Proceedings of 2020 IEEE 10th International Conference on Intelligent Systems

Software hardware combination for data gathering

Mika Saari
Tampere University
Pori, Finland
mika.saari@tuni.fi

Petri Rantanen
Tampere University
Pori, Finland
petri.rantanen@tuni.fi

Sami Hyrynsalmi
Tampere University & LUT
Pori, Finland & Lahti, Finland
sami.hyrynsalmi@lut.fi

Abstract—**The Internet of things (IoT) is a growing area in everyday life. New applications under the umbrella term IoT are being developed continually. This development has raised the need for framework definitions for different purposes. This research introduces a special software/hardware (SW/HW) framework for data gathering systems to be used in IoT related systems. The purpose of the research is to show the usability of a certain software hardware combination in prototype development.**

The SW/HW framework has been developed during several research projects. These projects have contained multiple iteration rounds. The main contribution of this research is the framework itself. The framework consists of a model of the system, sensor network, and cloud service. This research introduces the framework, its main components, and the usability of the framework in selected use cases.

Keywords - IoT; SW/HW framework; Raspberry Pi; Sensor network

I. INTRODUCTION

The Internet of things (IoT) is a growing area in everyday life. New applications under the umbrella term IoT are being developed continually. The IoT paradigm is the integration of several technologies and communications solutions [1]. This development has raised the need for framework definitions for different purposes. The draft of the IEEE standard [2] defines an architectural framework for the Internet of Things (IoT).

This research introduces a special software/hardware (SW/HW) framework for data gathering systems to be used in IoT related systems. The research question can be asked as follows:

RQ1: Is it possible to do rapid sensing prototype development with cost efficiency?

And a follow-up question may be asked:

RQ2: How can this prototype system be modeled?

Answering these research questions started with the first prototype system [3]. However, the questions themselves were not the main target at that time. The idea was to build a prototype to collect sensory data of physical features including temperature, humidity, brightness, or air pressure. The idea of modeling methods rose after the first steps of prototype building.

This study is part of the research related to the Internet of Things (IoT) carried out by the Software Engineering and Intelligent Systems (SEIntS) group at Tampere University, Pori. The SW/HW framework has been developed during several research projects. These projects have contained multiple iteration rounds. Many of these rounds have produced a research article, whose main target was to describe the working prototype. The first prototype system was introduced by Saari et al. [3] in 2015. The research introduced the initial idea of a framework and a working implementation from it. The latest research [4] introduces the advantages of rapid prototyping with off-the-shelf devices and open source software. Because of the long-term development process, the majority of the references are from our own research group. The work done is clarified in these references.

The main idea of prototype development has been to start with off-the-shelf devices and open source software. These key software and hardware components are modified in the desired direction and usually a working prototype system has been produced.

The main finding made during the research is the framework itself. This has the ability to act as a guiding principle when developing new prototypes for gathering data. This framework also aims to represent the development of software and hardware usage in data gathering systems; in particular the evolution in the usage of both software and hardware is considered in different parts of the system. The framework could be used as a model when planning new wireless sensor networks. The framework consists of a model of the system, sensor network, and cloud service. These three components together construct the framework of the IoT system.

Furthermore, the framework includes both software and hardware components. The components are introduced at an abstract level, which means a list of features and why a specific component needs the selected features. For example, the hardware components are mostly off-the-shelf devices, and therefore the design efforts put into the new hardware solutions have been kept to a minimum. Open source software components have been widely used, starting with the Linux operating system. The developed software consists of device and application interaction management.

The structure of this paper is as follows: In Section II, we review the related research about IoT, frameworks and sensors. In Section III, we introduce the developed SW/HW framework. This includes a discussion about the issue from various perspectives. Section IV continues the discussion about testing the SW/HW framework. The testing has been done in the authors' earlier research projects. Section V includes a discussion and suggestions for future research on the topic and finally, Section VI summarizes the study.

