Energy-efficient Internet of Things Monitoring with Low-Capacity Devices

Rodrigo J. Carbajales, Marco Zennaro, Ermanno Pietrosemoli
T/ICT4D, Wireless Communications Laboratory
International Centre for Theoretical Physics (ICTP)
Trieste, Italy
{rcarbaja, mzennaro, ermanno}@ictp.it

Felix Freitag
Department of Computer Architecture
Universitat Politècnica de Catalunya
Barcelona, Spain
felix@ac.upc.edu

***Abstract*—The Internet of Things (IoT) allows users to gather data from the physical environment. While sensors in public spaces are already widely used, users are reluctant to deploy sensors for shared data at their homes. The deployment of IoT nodes at the users premises presents privacy issues regarding who can access to their data once it is sent to the Cloud which the users cannot control. In this paper we present an energy-efficient and low cost solution for environmental monitoring at the users home. Our system is built completely with open source components and is easy to reproduce. We leverage the infrastructure and trust of a community network to store and control the access to the monitored data. We tested our solution during several months on different low-capacity single board computers (SBC) and it showed to be stable. Our results suggest that this solution could become a permanently running service in SBCs at the users homes.**

***Keywords*-Internet of Things, Community Networks;**

I. INTRODUCTION

The Internet of Things (IoT) allows users to gather data from the physical environment. While sensors in public spaces are already widely used, users are reluctant to deploy sensors for shared data at their homes. Similar as public sensors have helped to optimize city resource management by reducing costs, there is a huge potential for services in the users' homes to assist in local ICT management.

Several reasons explain this lack of users' willingness to participate: First, privacy issues play an important role. Many of today's IoT application foresee the sensors at the users home to send their data directly to the cloud service offered by the commercial provider (often, sensors and cloud service are offered by the same company). After uploading, users can access and visualize their data. The users, however, are concerned about their lack of control on their data once it is in the provider's platform. Second, the cost of commercial solutions may be another obstacle that has so far hindered the massive take-up of the existing commercial offers. Monthly fees for data storage and vendor lock-in further discourage user engagement.

Technical solutions and application areas for local IoT services have been identified within the area of Fog Computing [9], where a resource-constraint device close to the data obtaining sensors carries out initial processing. Concrete solutions for data transformations by devices at the users homes have been proposed for instance in [10]. In their work, the data from community facilities is gathered and processed locally, resulting in important traffic savings. A community context for cloud computing has been shown in [3]. In that work, the trust that exists within a community of users helped to bring together local computing resources form participants on which distributed services such as storage applications are run. In [2] it was pointed out that resources from many distributed nodes located on user premises could host local services more energy-efficiently than in data center solutions.

In this paper, we propose and analyze a solution for energy-efficient and low cost environmental monitoring at the users homes. Our system is built completely with open source components and is easy to reproduce. We leverage the infrastructure and trust of a community network to store and control the access to the monitored data. We tested our solution during several months on different low-capacity single board computers (SBC) and it showed to be stable. Deploying an open IoT infrastructure in a local context lets the users control and manage the stored information. It offers flexibility in the privacy settings by enabling user data management. From the obtained results, we see the proposed solution as a suitable candidate to become a permanently running service in SBCs at the users homes.

II. PROPOSED SYSTEM

A. Overall scenario

We consider wireless environment sensors that are installed by users in places of their interest, e.g. offices, houses, neighbourhood, with the purpose of assessing environmental parameters. These sensors are connected though Wifi with their LAN. They can thus transmit their data either to computing devices located in the same LAN or through a router to devices in other networks. On these computing devices, the data is stored and can be further processed.

While our solution can be applied in general, Figure 1 illustrates the concrete case of a community network where we have deployed our system. In community networks, users have built a network to interconnect nodes with each other through wireless links. While the interconnected nodes form the backbone network, at the users' homes local access

