Internet of Things and the Open Source Approach Towards Smart Cultural Objects

Serena Pastore*

INAF-Astronomical Observatory of Padova, vicolo Osservatorio 5, 35122, Padova, IT

*Corresponding author: serena.pastore@inaf.it

Abstract:
The topics of smart things and Internet of Things (IoT) is a recent technological trend that has many application fields, particularly in terms of Information and Communication technologies (ICT)-enabled benefits for society (e.g., smart cities, mobility, active and healthy ageing). The term IoT refers to a world-wide network of interconnected objects uniquely addressable based on standard communication protocols. The paper is a report on the work in progress in implementing IoT topics in order to create a smart cultural object as a physical object (e.g., a piece of a museum collection) transformed in a virtual object capable of sensing and processing capabilities and source and recipient of different web-based information. The project design and its steps together with the description of the first phases consisting in choosing technologies as part of the open source hardware and software environment to realize the smart cultural object are described.

Keywords:
Internet of Things; Web of Things; Open Source Hardware; Open Source Software; Smart Things; Ubiquitous Computing; Pervasive Computing

1. INTRODUCTION

The topics of “smart things” as applications of the Internet of Things (IoT) [1] is a recent technological trend. IoT is often associated with the Web of Things [2], and the two terms feature the Internet connection of different kind of things, not only of technological devices. The IoT defines a global network of devices (things like sensors and actuators) connected by standard Internet protocols, while the WoT includes a set of RESTful services [3], which are web services used with standard HTTP protocols. Web services permit the physical properties of things to be measured and manipulated. Another term machine to machine (M2M) [4] is part of the IoT and defines technologies that allow device communication through wired or wireless connections. Finally, the company Cisco coined the term Internet of EveryThing (IoE) [5] to broaden the concept. In IoE, “new or revisited technologies should bring people (humans), process (manages the way people, data and things works together), data (rich information) and things (inanimate objects and devices) to make networked connections more relevant and valuable than ever before” [5]. The idea is the application of pervasive or ubiquitous computing [6] that considers computing as appearing everywhere and anywhere and enhancing citizens’ life in personal and work activities. Thanks to the
Internet connection, the IoT has many applications in Information and Communication technologies (ICT)-enabled benefits for society such as smart cities, mobility, active and healthy ageing, and energy. Through Internet and web technologies, each “thing” contributes to provide information in term of web services on the Internet being uniquely identified and always online. However, other potential fields of IoT application could be cultural heritage and dissemination topics. In this context, a digital approach to cultural promotion usually passes through the creation of a website or an application that describes the cultural object in the form, for example of a virtual exhibition or a virtual application (e.g., game, virtual or augmented reality [7]). In our research, we are trying to take a different approach to these topics and to address IoT and WoT issues with practical applications related to cultural and virtual heritage. We focus on these issues with the aim of creating what we have called “smart cultural objects”. We refer to the concepts of the IoT/WoT where there a physical object becomes a virtual object capable of sensing and processing capabilities through an embedded device that fully integrates the object into the web. We conceptualize a cultural object into a broader sense. In the environment of a scientific research institute, we consider an ancient or modern instrument used for conducting scientific and technological research. The idea is to focus on each object giving it a unique identification and to collect, store in a web-based way, and process all different kinds of information related to it. In past studies, we have analyzed different web languages useful to store such objects [8] in a web-based manner. Moreover, by integrating sensing and processing capabilities, such an object becomes a source and a recipient of various information. Web technologies and tools are used to process the enriched and aggregated information related to the object. This gives users a cultural experience that is both real with a physical visit to the object location and virtual by means of specific mobile applications that could provide several types of information. The paper is a report on the work in progress in implementing a prototype of a smart cultural object. The different steps in the design and implementation, include the choice of technologies, the identification of the object (e.g., tagged with some kind of label technologies such as QR-codes [9]), and the integration of sensor and processing capabilities by means of the assembly of embedded devices. Then we consider the collection and storing in a web-based way of all data related to the object, and finally the use of such data to create applications targeted to each stakeholder (e.g., museum visitors or curators). The first phases of the project are described, involving technologies and methods to create the smart cultural object that is its identification in the web world and the integration of sensing and processing capabilities. Consideration about the cost of such solution, taking into account that a single object is replicated many times, and an open approach [10] both in hardware and software implementation to make the design available in other contexts are evaluated. Our aim is to obtain a low cost and open source prototype.