978-1-7281-5456-5/20/$31.00 ©2020 IEEE

II. Related Research

This study is based on the idea of rapid prototyping, which is used in our projects. The working prototype solution in the context of IoT needs is as follows: hardware to run the software; software to collect, save, and transmit the data; the right technologies for the use cases to make things easier for both developer and user. [4]

The importance of prototyping embedded hardware/software systems was introduced by [5]. That was because the system differences were increased, and the product relied mainly on variations in software and system features. In addition, integrating users into the specification process is important because more and more customers expect solutions and services tailored exactly to their particular needs. In a more recent study [6], the rapid prototyping process was developed with a working prototype using off-the-shelf components. This also showed that a lengthy product development life cycle is not required.

The software/hardware framework idea is not a new issue in the research field. For example, earlier studies [7-9] have addressed the framework subject from a real-time system perspective. In these studies, the design was at microcontroller level whereas our prototypes use off-the shelf single-board microcontrollers. Furthermore, our prototype systems use single board computers (SBC) with open source operating systems.

The IoT architecture consists of several components. The components can be divided into layers as follows: Sensing layer, Networking layer, Service layer, and Interface layer [10,11]. The SW/HW framework is focused on the Sensing and Service layers. The Networking layer exists but not is not the focus of this research. The Interface layer is described to the user in the SW/HW framework and is excluded from the study.

A wireless sensor network (WSN) can be used in various application areas. A WSN includes the sensor nodes, which consist of sensing, data processing, and communicating components. A sensor network is composed of a large number of sensor nodes, which send data to the data storage. Since sensor nodes have data processing ability, the uploaded data could be either raw or pre-processed [12].

III. The SW/HW frameworkIntroduction

The framework consists of several hardware and software components. Fig 1 shows the overall architecture of the framework. The data are not strictly specified. Simple basic data could be produced by sensors, such as temperature, humidity, luminosity, and so on. Complex data are photos or data processed from photos.

This chapter first introduces the hardware components at a higher level and later describes their features. The second part deals with the required software and its features.

A. The Hardware of the SW/HW Framework

The framework uses off-the-shelf hardware and devices. This limitation accelerates prototype development as at least partially tested devices can be used. The hardware can be categorized in three parts:

- Sensor hardware

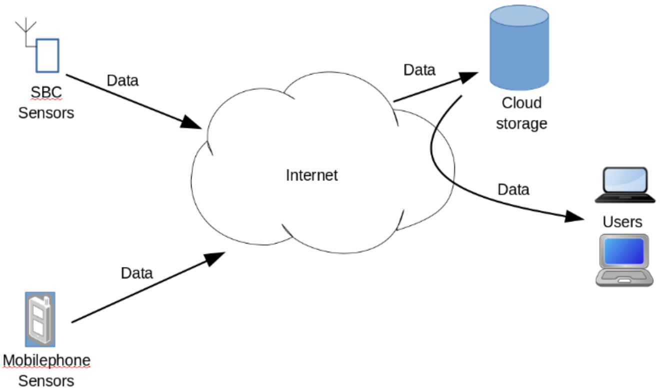

Figure 1. Architecture of SW/HW framework

- Data gathering devices
- Data storage devices

The sensor hardware collects the data from sensors. Data could be simple data such as temperature and humidity. On the other hand, the data could be more complex, such as photos. The hardware is selected according to the need for data–for example, the Arduino and other single-board microcontrollers can handle simple data. An SBC such as Raspberry Pi with a camera or basic Android phones could handle a lot of simple data. If it is necessary to collect photos, the Raspberry Pi and mobile phones could handle this. The data transfer rate should be selected according to the data requirements: Bluetooth, ZigBee, and LoRa for low rates and 3G / 4G or WiFi for faster data transfer speed.

Mobile phones are a special device group, which can be used to collect data. A typical smartphone has several sensors: gyroscope, accelerometer, ambient light sensor [13]. These sensors could be used to gather data from mobile devices [14].

