The Role of Internet of Things for a Continuous Improvement in Education

Constantin-Eugen CORNEL, Ph.D
Department of Exact and Engineering Sciences, Hyperion University of Bucharest
eugen.cornel@hyperion.ro

Abstract: *This research is concentrated on the value that Internet of Things (IoT) can add to the education process, by using it in development of online virtual laboratories, which is a major requirement for any education system, in order to be qualitative and competitive. Practical experimentation is possible even with distance learning approach and today students may have access to a multitude of teaching resources, including IoT services used for various real world experiments. This is possible due to low costs and high performance of new electronic modules, on one hand, and the development of many high scalable web services, which permits data processing and communication over the Internet, on the other hand. The paper also presents an example of using IoT, by connecting an Arduino platform with the Xively web service, in order to read and display data received from a temperature sensor.*

Keywords: education, internet of things, communication, web service

JEL Codes: L63, L86

1. Introduction

Today our society becomes more and more sophisticated and demanding. On one hand, almost any business has to make some upgrades in their approaches and thinking for keeping themselves present in a vast and competitive market. On the other hand, the education institutions worldwide are forced to keep up with the industrial and technological innovations, in order that education can produce well-trained employees in any economic or industrial field. Thus, Internet of Things (IoT) is a high capable network that meets this goal for both business and education players. With a continually increasing number of Internet connected devices and robust web services available in IoT, our world today has a new great resource for changing the education process for a much better and continuous improvement for present and future generations' benefit.

Since the earlier stages of *Internet of Things (IoT)* network development many authors have attempted to define this system in many forms, including *Internet of Everything*, *Internet of Anything*, *Internet of People*, *Internet of Signs*, *Internet of Services, Internet of Data* or *Internet of Processes,* according to (Oriwoh & Conrad, 2015), and the study conclusions led to a definition that is currently satisfactory, i.e., IoT represents '*anything at all, depending on requirements'*.

Many market analysts worldwide are amazed indeed by the great impact of Internet of Things (IoT) in our day lives. According to statistics depicted from Internet resources, by the year of 2020, IoT network will integrate over 25 billion electronic objects. However, due to continuous development for Internet communications, the number of devices that will connect to the IoT will surpass 30 billion for the same period forecasted. Figure 1 reveals a report of Mario Morales, IDC statistics company (http://www.idc.com), showing an upward trend in people and devices connectivity over Internet.

Internet of Things (IoT) is a global network consisting of an array of objects containing electronics, software, sensors and connectivity features to allow a high level of satisfaction in providing various services worldwide. This is achieved through a continuous exchange of data between manufacturers, users and other computing devices. Each *thing* in the IoT is uniquely identifiable by its name or symbol in its embedded computing system and interoperates on the Internet infrastructure, feature that makes the IoT represent a global-scale distributed system.

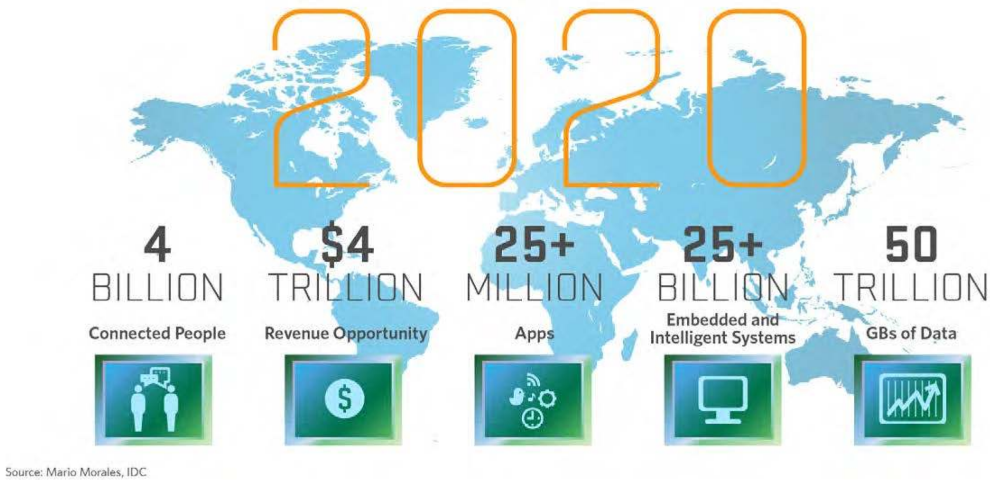

Fig. 1 - Statistics on internet users interconnection and the number of devices integrated into the IoT by 2020

(Source: Mario Morales, IDC, http://www.idc.com)

The objects of the IoT network may include a variety of devices such as:

- implants to monitor heart rate, blood pressure monitoring devices;
- monitoring biochip bracelets for farm animals or pets;
- autonomous vehicles and robots;
- devices for assisting emergency service personnel;
- automatic irrigation systems, home appliances (smart refrigerator, smart TV, air conditioning, intelligent thermostat etc.)

