traditional", such as analog/digital converter, serial/parallel communication ports, such as General-Purpose Input/Output (GPIO) [6], Universal Asynchronous Receiver-Transmitter (UART) [7], Inter-Integrated Circuit (I^2C) [8], Universal Serial Bus (USB) [9] and Serial Peripheral Interface (SPI) [10]. Interfaces can also be computer networks, such as Ethernet for wired communications. Wireless communication interfaces are more important for IoT things. Wireless interfaces can be satellite, cellular, Wi-Fi, Bluetooth, and ZigBee etc.

D. Local Control Logic

Local control logic enables thing autonomy. This can be algorithms for automatic control systems, adaptive control algorithms, optimal control algorithms, path planning algorithms, pattern recognition algorithms, and decision making algorithms.

E. Actuators

Actuators generate motion, movements, and initiate signal transmission. Actuators can be electrical motors, pneumatic devices, hydraulic devices and signal transmitters.

F. Analytics and Machine Learning Software

Such software is located locally inside the things, or remotely in the Cloud. These are typically mathematical tools for data analytics which can be descriptive, prescriptive or predictive. They can also be artificial intelligence and machine learning algorithms.

G. Multitasking or Multithreading

This is typically software that enables task concurrency locally inside the things, frequently implemented as part of the functionalities found inside an operating system.

H. Security Software

Security is extremely important for IoT things. There is need to have access control to the electronics and mechanical components of the things. Software in the things need to be immune to viruses. Communication information needs to be encrypted. Security algorithms can be embedded at different levels of the IoT system, such as in the electronics and in communication protocols.

I. Energy Management and Harvesting

Circuit modules are used for power management and energy harvesting. Power management modules optimize the use of power in power sources. Circuit modules are used to harvest energy from lights, heat sources, motion, and electromagnetic signals etc.

J. Digital Twinning and Device Shadows

Parameters, mathematical or statistical models, and status information of physical objects are used to build the twin image of a device. The image is also called a device shadow. The image can be stored in local memory. It can be used to diagnose problems, answer queries, perform simulations and provide predictions. Twin images can be loaded to the Cloud for further analysis or can be shared among devices.

V. INDUSTRIAL TRENDS IN IOT HARDWARE AND SOFTWARE IMPLEMENTATION

The actual hardware and software implementation of IoT things with the above smartness attributes depend on the technologies available and the cost allowed. The ideal solution is that all attributes are implemented on a single SoC module. This will minimize interference, power consumption, and thing sizes. The implementation can also be several electronic chips installed on a circuit board. In this

 ISBN: 978-1-61208-605-7

case, the implementation is called an IoT hardware platform. Some attributes may not be feasible to be integrated and therefore have to remain separate. Software also has to be carefully designed and implemented to realize the smartness attributes. A thing essentially makes an embedded system.

A. IoT Processors

An IoT processor is a SoC device designed for IoT applications. It is very much like a microcontroller and has most functionalities that a microcontroller has, plus many other functionalities.

Components of IoT processors generally include computing power, memory, on-chip sensors, input/output interfaces, wired and wireless networking interfaces, security measures, IoT operating systems (OSes), and power management modules. Companies that are making IoT processors include ARM (Cortex-M23 and Cortex– M33) [11], Qualcomm (Snapdragon processors and LTE modems) [12], Texas Instruments (CC3320) [13], and Cypress (PSoC 6) [14].

B. IoT Boards or "IoT Hardware Platforms"

IoT boards, also called IoT hardware platforms, usually have more functionalities than what IoT processors have. An example is the mangOH Red [15] jointly developed by Sierra Wireless and its partners. The mangOH Red is smaller than a credit card. It has built-in Wi-Fi b/g/n and Bluetooth 4.2 BLE (Bluetooth Low Energy) and built-in light, accelerometer, gyroscope, temperature and pressure sensors. However, it has only sockets for 2G to 4G and LTE-M (Long Term Evolution Category M1) and NB-IoT (NarrowBand IoT) wireless modules. It can connect to the AirVantage IoT cloud Platform. Legato is the Linux based open source development tool.