 DOI 10.1109/WF-IoT.2015.7389071

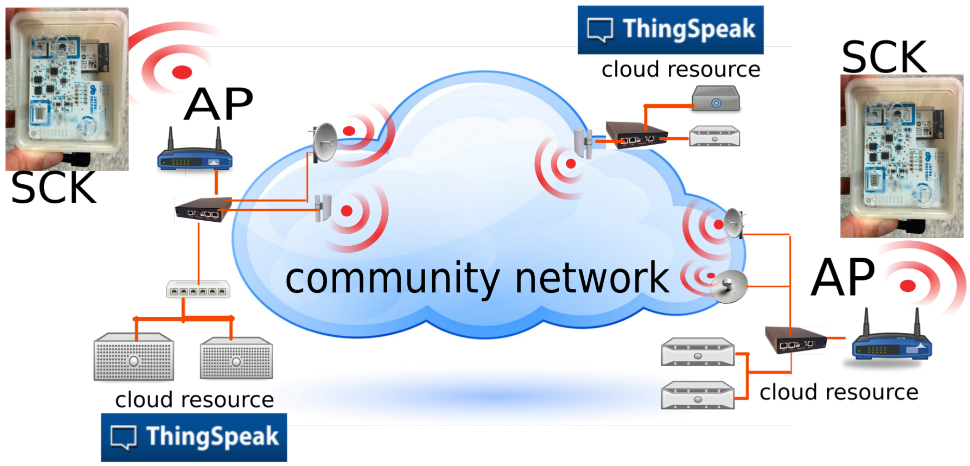

Figure 1. IoT monitoring with low-capacity devices in community network

points (APs) are installed to which the user's devices can be connected. In our scenario, the environmental sensor board (SCKs) connect through WiFi to such APs. In addition, the Single Board Computers (SBCs) at the user homes also connect to the AP. At some locations, more powerful resources like desktop PCs are connected. The SBCs themselves host the data gathering platform, for instance ThingSpeak. Optionally, additional storage capacity for data can be obtained from more powerful desktop PCs which are also provided by other participants within the community network cloud.

A distinguishing feature of our scenario as compared to the typical IPv4 DSL Internet access for end users is that each node in the community network has a range of routable addresses (typically /27) available for local devices. As a consequence, the devices connected to a node can be servers and can be directly reached from other nodes. For instance, the individual user can grant to other users or application processes access to certain data that is stored at the local device, and the retrieval of this data may allow external services to elaborate further for extracting useful information. This capability is different from the typical situation in DSL Internet connection for home users, which do not easily allow external access, since the devices are located behind a NAT and no static public IP address is assigned to them.

B. Smart Citizen Kit

The wireless sensor kit from SmartCitizen [6] was chosen to measure the data. With nine environmental sensors it offers a variety of possibilities for measuring air quality. The Smart Citizen Kit (SCK) is a project which provides low-cost hardware and open source software. The embedded solution of the SCK is an Arduino AtHeart [7] which is easy to program and communicates with the computer over a USB interface. On the software side, Arduino provides a number of libraries to make programming the micro-controller easier. The simplest of these are functions to control and read the I/O pins rather than having to fiddle with the bus bit masks normally used to interface with the micro-controller.

The kit is composed of two boards, the ambient board with sensors and the Arduino data-processing board. The ambient board as seen on figure 2 is equipped with the following sensors:

- Amount of gases (CO & NO_2)
- Temperature
- Sound level
- Humidity
- Light intensity

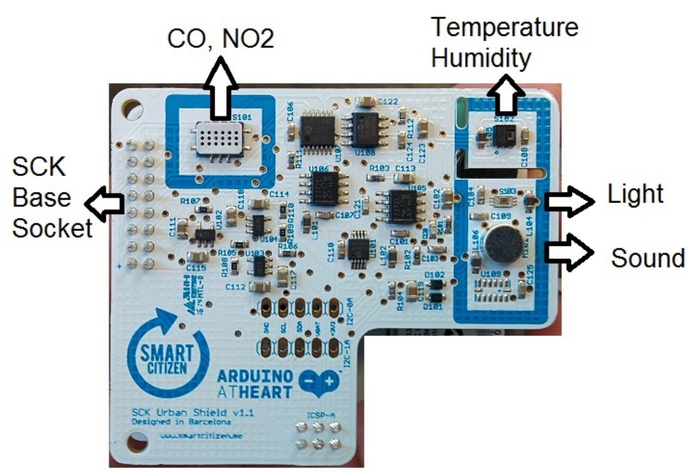

Figure 2. Smart Citizen Kit Environment Board

In addition, the Arduino data-processing board contains a voltage regulator that allows it to be fed by a photovoltaic panel, facilitating grid independent installation. It is equipped with a WiFi radio as seen in figure 3 that allows

Table I
SBCs USED.MODEL, μPROCESSOR AND RAM MEMORY.

