SoC as IoT sensor network hub

Alexandru Alexan
Department of Electric, Electronic
and Computer Engineering,
Technical University of Cluj-
Napoca, North University Center
Baia Mare, Romania
alexanalexandru@gmail.com

Anca Alexan
Department of Electric, Electronic
and Computer Engineering
Technical University of Cluj-
Napoca, North University Center
Baia Mare, Romania

Oniga Ștefan
Department of Electric, Electronic
and Computer Engineering
Technical University of Cluj-
Napoca, North University Center
Baia Mare, Romania

Alin Tisan
Royal Holloway, University of
London, United Kingdom

Abstract—**Nowadays SoC's miniaturization provide smaller yet more powerful devices that are perfect to be used as local hubs for small to medium sensor networks. Although sensors can now be easily connected directly to the cloud, a hub can simplify the process of bringing sensor to the IoT cloud. One of the most popular SoC board, Raspberry PI, is perfect for the hub role due to its small form factor, price, processing power and connectivity. Our proposed system consists in a SoC based low cost raspberry pi hub that connects two Bluetooth sensortag CC2650 modules to a mongoDB cloud database.**

Keywords— SoC, IoT, sensor network hub, sensor gateway, cloud, sensor network, sensortag CC2650, raspberry pi, python, mongoDB

I. INTRODUCTION

In today's digital era one of the most important thing is data, so it's no surprise that gathering and uploading data from every possible source is a common occurrence. This enables us to improve the quality of life and creates a digital environment that is always online and connected [1][2], the IoT environment.

The IoT environment captures data from everyday devices, data that is then collected, analyzed and stored [3]. Some areas like healthcare security and home automation are driving the expansion of IoT enabled devices in our lives but almost all-important areas like environment protection, industry and agriculture benefit from the immerging IoT coverage [4].

More and more everyday devices become IoT nodes, generating a device mesh that creates the digital environment in which we are living today.

There are multiple options for connecting the IoT node to the internet, two of them being via a hub or directly. Each option has advantages and disadvantages and choosing the right architecture is crucial for the data acquisition system's extensibility and performance.

Since the number of IoT devices is rapidly increasing, the need to store large amounts of data is also increasing. One of the most viable solution is to use the one storage platform that can almost provide virtually unlimited storage capacity and a great speed, a distributed cloud database [5]. This kind of database can be easily expanded and can accommodate from small to large sensor networks [6].

Our objective is to implement and test the usage of a sensor hub that has the role of a gateway, relaying and processing data from the sensors and sending it to the cloud. For keeping the cost down, we choose the hardware platform of the data hub to be system on a chip based.

II. RELATED WORK

A. Sensors

There are three main types of sensor categories for activity detection [7]:

- Wearable sensors
- Ambiance sensors
- Vision based sensors

For wearable sensors an important aspect is the size and battery life. Due to these two constrains it's important that the sensor should be as small as possible and provide the most functionality the end user. Usually these the sensor used are movement ones, accelerometers, gyroscopes and magnetometers [8]. Another important aspect that must be taken into consideration is the sensors connectivity that must be also low-power due to the battery size restrictions. In this case since the sensor module is wired by the end user, the connectivity is restricted to a wireless one.

Ambiance sensors are mostly based on pressure sensors, PIR or ultrasonic and provide a simple solution that is very easy and quick to install [9]. These kind of sensors do not have the size and battery life restrictions, and can be connected wired or wireless. Also the sensor nodes can have more powerful wireless transceivers than the wearable ones.

Vision based sensors provide a way of monitoring any kind of object in range as it uses video cameras and other combined sensors[12].

B. Sensor network hub and cloud

One of the most used network hub gateway is represented by a mobile device, like phone or table, which connects to the sensor modules and process or relays the data to a server [13].

Dedicated hubs can provide more flexibility and extensibility but they come with the cost of requiring a custom device as opposed to using an already omnipresent device, like

a smartphone. They support a wide range of operating system like Linux or windows based operating systems [14].

III. PROPOSED SYSTEM

The proposed system (Fig.1) consists of at least two sensors that form a sensor network that connect to a local hub that collects data and sends it for storage to the cloud.
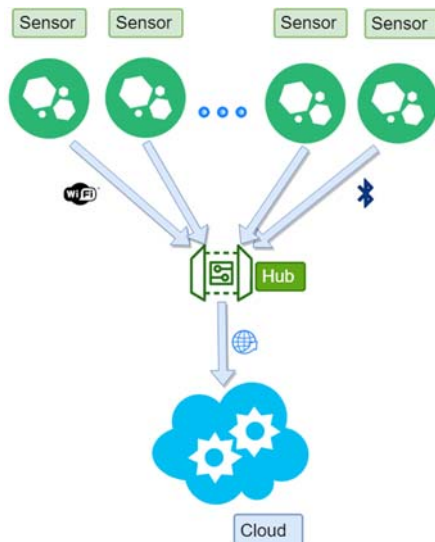

Fig. 1. Sensor network hub

Our implemented system consists of two Texas Instruments CC2650 sensor modules that connect via Bluetooth to a Raspberry PI hub that collects, pre-process and send the data to a mongoDB cloud database. All the hardware components of the system are chosen for their low power design and low cost. This creates a cloud enabled data acquisition system that can be cheaply and easily replicated and expanded if needed.

