RevoCampus: a Distributed Open Source and 
Low-cost Smart Campus

Olivier Debauche*
Faculty of Engineering - ILIA / Infortech
University of Mons
Mons, Belgium
BioDynE / TERRA
Gembloux, Belgium
Orcid: 0000-0003-4711-2694

Rachida Ait Abdelouahid*
Faculty of Sciences Ben M'Sik
University Hassan II
Casablanca, Morocco
Orcid: 0000-0002-9582-7988

Saïd Mahmoudi
Faculty of Engineering - ILIA / Infortech
University of Mons
Mons, Belgium
Orcid: 0000-0001-8272-9425

Yahya Moussaoui
EILCO Student
University of Coastal Opal Coast
Calais, France
Orcid: 0000-0002-4585-0370

Abdelaziz Marzak
Faculty of Sciences Ben M'Sik
University Hassan II
Casablanca, Morocco
Orcid: 0000-0002-6314-1498

Pierre Manneback
Faculty of Engineering - ILIA / Infortech
University of Mons
Mons, Belgium
Orcid: 0000-0003-3990-3621

Abstract—Smart Campus can be assimilated to small smart 
cities in which learning experience and living conditions are 
improved by smart environment and IoT concepts. In this paper, 
we present (R)evoCampus our Smart Campus solution based 
diverted smart Home interoperable protocol platforms, micro 
controllers ESP32, low-cost sensors. This architecture uses at 
same time the principles of IoT, smart environment technologies, 
and smart city concepts to develop an effective use of the 
resources, and to improve the quality of life inside the whole 
University. In the proposed solution, Wi-Fi protocol is used 
for communication in indoor while outdoor communications are 
ensured by LoRaWAN protocol.

Index Terms—Smart Campus, Smart City, Smart Environment, 
Internet of Things

I. INTRODUCTION

Smart Campus usually refers to buildings, ground, and 
places where university is located. Emergence of smart en-
vironment technologies and the presence of digital native 
equipped lively student community have enabled the develop-
ment of smart campus to improve experience of studying, 
sharing learning contexts, in time and space [1].

The main technologies supporting the smart campus include 
Cloud computing and Mobile Edge Computing, Internet of 
Things, Augmented Reality, Artificial Intelligence and Ma-
chine Learning [2] [3].

Abuarquoud et al. [4] have identified important benefits 
which can be obtained in a smart campus. Among this one, 
we keep up: (1) The promotion smart energy, water and waste 
management thought IoT-based services; (2) The monitoring 
of electrical device to notify automatically maintenance team 
to improve response time to operation; (3) The detection of 
non-authorized people in an area or opened windows and 
doors to prevent intruders; (4) The assistance of students and 
staff to find a parking place; (5) The automation of students 
attendance.

Designing a Smart Campus asks to tackle some challenges: 
social, economical, juridical and technical [5]. In a Smart cam-

* Olivier Debauche and Rachida Ait Abdelouahid are co-first authors
978-1-7281-8704-4/20/$31.00 ©2020 IEEE

Prandi et al. [1] have proposed a categorization of smart 
campus in three mains approaches: The first group, tech-
nology driven, exploits IoT services providers and Cloud 
Computing to transform a university into an intelligent campus 
environment. The second group adopts smart city concept 
considering smart campus as a small and self-contained city 
where users play a key and an active role int the crowdsourcing 
and/or crowdsensing. The third based on the development 
of an organization or business process collect data about 
environmental aspect to provide services, reduce costs and 
 improve the quality of life inside and outside the campus.

The aim of a (R)evocampus is to improve the everyday life 
for the University while decreasing the ecological footprint 
of buildings sent by reducing the costs of functioning (water, 
electricity, waste, etc.). This campus is digital, sustainable and 
intelligent combining innovative teaching equipment, a wide 
range of sensors, systems of communications, location, stor-
age, and simulation. It applies the three approaches described 
below in using IoT, smart environment, crowdsourcing and 
crowdsensing to develop an effective use of the resources of 
the whole university, and improve quality of life.