The remainder of the paper is organized as follows: sections 2 includes a description of the project, its objectives and the planning and design of the smart cultural object. Section 3 describes the analysis of the chosen boards and sensors that should act as a smart object. Section 4 includes an analysis and considerations about the hardware and software implementation by means of open source and the basic framework needed to collect, store and process the data about the object. The last section concludes and described future work in the subsequent phases of the project.

2. DESIGN OF THE SMART CULTURAL OBJECT

In the framework of Internet evolution toward the ubiquitous or pervasive web, we would like to experience tools and technologies that could allow smart objects to be created. Figure 1 depicts the Internet’s evolution toward the ubiquitous and pervasive web that is the basis of the IoT concept: a network of smart objects made by communication devices and sensors that become small and inexpensive
We focus on an object that is a tool used by scientific researchers. For testing purposes, we choose some instrumentations of the 17th and 18th centuries present in the La Specola Museum [11]. This museum hosts an exhibition of different tools and instruments used by astronomers in several rooms in the Padova site of our scientific research institute. The idea is to create the smart cultural object around an astronomical instrument. We have chosen, as Figure 2 shows, the “mural quadrant” of the Meridian room, which is an instrument of 1776 generally used to observe the transit of the sun and planets at the meridian. The idea is to enhance the virtual dimension of each different cultural object present in the museum giving it a digital dimension and a network connection.

There are two main approaches used for the digital description of a cultural object (Figure 2): a website-like approach (e.g., specific websites of virtual exhibitions) and a mobile approach with the development of mobile applications that could offer a personalized and always available digital view of objects.

In a virtual heritage context and in a mobile strategy, the usual approach is the development of a virtual exposition done through a website that, since the evolution of mobile devices, could respond [12] to the
different sizes of screens devices (i.e., responsive websites). The Omeka software [13] is an example of a framework used in the cultural world.

Developed by the Roy Rosenzweig center for history and new media of the George Mason University, Omeka is a free and open-source web-publishing platform design as a Content management system (CMS). It was adopted also in the Europeana project [14], Europe’s digital library whose portal, launched in 2008, has become the most visible representation of European cultural heritage online. The software allows in a virtual exhibition to be created in an easy manner that could promote cultural object. The other approach focuses on mobile devices. Considering the spread of mobile devices, applications are a tool used to achieve our objectives. Mobile application development should take into account the sensors embedded in the various devices (e.g., location, position, and camera) to provide an enhanced experience for users that visit the museum. Specific applications exploiting the augmented reality technologies could give a virtual experience by focusing on an object with the use of a mobile device camera. With the IoT approach, we propose to transform the object in a smart cultural object. This means to link the object to an assembled embedded networked device with sensing and processing capabilities in order to collect information in a web-based format about the object and its environment. Such data will be processed in order to give users tailored and personalized applications and/or services.

In the project plan, we analyze and test the tools and the technologies suitable for creating an embedded networked device with sensing and processing capabilities by connecting single board computers and sensors. We chose single board computers rather than assembly different shields for the practical issues of creating a first prototype. Our constraints were a compact device, possibly low cost, with the aim of replicating such an object. Moreover, we would like to adhere to the open source movement regarding hardware and software aspects for the wider adoption of solutions in other contexts. Ideally, the single object is replicated in each room for a single or group of cultural objects and thus is embedded into the environment. In such an approach, public engagement is both toward people who may be physically present in the places where the cultural assets are located from and who could experience the objects only through virtual web/mobile applications.

