

INTERNET OF THINGS (IoT): WIRELESS DIAGNOSTICS SOLUTIONS

R. Homs-Puron, S. Astorga, G. Cuní, D. Fernandez-Carreiras, O. Matilla, A. Rubio, ALBA-CELLS
Synchrotron, Cerdanyola del Vallès, Spain
R. Montano, ESS ERIC, Lund, Sweden

Abstract

ALBA requires a diagnostic system, where mainly include the temperature acquisition around the facility, such as tunnel, service area, experimental area, laboratories and auxiliary facilities. There is a big area to be covered and the location of the sensors may not be fixed, those measurement spots require a strong correlation to the machine startup configuration. This has an impact on the size whether a traditional wired installation is used, due the huge of measurement points to be covered; in addition, the restricted machine access schedule makes difficult their installation. In this paper we intend to describe one solution based on ESP8266 system-on-a-chip (SoC).

INTRODUCTION

The ALBA Equipment Protection Systems (EPS) is a homogeneous system based on B&R PLCs distributed system independent of from the Tango Control System [1] and became a versatile system that has been adapted for interlock, diagnostic acquisition and motion and collision control in both, accelerators and beamlines.

EPS system is under continuous improvement, these are the most significant changes [2]:

- Upgraded to use Powerlink version 2.
- Implemented the possibility to have differential temperature interlock.
- The 512 storage ring temperatures are treated as interlock, for that reason there are no temperatures for diagnostics.

Adding more temperatures for diagnostics will require an upgrade of the current PLC infrastructure, which will consume a lot of intervention time inside the tunnel and high economic cost.

These are the characteristics that were taken into account for the study and evaluation of the different solutions:

- Price
- Wireless communication
- Hardware and software complexity.
- Documentation.

HARDWARE

ESP8266

ESP8266 [3] is a system-on-a-chip (SoC) based on a 32-bit RISC CPU at 80 MHz equipped with 64 KB (code) + 96 KB (data) of RAM. It features 2.5GHz Wi-Fi(802.11 b/g/n) with full TCP/IP stack, 16 general purpose input/output (GPIO), one 10-bit analog-to-digital converter (ADC), Inter-Integrated Circuit (I2C), Serial Peripheral

Interface (SPI), UART, Integrated Interchip Sound (I2S), pulse-width modulation (PWM). Its price is ~10 euros.

Raspberry Pi

The Raspberry Pi family [4] is a series of small single-board computer (SBC). The Raspberry Pi 3 is bundled with on-board WiFi 802.11n, Bluetooth 4.1. It has 46 GPIO, I2C, I2S, SPI, UART, pulse-code modulation (PCM), PWM, USB 2.0 ports, 10/100 Ethernet port. The CPU is a 1.1 GHz ARM with 1GB of RAM. Its price is ~40 euros.

Xbee

Xbee is a family [5] of form factor compatible radio modules based on the IEEE 802.15.4-2003 (ZigBee) standard designed for point-to-point and star communications at over-the-air baud rates of 250 kbit/s. The Xbee Pro S2B includes 10 GPIO, 4 10-bit ADC, UART, SPI, PWM. Its price is ~ 30 euros.

Proof of Concept for ALBA

The three types of hardware evaluated are too different between applications. The most oriented to the industrial application is the Xbee, for that reason we implemented two systems: the first one to measure temperature inside the tunnel with a PT100 sensor, and the second one to measure a high current on CIRCE beamline, but, as they require a coordinator for each one, the cost of the components was ~400 euros and the configuration and installation is not friendly for the user. The Raspberry Pi is a powerful board, but the configuration of the operating system (OS) is more complex to use it as standalone device without monitor, keyboard, etc. The ESP8266 suits [6, 7, 8], as first proposal, to test a wireless diagnostic system. Hardware and software configuration is done in the laboratory and the installation will be carried on by the users, without any support of the computing section. The low cost and the reusability are the strengths of this chip.

ESP8266 FIRMWARE

To implement the firmware we used the Arduino framework [9, 10]. The main goal of the firmware is to be object oriented, to be more scalable, and reusable. The code is divided on three groups of classes: communication, sensors and helpers. The main loop of the program implements a telnet server based on ASCII commands, these commands are verified and processed by the communication classes.