Smart Home Automation systems such as Home Assistant
or openHAB have been developed to interconnect large wide amount of sensors using different network protocols. Home Assistant is an open source automation solution hosted on a simple Raspberry Pi or a similar computer. It allows to survey the state and control numerous devices from a simple and user-friendly interface which respect privacy and don’t publish any information on the cloud. This platform can be diverted to be implemented in each auditorium, office, laboratory, room of the University. These platforms can publish and subscribe to MQTT topics to exchange between them and transfer data to a central point located in the building.

ESPHome is a system allowing to control and manage ESP32 by simple yet powerful configuration files and control them remotely through Home Automation systems like Home Assistant. ESP32 code can also be updated through ESPHome in over-the-air programming (OTA).

The novelty of this paper is actually diverting an existing open source smart home platform (Home Assistant) and network of sensors/actuators controlled by ESPHome to manage rooms and auditoriums of university campus.

The remaining of this paper is organized as follow: We present in section II, a literature review composed of two parts. The former, describe our background in Internet of Things and mentioned previous works achieved in this domain. The second presents related works about Smart Campus. In section III, we describe our proposition of Smart Campus. Then, the section IV explains our experimentation. Results are presented and discussed in section V. Security aspects are discussed in section VI. Then, limitations of our proposition are reported in section VII and finally, we conclude and consider future works.

II. LITERATURE REVIEW

This section is divided into a background part describing our previous papers and their respective contributions and a second part related to the existing state of the art in this field with major contributions on smart campus.

A. Background

In our previous papers, we have successively developed a cloud centric Lambda Architecture [6]. Currently, we complete our central cloud architecture with an edge AI - IoT architecture [7]. In parallel, we have developed several IoT applications: smart home [8], smart city [9], smart building [10].

The cloud architecture and its complement at the edge level and protocol tests allowed us to develop a data storage and processing architecture for data collected by wireless sensors network.

B. Related Works

The Literature contains a tremendous amount of Smart Campus initiatives that address wide range of challenges in the Smart Campus such as energy optimization, Smart Parking, services, mobility, etc. Alonso et al [11] use motion sensors (accelerometer, gyroscope), position sensors (gps, magnetometer), and environmental sensors (light, ambient temperature, relative humidity) of smartphone stored locally in a SQLite database, and afterwards sent to Ubidot platform.

Ward [12] demonstrated that it is possible to use retired smartphone to develop applications such as the availability of lab workspaces, detect and monitor Wi-Fi signal in study areas, follow current location of public transport.

Prandi et al [1] have proposed an architecture based on 3 layers : (1) The sensors layer collects and validates data by comparison with different data sources, and make sensors data available thought open-data repositories; (2) The database Layer stores sensed data in MySQL and use ckan an open-source DMS to allows the interaction with open data; (3) The data visualization layer is composed on one hand of a rich web-based application which allow the interaction with the hyperlocal data and on the other hand a log management web interface that enables to perform analysis and visualize data.


Liu et al [3] proposed a WiCloud platform built around several MEC servers and using Network Functions Virtualization (NFV), Software-Defined Network (SDN).

III. OUR PROPOSITION

Smart Campus generate a large amount of heterogeneous data (numerical or not) of type continuous, discrete, multidimensional or not [5] which must be collected and treated. WiFi is widely available inside of university building, it is easy to connect microcontrollers equiped of sensors on the WiFi network. In order to obtain autonomous units taking local decision, we have implemented a home assistant in each room, auditorium and laboratory which collect and process data produced by a set of ESP32 equiped of sensors managed by ESPHome an add-on for Home Assistant. Each micro controller ESP32 is connected on Wi-Fi and support Over-the-air programming (OTA). The Home Assistant installed on an Raspberry Pi 4 process data and afterwards send them to a central point in the building by means of MQTT protocol.

A. Campus interoperability Architecture

The Fig. 1 presents our IoT interoperability architecture apply to this project of smart campus. This project implements the generic IoT interoperability architecture proposed by Ait Abdeloualid et al. in 2018 [14] and composed of 7 layers:

- **Infrastructure Layer** represent different connected things which acquire environmental information or achieve an action on this environment.
Fig. 1: Smart Campus Interoperability Architecture.

- **Information Layer** allows to communicate data between different connected things by means of microcontrollers. It allows to discover and identify connected objects via microcontroller.

- **Communication Layer** allows the transmission of data between connected things by means of various communications protocols and technologies such as MQTT, Wi-Fi, Z-Wave, NFC, Zigbee, 4G, Bluetooth, etc.

