Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 55 (2021) 72–79

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
7-10 September 2021, Athens, Greece.

# Simulation and Control of a Cyber-Physical System under IEC 61499 Standard

Adriano A. Santos*, António Ferreira da Silva

*CIDEM – Centre for Research & Development in Mechanical Engineering, Mechanical Engineering Department*
*School of Engineering of Porto (ISEP), Polytechnic of Porto, 4249-015 Porto, Portugal*
*INEGI – Institute of Science and Innovation in Mechanical Engineering and Industrial Engineering,*
*Rua Dr. Roberto Frias, 400, 4200-465, Porto, Portugal*

* Corresponding author. *E-mail address:* ads@isep.ipp.pt

**Abstract**

IEC 61499 standard provides an architecture for control systems using function blocks (FB), languages, and semantics. These devices can be interconnected and communicate with each other. Each device contains several resources and algorithms with a communication FB at the end, which can be created, configured, and deleted without affecting other resources. Physical element can be represented by a FB that encapsulates the functionality (data/events, process, return data/events) in a single module that can be reused and combined. This work presents a simplified implementation of a modular control system using a low-cost device. In the prototyping of the application, we use 4diac to control, model and validate the implementation of the system on a programmable logic controller. It is proved that this approach can be used to model and simulate a cyber-physical system as a single element or in a networked combination. The control models provide a reusable FB design.



*Keywords:* Cyber-Physical Systems (CPS); IEC 61499; Industry 4.0; Low-cost devices; Process control; Reconfiguration

## 1. Introduction

Currently, at the industrial application level, the process automation control systems are, majority, carried out based on Programable Logic Controllers (PLCs). These are programmed in a traditional Ladder languages [1] or in any of all the 5 distinct languages of IEC 61131-3 [2] focused on standalone PLC application, i.e., a centralized application running on a single PLC [3]. However, current industrial physical process require interconnection between machines, communication machine-to-machine (M2M) [4], to coordinated actions, transfer data and/or events between itself. These communications are usually carried out by a dedicated line, or by a data network of physical devices – considered the first application domains of Internet of Things (IoT) [5] and Industrial Internet of Things (IIoT) [6] – or base on wireless data transmission (Bluetooth, WiFi, RFID) [4].

The latest standard for automation is the IEC 61499 Function Blocks [7], an open standard, provides a robust, model-driven engineering approach to model centralized and/or distributed system applications, i.e., design automation controllers [8]. This standard was proposed by the International Electrotechnical Commission (IEC) to extend the functionality of FBs, already used in IEC 61131-3, for programming PLC systems. The standard defines a Function Block (FB) as reusable software components, for distributed Industrial Process Measurement and Control Systems (IPMCS) [9], used to encapsulated all the functionalities of mechatronics units [10]. FBs, executed by events-driven (asynchronous when using Publish/Subscribe or synchronous when using Client/Server approach) [11], are defined by input


10.1016/j.promfg.2021.10.011

and output interfaces associated with a head and a body that contains the algorithms.

They can interact with each other through specific communication interfaces FBs based on UDP or TCP protocol [11]. For this reason, IEC 61499 can be understood as an excellent architecture for the development of Cyber-Physical Systems (CPS) given its modular structure [10] as well as an excellent platform for emulating plant specifications [12].

CPS it is an emerging paradigm for networked and distributed control system (NCS/DCS – Network Control Sytem/Distributed Control System) [13], such as the IEC 61499 standard, and it is considered one of the technologies that integrated Industry 4.0 [14]. CPSs or Cyber-Physical Production Systems (CPPS) are composed of control devices with high processing and communication capabilities, either local or remote, i.e., they are a heterogeneous distributed real-time system [15]. So, due essentially to its specific characteristics, one of the standards that can fit into the development of CPSs applications is the IEC 61499 standard [16]. On the other hand, since it is a modular architecture (FB), communicating with each other through standard Ethernet (when implemented in a low-cost device) and also through industrial Ethernet, more robust [17], it fulfills one of the main requirements of CPSs. These systems combine cybernetic and physical resources to solve problems that could not be solved isolated by one of the parts [18].

