Development of a Smart Gateway for a Label Loom Machine using Industrial IoT Technologies

https://doi.org/10.3991/ijoe.v16i04.11853

Rogério Dionísio (✉),
Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal
DiSAC - R&D Unit in Digital Services, Applications and Content
rdionisio@ipcb.pt

Sérgio Malhão
Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal

Pedro Torres
Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal
SYSTEC - Research Center for Systems and Technologies

Abstract—Constant search for efficiency and productivity has led to innovation on the factory shop floor, representing an evolution of the current production systems combined with new technologies of industrial automation and information technology. This work presents a versatile gateway for experimental demonstration of Industrial IoT technologies in a loom machine, allowing sensing, monitoring and data acquisition that was not originally available. We have implemented an approach, based on the OPC UA communication protocol for real time applications, and OPC UA to MQTT conversion mechanism. Raspberry Pi's platform act as an OPC UA server. From the measurements, data stored in a cloud server can be accessed remotely with improved security and visualized from a computer dashboard. One of the conclusions that can be drawn is that the proposed gateway allows data to be stored and easily monitored from a smartphone application or a computer web interface.

Keywords—Industrial IoT, OPC UA, MQTT, Embedded Systems, Node-RED

1 Introduction

Recent years witnessed the forthcoming of Machine-to-Machine (M2M) networks as an efficient means to provide automated communications between distributed devices. The idea of Smart Industry is gaining increasing importance in the current context because of its ability to automate industrial environments with great effectiveness using smart systems. These systems incorporate detection, actuation and control functionalities.

The Portuguese industrial sector follows this trend and searches to technologically re-adapt its production processes to include sensing, monitoring and control functionalities, allowing efficient communication with the equipment on the shop floor with Industrial Internet of Things (IIOT) platforms [1-3]. This is the case of the label loom machines industry.

The loom machine, also known as the Jacquard machine, is an industrial device fitted to a power loom that simplifies the process of manufacturing textiles labels with complex patterns. Although some recent loom machines are equipped with all the necessary sensors and communication ports to connect it to a cyber-physical environment, many industries continue to use old but reliable loom machines on the shop-floor. However, the manufacturing process of these machines cannot be effectively supervised during the waving process, and the common monitoring methods require a number of manpower and material resources.

The remaining of the paper is structured as follow: section 2 describes the system architecture, section 3 presents the gateway design, section 4 follows with the prototype test and results and section 5 present the conclusions.

2 System Architecture

2.1 Label loom requirements

The electrical power box of the looms is cooled by a fan. Whether due to mechanical wear or dust in the air (generated by friction of the textile lines when processed) the fans fail to cool. The looms are not equiped with sensors to react to this problem, that can damage the loom if it goes undetected.

The looms have a label cutting system (calender) based on electrical resistances that can be positioned according to the desired cut. These resistors can fail, and the machine has no mechanism of detecting the problem.

The part of the loom that controls the needles is an electro-mechanical system called Jacquard. This section of the machine can sometimes overheat, leading to premature wear of its internal components and failure. The machine cannot predict or warn of these problems due to a lack of sensors.

There are several problems with the looms, such as textile lines that break frequently, leading to a halt in production. When these failures are detected, the loom illuminates a red light in a tower of lights used for operators to understand the state of the machines. These stops must be recorded manually as there is no connection between the machines and the factory information systems.

The production meter is counted by an inductive sensor placed on a loom shaft, but the sensor is prone to counting errors.

During the production process, the looms produces mechanical and acoustic vibration. Some particular malfunctions can be perceived by sound analysis.

2.2 Proposed solution

Taking into account all the requirements, the proposed architecture is depicted in figure 1. This architecture has incorporated the loom alarm light tower in order to process the tower signals and correlate them with the sensor data. The solution incorporates:

1) Three temperature sensors per loom (one for the Jacquard, one for the calendar and one for the electrical Power Box).
2) One relative humidity, pressure and temperature sensor for air measurements near the loom.
3) An encoder to accurately count production meters.
4) One acoustic sensor and one vibration sensor per loom.
5) Integrate light tower alarm signals through the gateway, using analog inputs.
6) A push button near the machines so the operator can signal malfunctions when they happen.