Data gathering can be divided into two layers: sensor nodes and master node. The sensor node–master node combination is a special case where several sensor nodes send simple data to the master node. The master node collects the data for bigger chunks and sends or offers the data for collection. This WSN type solution could collect data from a large number of sensor nodes. The data transfer speed could be low and communication between sensor node and master node handled by license-free radio frequencies. Techniques such as Bluetooth, ZigBee, and LoRa could be used.

This SW/HW framework relies on communication to the public Internet. The collected data is transferred via the Internet to the data storage devices. These could be cloud servers with a database or dedicated Open source Linux servers for saving data. There are several database models for sensor data storing and each of these have a special use case where they are best. The SW/HW framework takes no stand about these techniques.

B. The Software of the SW/HW Framework

The hardware of the SW/HW framework also requires software. The software used is mostly open source. In this way the selected software is community tested and the source code is freely available. Open source software is also free to use.

Therefore, several software combinations can be used for testing purposes without extra costs.

The software components are divided into three parts by the hardware parts:

- Sensor software
- Data gathering and preprocessing software
- Data storage software

The list of software components starts with sensor software and ends with data storage software. The sensing and sensor software typically have low-level programming with C++, Python, or Perl scripts, which are dedicated to do few tasks, for example read sensor data, review the data, and store the data. Sensor devices, such as Arduino with sensors, are typically a micro-controller board that runs a dedicated program. The Raspberry Pi based small computers can do both the sensing and data gathering. The data gathering and preprocessing software are more complex and usually these devices are equipped with the full Linux operating system (OS). In the data gathering and preprocessing phases, the framework uses premade software and libraries. For example, Raspberry Pi could offer the gathered data to the Internet with a server application. Preprocessing in this scenario could be image recognition with image recognition software and library.

The database model for data storage is a relational model. The collected data could be stored locally in the sensor device, for example Raspberry Pi with Linux OS, MariaDB database, and a RESTful API combination. The RESTful API (Application Programming Interfaces) method allows remote control or management of a device over the network. Cloud storage for data is the alternative to local storage. Cloud storage could be, for example a Linux server or maintained cloud service such as Google Firebase. These both have more capabilities to store a bigger amount of data than the local database in Raspberry Pi.

Mobile phones equipped with the Android OS have been tested with this SW/HW framework. The Android OS has a software development kit (SDK), which enables the wide use of mobile phone capabilities. For example, the SDK enables phone camera usage [15]. The SDK also enables usage of the mobile phone's accelerometer sensor [14].

Data storage on the mobile phone is enabled by the OS. The SDK provides the capabilities to use files and databases for data storage. In terms of the SW/HW framework, the data should be usable by the user. The SDK also enables data transfer to cloud services.

IV. Testing the SW/HW Framework

The SW/HW framework has been developed and tested during several projects. These projects have contained multiple iteration rounds. Each iteration has produced a working prototype system. This section describes the prototype systems and their testing. In addition, a few studies about key technologies are included here.

A. Raspberry Pi related prototype systems

Raspberry Pi is a prototype device well known among research groups. The ieeexplore.ieee.org research document base gives over two thousand hits for the keyword "Raspberry Pi". Our first prototypes used similar devices – BeagleBone Black [16, 3] and Intel Galileo Gen 2 Development Board [17, 18]. Both of these devices have a Linux OS and several ways to connect sensors and other devices. These SBCs have similar features to the Raspberry Pi.

The studies [3, 18] introduced an example of how cost-efficient SBCs can be used to gather sensory data. Both of these prototype systems save the data locally and only provide the data to the user over the public Internet with web services.

B. Mobile phone related prototype systems

The mobile phone is an excellent WSN sensor node. It has a hardware working packet: power source, communication skills, and sensor devices. It also has an OS which allows the usage of hardware. In the presented prototype systems we used Android mobile phones. The studies [14, 15] introduce mobile phone related data gathering prototype systems. During the first study [14], the ShockApp software was developed. The software collected data from the accelerometer and GPS and the data were sent to a cloud service. This protype system points out the ability of mobile phone hardware to collect a large amount of data effectively: ShockApp measures one data point per second and almost half a million measurement points were collected during the test period.