Sensors

The sensors classes inherit from the BasicSensor class which is used on all functions; the polymorphism abstracts the temperature reading of each type of sensor. The BasicSensor class implements a linear correction and checks if the temperatures are inside the working range. The UML class diagram is showed on Figure 1.

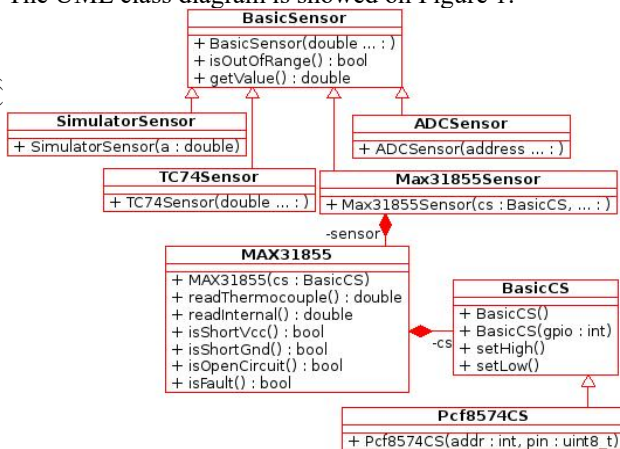


Figure 1: Sensors classes UML diagram class.

In the first iteration to implement a device as remote sensor, we used two type of sensors, TC74(I2C) [11] and Max31855(SPI) [12], and the Analog to Digital Converter (ADC) as a temperature sensor. The typical temperature sensors at Alba are thermocouples, for that reason we select the chip Max31855 which allows to read different types of thermocouples, the board used on the test is MAX31855PMB1 [13] (Figure 2).

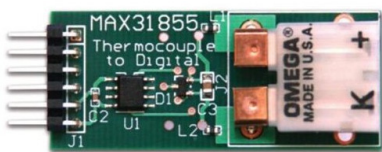


Figure 2: Max31855PMB1 thermocouple sensor board.

The ESP8266 has 16 pins which are used for communication, programming and power. One requirement of the project is to have sensors per controller, for that reason we expanded the GPIOs by using the chip PCF8574CS [14] (Figure 3)

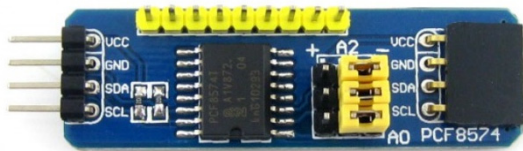


Figure 3: Pcf8574CS GPIOs board.

This board allows us to have eight input/output pins more, which are used as chip selects for the Max31855 chips. To have the possibility to control the Max31855

with the ESP8266 GPIOs and the PCF8574CS the class MAX31855 uses as chip select variable an instances of the BasicCS.

All boards are connected to the mainboard (see Figure 4) which has the ESP8266.

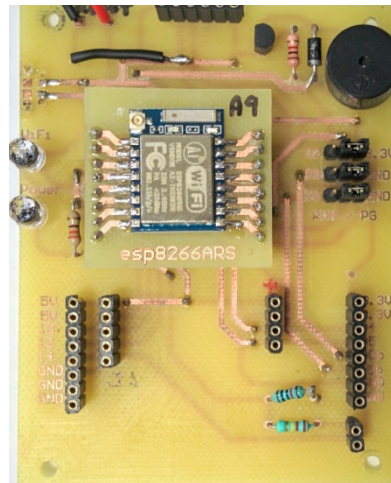


Figure 4: Mainboard for the ESP8266 and the sensors.

CONTROL SYSTEM INTEGRATION

We developed a python library (esp8266ARS) which implements the control of the ESP device in a friendly way. It has the TemperatureSensors class which generates dynamic temperatures attributes for each connected sensor to the ESP device and underpinning the other components of the software. In addition, it provides a basic bash script (see Figure 5) to read all the sensors and state of the device and it state.

```
$> esp8266ARS 192.168.1.73
NrSensor: 2
T1: 25.4
T2: 27.0
Device State: 'OK' 'All sensors are in range'
Device is finding: False
```

Figure 5: esp8266ARS bash script outputs.

To use the python library is really easy as you can see on Figure 6:

```
>> from esp8266ARS import TemperatureSensor
>> dev = TemperatureSensor('192.168.1.35')
>> dev.T1
32.8
>> dev.state
OK; Finding False;
```

Figure 6: Using esp82ARS python library.