This article is structured as follows: Section II presents an overview of the IEC 61499 standard. Section III illustrates the methodologies used in the modulation and simulation. This also describes the hardware and software units to archiving the implementation in a PLC and a system control base in a Low-Cost Devices (LCDs), with IEC 61499 standard. Section IV present the case study and control models developed and its implementation. Section V concludes the paper with a brief summary of the work developed and proposed future developments.

## 2. IEC 61499 Standard

IEC 61499 is the new graphic architecture developed for the configuration of distributed industrial systems - Process Measurement and Control Systems (IPMCSs), presenting itself as an alternative to the IEC 61131-3 standard.

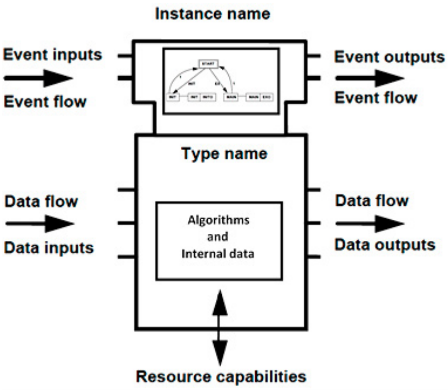


Fig. 1. Basic characteristics of FBs (adapted from [7]).

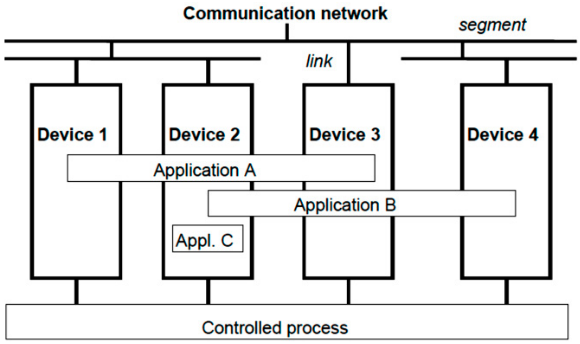


Fig. 2. Applications distributed among several computing devices [7].

The standard defines the terms of syntax implementation of textual models or graphical representation based on FBs characterized by their inputs and outputs. FBs are encapsulations of control algorithms that are executed according to the invocation of the Execution Control Chart (ECC) based on triggering the input event (Fig. 1).

FB is the base of the standard, a single unit, which encapsulate algorithms working with an event and/or data inputs/output. In this sense, a distributed control application consists of a set of FBs that interact with each other transferring data through the propagation of events, distributed over several nodes connected by a FB network.

Developer creates applications based on Basic FBs (BFBs, primary and elementary model), Composite FBs (CFBs, encapsulation of one or more connected FBs) and Service Interface FBs (SIFBs, communications and management FB), distributing it to one or more devices interconnected by the communication SIFBs (Fig. 2). The device model contains at least one interface (communication or process interface) and one or more functional units, resources.

IEC 61499 is an open standard that provide a reference architecture to modeling distributed control systems, by encapsulating FBs information, which aims portability, re-configurability, interoperability, and distribution [19].

## 3. Modulation and Simulation Methodologies

The proposed case study describes a section of many of the parts transport sequences in an industrial process or baggage handling automation system. This work presents the modeling of a prototype, using *Eclipse 4diac*$^{TM}$ [20], as implementation environment. The presented work model is implemented in the Project Kit Conveyor/Buffer development kit (from FESTO), considering the scalability of the system, as well as the ease of development of the model.

The prototyping of the model was performed in C++ language in a FORTE execution environment (Eclipse 4DIAC-RTE), a virtual machine, running in an IEC 61499 environment supported by a network of FBs. A small, low-cost embedded device (Raspberry Pi3 with an ARMv7 processor, running at 1.2 GHz - RPi) is used, connected by a 10/100 Mb local area network (LAN) (RJ45 connector), to test and validate the system control model.

### 3.1. Physical model, conveyor

In Fig. 3 shows a possible model of conveyor used to transfer parts in an industrial plant. The conveyor is considered as mechatronic system that includes conveyor belt, sensors – used to control the input and output – and actuators. Conveyor moves the parts, according to the activation of the photo-sensors and their position, based on the *Boolean* motor control signals (run or stop signs). So, the controller of the low-level mechanical features of the conveyor must deal, not only with linear transport (from *S_In* to *S_Out*), but also with situations of deviation (from *S_In* to *Deviation*) or reception of parts.