C. Models and methodologies

The model for a sensor network were discussed in a study by [18]. This model was developed for an SBC related data gathering prototype system to clarify the architecture. The model included hardware and software components. The architecture of the software components was presented in a sequence diagram. The main idea of this was to show validation and the movement of the gathered data between software components.

The modified data gathering model for use in a mobile phone environment was introduced in [19]. The mobile phones used during the tests did not have hardware modifications. This prototype system consists of Android software where the hardware components of the phones were utilized. The software used the phone's GPS and accelerometer sensor. The model takes into account the mobile phone's hardware and software combination and connectivity to the cloud server, since the phone sends data to the cloud server.

Prototype testing methodologies have been discussed in [20]. The study shows that there has been a lack of formalized approaches, methods, and tools in research studies. Often only some methods were found: software testing, software performance testing, and validation of data tests. In addition, the study highlighted the minimal use of testing practices and methods with Raspberry Pi based protype systems.

V. Discussion

This study aims to resolve two research questions: Is it possible to do rapid sensing prototype development with cost efficiency (RQ1); and the follow-up question, How can this prototype system be modeled (RQ2)?

To answer the first questions, it can be said that our former research answered yes to this question. The several research papers are introduced where the rapid prototyping development is made with a cost efficiently. This was enabled by using off-the-shelf embedded devices such as mobile phones and Raspberry Pi SBCs. Furthermore, these devices have the such OS, which could be modified for the sensor usage.

This study answers the second question: How can this prototype system be modeled? At first two related studies were mentioned. These studies introduce the IoT components by layering those in five layers. These layers where mirrored to the SW/HW framework. Furthermore, the SW/HW framework hardware components and software components were discussed in detail.

The results of the RQs and the development of SW/HW Framework raised several new research topic ideas. These topics are shortly discussed here.

The topics of user and user experience are beyond the scope of the study. Our prototypes have been developed due to the need of some project partner. The prototypes have been tested with use case testing and once the customer has received a reasonable answer to a certain need, the development has stopped (except for one example [21] where was long piloting period in a real usage environment). The project outputs and prototypes are freely exploitable by the project partners.

The sensor prototypes produce a large amount of sensor data. Data processing and data mining are important issues, which this study leaves to ideas for future research, as it is such an extensive research area. The issues of data visualization have been handled in some studies [14]. In addition, sensor data will come more usable if merged with other publicly available data. This kind of data could be weather data or map data [19, 21].

Security issues are an important issue for IoT devices. Security vulnerabilities and attacks on IoT systems have been covered extensively by [22]. The SW/HW framework does not pay attention about security except for the communication channel. This concern was raised in [18] and the proposed, more secure, communication technology LoRa has been discussed by [23].

VI. Summary

The paper presented a SW/HW framework for data gathering systems to be used in IoT related systems. The SW/HW framework has been developed during several research projects. The main finding was the framework itself. The framework consists of a model of the system, sensor network, and cloud service. This research introduced the framework, its main components, and the usability of the framework in selected use cases. The research shows the usability of a certain hardware–software combination in prototype development.

Acknowledgement

This work is part of KIEMI -project and has been funded by the European Regional Development Fund and the Regional Council of Satakunta.

References

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Comput. Networks, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] IEEE Approved Draft Standard for an Architectural Framework for the Internet of Things (IoT)," in IEEE P2413/D0.4.6, March 2019 , pp.1-265, 21 May 2019

[3] M. Saari, P. Sillberg, P. Rantanen, J. Soini, and H. Fukai, “Data collector service - practical approach with embedded Linux”. In 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1037–1041). IEEE.