Raspberry Pi is another popular IoT hardware platform [16]. A Raspberry Pi has memory (RAM and a SD memory card), networking interface (Ethernet, Wi-Fi, and Bluetooth), Input/Output ports (USB, Camera interface, video interface, audio interface, GPIO pins which can be used as UART, I^2C, and SPI). By default, a Raspberry Pi runs a Linux-type OS called Raspbian [17]. Numerous smartness attributes can be readily implemented locally using the C, JavaScript, Java, or Python programming languages. Other OSes, such as the Windows 10 IoT Core [18], can also run on a Raspberry Pi.

C. IoT OSes

An OS is usually multithreading, has a task scheduler, and can frequently run real-time. For example, the ARM mbed OS [19] is an open-source embedded operating system designed to run in the things of an IoT system, specifically, on IoT processors. It is modular and configurable to reduce memory usage. The ARM mbed OS provides drivers for sensors, I/O devices and networking. It also provides cloud management services and security services. The ARM mbed OS is designed for ARM Cortex-M microcontrollers. Other IoT OSes include Contiki [20], Google Brillo (now called Android Things) [21], RIOT OS [22], and Windows 10 IoT [18]. Common features of IoT OSes include support for network communications, sensor interface, security, multi-tasking, and real-time.

D. Other Software

Signal processing software processes data from sensors, such as software for digital filters. System identification and modelling software build models for the things. Automatic control system algorithms and control software control the things. There are software that process audio and video signals and for pattern recognition. Web software now enables the Web interface for the things. Most such IoT software are based traditional software, with customization to support contemporary hardware, novel programming languages, and real-time applications. An example is OpenCV [23].

E. IoT Software Platforms

IoT software platforms are frequently called IoT platforms. They are the upper-level software in an IoT system. An IoT platform is a cloud-based software service that connects the things, collects data, manages the things, visualizes data, and performs data analytics. Additional smartness for the things is provided by IoT platforms. IoT platforms are also called middleware. Example commercial platforms are Amazon IoT [24], Microsoft Azure [25], and Xively [26].

F. Other Emerging Technologies

As the communication medium between IoT things and the Cloud, wireless communication networks are critically important. For this reason, tremendous efforts have been placed on developing new wireless communication networks that are suitable for IoT applications. Numerous new wireless technologies are on the horizon. These include NB-IoT [27], LTE-M [28], 6LoWPAN [29], Sigfox [30], BLE [31], and Bluetooth mesh networking (BLE-Mesh) [32]. Characteristics of these new technologies are low power consumption, small data packets and low data transmission overheads. However, IoT involves almost every aspect of the IoT ecosystem. Therefore, other technologies are also being developed. These include new sensor technologies, new software technologies, and new security technologies, as well as new applications.

VI. CONCLUSIONS

This paper has discussed the attributes that can make the IoT things smart, as well as hardware and software resources essential for implementing these attributes. Current industrial efforts in building powerful hardware and software for implementing smart things are also reviewed. Major technologies and products are identified.

REFERENCES

[1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of Things", IEEE Internet Computing, vol.14, no.1, pp. 44-51, 2010.

[2] Y. Karam et al., "Security Support for Intention Driven Elastic Cloud Computing", The Sixth UKSim/AMSS

European Symposium on Computer Modeling and Simulation (EMS), 14-16 Nov., 2012.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, Internet of Things for Smart Cities, IEEE Internet of Things Journal, vol.1, no.1, pp.22-32, 2014.

[4] G. Fortino, A. Rovella, W. Russo, and C. Savaglio, "On the classication of cyberphysical smart objects in the Internet of Things, Proceedings of the 5th International Workshop on Networks of Cooperating Objects for Smart Cities (UBICITEC 2014), pp.86-94, Berlin, Germany, Apr 14, 2014.

[5] M. Beigl and H. Gellersen, "Smart-Its: An Embedded Platform for Smart Objects", Smart Objects Conference (sOc), Grenoble, France. May 15-17, 2003.

[6] Adafruit, The GPIO Connector. [Online]. Available from: https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/the-gpio-connector. 2017.11.02.

[7] Cypress, Universal Asynchronous Receiver Transmitter (UART). [Online]. Available from: http://www.cypress.com/file/132486/download. 2017.11.02.