SBCs	RaspberryPi	BeagleBone	Alix
Model	2-B	Black	3D2
μProc.	ARM Cortex A7 900MHz	ARM Cortex A8 1GHz	AMD LX800 500MHz
RAM	1GB	512MB	256MB
Price($)	35	45	103

to upload data from the sensors in real time to an on-line platform.

Once it is set up, the ambient board streams the measurement values over the WiFi module of the Arduino data-processing board. Power to the device can be provided by a battery, replenished by solar panel or other voltage source. The devices low power consumption allows for placing it on balconies and windowsills. The SCK device can be fitted with a 3D printed enclosure that makes it suitable to be placed on the open air.

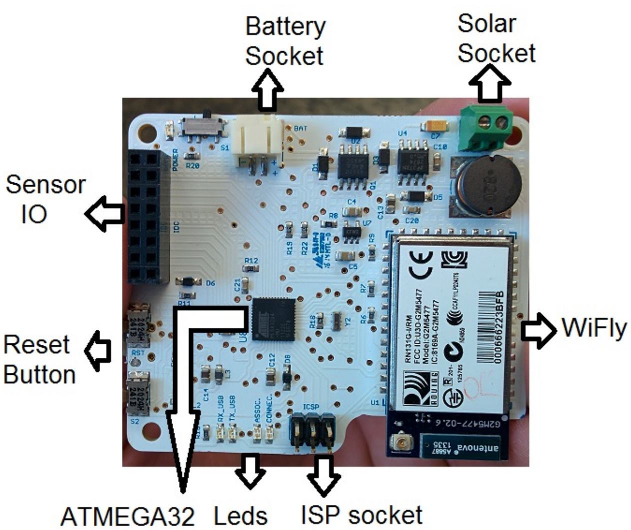

Figure 3. SCK Arduino data-processing Board top.

C. Single board computer

For this work, several SBCs were used. We tested our solution to run on the Raspberry Pi 2 model B, BeagleBone Black and Alix, shown in fig 4. The main characteristics of these SBCs are displayed in Table I.

D. ThingSpeak data platform

The platform we use to gather the monitored data is ThingSpeak (TS) [5]. It is a free open source IoT application with an Application Programming Interface (API) designed to store and retrieve data from sensors using HTTP over the Internet or via a Local Area Network. Sensor logging applications, location tracking applications, and a social network of things with status updates can be created.

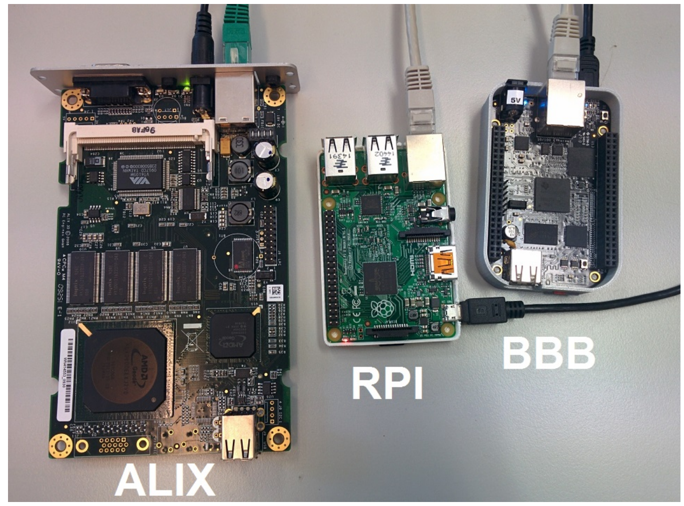

Figure 4. SBCs used

In addition to storing and retrieving numerical and alphanumerical data, the API allows for numerical data processing such as time-scaling, averaging, summing, and rounding. Each channel connected to a sensor supports data entries of up to 8 data fields, including latitude, longitude, elevation, and status. The channel feeds support JSON, XML, and CSV formats for integration in a variety of applications.

The TS on-line platform has some restrictions, like a minimum time of 15 seconds to update measurement data and it also requires a permanent Internet access, which is not always guaranteed in community networks often built with low cost and low reliability devices. Therefore, we installed our own version of the TS server that does not require an Internet connection, while also removing the 15 seconds limitation in upgrading time.