When talking about embedded devices, we are witnessing a proliferation of open source hardware solutions [15]. From the idea of low-cost devices and/or the application of open source in the hardware area, in recent years, single board microcontrollers (e.g., Arduino shields [16]) or single board computer designed completely (e.g., Cubieboard [17], Beaglebone [18] and Minnowboard [19]) or partially (e.g., Raspberry PI [20]) as open-source hardware boards [21] are available in the market. These solutions represent low cost Internet networked computer devices with minimal processing capabilities, small sizes, and low prices. The integration of such devices with sensors will produce the smart objects. In our project, we selected some of these solutions according to the criteria based on cost, availability and performance. The subsequent entails the selection of open source software i.e., network operating system, and when possible, HTTP services) targeted to devices with Internet-connectivity (e.g., Ethernet shield or Wi-Fi receiver), sensors equipped (e.g., temperature and humidity) and providing audio/video capabilities through a webcam. The choice is among different GNU/Linux distributions available for the specific single board computers and lightweight web frameworks. Then, we address how to collect and store information about the object in a web-based way.

In this context, we take advantage of our previous studies [22], but also of others such as those on digital object memory (DOMe) [23] that is used to save information about a smart object in a web-based way. A DOMe is defined as a digital storage space intended to keep all information about a physical object. It involves hardware and software components, which together provide an open platform for capturing,
associating, and interacting with the data on the connected objects, including storage, and the actions an object might be involved in. For this goal, each physical object has an identification number (ID) obtained by applying tagging technologies (e.g., QRcode [9]). Object memories are divided into passive digital object memories providing passive storage and active digital object memory based on embedded systems.

The Object Memory Modeling (OMM) Incubator Group [24] of the World Wide Web Consortium (W3C) introduced an XML-based format to describe such objects. The model supports the data storage of the smart labels attached to a physical artifact. It may contain data concerning information about a physical artifact or semantic information. In this way, object memories are stored on the web, facilitating building and communication with other object-related data. There are different projects, which are also funded by European Commission about such topics [25] and literature about such new ways of storing information about smart objects that we are considering in our project.

Finally the design and development of web/mobile applications using cross-platforms tools to enhance the cultural object chosen among our heritage integrating different information (e.g., data coming from the sensor and data about the object present in the web platform) will be the final step of our project.

We have finalized the first step of the project, which is the scope of this paper, and we are focusing on the implementation of the software environment and of the framework where information about the cultural objects is saved.

### 3. STEP ONE: HARDWARE IMPLEMENTATION

There are many platforms for creating digitally enabled objects, but some successfully prevail for their availability. There are two main categories of hardware platforms: those completely developed in the open hardware area and those that share some aspects of open source features, but still have some proprietary component. Each platform, however, belongs to the low-cost category.

#### 3.1 Open Source Hardware Solutions

The open source movement, born for software as the open source software (OSS) [26] movement, expanded for the access, the data and, in the recent years, hardware. Open source hardware or open hardware (OSH) includes the proliferation of many projects, initiatives (i.e., the Open Source Hardware association (OSHWA) [27]) and solutions (i.e., the Arduino platforms). OSHWA, which claims to be the voice of the community, developed the set of principles and definitions of an open source hardware solution [28]. The term refers to the availability of the design of the tangible artifacts (e.g., machines, devices or other technical things). It should be released to the public in such a way that anyone can study, modify, distribute, make, and sell the modifications based on the original design. When designs and processes can also be open and shared, different kind of users could test, alter, and iterate the products, allowing for hardware completion and innovation in the same way as we have done in software. As Figure 3 shows, a hardware product must comply with twelve criteria to be distributed as open source hardware.

According to these, all information about a product (i.e., design files, documentation, etc.) should be released with the product through a license that allows for modification and free distribution. There could be two kind of projects: open source IP (electronic components such as core, controllers, processors)
and open source designs (*i.e.*, more complex design such as specialized boards, prototyping boards, and machines). The protected items are the source code for electronics or associated software, the hardware design, the aesthetic value, documentations and brands. In additions, various IP tools (*i.e.*, protection for the semiconductors topographies, industrial design rights, patents, copyrights, and trademarks) are included. Open hardware projects take place in a range of various environments with different goals. For the aim of our project, we focus on computer boards or microcontrollers. In this area, Arduino projects are becoming famous for their microcontrollers products and for the Integrated Developer Environment (IDE)
software platform developed to help people making electronic prototypes. Arduino sells different boards and shields that are highly valued in the environment of so-called “makers” for their assembly capacities. Arduino includes the open-source electronic prototyping platforms that are flexible and easy-to-use both for their hardware and software aspects. A great community is born around Arduino projects, particularly with regard to home automation and 3D printers [29]. Different boards have been implemented, starting from the original ones (i.e., Arduino) that are compatible with the previous one, and many so-called shields to extend the boards’ functionalities (e.g., Ethernet or wireless shields).