These devices collect useful data using multiple sensors and data acquisition technology, then transport it to other computing devices or systems for further processing and interpretation.

2. Data processing methodology in IoT for education

The IoT can be used to develop online virtual laboratories for various faculty specializations, like electronics or automatics. Examples can include electronic circuity testing and automatization process monitoring and control. Thus, education institutions can implement such virtual labs for various study fields and enable distance learning facilities even for technical specializations. Here comes the important role of IoT, which is the fact that it provides the suitable framework for developing online virtual laboratory platforms.

The most used approach in getting devices connected in IoT and obtaining useful information is to have an electronic module as device, which needs to have built-in Internet connectivity feature, and a web service capable to receive and send data from and to the connected devices.

Hardware requirements

Hardware technologies used to implement a virtual laboratory for automation control processes are based on AVR[3] or ARM[4] microcontroller systems, and some on PLC[5] controllers. As a common solution in universities' virtual laboratories, AVR and ARM microcontrollers systems are sufficient and more affordable than platforms using PLCs. Thus, companies like Texas Instruments or STMicroelectronics have made open development platforms with ARM microprocessors that can be acquired by anyone. At the moment (June 2015) in our country one can buy STM32 NUCLEO-F401RE platform with only 20 Euro, given that it is a complete development platform with high performance.

On the other hand, Arduino[6] platforms, having less performance, keep an increasing trend of purchasing costs. For example, an original Arduino UNO R3 platform sells with 25 Euro, although its specifications are low compared to STM32 NUCLEO-F401RE development platform. However, Arduino modules are widely spread due to its ease of programming, implementation and operation and many Chinese manufacturers produce Arduino clones at lower prices (less than half), as Arduino offers open source licenses for their architectures. ARM microcontrollers programming is more difficult, but the results are superior to working with AVR microcontrollers (see table 1 for a comparison).

Table 1. STM32 NUCLEO-F401RE vs. ARDUINO UNO R3 specifications

• STM32 microcontroller: Core Processor ARM® Cortex®-M4 • STM32F401RET6 in LQFP64 package • ARM®32-bit Cortex®-M4 CPU • 84 MHz max CPU frequency • VDD from 1.7 V to 3.6 V • 512 KB Flash • 96 KB SRAM • GPIO (50) with external interrupt capability • 12-bit ADC with 16 channels • RTC • Timers (8) • I2C (3) • USART (3) • SPI (3)	• Microcontroller: ATmega328 • Operating Voltage: 5V • Input Voltage (recommended): 7-12V • Input Voltage (limits): 6-20V • Digital I/O Pins: 14 (of which 6 provide PWM output) • Analog Input Pins: 6 • DC Current per I/O Pin: 40 mA • DC Current for 3.3V Pin: 50 mA • Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader • SRAM: 2 KB (ATmega328) • EEPROM: 1 KB (ATmega328)

[3] 8-bit RISC AVR microcontrollers developed by Atmel (http://www.atmel.com/)

[4] 32-bit/64-bit microcontrollers, licensed by the British company ARM Holdings (http://www.arm.com/)

[5] Programmable Logic Controller - used for industrial electromechanical automation processes

[6] Arduino develops widely used digital electronic devices with open source license (http://www.arduino.cc/)

• USB OTG FS • SDIO	• Clock Speed: 16 MHz • Length 68.6 mm • Width 53.4 mm • Weight 25 g

All the information in Table 1 is depicted from STMicroelectronics, respectively, Arduino official websites.

These platforms offers a high connectivity with a variety of measurement and control devices, such as data acquisition modules, sensors, transducers or actuators. Any automation control system contains such electromechanical elements, by which a process parameters (temperature, flow, pressure, level, etc.) can be monitored and controlled.

Software requirements

Interface software applications for accessing physical lab platforms can be developed in various ways, choosing from proprietary visual programming software, like NI LabVIEW or MATLAB to free or open source available tools. The National Instruments (NI) company offers the LabVIEW design & development software, which represents a strong tool for developing complex graphical user interfaces for various applications with all the components an online virtual lab requires, including the Internet connection feature for distance monitoring and control.