A. Sensors

The chosen sensors are Texas Instruments CC2650 SimpleLink SensorTag sensor modules showed in figure 2. These modules were chosen due to the low power design and multiple build-in sensors. The small form factor also proves to be a big advantage since they can be incorporated and used in multiple areas for data acquisition. They communicate over a Bluetooth low energy interface, perfect for small to medium sensor networks. Due to the low energy design the battery life ensures a decent run time between battery changes.

Fig. 2. CC2650 SimpleLink SensorTag[10]

The main sensors on this module are the movement sensors:

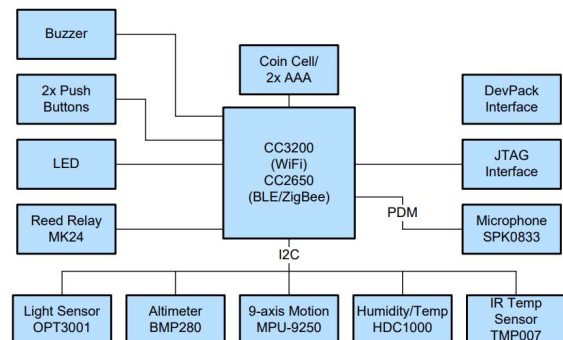- Accelerometer
- Gyroscope
- Magnetometer

Fig.3. CC2650 SimpleLink SensorTag sensors[11]

The rest of the environmental sensors like light, humidity, temperature and infrared light make this sensor module a suitable candidate for data acquisition and further data correlation.

Our implementation approach uses two identical CC2650 sensor modules for data acquisition that form a small sensor network. The sensors run on a custom firmware in order to increase performance, mainly to increase the number of readings per second. This firmware update was done using the Code Composer Studio IDE and the number of samples per second was increased from 20 to 50 samples per second.

The sensors are identified by the Bluetooth MAC address and based on this address the hub assigns a friendlier name identifier that is further used by the entire data acquisition system. This makes development and debugging a lot easier since each sensor has a meaningful name.

Currently only accelerometer and gyroscope data is sent to the cloud, via the hub, but all sensors data can be used if needed.

B. The sensor network hub

The chosen hardware platform for the hub, mainly for its small cost and footprint, is the raspberry pi SoC. This provides a cheap and versatile development platform that can run multiple operation systems:

- Raspbian
- Windows 10 IOT Core
- Ubuntu (mate, core, server)
- OSMC
- LibreELEC
- PiNet
- Risc OS

The heart of this SoC is the Broadcom BCM2837 chip with a quad-core ARM Cortex A53 architecture, with each core running at 1.2GHz. The powerful "Broadcom VideoCore IV" GPU enables this SoC to provide great display

capabilities. The RAM memory is also decent, featuring 1GB of LPDDR2 memory running at 900 MHz.

The module's connectivity also contributes to be a great choice for the sensor network hub since it features Bluetooth 4.1 classic and Bluetooth low energy sensor module. The built 10/100 Ethernet and 2.4GHz 802.11n wireless capabilities enables this board to connect to a multitude of sensors. The USB available ports allow for 3rd party USB sensor interfaces and the existing GPIO ports make possible the connection of additional modules for supporting virtual any sensor type, wired or wireless.

Our proposed hub used the Raspbian OS and the software application for the hub is written in Python, one of the best suited programming languages that can be used on this OS.
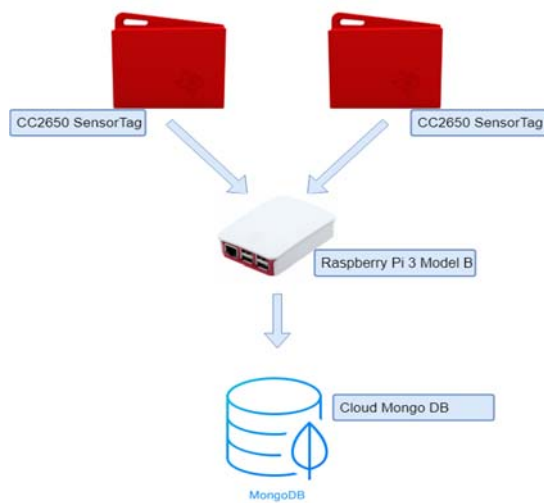

Fig. 4. The implemented sensor network

For connecting to the two SensorTag modules via a low energy Bluetooth connection, the bluepy library is used. This library represents a Python interface to Bluetooth LE allowing access to Bluetooth Low Energy devices from Python.