A Tango Device Server is the interface to the archiving system, alarm system [15], web report [16] and the Taurus widgets. The tango device uses dynamic generation of the

temperatures by the library and creates an attribute for each one. This functionality allows to the user change the number of sensor on the hardware without changing any configuration of tango device. The device server exports only one device which has as properties the host IP (see Figure 7) of the hardware and the port used for the communication, by default, the port 23 is used.

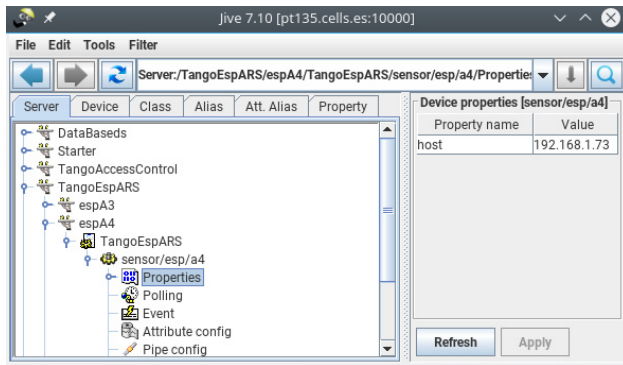


Figure 7: Jive views of the devices tree.

FUTURE ROADMAP

The next targets for the project are:

- Do stress test inside the tunnel.
- Implement 4-20mA inputs.
- Implement analog inputs +/- 10V.
- Test MicroPython: evaluate micro-python allows us to reuse more the code to other devices.

CONCLUSION

The entire system design had as main premises being functional and economic. From the beginning, a special attention was paid to cost and effectiveness.

Next step will be to produce some devices with 4 thermocouples sensors per unit to be used by accelerators and beamlines groups, with the main goal to do a punctual diagnostic and without the cost on installation time and equipment on the EPS system. This first diagnostic allows them to identify the future sensors installation on the EPS system.

In the future scope this kind of technology can be used to monitor several kind of sensors.

ACKNOWLEDGEMENT

We would like to thank all controls, electronics, network system administration groups, but especially to Carlos Falcón, Daniel Roldan, Sergi Pusó, Zbigniew Reszela, Jordi Andreu for their help in project design, development and commissioning.

REFERENCES

- [1] Tango, <http://www.tango-controls.org/>
- [2] A. Rubio, G.Cuní, D. Fernandez-Carreiras, S. Rubio-Manrique1, J. Villanueva1, N. Serra, "ALBA Equipment Protection System, Current Status", ICALEPCS'17, Barcelona, Spain, 2017.
- [3] ESP8266, <https://es.wikipedia.org/wiki/ESP8266>
- [4] Raspberrypi, <https://www.raspberrypi.org/documentation/>
- [5] Digi, <https://www.digi.com/xbee>
- [6] M. Mehta, "ESP 8266: A breakthrough in wireless sensor networks and internet of things", *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, vol. 6, no. 8, pp. 07-11, 2015.
- [7] J. A. Stankovic, 2014. <http://www.intercom.virginia.edu/~stankovic/psfiles/IOT.pdf>
- [8] T. He, J. Stankovic, C. Lu and T. Abdelzaher, A Spatiotemporal, "Communication Protocol for Wireless", *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, pp. 995-1006, 2005.
- [9] B. W. Evans, *Arduino Programming Notebook*, San Francisco, California, USA: Creative Commons, 2007.
- [10] Arduino, <https://www.arduino.cc/en/Guide/HomePage>
- [11] Microchip, "TC74 Tiny Serial Digital Thermal Sensor", 2012.
- [12] Maxim, "MAX31855 Cold-Junction Compensated Thermocouple-to-Digital Converter", 2012.
- [13] Maxim, "MAX31855PMB1 Peripheral Module", 2012.
- [14] Texas Instruments, "PCF8574A Remote 8-Bit I/O Expander for I2C", 2005.
- [15] S. Rubio-Manrique *et al.*, "Extending Alarm Handling in Tango", ICALEPCS'11, Grenoble, France, 2011.
- [16] M. Broseta, *et al.*, "A web-based report tool for Tango control systems via websocket", ICALEPCS'17, Barcelona, Spain, 2017.