- **Connectivity Layer** ensures the connectivity and interoperability of exchanged data between connected objects thanks to IoT platforms which are for example home automation solutions. These platforms contain services which can store, correlate, analyze, and exploit data. In our use case of smart campus, we have chosen to use Home Assistant as IoT interoperability platform.

- **Middleware Layer** ensures collect data in the cloud, data processing ETL (Data cleaning, integrity verification, normalization, data sorting and transformation), data interpretation with machine learning algorithms, and finally storage in Big Data.

- **Service Layer** provides critical and reliable services for various applications i.e.: weather station, open data, University backend.

- **Application Layer** proposes different manners of data presentation, visualization, and consultation to end-users.

**B. Material**

The material is composed of four packages. The first package contains the microcomputer hosting applications which manages the sensor network. The second package integrates microcontrollers used to sense environment and operate actuators. The third collects different sensors used in this application. Finally, the fourth package gathers actuators.

a) **Micro computer**: is used to host software need to manage and monitor the network of sensors.

The Raspberry Pi 4 is used to host Home Assistant which can take local decision and turn on/off actuator. It also plays the role of local gateway and transfer data to central point of the building with MQTT protocol. The Raspberry Pi 4 is powered by a Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz with 2GB LPDDR4-3200 SDRAM, a Wi-Fi 2.4/5.0 GHz IEEE 802.11ac wireless, Gigabit Ethernet, Bluetooth 5.0, Bluetooth Low Energy (BLE), and a Micro-SD card slot for loading operating system and data storage. (See Fig. 2a)

b) **Micro controllers**: are used to connect sensors and transmit data.

The ESP32-Wroom-32 is a microcontroller that is programmed like an Arduino using Arduino IDE. It is equipped with a Wi-Fi and Bluetooth interface which allows it to communicate with the local gateway configured as an access point. ESP-Wroom-32 contains an Xtensa dual-core 32 bit LX6 240 MHz microprocessor, 520 KiB SRAM, 4 MiB Flash Memory. It is also equipped with 10 GPIO, 4
Serial Peripheral Interface (SPI), 2 Inter-IC Sound ($I^2S$), 2 Inter-integrated Circuit ($I^2C$). It also contains a 12-bit SAR ADC up to 18 channels, 2 8-bit DACs. This microcontroller is used to connect sensors inside buildings (Fig. 2b).

The ESP32-CAM AI-Thinker is equipped with an ESP32-S chip, an OV2640 camera, microSD card slot and several GPIOs to connect peripherals (Fig. 2c).

ESP32 Lora is equipped in addition with an Oled 0.96” display and a Semtech chip to communicate in LoRa with a gateway and transmit data on the things network (TTN) (Fig. 2d).

c) Auditorium Sensors: measures environmental parameters.

The AM2315 a $I^2C$ Sensor measures the Air temperature with an accuracy of ±0.1°C between -20°C to 80°C and the humidity with an accuracy of 2% between 0 to 100% (Fig. 3a).

The TSL2591 (Adafruit Industries LLC) is a high dynamic range digital light sensor which can detect light ranges from up to 188 µLux up to 88,000 Lux (lumens per square meter). This sensor provides full spectrum, visible and infrared (IR) measures, and also the light intensity expressed in Lux (Fig. 3b).

The PIR Motion Sensor HC-SR501 is a adjustable sensitivity module that allows for a motion detection range from 3 meters to 7 meters (Fig. 3c).

The MH-Z19 (Zhengzhou Winsen Electronics Technology CO., LTD) is non-dispersive infrared (NDIR) principle to detect the existence of CO$_2$ in the air. This sensor is connected to the UART at 9600bps and measures carbon dioxide in a range of 0-5000ppm (Fig. 3d).

The contact switch allows to detect the opening or closing of a window (See Fig. 3e).

The DS3231 is a low-cost, and extremely accurate $I^2C$ real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO). The RTC plays a crucial role in the automation of control processes of environmental condition. This RTC is also equipped with a 32Kbits EEPROM allowing to store next step in the plant growing process which guarantees the operation of the installation in the event of a network failure. A DC 3V lithium battery ensures the power to the real-time clock for 10 years (Fig. 3f).

d) Weather Station Sensors: 
The BME680 is a sensor which allows to measure at same time the air temperature with an absolute accuracy of 1°C, relative humidity with an absolute accuracy of 3%, and air quality with an accuracy of 15% (See Fig. 4a).