The bluepy library needs at least Python version 2.7 to run and it will work with any version higher than this. This library supports multiple sensor platforms out of the box including the SensorTag platform.

The datetime python module is used for date and time management, providing classed for dates and times manipulation.

For accessing the cloud database, the pymongo packages is used, a Python distribution that contains tools for working with MongoDB. This library is thread safe, which means that we can acquire data from the sensors using multiple threads, thus increasing performance.

The main program is written in python and uses the MAC addresses of the two SensorTag modules for connection, waiting for each SensorTag module to be available. Currently the MAC address of the SensorTag modules used is hardcoded in the application, as a white list, making sure that other SensorTag modules cannot connect if available. Also, this provides a secure Bluetooth LE network since all the allowed

devices are whitelisted. This can be easily changed to allow any SensorTag device to connect regardless of it's the MAC address.

For each connected SensorTag, the sensors that will be read are enabled and, in this case, the following movement sensors will be enabled: accelerometer and gyroscope. We chose only these sensors since this data can enable activity detection and provide the most dynamic sensor readings that are very good for debugging.

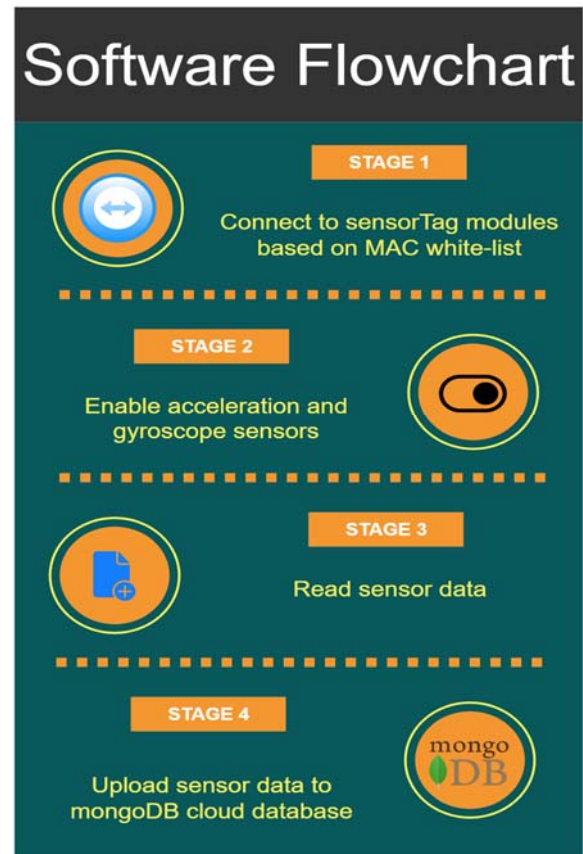After the sensor modules are enabled, they are available

Fig. 5. Software flowchart

for data read. The read process takes places continuously and without any added delay to obtain as much data as possible. This is great for debugging purposes and can be easily adjusted to restring the number of samples per second if needed.

Once the desired sensors are enabled, the program starts reading data continuously, preprocessing and uploading the data directly to the cloud. Each record receives a timestamp when it reaches the sensor network hub since the current used sensor modules do not have time capabilities. Even though the cloud database can also attach a timestamp for each record, the timestamp added by the hub is currently used since it's more accurate due to the relative high-speed of local sensor network compared to the intricate route to the cloud database servers.

C. Cloud database

The chosen cloud database is MongoDB since it's an open-source document-oriented database. It's one of the most used and known NoSQL database and it uses a JSON like format named BSON (Binary JSON) to work with data. This database system is capable of high performance and availability, yet it can be easily scaled so it's perfect for sensor network that can start small and grow beyond the initial estimation. It runs on all the major operating systems like Linux, Windows, macOS, Solaris and FreeBSD.

In our implementation we've created a collection for storing sensor data, with entries collected at the same time.