Together with other tools, many authors developed a series of virtual labs, some examples include:

- Design of online v-labs for automation control engineering education with NI LabVIEW (Stefanovic et al., 2011);
- Robotics virtual labs with remote access using EJS[7], MATLAB and LabVIEW (Chaos et al., 2013);
- Design and development of distance (and mobile) access to remote reconfigurable electrical engineering laboratory platforms for e-Learning within education (Sandu et al., 2008).

In addition to proprietary applications mentioned above there are also free or open source software communication tools, mature enough to satisfy any developer of web applications for remote control laboratory experiments. For instance, tools developed with WebSocket protocol and related API's (as Socket.IO API within Node.JS web application framework) can offer full-duplex communication over a single TCP connection, required by client-server applications like online virtual labs. Such tools are developed using client-side web programming languages, such as HTML, JavaScript, CSS, and server-side languages, like PHP, Perl, Ruby or Python.

3. Case study

For the demonstration of IoT benefits, let's present an example of connecting an electronic platform to a web service and see how the data received from a temperature sensor is nicely displayed in the browser, almost in real-time. The new monitoring system achieved is based on Arduino developer platform and Xively web service and can be easily integrated in a virtual lab experiment.

The hardware platform

The chosen components of the hardware needed for the lab include:

[7] Easy Java/JavaScript Simulations - http://www.um.es/fem/EjsWiki/pmwiki.php .

- Arduino UNO R3 development platform (http://www.arduino.cc);
- ENC28J60 Ethernet shield (controller produced by Microchip, http://www.microchip.com/wwwproducts/Devices.aspx?product=ENC28J60);
- LM35 temperature sensor;
- Li-Po battery with 7.4V and 3600 mAh for power supply. It can also use a 9V battery.