III. EXPERIMENTS

Several experiments were carried out to assess the performance of the proposed system. We found it user friendly for data visualisation and quite stable. We show results of energy consumption while running ThingSpeak with RPi2 which is the cheapest and more powerful of the SBCs tested.

A. Monitoring SCK data with ThingSpeak

ThingSpeak allows the user to display graphically the sensor data gathered in the Web interface using channels. The user can configure a public or private access to these channels. Figure 5 shows the graphical display of data received from the SCK, corresponding to six different sensors. ThingSpeak (TS) in this case was running in a Raspberry Pi. While the data shown corresponds to zooming

Table II
SCKs AND DATA SENDING PERIOD

num. of SCKs	Sending period (sec)
1	20 s
2	20 s
10	20 s
1	1 s
2	1 s
10	1 s

into a small time period, our experiment in fact was run during several weeks. In this time, the system showed to be stable and was permanently operational. When comparing the values measured by several SCKs which were close to each other, we noticed however that there were deviations among their measured values. We attribute this fact to a lack of calibration of the sensors.

B. Use of CPU in SBC running ThingSpeak

To measure the CPU usage of RPi by the TS server, 6 different situations were created, changing the sending period and the number of SCKs. Table II summarizes the experimental configurations.

Figure 6 shows the CPU consumption of the RPi with TS in the different situations. It can be seen that with a sending period of 20s, the CPU can cope and finish the processing of the data sent from different numbers of SCKs. When the sending period was reduced to 1s, the CPU is more taxed and the CPU usage increases with the number of SCKs sending data. It seems thus that for typical situations of 1 or 2 SCKs per home, the RPi works correctly, while its resources may be too limited to process data from many SCKs, as would be the situation when monitoring environmental data in many houses within a neighbourhood.

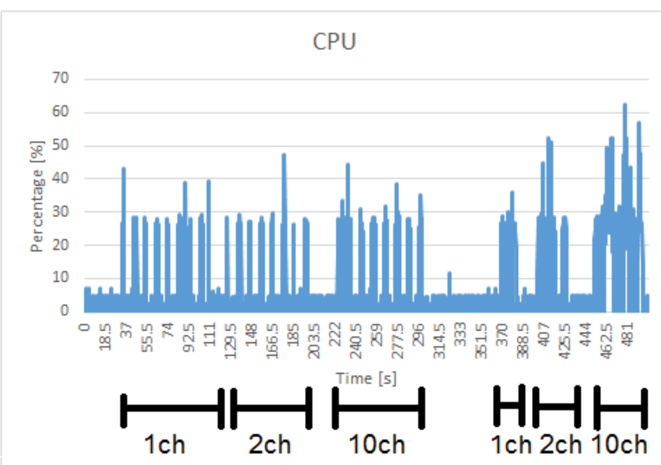

Figure 6. CPU percentage.

C. Power consumption running TS.

In this experiment we study the energy consumption of the RPi during the TS execution. Figure 7 shows the experimental setup where the RPi is connected to power supply. A multimeter is used for measuring the current consumption.

Figure 8 shows the current used by the RPi during boot and when the TS server is started. The change of the current consumption in different states of operation can be observed.