However, a more complex transport system will be an aggregation of multiple models of interconnected conveyors controlled by a centralized database using PLCs (traditional approach) or by a distributed system composed of reusable software models (CFB) interconnected by a network (ideal environment for the use of IEC 61499). In this context, the operations of the transport system will be as follows:

- First parts arrive randomly at the conveyor entrance activating the entrance sensor (S_In).
- Motor is activated immediately, due to the digital signal processing of the *S_In* sensor, remaining ON (motor *Set*) until the parts leaves the conveyor.
- Sensor output (S_Out) deactivates the motor, based on the negative flank of the digital output signal (motor *Reset*).

Note that, in a more complex transport system, the previous conveyor sends information about the part transported to the next conveyor. However, in the case study we present, we intend that the transport process becomes as simple as possible, so that the model will transport only a single piece in each activation. The system will remain stopped (put out the part in transport) until a new part arrives.

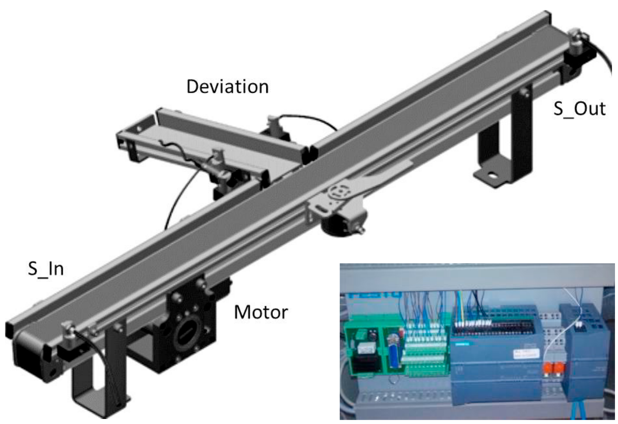


Fig. 3. Project kit Conveyor/Buffer, FESTO [21].

### 3.2. Modelling plant

Plant modeling consists of an accurate representation of the formal structure of the continuous dynamics of the real system. So, the model is used to simulate physical behavior of the system in detriment of experiments that may be carried out on the real plant. In this sense, the modeling of the continuous dynamics of the physical system under study should be developed considering the requirements of the IEC 61499 architecture, that is, considering its event driven execution. Therefore, the information as well the dynamics of the application must be modeled for Verification & Validation using formal model such a Discrete Time/Event approach (DT/DE) Timed Net Condition, Event System, Finite-Stat Machine, Timed Automata [15], Petri nets [22], or others. However, these models can present major disadvantages, since significant changes to the plant model, as well as to the controller, may be necessary.

In fact, to deal with the real problems of systems modeling, the DT or DE approach is more appropriate [23] so, the modeling plant can be performed according to a monolithic or modular approach. These two main approaches can be used interchangeably in system modeling, as they can concatenate the entire plant model in a single block (monolithic approach, used in small systems) or in a modular way where the physical plant is followed in several parts that complement each other. However, both approaches can be represented as a single block (CFB – a set of interconnected modules) that, in an IEC 61499 context, can be reused as many times as necessary (e.g., modelling complex conveyor system).

### 3.3. Modelling controller

The model of the system controller is the representation of the expected behavior of the system. The controller interacts with the process (plant) by sending and receiving signals (inputs/outputs), confirming the fundamental actions for the evolution of the operating logic of both models, IO interaction. On the other hand, we will have to consider that it will be from these interactions that the systems can be simulated and tested for reliability and robustness, ensuring the requirements for reliability, availability, and maintenance of the CPS/CPPS control.

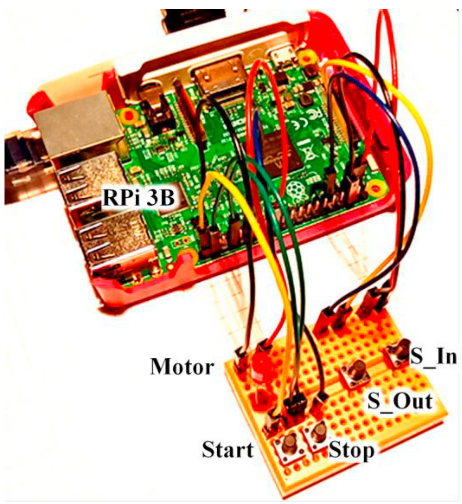


Fig. 4. Simulation of physical behavior of the system.