The DPS310 (Infineon Technologies) is an I²C high precision Barometric Air Pressure Sensors which can acquire the pressure with ±1 hPa absolute accuracy (Fig. 4b).

The GL5516 is a photoresistor with a spectrum peak value at 540nm operating between -30°C to 70°C (See Fig. 4c).

The anemometer (Wind Speed Sensor) for weather station WH1080 (See Fig. 4d).

The TP40556 is a rechargeable lithium batteries module with protection circuits using the constant-current/constant-voltage (CC/CV) charging method. It can be powered by USB or power supply between 4.5 and 6.0V (See Fig. 4e).

The regulator DC12V-DC5V transform an input power supply of DC 8 to 40V in DC 5V maximum 25V. It is placed between the TP4056 charging module and the DC 12V 5W solar panel (See Fig. 4f).

The DS3231 was described in previously (See Fig. 3f).

e) Actuators: controls external equipment.

To turn on/off actuators, we use also ESP32-Wroom-32 with relay card to activate and deactivate air extractor and electric blinds (See Fig. 5a).

While Z-Wave devices are directly connected and controlled by the Raspberry Pi 4 by means of a USB Z-Wave controller (See Fig. 5b).

C. Implementation

The implementation of our proposition is composed of four parts. A central point placed in each campus collect all data transmitted by all Home Assistant/ESPHome placed in each meeting room, auditorium, office, laboratory, and office.

a) A2IoT Architecture: centralizes all data from all nodes.

Our A2IoT is completely described in [7]. This architecture uses containerization and kubernetes an container orchestrator. It is composed of a Odroid N2 master which manages a mixed cluster of 3 Nivida Jetson to hosts adapted algorithms of Artificial Intelligence and 3 Odroid N2 to hosts micro-services.

The A2IoT architecture received data from all Home Assistant / ESPHome managing a local network of sensors
and actuators. Each Raspberry PI is by the Ethernet Network of the University Network to A2IoT architecture and transmit them sensing data by MQTT Protocol and images taken by AI-Thinker to detect person presence (See Fig. 6). A2IoT architecture centralize all data produce and allows to remotely manages Raspberry PI and operate maintenance on ESPHome using OTA firmware deployment.

b) Home Assistant / ESPHome: acquires environmental data and operates actuators.

The sensors deployed in classrooms enable to collect ambient parameters (air temperature [°C], air humidity [%], CO₂ rate [ppm], light intensity [lux], passive infrared (PIR) presence sensor [-]) connected to a ESP32-Wroom-32. This cheap micro controller transmit data on MQTT server by mean of dedicated Wi-Fi SSID in WAP2-PSK deployed in all buildings of the Smart Campus (See Fig. 7). The same micro controller is used to actuate air extractors, control lights and connected radiator valves on basis of sensed data and auditorium occupancy planning recovered from hyper planning \(^1\).

The Table I shows sensors typically used for the sensing of ambient parameters in an auditorium. Air temperature, Air humidity are acquired by means of a am2315 (Aosong), the light intensity is measured with a TSL2591 (ams AG) in a range comprise 0.000118 to 88,000 lux, the CO₂ rate between 0 to 5000 ppm is determined with a MH-Z19 Winsen, HC-SR501 detect presence of persons in a range of 3 to 7 m.

\(^1\)http://www.index-education.com/fr/presentation-hyperplanning.php

---

2http://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete
3https://www.sigfox.com/
4http://www.ingenu.com
5http://www.weightless.org/
6https://lora-alliance.org/about-lorawan
7http://www.telensa.com
8http://waviot.com/technology/whats-is-nb-fi
We have selected LoRaWan because it is free via the Things Network (TTN) and it does not require a sim card and to take out a subscription to access the network. In addition, the university has several gateways of the TTN network deployed on its campuses.

A mesh of antenna covers the campus in which each antenna covers a range up to 2 km. We use LoRaWan protocol to send ambient parameters to the implantation MQTT server and activate actuators in order to act to environment.

d) Weather Station: measures external environmental conditions.