Moreover, due to restriction on the equipment in use on the factory, that uses MQTT [4] protocol to communicate with an existing IIoT platform, a conversion module from OPC-UA [5] to MQTT is also developed and further described in section 3. The IIoT platform, following the Reference Architecture Model for Industrie 4.0 (RAMI 4.0) [6], connects the shop floor to an external cloud server and database (not shown in the diagram of figure 1).

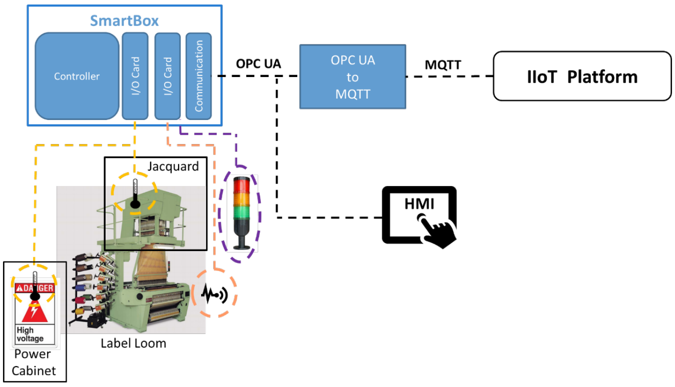

Fig. 1. Physical network architecture of the loom label machine, incorporating a smart gateway and cloud server

3 System Design

Several low-cost gateways have been proposed in the literature [7], but the lack of flexibility of the programming environment is a limitation. The option we propose is a gateway based on Raspberry Pi hardware and OPC UA communication protocol. The system architecture is built upon Ethernet and Wi-Fi networks, as can be seen in figure 2. In this scenario, each smart gateway (Raspberry Pi) are configured as an OPC-UA server to provide information to clients (shop floor equipment / devices, such as computers or mobile devices).

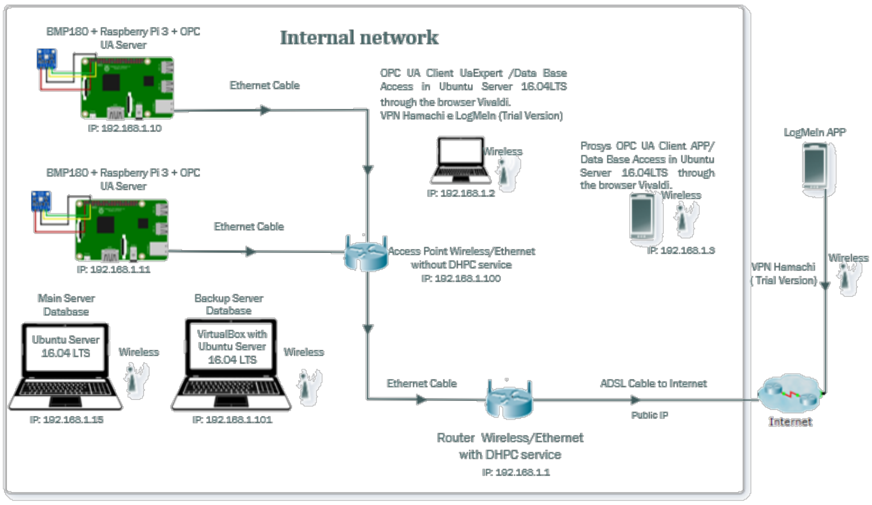
Internal network
BMP180 + Raspberry Pi 3 + OPC UA Server
IP: 192.168.1.10
Ethernet Cable
OPC UA Client UaExpert /Data Base Access in Ubuntu Server 16.04LTS through the browser Vivaldi.
VPN Hamachi e LogMeIn (Trial Version)
Wireless
IP: 192.168.1.2
Prosys OPC UA Client APP/ Data Base Access in Ubuntu Server 16.04LTS through the browser Vivaldi.
Wireless
IP: 192.168.1.3
LogMeIn APP
BMP180 + Raspberry Pi 3 + OPC UA Server
IP: 192.168.1.11
Ethernet Cable
Access Point Wireless/Ethernet without DHPC service
IP: 192.168.1.100
VPN Hamachi (Trial Version)
Wireless
Main Server Database
Ubuntu Server 16.04 LTS
Wireless
IP: 192.168.1.15
Backup Server Database
VirtualBox with Ubuntu Server 16.04 LTS
Wireless
IP: 192.168.1.101
Ethernet Cable
ADSL Cable to Internet
Public IP
Internet
Router Wireless/Ethernet with DHPC service
IP: 192.168.1.1

Fig. 2. Network architecture using Raspberry Pi and OPC UA

The Raspberry Pi SD Card was installed with the Raspbian operating system as well as SSH access, remote VNC connections, and I2C communication enabled. For the Raspberry Pi's to act as an OPC UA server, several libraries were previously installed: libxml2-dev, libxmlsec1-dev, libffi-dev, Freeopcua and cryptography [8].