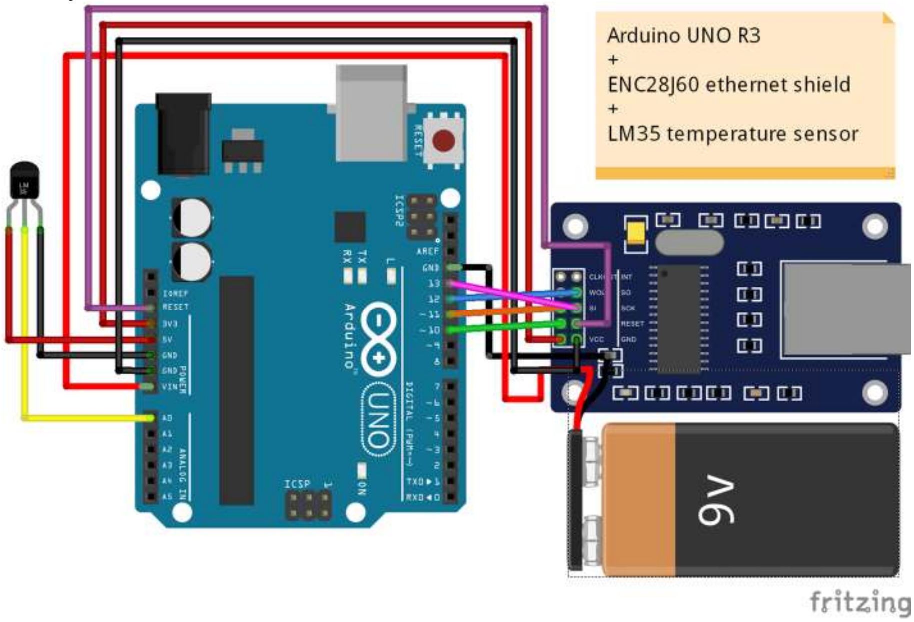

Fig. 2 – The schematic electronic representation of virtual lab platform

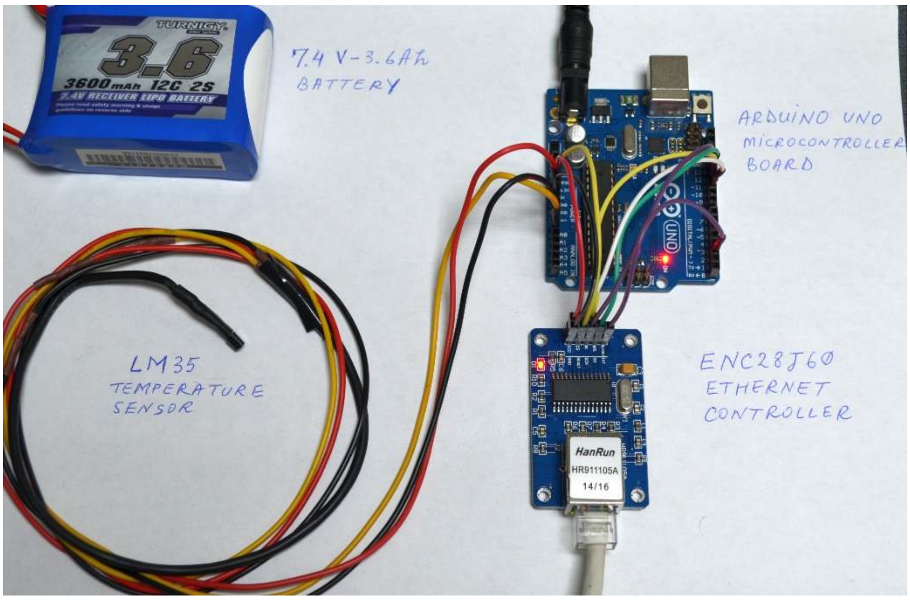

Fig. 3 – The physical platform of the virtual lab

Figure 2 shows the schematic diagram containing all the components used for the demonstration and it was designed using the Fritzing electronic prototyping software (http://fritzing.org/). All the connections were realized respecting the wiring methodology standards. In figure 3, we have the real platform in action with all the components connected in the same way as in figure 2.

The software application

In order to connect the platform with the Xively web service we need to program the Arduino microcontroller board with the corresponding source code, as the following:

```
#include <EtherCard.h>
// The FEED ID and API KEY created in Xively web platform
#define FEED "1175311349"
#define APIKEY "xuIM3MbWh0JIrmN8znZIlGiGkMwHADKyDOVbsmwIw3tBiqwy"
float tempC;
int tempPin = 0;
// ethernet interface mac address for ENC28J60
byte mymac[] = { 0x74, 0x69, 0x69, 0x2D, 0x30, 0x31 };
// Xively URL address
const char website[] PROGMEM = "api.xively.com";

byte Ethernet::buffer[700];
uint32_t timer;
Stash;
static char statusstr[10];
void setup () {
  Serial.begin(57600);
  Serial.println("\n[webClient]");
  if (ether.begin(sizeof Ethernet::buffer, mymac, 10) == 0)
    Serial.println( "Failed to access Ethernet controller");
  if (!ether.dhcpSetup())
    Serial.println("DHCP failed");
  ether.printIp("IP: ", ether.myip);
  ether.printIp("GW: ", ether.gwip);
  ether.printIp("DNS: ", ether.dnsip);
  if (!ether.dnsLookup(website))
    Serial.println("DNS failed");
  ether.printIp("SRV: ", ether.hisip);
  Serial.println("\n");
  Serial.println("... Start Reading Data and uploading to Xively  ...\n");
}
void loop () {
  word len = ether.packetReceive();
  word pos = ether.packetLoop(len);
  if (millis() > timer) {
    timer = millis() + 2000;
    // convert the analog data to temperature values
    tempC = (5.0 * analogRead(tempPin) * 100.0) / 1024.0;
    Serial.println(tempC);
    delay(1000);
    dtostrf(tempC, 3, 1, statusstr);
    // we can determine the size of the generated message ahead of time
    byte sd = stash.create();
    stash.print("LM35_sensor,");
    stash.println(statusstr);
    stash.save();
    // generate the header with payload - note that the stash size is used,
    // and that a "stash descriptor" is passed in as argument using "$H"
    Stash::prepare(PSTR("PUT http://$F/v2/feeds/$F.csv HTTP/1.0" "\r\n"
                        "Host: $F" "\r\n"
                        "X-PachubeApiKey: $F" "\r\n"
                        "Content-Length: $D" "\r\n"
                        "\r\n"
                        "$H"),
                   website, PSTR(FEED), website, PSTR(APIKEY), stash.size(), sd);
    // send the packet - this also releases all stash buffers once done
    ether.tcpSend();
  }
}
```

This code was written, compiled and uploaded using Arduino IDE, thus enabling the platform to connect to Xively web service and transmit sensor data.

4. The results

Xively web service is developed by LogMeIn and offers an Internet of Things product relationship management solution, both for personal and enterprise use (https://xively.com/). For the case study we chose the Xively personal service, being free for developers worldwide.