[8] I2C-bus.org, I2C – What's That? [Online]. Available from: https://www.i2c-bus.org/. 2017.11.02.

[9] usb.org, USB 3.2 Specification. [Online]. Available from: http://www.usb.org/developers/docs/. 2017.11.02.

[10] Texas Instruments, Serial Peripheral Interface (SPI). [Online]. Available from: http://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf/. 2017.11.02.

[11] ZigBee Allaince, ARM Accelerates Secure IoT from Chip to Cloud. [Online]. Available from: http://www.zigbee.org/arm-accelerates-secure-iot-from-chip-to-cloud/. 2017.09.29.

[12] Qualcomm, Internet of Things. Available from: https://www.qualcomm.com/solutions/internet-of-things. [Online]. Available from: http://www.zigbee.org/arm-accelerates-secure-iot-from-chip-to-cloud/. 2017.0.9.29.

[13] J. Wyatt and B. Fankem, SimpleLink Wi-Fi CC3220 and CC3120 Product Overview. [Online]. Available from: https://training.ti.com/simplelink-wi-fi-cc3220-and-cc3120-product-overview. 2017.09.29.

[14] Cypress, Welcome to the PSOC 6 Community. [Online]. Available from: https://community.cypress.com/community/psoc-6. 2017.11.02.

[15] Wavefront, Sierra Wireless launches mangOH Red open source platform for industrial IoT devices. [Online]. Available from: http://wavefront.ca/sierra-wireless-launches-mangoh-red-open-source-platform-industrial-iot-devices/. 2017.11.02.

[16] Raspberry Pi Foundation, Raspberrry Pi Hardware Guide, [Online]. Available from: https://www.raspberrypi.org/learning/hardware-guide/. 2017.11.02.

[17] Raspberry Pi Foundation, Raspbian, [Online]. Available from: https://www.raspberrypi.org/downloads/raspbian/. 2017.11.02.

[18] Microsoft, Windows 10 IoT Core - The operating system built for your Internet of Things. [Online]. Available from: https://developer.microsoft.com/en-us/windows/iot. 2017.11.02.

[19] ARM, mBed OS. [Online]. Available from: https://www.mbed.com/en/platform/mbed-os/. 2017.09.27.

[20] Contiki-os.org, Contiki: The Open Source OS for the Internet of Things. [Online]. Available from: http://www.contiki-os.org/. 2017.09.27.

[21] developer.android.com, Android Things, [Online]. Available from: https://developer.android.com/things/index.html. 2017.11.02.

[22] riot-os.org, RIOT: The friendly Operating System for the Internet of Things. [Online]. Available from: https://riot-os.org/. 2017.11.02.

[23] opencv.org, OpenCV. [Online]. Available from: https://opencv.org/. 2017.11.02.

[24] Amazon, AWS IoT. [Online]. Available from: https://aws.amazon.com/iot-platform/. 2017.11.02.

[25] Microsoft, Azure IoT Suite. [Online]. Available from: https://www.microsoft.com/en-us/internet-of-things/azure-iot-suite. 2017.11.02.

[26] LogMeIn, Profit from the Connected Product Conversation with the Xively IoT Platform. [Online]. Available from: https://www.xively.com/xively-iot-platform. 2017.11.02.

[27] 3GPP, Standardization of NB-IOT completed. [Online]. Available from: http://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete. 2017.11.02.

[28] GSMA, Long Term Evolution for Machines: LTE-M. [Online]. Available from: https://www.gsma.com/iot/mobile-iot-technology-lte-m/. 2017.11.02.

[29] IETF, 6lowpan Status Pages. [Online]. Available from: https://tools.ietf.org/wg/6lowpan/charters. 2017.11.02.

[30] Sigfox, the world's leading Internet of things (IoT) connectivity service. [Online]. Available from: https://www.sigfox.com/en. 2017.11.02.

[31] Cypress, Bluetooth® Low Energy (BLE) Profiles and Services. [Online]. Available from: http://www.cypress.com/documentation/software-and-drivers/bluetooth-low-energy-ble-profiles-and-services. 2017.11.02.

[32] Bluetooth SIG, Inc, Bluetooth LE: mesh. [Online]. Available from: https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/le-mesh. 2017.11.02.