To subscribe and store all data sent from the OPC UA server, a MySQL database is installed on an Ubuntu Server 16.04 LTS together with Unified Automation UA Expert software [9] on a Windows 10 virtual machine. On the Client side, we use a Prosys Smartphone application [10] to subscribe and visualize the sensor data parameters.

A Python script was written to enable the acquisition of all environmental and machine parameters available with each sensor installed, through the OPC UA Servers. It was also necessary to set up a set_endpoint consisting of an IP address, communication port, library and class, and also to define the object parameters, such as temperature, pressure, relative humidity, vibration, sound, as well as access and permission rules of the OPC UA Clients.

For data acquisition via OPC-UA, the OPC-UA Server endpoint address were programmed to be accessed by UA Expert software on a PC and Prosys App for mobile devices. The Endpoint Address Configuration for a gateway is:

```
  server.set_endpoint
("opc.tcp://192.168.1.10:4840/freeopcua/server/")
```

When configuring *server.set_endpoint*, the communication is done via the OPC-UA TCP protocol; this is essentially a light protocol that establishes a low-level communication channel between a client and a server and provides end-to-end transfer. OPC-

UA TCP transport negotiates maximum buffer sizes between devices and ensures data delivery.

The *get_objects_node ()* function returns the existing Object Nodes on the OPC UA Server, and the *add_object ()* function adds objects through node ID and their name. Variables that contain information about sensor data are added via the *add_variable ()* function and can always be accessed by *get_value ()*. The connection to the database is done through the following instruction:

```
  db=mysql.connect(host='192.168.1.15', port=3306, user =
'root', password = 'admin', database = 'OPCUA')
```

The host is the server IP address (Ubuntu Server), with the default port in the *mysqld.cnf* configuration document on Ubuntu Server, user is the user ("root") password ("admin"). The database parameter is the name of the database created in phpMyAdmin. As an Example, the table definition for storing data from sensor BMP180 is as follows:

```
  sql = """CREATE TABLE sensorBMP180_raspberrypi2(
  temperature DOUBLE,
  altitude DOUBLE,
  pressure INT
  )"""
```

The *cursor.execute (sql)* command makes it possible to create the table within the OPC UA database (query or command). The parameters found in the tuple or dictionary are linked to the variables in the operation. The command *cursor.execute ("" "INSERT INTO sensorBMP180_raspberrypi2 VALUES (% s,% s,% s)" "", (temp, altit, press)* is used to store the values of *temp*, *altit* and *press* variables. This command is programmed within a loop, so as to continuously send the acquired data to the database. The *db.commit ()* command will send a COMMIT statement to the MySQL server, confirming the current transaction.

Data from the sensor is monitored with a Node-RED [11-12] dashboard, through time-varying diagrams. It allows to analyze sensor past values through the interaction between Nod-RED and the Database. Access to data from OPC-UA servers is ensured through the *node-red-contrib-opcua* library, and the access to data from a MySQL database through *node-red-node-mysql* library.

1.1 OPC UA and MQTT bridge

The OPC UA to MQTT bridge was developed with Node-RED flows. figure 3 depicts the conversion process from OPC UA to MQTT topics. figure 4 presents the flow diagram form MQTT topics back to OPC UA nodes.

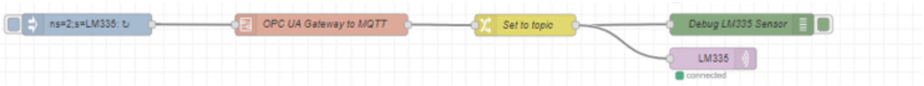

Fig. 3. Conversion of OPC UA node values to MQTT topics for LM 35 sensor temperature