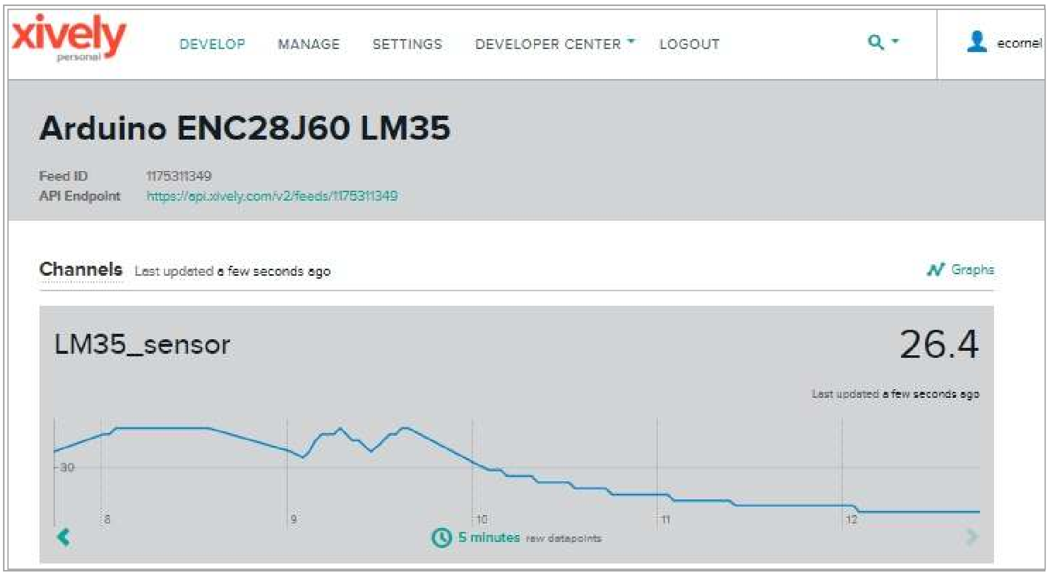

Fig. 4 – The temperature graph displayed in Xively web platform

The philosophy beyond Xively is adding Internet enabled devices inside the web platform and using the FEEDs and API keys provided to identify devices and communicate over Internet. Any communication can be either public or private, depending of the API key used and if someone uses multiple sensors Xively supports different channels of communication for each of them.

After creating the personal Xively account and adding the physical platform as the device, the web service has provided us a FEED ID and API key, necessary for communicating with our platform. Figure 4 shows how the graph with temperature values is displayed inside Xively web platform in the browser. The data is updated in real-time, the refresh interval being set in the Arduino code as needed; for the demonstration it could be observed that the Xively web service can efficiently refresh and display data even at 2 seconds update refresh interval.

As for the whole system, it is one robust enough and a very low cost solution, and it can be used by any laboratory development team in education institutions worldwide. The platform delivers sensor data as long as the battery can provide energy, or can send data continually without being turned off, when using the normal 110/220V electricity from a laboratory room.

5. Conclussions

The study presented here demonstrates that IoT is the future in many fields where Internet enabled devices are needed to communicate and transmit useful data, as in the case of online virtual laboratories. The results of the case study proves that IoT plays an important role for a continuous improvement in education systems in Romania, in order to have a competitive and high quality educational process, which can be acknowledged and accredited worldwide.

6. Bibliography

[1]. Chaos, D.; Chacón, J.; Lopez-Orozco, J.A.; Dormido, S. (2013), Virtual and Remote Robotic Laboratory Using EJS, MATLAB and LabVIEW. Sensors 2013, 13, 2595-2612.

[2]. Oriwoh, E., & Conrad, M. (2015), 'Things' in the Internet of Things: Towards a Definition, International Journal of Internet of Things, Vol. 4 No. 1, pp. 1-5. doi: 10.5923/j.ijit.20150401.01.

[3]. Sandu, F., Nicolae, P., Kayafas, E., & Moraru, S. A. (2008). LabVIEW-based remote and mobile access to real and emulated experiments in electronics. In Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments (pp. 1–7). Athens, Greece: ACM.
[4]. Stefanovic, M., Cvijetkovic, V., Matijevic, M. and Simic, V. (2011), A LabVIEW-based remote laboratory experiments for control engineering education. Comput. Appl. Eng. Educ., 19: 538–549. doi: 10.1002/cae.20334.
[5]. Mario Morales, IDC, http://www.idc.com .
[6]. http://en.wikipedia.org/wiki/Internet_of_Things .
[7]. http://en.wikipedia.org/wiki/WebSocket .

Acknowledgements

This paper has been financially supported within the project entitled "Horizon 2020 - Doctoral and Postdoctoral Studies: Promoting the National Interest through Excellence, Competitiveness and Responsibility in the Field of Romanian Fundamental and Applied Scientific Research", contract number POSDRU/159/1.5/S/140106. This project is co-financed by European Social Fund through Sectoral Operational Programme for Human Resources Development 2007-2013. Investing in people!