The weather node is built around an ESP32-Lora V2 equipped of a 0.96” OLED Display is powered by a 3.7V Li-Ion battery. A 5W, 12 V solar panel mono crystalline is connected to a DC12V-DC5V converter which powers the TP4056, a Li-ion battery charger module. A BME680 Digital Sensor Temperature, Humidity, Atmospheric Pressure, Air Quality Sensor Module is connected to the ESP32-Lora microcontroller by means of I2C bus. A Froggit wind speed sensor acquires the wind speed. A Photoresistor GL5516 allows to measure the light Intensity (See Fig. 8).

![Fig. 8: Block Scheme of weather station.](image)

IV. EXPERIMENTATION

We have chosen to equipped a computer laboratory with 6 windows, 1 air extractor implanted in 1 window and 3 radiators. Moreover, it contains also 25 personal computer and a video projector which can be tun on / off manually.

We have installed the last release (109.6) of Home Assistant and the last release of ESPHome (1.14.3) on the local micro computer, a Raspberry Pi 4 4gb. Specific libraries for ESPHome have been developed to allows the support of TSL2591 and the DS3231 Real-Time Clock.

We have changed radiator valve with Z-Wave connected radiator valves which can be managed directly by the Raspberry Pi via a USB Z-Wave dongle.

Two environmental Sensor Node built around a ESP32-Wroom-32 on which are connected to I2C bus a Real Time Clock (DS3231), a light intensity (TSL2591), a Pressure Sensor (DPS310), a CO2 Sensor (MH-Z19) in UART, and a temperature & humidity sensor and Motion Detection Sensor on digital pin. These nodes are placed on the ceiling and sent their data at intervals of 60 seconds.

An ESP32 operates the relay card which tun on/off the air extractor, turn on/off the lights, turn off the video projector and go up/down electrical blinds.

An ESP32 is relied with contact switches in order to verify that windows are closed or opened.

Two AI-Thinker nodes are installed front of door to detect people presence. Images are sent to Home Assistant which relay them when it is necessary to the A2IoT Architecture where people detection is operated with Tiny Yolo V3.

A weather station built around an ESP32-Lora V2 equipped of temperature, humidity, air quality sensors (BME680), a Real Time Clock (DS3231), a High precision Atmospheric pressure sensor(DPS310) connected on I2C bus.

All ESPHome nodes status and wireless signal strength are monitored by Home Assistant in order to detect failed or temporary disconnected nodes.

An automation has been implemented in Home Assistant with following rules (See Table III).

Rule 1 activates the air extractor when the air temperature measures with the am2315 is more than 23°C or air humidity exceeds 70% or CO2 rate exceeds the limit of 1500ppm. Rule 2 stop the air extractor when all the three following conditions are met: temperature < 20°C, air humidity <40%, and CO2 rate < 1000ppm. Rules 3 and 4 increase or decrease the

---

TABLE I: Power consumption according to manufacturer’s data and interface of connection.

<table>
<thead>
<tr>
<th>Component</th>
<th>Interface</th>
<th>Operation mode</th>
<th>Supply Current (Max)</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time Clock (DS3231)</td>
<td>I2C</td>
<td>Active / Stand-by / Conversion</td>
<td>300 µA / 170 µA / 650 µA</td>
<td>5V</td>
</tr>
<tr>
<td>High Dynamic Range Digital Light Sensor (TSL2591)</td>
<td>I2C</td>
<td>Sleep / Active</td>
<td>4 µA / 325 µA</td>
<td>5V</td>
</tr>
<tr>
<td>Temperature &amp; Relative humidity (AM2315)</td>
<td>Digital</td>
<td>Stand-by / Measuring / Converting</td>
<td>50 µA / 1.5 mA / 10 mA</td>
<td>3 to 5V</td>
</tr>
<tr>
<td>Distance sensor (HC-SR501)</td>
<td>Digital</td>
<td>Sleep / Normal</td>
<td>2.5 mA / 20 mA</td>
<td>4.5 to 5.5V</td>
</tr>
<tr>
<td>CO2 Sensor (MH-Z19)</td>
<td>UART</td>
<td>Average</td>
<td>&lt; 18 mA</td>
<td>3.6 to 5.5V</td>
</tr>
</tbody>
</table>

---
TABLE II: Power consumption according to manufacturer’s data and interface of connection for weather station.