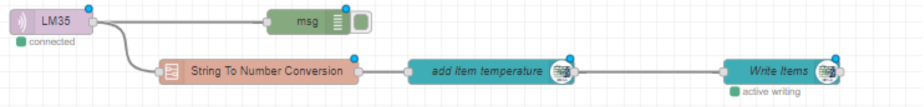

Fig. 4. Conversion of MQTT topics to OPC UA nodes for LM 35 sensor temperature

4 Gateway Demonstration

The prototype module is shown in figure 5. The compact size was intended for demonstration purposes only. In a real scenario, all the components are distributed on the loom machines of the factory shop floor. It incorporates two OPC-UA servers based on the Raspberry Pi platform (white boxes in each side of the module). Each OPC-UA server is equipped with temperature and humidity sensors. The monitoring dashboard runs on a laptop PC. The same laptop is also running the gateway program to convert between MQTT topics and OPC UA nodes. MQTT nodes. Two modules, using MQTT protocol, were added and configured to emulate the equipment in use on the factory (front side of the module). They are connected to additional sensors (one acoustic sensor, one vibration sensor, light tower) and actuators (a push button to signal malfunctions).

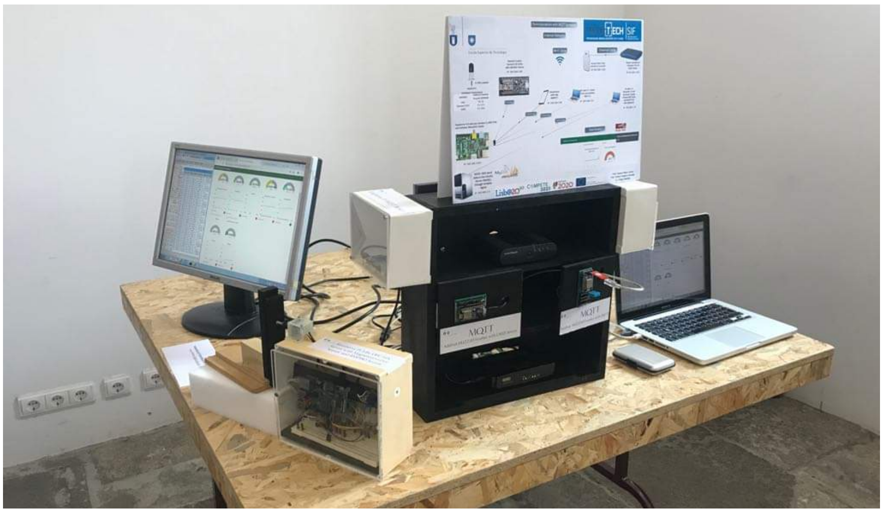

Fig. 5. Complete workbench running during a demonstration event

4.1 Dashboard page test

Figure 6 illustrates the dashboard, created in Node-RED, monitoring sensor values through OPC-UA servers.

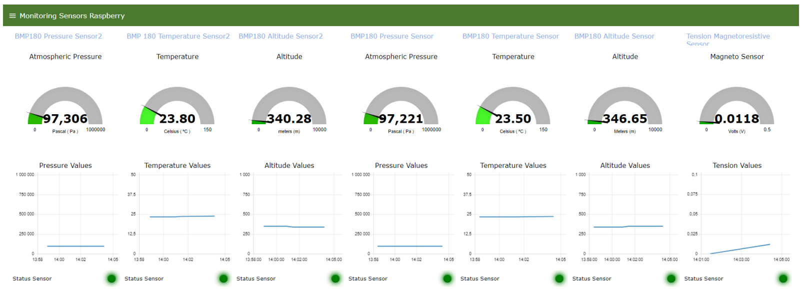

Fig. 6. Node RED dashboard with values and graphs for each of the sensor parameters

Node-RED flow is accessible through a localhost with IP address 127.0.0.1:1880 and the corresponding dashboard is accessible through IP address 127.0.0.1:1880/ui/#0. With Node-RED, we can observe the sensor data and simultaneously send it to a MySQL database configured on UBUNTU Server 16.04 LTS, and then export it to PDF, Excel or CSV files.

5 Conclusion

The development of a versatile gateway has created an affordable solution for industrial IoT applications, based on the Raspberry Pi platform. It can monitor critical parameters of the shop-floor factory through open-source software, both for smartphone and Desktop / Laptop computers, as well as storing data for remote analysis. Some limitations remain to find libraries suitable to program alarm features in the OPC UA Servers. OPC UA protocol is adequate to implement M2M solution in factory shop-floor.