Arduino shields address the projects dealing with creating interactive objects or environments, but it requires some efforts in assembling the hardware components and programming skills in integrating the connection with the sensors. However, Arduino is a programmable microcontroller, and the supported software does not have the typical operating system style and features. For practical reasons, at this stage, we decided to choose solutions that did not require large hardware assembly. Therefore, we focus on single board computer equipped with an Ethernet or Wi-Fi adapter rather than assembly Arduino shields, although we may evaluate them in the future improvements of the prototype with and integration with other boards (e.g., Raspberry). Nevertheless, for now, aiming at building working prototype, we focus on single board computers. Since there are constraints on purchases in a public research institute, we choose boards (System on Chips or SoC) present in the electronic market for the Italian public administrations. The solution selected on such a market are two open-source hardware boards (i.e., Cubieboard2 from Cubieboard and BeagleBone Black from beagleboard.org) and the Raspberry PI model B board. While the first boards are completely open (e.g., the hardware specifications like schematics, system reference manual, and so on) are available, the second one is not open source hardware since the SoC firmware is already closed.

3.2 The Chosen Boards

The selected boards (Figure 4) manifest similar features: they are low-cost development platforms, small-form factors, and have the same expansions, even if they have different sizes, and comparable low-cost prices. They use the same processor family (ARM) and uses the ARMv7 instruction set, which is the most common architecture among embedded systems. They support an internal flash, Ethernet, and Wi-Fi connectivity (the last through the USB connection), and an HDMI display output, even for different sizes.

The Cubieboard2 features a dual core processor that offers an enhanced processing capability, a dual core GPU, 1GB of RAM, and 3.4GB of internal flash expandable by slots with an SD card and with an external 2.5 SATA disk (up to 2T). The Beaglebone Black rides the 1GHZ edge with the Sitara ARM Cortex-A8 processor, a RAM of 512MB, and an on-board storage capacity of 2GB. Furthermore, with two 46 pin headers, it has a total of 92 possible connection points that can be configured for I2C (Inter Integrated Circuit Communication) or SPI (Serial-Pe ripheral interface) [30] buses, GPIO (general purpose input/output) pins, and other kinds of connections. Many external sensors interface with this board. Beaglebone Black manifests better graphics capabilities with a 3D graphics accelerator. Finally, these open hardware built-in boards are sold as they are; thus, for their adoption in a production environment, they need the necessary packages to protect electronic components from the environment (e.g., a case) and cables for connections to peripherals and other components. The actual costs of these solutions should take into account all of the components.

The Raspberry PI is a single-board computer based off a proprietary processor platform (e.g., the full
It has a relatively high performance, but the BeagleBone Black processor is nearly twice as fast as the Raspberry PI one. It was the first inexpensive single-board computer appearing in the market for the public, and thus many projects (i.e., media server aim and multimedia-based projects) adopted it. The model B version is equipped with Ethernet functionality, but compared to other solutions, it lacks an internal storage area where at least the operating system can be saved, and all of the software is stored on a removable SD card. Instead, its integrated videocore graphics processor gives enhanced functionalities in graphics processing as regards decoding video streams, thus enabling the connection to a webcam in order to provide video information about the object and its environment. It is capable of decoding 1080p video streams, rendering OpenGL and offers a full-sized HDMI connector and a composite video output for lower quality connection. The Beaglebone has built in graphics support, but with a lower processing power. Finally, Raspberry PI has an add-on compatibility with Arduino shields that could be very useful, considering the number and availability of such shields.