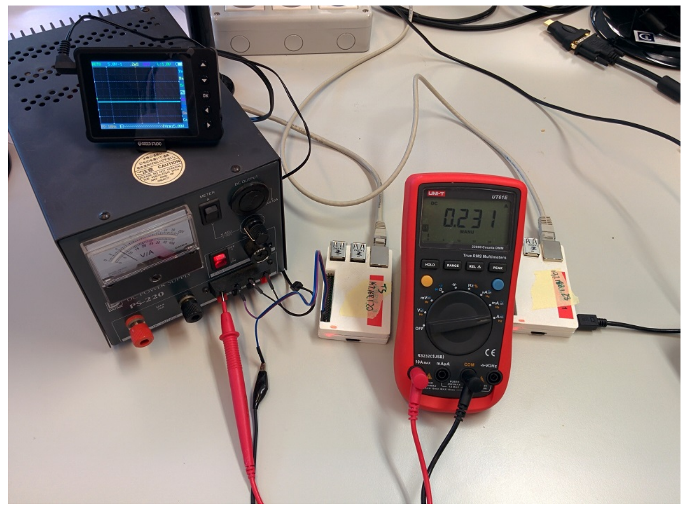

Figure 7. Measuring RPi power consumption and CPU percentage

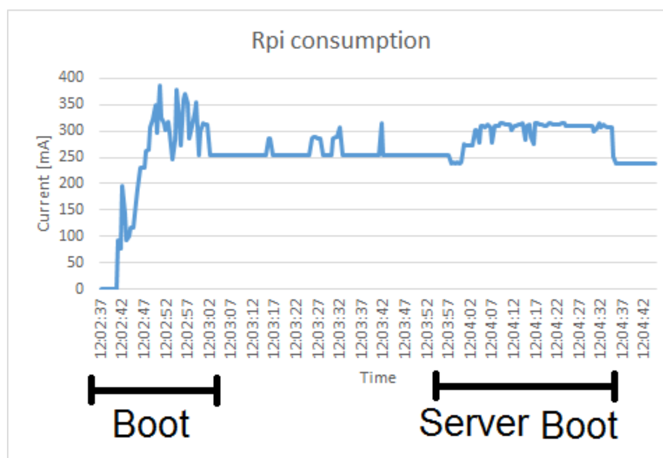

Figure 8. Boot consumption on RPi.

The next experiments compares the current consumption when one channel and ten channels of data were sent to the TS in the RPi every 20 seconds and every second. Figures 9, 10, 11, 12 show the RPi current consumption measured. It can be seen that the reception of data at each sending period leads to an increase of the current consumption, as it does when going from 1 to 10 channels.