Future work will include additional sensors for industrial environment (angular rotation speed), alternative hardware platforms with more computational capabilities, and the implementation of Machine Learning algorithms for predictive maintenance based on the collected sensor data.

6 Acknowledgement

This work is funded by the Portuguese programme PORTUGAL 2020, under grant agreement n. º 024541 (PRODUTECH SIF - Soluções para a Indústria do Futuro).

7 References

[1] Rubio, E.M., Dionísio, R.P., & Torres P.M.B. (2019) -Industrial IoT devices and cyber-physical production systems: review and use case. In: Machado, J., Soares, F., Veiga, G. (eds) - Innovation, Engineering and Entrepreneurship. HELIX 2018. Lecture Notes in Electrical Engineering. Cham: Springer. ISBN: 978-3-319-91334-6. Vol. 505, p. 292-298 https://doi.org/10.1007/978-3-319-91334-6_40

[2] Rubio, E. M., Torres, P. M., & Dionísio, R. P. (2019) - Smart gateways for IOT-Factory integration: trends and use case. In FERREIRA, L. [et al.], ed - Technological Developments in Industry 4.0 for Business Applications. Hershey, PA: IGI Global. ISBN 9781522549369.p. 149-170. https://doi.org/10.4018/978-1-5225-4936-9.ch007

[3] Neto, L., Gonçalves, G., Torres, P., Dionísio, R., & Malhão, S. (2019). "An Industry 4.0 Self Description Information Model for Software Components Contained in the Administration Shell", 8th International Conference on Intelligent Systems and Applications (ITELLI 2019), June 30 – July 4 2019, Rome, Italy, pp. 24-29.

[4] MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. Retrieved from: http://docs.oasis-open.org

[5] W. Mahnke, S. H. Leitner, and M. Damm. 2009. OPC Unified Architecture (1st ed.). Springer Publishing Company, Incorporated. https://doi.org/10.1007/978-3-540-68899-0

[6] J. D. Contreras, J. I. Garcia, and J. D. Pastrana. Developing of industry 4.0 applications. International Journal of Online Engineering, 2017, 13(10), 30–47. https://doi.org/10.3991/ijoe.v13i10.733

[7] Sahidi T T, Basuki A, Tolle H. MIOT Framework, General Purpose Internet of Things Gateway using Smartphone. International Journal of Online Engineering, 2018, 14(2):6. https://doi.org/10.3991/ijoe.v14i02.7326

[8] M. Schleipen (2008). "OPC UA supporting the automated engineering of production monitoring and control systems", 12th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008), September 15-18 2008, Hamburg, Germany, pp. 640-647. https://doi.org/10.1109/etfa.2008.4638464

[9] Unified Automation [Computer Software]. (2018). Retreived from: https://www.unified-automation.com/

[10] PROSYS [Computer software]. (2018). Retrieved from: https://www.prosysis.com

[11] R. Krishnamurthi, "Teaching Methodology for IoT Workshop Course Using Node-RED," *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Noida, 2018, pp. 1-3. 0.1109/IC3.2018.8530664 https://doi.org/10.1109/ic3.2018.8530664

[12] Z. Chaczko and R. Braun, "Learning data engineering: Creating IoT apps using the node-RED and the RPI technologies," *2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET)*, Ohrid, 2017, pp. 1-8. https://doi.org/10.1109/ithet.2017.8067827

8 Authors

Rogério Dionísio (corresponding author) is an adjunct professor of the School of Technology in the Polytechnic Institute of Castelo Branco, Portugal. He has received a M.Eng. and a Ph.D. in electronic and telecommunication engineering from the University of Aveiro, Portugal. His current research interests include wireless sensor networks, smart agriculture technologies and Industrial IoT applications.

Sérgio Malhão receives a B.Sc. from the School of Technology in the Polytechnic Institute of Castelo Branco, Portugal. His current research interests include cyber-physical systems, industrial communication protocols and electronics.

Pedro Torres is an adjunct professor of the School of Technology in the Polytechnic Institute of Castelo Branco, Portugal. He has received a Ph.D. in mechanical engineering from the University of Lisbon, Portugal. His current research interests include cyber-physical systems, robotics, machine learning and automation.

Article submitted 2019-10-14. Resubmitted 2019-12-26. Final acceptance 2019-12-28. Final version published as submitted by the authors