<table>
<thead>
<tr>
<th>Component</th>
<th>Interface</th>
<th>Operation mode</th>
<th>Supply Current (Max)</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time Clock (DS3231)</td>
<td>I²C</td>
<td>Active / Stand-by / Conversion</td>
<td>300 µA / 170 µA / 650 µA</td>
<td>5V</td>
</tr>
<tr>
<td>Photoreistor (GL5516)</td>
<td>Analogic</td>
<td>Unknown</td>
<td>Unknown</td>
<td>3 to 5 V</td>
</tr>
<tr>
<td>Wind Speed (Foggit)</td>
<td>Analogic</td>
<td>Unknown</td>
<td>Unknown</td>
<td>3 to 5 V</td>
</tr>
<tr>
<td>Tmp. Hum. Air Quality Sensor (BME680)</td>
<td>Digital</td>
<td>Sleep / Standby / Normal</td>
<td>1 µA / 0.8 µA / 0.09 mA to 12 mA</td>
<td>4.5 to 5.5V</td>
</tr>
<tr>
<td>Pressure sensor (DPS310)</td>
<td>I²C</td>
<td>Average / Standby</td>
<td>1.7 µA / 0.5 µA</td>
<td>1.7 to 3.7 V</td>
</tr>
</tbody>
</table>

Fig. 9: Example of few parameters monitored an auditorium.

TABLE III: Automation expressed in High Level Semantic Rules.

<table>
<thead>
<tr>
<th>#</th>
<th>Fact Triggering Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ExtractorStart Observation hasTemperature &gt; 23°C ∨ Observation hasHumidity &gt; 70% ∨ Observation hasCO2 &gt; 1500ppm → ns:ExtractorStart</td>
</tr>
<tr>
<td>2</td>
<td>ExtractorStop Observation hasTemperature &lt; 20°C ∧ Observation hasHumidity &lt; 40% ∧ Observation hasCO2 &lt; 1000ppm → ns:ExtractorStop</td>
</tr>
<tr>
<td>3</td>
<td>ValveUP Observation hasTemperature &lt; 20°C → ns:ValveUp</td>
</tr>
<tr>
<td>4</td>
<td>ValveDown Observation hasTemperature &gt; 22°C → ns:ValveDown</td>
</tr>
<tr>
<td>5</td>
<td>LightOn Observation hasPeopleDetected true ∨ hasPIRDetection true → ns:LightOn</td>
</tr>
<tr>
<td>6</td>
<td>LightOff Observation hasPeopleDetected false ∧ hasWindowClosed true → ns:LightOff</td>
</tr>
<tr>
<td>7</td>
<td>LightDimmed Observation hasProjectorOn true → ns:LightDimmed</td>
</tr>
<tr>
<td>8</td>
<td>BlindsUp Observation hasLightIntensity ≤ 1000lux → ns:BlindsUp</td>
</tr>
<tr>
<td>9</td>
<td>BlindsDown Observation hasLightIntensity &gt; 1000lux ∧ hasWindowClosed true → ns:BlindDown</td>
</tr>
<tr>
<td>10</td>
<td>ProjectorOff Observation hasPeopleDetected false ∧ hasProjectorOn true → ns:ProjectorOff</td>
</tr>
</tbody>
</table>

...temperature by operating the Z-Wave radiator valves via the Raspberry Pi 4. Rule 5 turn on light when a people is detected by the HC-SR501 motion sensor or at least one people is detected at A2IoT Architecture level by Tiny Yolo V3 on images sent by AI-Thinker. Rule 6 turns off lights when nobody is detected on AI-Thinker images and all windows are closed. This rule prevents the forgetting to close windows. Rule 7 dims lights when the video projector is turned on. Rule 8 goes up electrical blinds when the light intensity measures by the TSL2591 sensor is under or equal to 1000 lux. Rule 9 goes down the electrical blinds when the light intensity measured by TSL2591 is more than 1000 lux and all windows are closed (verified with contact switches placed on windows). Finally, rule 10 closes the projector when it is tuned on and there is nobody in the laboratory.