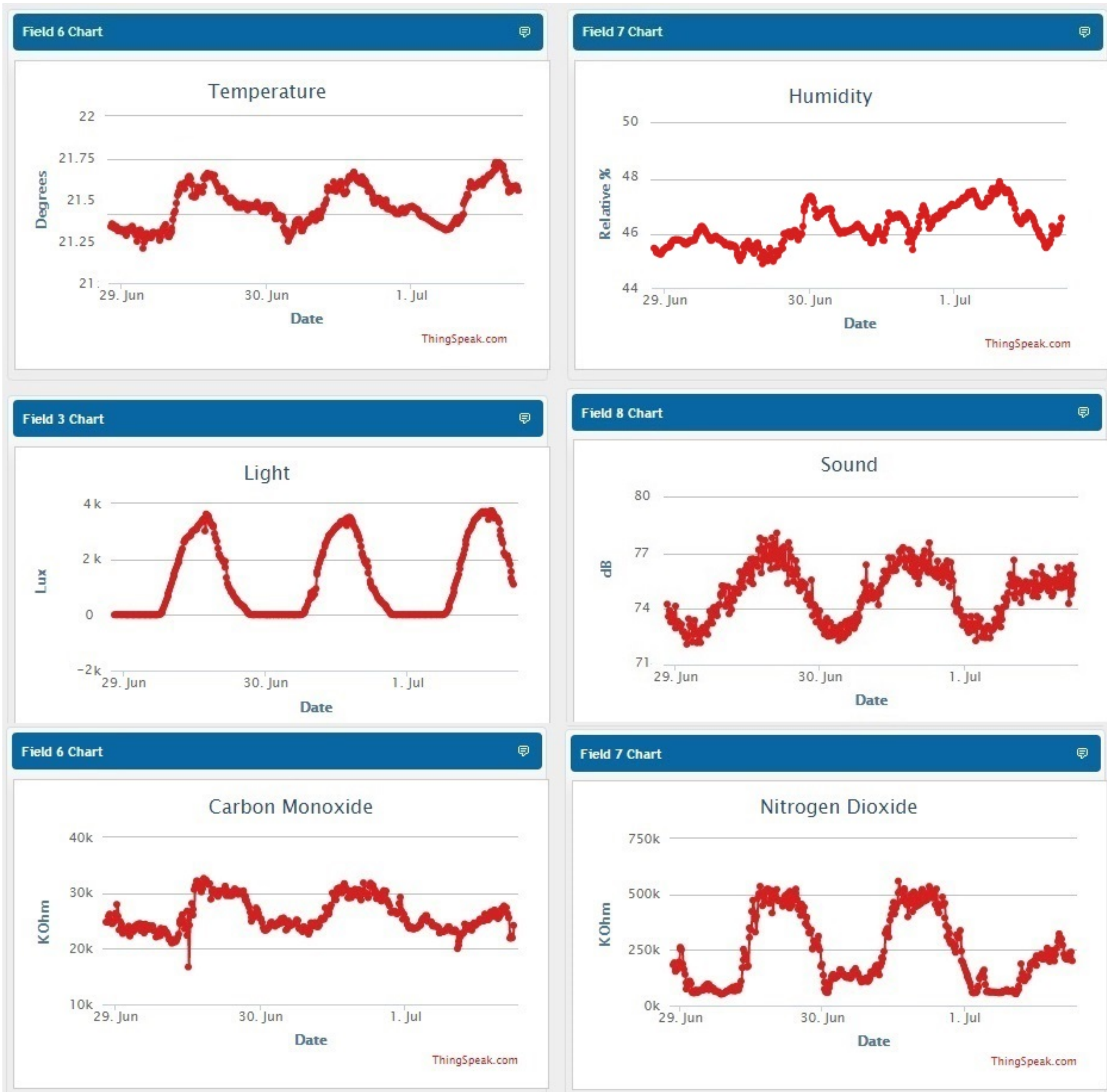

Figure 5. Example of ThingSpeak channel with sensors values

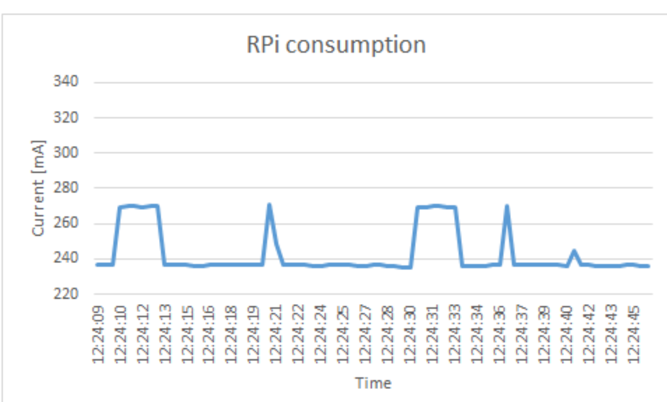

Figure 9. RPi current consumption in scenario 1ch20s.

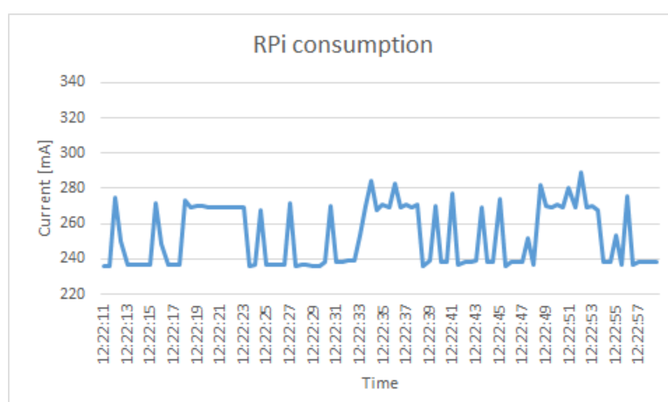

Figure 10. RPi current consumption in scenario 1ch1s.

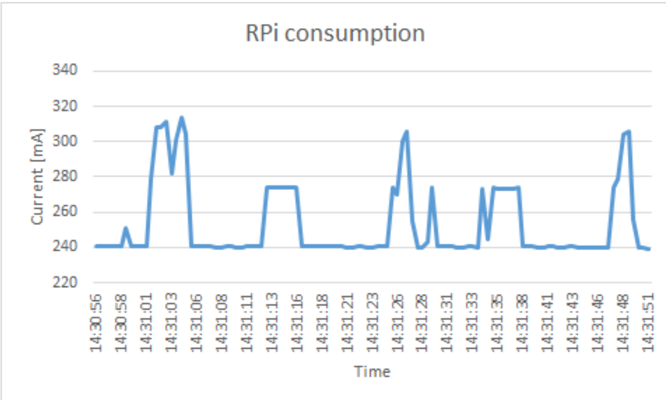

Figure 11. RPi current consumption in scenario 10ch20s.