In this paper, the analysis of the CPS simulation will focus on the use of different simulations that combine real (Fig. 3, conveyor model) and virtual (Fig. 4, breadboard simulation) techniques, implementation of several control models. Control is performed in a real controller, interacting with a representation of the plant model (Laboratory-Test – LT) on the workbench or/and in a virtual simulation (Simulation-in-Loop – SiL).

The model of the real plant is controlled and implemented using the SIMATIC TIA Portal S7-1200 software from Siemens. It is located at the Industrial Automation and Robotics Laboratory (LARI) of the Mechanical Engineering Department of the Polytechnic of Porto, in Portugal. Thus, to carry out this analysis (LT and SiL) it is intended to implement the control specifications using different software tools, basing its development on Sequential Function Charts (SFC) [24], and their interactions.

All the steps of plant model are associated with actions interconnected with transition and receptivity's, oriented links connecting actions following the rules given in [24]. Considering this formalism, the global model is converted to Ladder language (LD) [2] in order to simulated process on the physical workbench (Fig. 3). Our intention, in the first stage, was to simulate the process using a physical model controlled by a real automaton to observe the behavior of the system process. Thus, to choose the most appropriate approach for the plant model, we chose to divide the model into two parts: i) representation of the control of activation and deactivation of the system and ii) representation of the control of the actuators, in this case the motor. The system is modeled based on TIA Portal V16 software using LD language, on a Programmable Logic Controller (PLC) SIMATIC S7-1200 CPU 1214C AC/DC/Rly, and a console for Human-Machine Interface (HMI) SIMATIC HMI KTP700 Basic from Siemens.

The virtual platform is developed using an RPi and a breadboard based on the same formalism used in the global model. Motor is simulated by a red LED, sensors (S_In and S_Out) and START/STOP system are simulated by a four mini push buttons (Fig. 4). The purpose of implementing breadboard using IEC 61499 is, first, to simulate and observe the behavior of the program to ensure that there is no unexpected behavior and, secund, to interact with the physical system and observe the behavior of the system. This implementation provides us with a virtual test environment that allows us to implement and test, on an unconventional PLC, the control developed, simulating, and verifying, simultaneously, the real behavior of the system mounted on the workbench. However, this independence of the physical system and the PLC used in the conveyor control requires the development of a new IEC 61499 controller, with or without temporal processing, equally representative of the plant's behavior. This may be a disadvantage since a controller with the same requirements will have to be developed.

The main idea presented in this paper is to present some approaches to simulate, test and validate PLC programs. A workbench is used to test and validate de LD language (LT approach) and the RPi is used to interacting and simulate, in a virtual environment (SiL approach) under the IEC 61499, the same code using FBs.

## 4. Case Study

Figure 3 show the station (conveyor) for transporting parts between two stations. Thus, it is intent to prepare the respective SFC command based on the required operation described as follows:

The parts arrive from a distribution station to the left of the conveyor. Sensor S_In detect part and put on de conveyor belt. On the right (output) the S_Out sensor stops the conveyor (the sensor acts in the negative edge of the digital signal, transition from 1 to 0) and the Start/Stop system is performed by two push-button. For simplicity, a single piece is carried between the conveyor input and the output system. The START button activates the system but will remain stopped until a single piece arrives.

TABLE I. DESCRIPTION OF THE DIGITAL INPUTS

| Inputs | Description | PLC address |
|---|---|---|
| START | Start system button | %I0.0 |
| STOP | Stop system button | %I0.1 |
| S_In | Input sensor | %I0.2 |
| S_Out | Output sensor | %I0.3 |

The STOP button deactivates the system and making it inoperative, the system will remain stopped even if a part is present at the entrance. PLC address for the digital inputs is presented in Table 1 and outputs signal in Table 2.