V. RESULTS AND DISCUSSION

The major advantage to use Home Assistant in each room is to have autonomous solution that control actuators such air extractors, radiator valves, dimmers. But this solution is most costly than a central approach because it need a Raspberry Pi to host Home Assistant / ESPHome in each room. On the other hand, the fact of distributing the resources locally makes the solution more resilient to the breakdowns of network or electricity cut which can occur in certain parts of the building. Moreover, Home Assistant and ESPHome are maintained by a important community which add continuously new integration and extend possibilities in term of sensors, actuator supported, and system interfacing.

VI. SECURITY

Home Assistant is not dependent on cloud services and use only a dedicate local network only for the smart campus. Wi-Fi Protected Access with pre-shared key (WPA2-PSK) is used to secure wireless network. Otherwise, the ssid of the Wi-Fi network is not broadcasted. Each raspberry pi use a SSL Letsecrypt certificate to secure http connection. Each ESP32 is protected by a security key which is requested during the ESP configuration updates via OTA. All the smart system is deployed in a specific Virtual Lan (VLAN) to which other
network users do not have access. Static external IP authorized to access to the VLAN are specified at router level.

VII. LIMITATIONS

The main limitation is the number of devices which can be simultaneously connected on the university Wi-Fi access points. Despite the implementation of a separate Wi-Fi network for the Internet of Things, it is shared by other IoT projects, which limits the pass band and/or the number of connectable ESP32s. The saturation of certain access points forced us to add a new private access point for the sensor network which is connected to the university’s Ethernet network. The use of access points when the number of ESP32 exceeds 16 is necessary. The maximum number of ESP32s that we have managed to connect to a single Raspberry Pi 4 is 32 nodes.

A second limitation is the support of a limited number of sensors and actuators by ESPHome currently, although it is possible to add support for new sensors and actuators with a few lines of codes.

Another limitation is the use of LoRaWan as an outside data transfer protocol. However, the latter is dependent on the availability of a gateway network on the campus entrance, all the more so as the things networks is a network of free gateways without any guarantee of availability. However, access providers now offer paid networks with guaranteed service.

Finally, the life duration of SD card used to store OS and applications on the Raspberry Pi 4 is an other problem. Our experimentation have shown that the duration of SD-card is limited to 5.5 months. To address this issue, we will change in our future developments, RPI 4 by Odroid N2+ up to twice as powerful and supporting eMMC cards which are faster and more durable than SD cards.

On The Things Network a Fair Access Policy that limits the uplink airtime to 30 seconds per 24 hours and per node, and the downlink messages to 10 messages per 24 hours and per node is applied.

The theoretical time of transmission can be calculated as follow: Each transmission begin with 8 preamble periods + 2 sync periods to which we must add (1 * 8 header bits + the frame size in bytes * 8 data bits + 4 * 8 integrity code bits + 4 * 8 CRC bits) / Spreading Factor = number of transmission periods.

The LoraWan network offers a Spreading Factor 7 (SF7) with a bitrate of 50,000 bit/s, allowing to transmit theoretically 937 frames per day or each 92s period for frames of 10 bytes. With the lower throughput at Spreading Factor 12 (SF12) and allowing to transmit a maximum of 42 frames of 10 octets per day or each 2010s. Duration of periods in SF7 (7bits) and SF12 (12bits) are respectively transmitted in 1ms and 32ms.

The number data bytes to transmit per day and the frequency impact directly the number of antenna and their choose of their implantation.

VIII. CONCLUSION AND PERSPECTIVES

In this paper, we propose a distributed smart campus solution based on diverted Smart Home solution: Home Assistant coupled with ESP Home used to manage a network of ESP32 microcontroller. Each room of the building is equipped of an autonomous Home Assistant which sense environmental parameters and automate actuators in function determined rules. The solution is easily extensible and the replacement of failed device, the adding of new device are easy. Moreover, devices in the ESPHome network are updated in Over-the-air programming (OTA).

In our future works, we will implement other outdoor protocols such LTE-M (CAT-M1 and NB-IoT), SigFox. These protocols allows to support new outdoor devices. Moreover, 5G will be used when it will be available and will allows to develop new applications such as pushing personalized content directly to the smartphones of campus students, geolocation and monitoring of student flows, especially for the detection of anomalies such as mass movement, leakage or mass concentration.

ACKNOWLEDGMENT

The authors would especially like to thank Mr Adriano Guttaduria for his technical support and setting up all the computing systems, and Mr Rudy Schartz for setting up all the electronic systems necessary for carrying out this research.

REFERENCES