This collection is composed of multiple documents, each document having the following fields:

- Object id: Unique generated identifier

- Sensor name : the sensor name identifying in our case one of the two used sensor modules

- Time stamp : the time stamp added by the hub upon data read

- Acceleration data for the x-axis

- Acceleration data for the y-axis

- Acceleration data for the z-axis

- Gyroscope data for the x axis

- Gyroscope data for the y axis

- Gyroscope data for the z axis

The Fig.6 shows one example of a document record containing the unique id, sensor name, timestamp and the movement data.

```
_id: ObjectId("5ce1bfc93afa6aa5027bdd7b")
accY: -267
sensorName: "sensorTag2"
accZ: 4091
gyroZ: 2
timestamp: 2019-05-19T20:42:49.790+00:00
gyroX: -1044
gyroY: -516
accX: 235
```

Fig. 6. MongoDb one document, sensor data record

Another option for the storing data schema would be to have a collection for each sensor type, option that is best suited if many sensors are read for each sensor module.

IV. CONCLUSINONS

This paper presents the usage of a SoC as a sensor network hub with the role of relaying and preprocessing information from the sensors and sending it to the cloud.

Choosing a dedicated device, the SoC, for the sensor network hub proved to be a good option since all the device's resources can be used for data gathering and relaying thus decreasing cost and improving performance.

We chose the SoC hub, a raspberry pi board, mainly due to its availability, price and performance. For testing we used the smallest sensor network composed of two sensors modules, Texas Instruments CC2650 SimpleLink SensorTag

In the current implementation, movement data is retrieved and sent to the cloud by the local hub.

The used cloud database is a NoSQL database perfect for projects that need to be rescaled in the future. Storing data into the cloud allows other systems to access and analyze it, further allowing the system to expand.

Thus using a SoC as a sensor network hub proved to be a great solution that simplified the data acquisition process and provided an extensible and cheap sensor network architecture.

REFERENCES

[1] Z. Arkady, Ch. Perera, and D. Georgakopoulos. "Sensing as a service and big data.", Proc. of the International Conf. on Advances in Cloud Computing (ACC), Bangalore, India, July, 2012.

[2] F. Sheikh, and X. Li. "Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications.", Procedia Computer Science 34, 2014, pp. 103-110.

[3] Singh K.J., Kapoor D.S. Create Your Own Internet of Things: A survey of IoT platforms. IEEE Consum. Electron. Mag. 2017;pp. 6:57–68

[4] N.Q. Mehmood, R. Culmone, L. Mostarda, "Modeling temporal aspects of sensor data for MongoDB NoSQL database", J. Big Data, 2017, pp. 4:8

[5] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A Survey", Computer Networks, vol. 54, no. 15, pp. 2787-2805, Oct., 2010. N.Q. Mehmood, R. Culmone, L. Mostarda, "Modeling temporal aspects of sensor data for MongoDB NoSQL database", J. Big Data, 2017, pp. 4:8

[6] M. Roopaei, P. Rad and K. R. Choo, "Cloud of Things in Smart Agriculture: Intelligent Irrigation Monitoring by Thermal Imaging," in IEEE Cloud Computing, vol. 4, no. 1, pp. 10-15, Jan.-Feb. 2017.

[7] K. Lih-Jen and C. Chih-Sheng, "A Smart Phone-Based Pocket Fall Accident Detection,Positioning, and Rescue System," Biomedical and Health Informatics, IEEE Journal of, vol. 19, pp. 43-60, 2015

[8] AP. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A High ReliabilityWearable Device for Elderly Fall Detection," Sensors Journal, IEEE, vol. 15, pp. 4545-4552, 2015.

[9] L. Yun, K. C. Ho, and M. Popescu, "Efficient Source Separation Algorithms for Acoustic FallDetection Using a Microsoft Kinect," Biomedical Engineering, IEEE Transactions on, vol. 61, pp. 740-760, 2014

[10] Texas Instruments Incorporated, Multi-Standard CC2650 SensorTag Design Guide, Texas Instruments Incorporated, pp. 2, March 2015.

[11] Texas Instruments Incorporated, CC2650 SensorTag Quick Start Guide (Rev. A), Texas Instruments Incorporated, pp. 1, 2016.

[12] AE. E. Stone and M. Skubic, "Fall Detection in Homes of Older Adults Using the Microsoft Kinect," Biomedical and Health Informatics, IEEE Journal of, vol. 19, pp. 290-301, 2015.

[13] Vamos Daniel, Oniga, Stefan , Alexan Anca. (2018). Personal data acquisition IOT gateway. Carpathian Journal of Electronic and Computer Engineering. 11. 44-47. 10.2478/cjece-2018-0008.

[14] Yuejiao Cheng, Yuejiao & Chenglong Jiang, Chenglong & Shi, Jiong. (2016). A Fall Detection System based on SensorTag and Windows 10 IoT Core. 10.2991/mse-15.2016.4

[15] J. Suto, S. Oniga, C. Lung, I. Orha, Comparison of offline and real-time human activity recognition results using machine learning techniques, Neural Computing and Applications, March 2018. https://doi.org/10.1007/s00521-018-3437-x