TABLE II. DESCRIPTION OF THE DIGITAL OUTPUTS

| Outputs | Description | PLC address |
|---|---|---|
| MOTOR | Motor of conveyor belt | %Q0.0 |

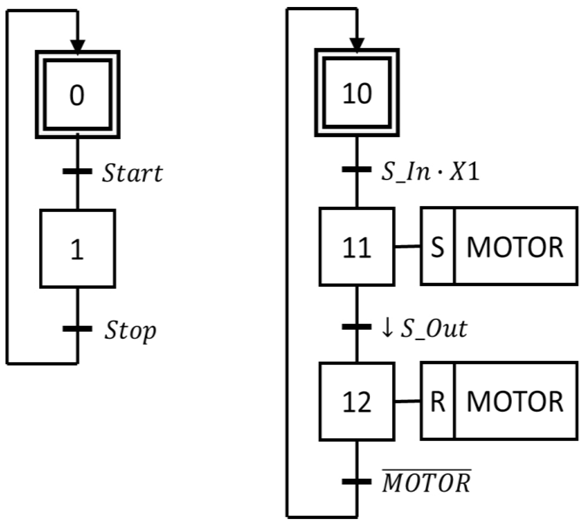


Fig. 5. SFC control model of the system.

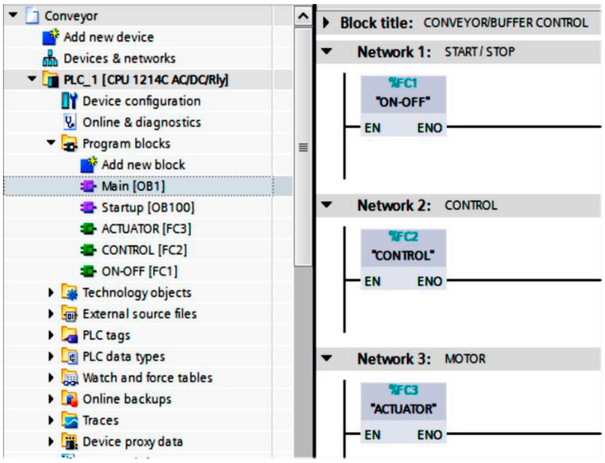


Fig. 6. Encapsulated LD control system.

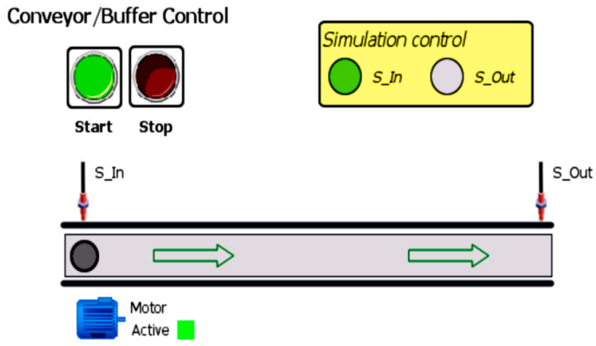


Fig. 7. HMI control system.

### *4.1. LT simulation, physical platform*

In order to simulating system control using Laboratory-Test simulation we use a modular approach to develop control program. Thus, to perform this we divide control system in to two parts representing the control buttons of the Star/Stop system and the motor control as show in Fig. 5. Laboratory-Test implementation is made using SIMATC TIA Portal software, running in a CPU 1214C AC/DC/Rly, where the visualization of the evolution of the system was performed using SIMATIC WinCC software running on a KTP700 Basic touch console.

All components of the plant, modelling in a Ladder language, it was divided into three main blocks that represent Start/Stop system buttons (block Function “ON-OFF [FC1]”), system control (block Function “CONTROL [FC2]”) that control all the system, and motor control of the conveyor belt (block Function “ACTUATOR [FC3]”), see Fig. 6. The Main Function, Organization Block [OB1], and the Startup Block [OB100] form the interface between the command operating system and the application program, activation of the initial steps of the SFC. These OBs are called by the operating system to control the cyclic execution and command initialization. FC are Functions that encapsulate the control system model.

All these FBs can be reused or retrofitted for a new application control. The control model used in the workbench implementation is presented in Fig. 5. It is one of the several possible control solutions of the Conveyor/Buffer station.

The control model has been translated to LD language. Simulation has performed at the physical platform (LT simulation) with SIMATIC TIA Portal V16 program combined way WinCC using a SIMATIC HMI KTP700 Basic from Siemens.

Fig 7 presented the HMI interface of the system model for the workbench developed in IEC 61131. Thus, when arrives a part, and it is detected by sensor S_In, the conveyor belt moves to the right until the part is detected by the sensor S_Out. Conveyor belt, operated by a 24V DC motor, stops when the negative edge of the digital signal from the sensor transiting between the values 1 and 0.