Raspberry relies on the open-source community for software. All of the boards support different open operating systems, most of them as specific versions of GNU/Linux distributions in customized versions of the Debian distribution or the Android operating system. Each operating system is customized according to its limited features (i.e., small size and limited internal memory) and hardware specificity. Considering the project requirements, we also focus on an HTTP access (e.g., including a web server service or a mechanism able to communicate with the machine with web and Internet protocols). We evaluated different operating systems/distributions to select the software frameworks suited to the objectives. We approached GNU/Linux distributions in the customized embedded version (e.g., Cubian or Raspian). Each distribution contains TCP and HTTP software and thus all the boards implement a service to send data to a server or acts as a server.

### 3.3 Open Source Hardware Solutions

A physical object embeds sensing capabilities through a network of sensors. Useful parameters to monitor a cultural object to protect are temperature and humidity that affect instrument preservation. Moreover, video monitoring through a webcam could improve the amount of information about the environment. Video stream captured and environmental data may be sent on a network storage system due to the limited processing and storing features of the boards. The equipment connection is chosen through the USB connectivity, and thus we selected products with this capability.

Sensirion products [31] are a good solution, providing performing sensors at an affordable cost. For our test, we choose a specific kit (the evaluation kit EK-H5) consisting of a USB stick with an adapter to the sensor cable. On one end of the cable, there is a connector where the sensor (e.g., SHTxx family) is attached. The included viewer software is available for the Windows platform, but the use of a standard USB connection allows for the interaction with such sensors by using other operating system platforms. The same considerations apply to the webcam selected with a USB connection. Regarding video monitoring capabilities, we also buy a standard USB webcam to take pictures and videos. Software tools that are able to provide video capabilities for the smart object are available (e.g., fswebcam [32]) in the GNU-based repository.
4. THE SOFTWARE ASPECTS

The open movement regarding software that anyone can freely use, study, share, and improve is conceptualized according to two major terms: free software and open software. There are different combinations and translations of these, including the FOSS (Free and Open Source Software), FLOSS (Free and Libre Open Source Software) [33], or simple OSS (Open Source Software).

Free software is the first kind of open source introduced by the Free Software Foundation (FSF) that imposes strict rules on what constitutes free software, while the Open Source Initiative (OSI) focuses on the advantages of realizing the source code under one of the compliant licenses. In this sense, they manifest different philosophies: free software stresses that the user has freedom to use the software, while open software emphasises the source code. The FSF license of choice, GNU GPL (general public license) is a copyleft license. According to it, modified versions of software should be distributed under the same license. The copyleft could be complicated and/or restrictive resulting in some more permissive licenses (e.g., the BSD license). Instead, OSI defines and maintains a list of approved open source licenses. Open source software is thus covered by licenses based on copyright (sometimes clauses about patents and trademarks), copyleft (LGPL, GPL) or not (BSD, MIT).

Business models are strongly influenced by licenses. In OSS, revenue consists of selling services (new developments, training, hosting) or software (dual licensing schemes or the sale of add-ons). The same concept applies to hardware licenses [34]: TAPR [35] and CERN [36] open hardware license (OHL) are examples. The first is not primarily a copyright license and does not prevent a third party from patenting inventions. The second defines conditions, in a similar way as GPL, for using or modifying the licensed material. Anyone should be able to see the design documentation of the hardware, and if distributed/modified, it must be under the same license conditions.

4.1 The Middleware: Choosing about GNU-Linux Based Distributions

Customized versions of operating systems and software depend on the board type. All boards support several GNU-Linux based distributions. Some of them, such as Raspian or Cubian that include the name of the supported board, are Debian-based customized version. Several Linux-based distributions focus on ARM processor support, even if the Linux kernel is supported throughout an organization (i.e., Linaro [37]) whose goal is to optimize Linux software on the ARM architecture. The available distributions adapted to such boards result from the major GNU Linux distributions such as Debian ARM, Ubuntu ARM, or Fedora ARM. Debian supports more hardware architectures including ARM. However, there are some issues in dealing with different versions of ARM and software packages, and there is no detailed documentation of installing and configuring the kernel software. Distributions based on Debian are Raspian and Cubian.