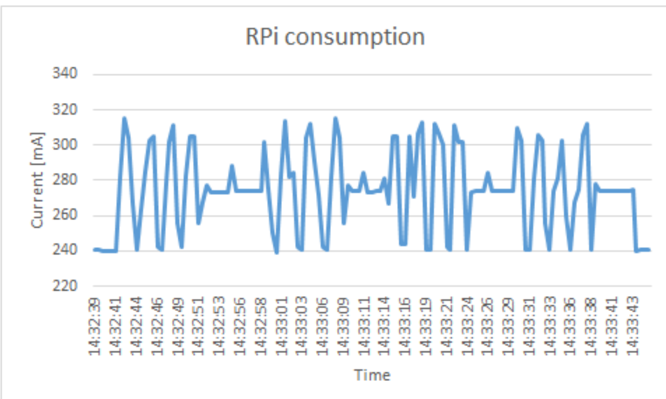

Figure 12. RPi current consumption in scenario 10ch1s.

IV. Conclusion

A system was presented facilitating citizens to conduct IoT environmental monitoring with low-capacity devices in home environments, extensible with the option for later sharing of the measured data among a community. The system was built with the Smart Citizen Kit (SCK) for measuring and sending the sensor data, and the Raspbberry Pi SBC for processing by means of the ThingSpeak platform.

Performance of the system was studied with respect to energy consumption, performance and stability. Results showed low energy consumption and cost, while being snappy and uses-friendly. Thus the electricity bill of the users is barely affected, allowing them to run this solution in a 24/7 mode. The system showed to be stable while it was run during several months in our experiments, and the measured values are easily available to the user through the ThinkSpeak Web interface. Regarding user friendliness, the configuration of the SCK proved to be well documented and ThingSpeak channels can be configured with a few steps by an average user. The proposed solution therefore seems suitable for running as a permanent service in SBCs that are deployed at users homes.

The proposed system could easily be extended to integrate and interact with more powerful cloud-based resources for larger volumes of data. While currently the SCK sensor data is of small size and sent with moderate frequency, additional IoT sensors at homes may produce larger volumes of data. Our next steps therefore will look at multiple data processing services running in the user home SBC, and how to combine local storage and processing in the SBC with external cloud services.

Acknowledgements

This work is supported by European Community Framework Programme 7 FIRE Initiative project CLOMMUNITY, A Community Networking Cloud in a Box, FP7-317879. Support is also provided by the Universitat Politècnica de Catalunya Barcelona TECH and the Spanish Government under contract TIN2013-47245-C2-1-R.

References

[1] Mineraud J., et al. *"Contemporary Internet of Things platforms."* arXiv preprint arXiv:1501.07438 (2015).

[2] Freitag F., et al. *"A Look at Energy Efficient System Opportunities with Community Network Clouds"*, in Workshop on Energy Efficient Systems. EES 2014 at 2nd International Conference on ICT for Sustainability (ICT4S), Stockholm, Sweden, August 27, 2014.

[3] Jimenez J., et al. *"Supporting cloud deployment in the Guifi.net community network"* Global Information Infrastructure Symposium, 2013 , vol., no., pp.1,3, 28-31 Oct. 2013

[4] Telecommunications Network Open, Free and Neutral, http://guifi.net (2015).

[5] The open data platform for the Internet of Things. https://thingspeak.com (2015)

[6] Open source technology for citizens' political participation in smarter cities. https://smartcitizen.me (2015)

[7] Arduino AtHeart program based on Arduino technology. http://www.arduino.cc/en/ArduinoAtHeart (2015)

[8] SGX Sensortech Limited. Gas sensor MICS-4514 datasheet. http://www.cdiweb.com/datasheets/e2v/0278-Datasheet-MiCS-4514.pdf (2015)

[9] F. Bonomi et al. *"Fog Computing: A Platform for Internet of Things and Analytics"*, in Studies in Computational Intelligence: Big Data and Internet of Things: A Roadmap for Smart Environments, pp. 169-186, vol. 546, 2014.

[10] Kenji Yoi, Hirozumi Yamaguchi, Akihito Hiromori et al. *"Multi-dimensional Sensor Data Aggregator for Adaptive Network Management in M2M Communications"*. IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada, May 2015.