[4] M. Saari, P. Sillberg, J. Grönman, P. Rantanen, H. Jaakkola, and J. Henno, “Survey of Applications for Apartment Energy Consumption Monitoring,” in 23rd IEEE International Conference on Intelligent Engineering Systems 2019, IEEE, 2019.

[5] K. Buchenrieder, “Rapid prototyping of embedded hardware/software systems,” in Proceedings. Ninth International Workshop on Rapid System Prototyping (Cat. No.98TB100237), 2000, vol. 5, no. 3, pp. 2–3.

[6] C. P. Kruger, A. M. Abu-Mahfouz, and G. P. Hancke, “Rapid prototyping of a wireless sensor network gateway for the internet of things using off-the-shelf components,” in 2015 IEEE International Conference on Industrial Technology (ICIT), 2015, pp. 1926–1931.

[7] C. Kreiner, C. Steger, E. Teiniker, and R. Weiss, “A HW/SW codesign framework based on distributed DSP virtual machines,” Proc. - Euromicro Symp. Digit. Syst. Des. Archit. Methods Tools, DSD 2001, pp. 212–219, 2001.

[8] D. Saha, R. S. Mitra, and A. Basu, “Hardware software partitioning using genetic algorithm,” in the Proceedings of the Tenth International Conference on VLSI Design, 1997, no. January, pp. 155–160.

[9] M. B. Srivastava and R. W. Brodersen, “Rapid-prototyping of hardware and software in a unified framework,” in 1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers, pp. 152–155.

[10] L. Da Xu, W. He, and S. Li, “Internet of Things in Industries: A Survey,” IEEE Trans. Ind. Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[11] A. Vakaloudis and C. O’Leary, “A framework for rapid integration of IoT Systems with industrial environments,” in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 601–605.

[12] I. Akyildiz, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Commun. Mag., vol. 40, no. 8, pp. 102–114, Aug. 2002.

[13] K. I. Satoto, E. D. Widianto, and Sumardi, “Environmental Health Monitoring with Smartphone Application,” in 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 2018, no. 2, pp. 281–286.

[14] P. Sillberg, J. Gronman, P. Rantanen, M. Saari, and M. Kuusisto, “Challenges in the Interpretation of Crowdsourced Road Condition Data,” in 2018 International Conference on Intelligent Systems (IS), 2018, pp. 215–221.

[15] J. Grönman, P. Rantanen, M. Saari, P. Sillberg, and H. Jaakkola, “Lessons Learned from Developing Prototypes for Customer Complaint Validation,” in CEUR Workshop Proceedings, 2018.

[16] G. Coley, BeagleBone Black System Reference Manual. 2014.

[17] Intel Corporation, “Intel Galileo Gen 2 Development Board,” 2014. Retrieved December 2, 2019 from https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/galileo-g2-datasheet.pdf

[18] M. Saari, A. M. Baharudin, P. Sillberg, P. Rantanen, and J. Soini, “Embedded Linux controlled sensor network,” in 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2016, pp. 1185–1189.

[19] P. Sillberg, M. Saari, J. Grönman, P. Rantanen, and M. Kuusisto, Interpretation, Modeling, and Visualization of Crowdsourced Road Condition Data. In Intelligent Systems: Theory, Research and Innovation in Applications, Springer, 2020. In press.

[20] M. Saari, A. M. bin Baharudin, and S. Hyrynsalmi, “Survey of prototyping solutions utilizing Raspberry Pi,” in 2017 40th International

Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 991–994.

[21] J. Soini, M. Kuusisto, P. Rantanen, and M. Saari, “A Study on an Evolution of a Data Collection System for Knowledge Representation”, In the Proceedings of the 29th International Conference on Information Modelling and Knowledge Bases, EJC 2019.

[22] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, “Proposed embedded security framework for Internet of Things (IoT),” in 2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011, pp. 1–5.

[23] M. Saari, A. M. bin Baharudin, P. Sillberg, S. Hyrynsalmi, and W. Yan, “LoRa — A survey of recent research trends,” in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2018, no. May, pp. 0872–0877.