Ubuntu is also a distribution customized for every category of computer systems (e.g., desktop, server, and laptop) and thus supports the different versions of the processor and targets its instruction set. ARM and Canonical, the organizations behind Ubuntu, are working together to deliver ARM processor support for the Ubuntu platform across many different classes of devices. Supporting software includes the management of video, audio, networking, processing power for running applications, and a web management framework. It allows the installation of a complete LAMP stack that is widely used to provide web server, database, and web programming services. Such a stack transforms the board into a
platform, deploying, executing, and providing web services. Finally, Fedora supports specific releases for different ARM-based boards and an optimized version for Raspberry Pi called Pidora.

Another distribution, Arc Linux, develops a customized version for ARM computers, supporting through packages the kernel and the software for the several versions of the processors and the instruction sets; it is thus customized for different consumer devices and development platforms. Alongside this classical Linux distribution, there are also the Angstrom Linux distribution and the Android operating system. The Angstrom distribution is a Linux distribution for a variety of embedded devices built using a cross-compile environment for embedded Linux (the OpenEmbedded [38]) and based on other similar projects (i.e., OpenZarus) of operating systems. On the selected boards, the s.o is hosted on the onboard storage or the SD card and thus affects the loadable software. For example, the OHS boards have an onboard of limited storage size, even if they are expandable, while Raspberry PI boots from the external SD card.

Beaglebone has an included mini USB cable to connect it to a computer in order to supply power, but it is also possible to interact with the board from the computer, since with such a method a new mass storage appears on the computer when connected. However, it is possible to use the board standalone with a keyboard, mouse, and monitor by plugging it to a 5V 1A DC power supply. In this case, it is necessary to have a micro-HDMI cable. The firmware distributions for the board are based on Angstrom, even if the other operating system are available as a compressed sector-by-sector image of an SD card (.img.vz extension) on the website. When installing the drivers, it is possible to give network-over-USB access to the board and serial access as well as the HTTP access through an installed web server software. Cubieboard has the same features, while Raspberry PI board comes with Raspian, a GNU/Linux distribution based on the debian wheezy. The distribution also contains a LAMP platform that serves as the basis for an OSS for processing, logging, and visualizing energy, temperature, and other environmental data.

4.2 Components of a IoT Application: TCP and HTTP Access

Being on the Internet and available through the web platform, an IoT application supports TCP/Socket in order to give programs access to the network and HTTP client/server to develop applications that interact with a smart object. When an endpoint HTTP server is not convenient for performance or there are security issues, it is possible to use an external service or a library that could implement an HTTP server or a web server that is run-time configurable. Yaler [39], for example, which is available as open source software, provides a relay infrastructure for access to embedded systems. The embedded computer provides a RESTful web service [3] to control its sensors and publish itself through a relay server hosted online using a TCP socket and standard HTTP. Regarding web server software, there are a variety of solutions available for Linux-based platforms, but it is necessary to take account of the boards' hardware limitations; therefore, it is best to avoid running resource-heavy software. There are many light httpd servers [40], which could be implemented in such devices. Nginx and lighthttp are widely used lightweight web server engines. Nginx offers a blend of applications in a compact-sized software. It is a FOSS software, even if also distributed in a Plus commercial version. Lighttpd software provides a web server software optimized with advanced features that are set distributed as open source under a revised BSD license. It has event-driven architectures optimized for a large number of parallel connections (keep-alive). Moreover, considering the type of hardware, it could be useful to focus on service such as publishing/subscribing and thus utilize the MQQT (MQ telemetry transport for M2M communication...
protocol [41]. MQTT defines a lightweight messaging publisher/subscriber protocol. It is used for M2M communication for low-profile and low-band devices. The protocol is under standardization by OASIS. There are many open broker, client libraries and tools to publish/subscribe and other bridges toward different platforms.

A MQTT message broker that implements the MQ telemetry transport is Mosquitto which is an open source software. The transport is the lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for M2M messaging, such as with low-power sensors or mobile devices such as phones, embedded computers, or microcontrollers.

5. FIRST RESULTS AND FURTHER DEVELOPMENT

The first tests consisted of the implementation of the Raspberry PI equipped with the Raspian operating system connected with a webcam in order to take pictures around the room and thus monitor the environment. The webcam package is available in the Linux distribution to interact with the Raspberry PI board, to take pictures at a given time interval (such as every minute), and to capture a time-lapse. Due to the limited dimension of storage, such files could be saved on a network storage system (e.g., using SSH protocol for picture transfer or rsync to synchronize the folder of pictures taken in a folder between the boards and the web storage). The same software could be used on the other boards. However, Raspberry PI has better graphics performance. We are trying to interact with the temperature and humidity sensors, collect the data that in the same way could be saved on a network storage system or be directly sent to one in the cloud. Libraries software module for the management of Sensirion sensors are available for the Python programming language, while the software could be used for data gathering and graphing tools. It is necessary to create a Python script, for example, to read temperature data through cacti. Another software system that can be used for log management is Fluentd [42], which is open, allowing data collection and storage pipeline. With such software it is possible to save the data on a cloud service, even if in this case it is necessary to evaluate the cost. Cloud services of this kind are usually offered for free, starting with an initial plan. Similar software will be used in the other boards in order to evaluate the performance of the different boards.

6. CONCLUSIONS AND FURTHER DEVELOPMENT

In this paper, a description of the first phase of implementation for a hardware and software prototype applying the concept of the IoT and WoT allows for enhancing the promotion and preservation of cultural objects is proposed. The description of the project and the division in steps providing the development of a smart cultural object are presented. The case used is the application of such concept in a virtual heritage area, even if each smart object could be applied in a broader context. The idea is to associate a physical object with a virtual object enabling the processing and sensing capabilities of such an object. The idea is to transform the collections in a set of things in the sense of the IoT that is a set of Internet-connected devices able to provide specific web services and capabilities. This is realized by choosing computer boards in the open source hardware environment to reduce their cost and maximize their impact since they could be adapted in different environments. Then, the idea is to add specific sensing capabilities by means of sensor devices and exploit the processing capabilities to collect different data (both information and environmental) about an object and its surroundings that, structured in a web-based way, could be used to
Internet of Things and the Open Source Approach Towards Smart Cultural Objects

develop applications and services available to the general public. In the first step, we focus on three boards (Cubieboard, Beaglebone Black and Raspberry Pi) that we have equipped with open source software in order to evaluate their features. Having established these three boards that manifest flexibility and openness, we are now focusing on structuring collected data to be stored and processed in a web-based way.

References


[28] “Open source hardware association website (OSHWA),”


[34] C. Oleg, *The basic of free software licenses, cardanCOs blog*. [www.cardanco.com/blog/license-understanding.html](http://www.cardanco.com/blog/license-understanding.html).


About This Journal

CS is an open access journal published by Scientific Online Publishing. This journal focus on the following scopes (but not limited to):

- Analysis and Verification Technology
- Broadband Networking and Protocols
- Broadband Wireless Access
- Communication and Information Theory
- Communication Network Security
- Communication QoS and Performance Modeling
- Communication Software and Services
- Communications Systems
- Cooperative Communications and Networking
- Cross-layer Design and Optimization
- Cloud Computing
- Embedded Software
- Emerging Communication Technology and Standards
- Hardware Architecture for Communications and Networking
- Internet Services, Systems and Applications
- Mobile Applications
- Multimedia Processing and Communications
- Cognitive Radio Communications and Networking
- Optical Communications and Networking
- Optical Amplifiers
- Open Source Software Development
- P2P Communications and Networking
- Parallel and Distributed Computing
- Pervasive Computing and Grid Networking
- Protocol and Algorithms for Communications
- Project Management
- Satellite and Space Communications
- Signal Processing for Communications
- Software Engineering
- Wireless Ad-hoc and Sensor Networking
- Wireless Communications and Networking

Welcome to submit your original manuscripts to us. For more information, please visit our website:
SOP welcomes authors to contribute their research outcomes under the following rules:

- Although glad to publish all original and new research achievements, SOP can’t bear any misbehavior: plagiarism, forgery or manipulation of experimental data.
- As an international publisher, SOP highly values different cultures and adopts cautious attitude towards religion, politics, race, war and ethics.
- SOP helps to propagate scientific results but shares no responsibility of any legal risks or harmful effects caused by article along with the authors.
- SOP maintains the strictest peer review, but holds a neutral attitude for all the published articles.
- SOP is an open platform, waiting for senior experts serving on the editorial boards to advance the progress of research together.