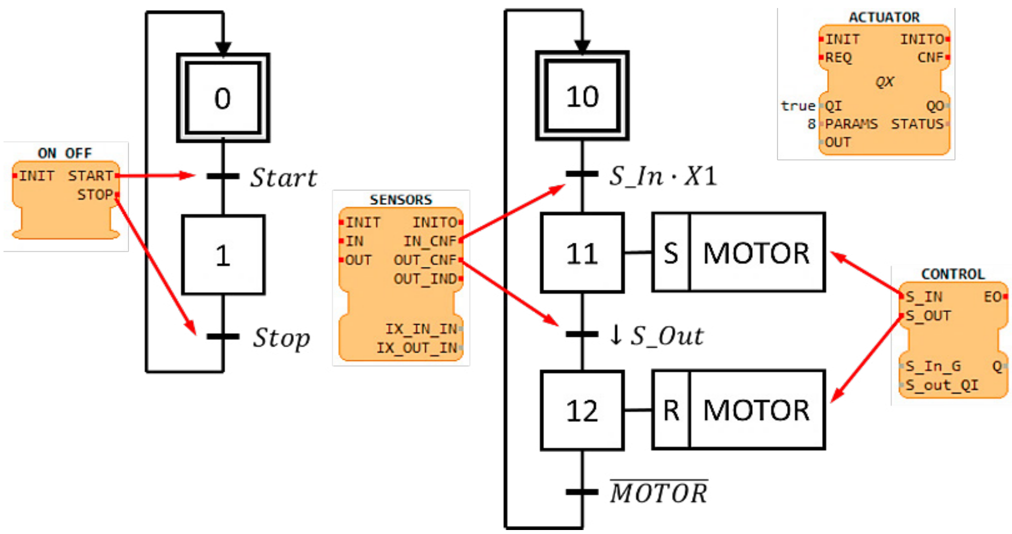


Fig. 8. Associated FBs interface to the SFC control model.

### *4.2. SiL simulation, virtual platform*

The virtual platform for simulation (SiL) was developed using the integrated development environment tools (IDE) of 4DIAC-IDE and the SFC control model presented in Fig. 5 to the IEC 61499 environment. So, after the development phase of the control model, using IDE, the encapsulated group of FBs (implementation of the I/Os functionalities, inputs and outputs events and data) necessary to perform services, must be deployed and executed in the RPi board.

However, to have access to the hardware of the RPi board it is necessary to have *root* rights [25]. The RPi's hardware access control model needs to be configured during the boot phase in order to get access to the GPIO pins. A possible configuration code of the GPIO pins is shown in the lines of code shown below:

```
gpio -g mode 8 out    #Motor output
gpio -g mode 2 up     #S_In sensor input
gpio -g mode 3 up     #S_Out sensor input
gpio -g mode 20 up    #Start button input
gpio -g mode 21 up    #Stop button input
```

The IDE implementation in *Eclipse 4diac*TM also has a modular approach, following the same philosophy of the implemented LD language so, it will be possible to implement similar modules to those implemented in the SIMATIC TIA Portal, IEC 61131 environment. Thus, as shown in Fig. 8, a case study of modeling control system is simulated, to validate the control information, using the FBs proposed in IEC 61499. It is an approach, based on object-oriented programming based on events trigger connected by a network.

Each part of the conveyor (ON-OFF, CONTROL, ACTUATOR, output to motor control, and Sensor FB) are modelled, as a network, inside an individual sub-application FBs (see Fig. 8 and Fig. 9). These FBs are connected according to their natural working connection and control model according SFC model presented in Fig. 5.

The configuration shown in Fig. 9 is the basis for reusing a conveyor control system. This approach can be replicated and used on many other conveyors however, it can only be reused with a few minor adjustments to its operation, CFB encapsulation. The same principle can be applied to the simulation and validation of a broader set of conveyors. Each conveyor control model has two *REQ* (corresponding to S_IN and S_OUT) event inputs for the proximity sensor, associated with GPIO pins 2 and 3, to detecting parts on the conveyor belt and a *E0*, event output, to control start/stop motor, GPIO control output pin 8.

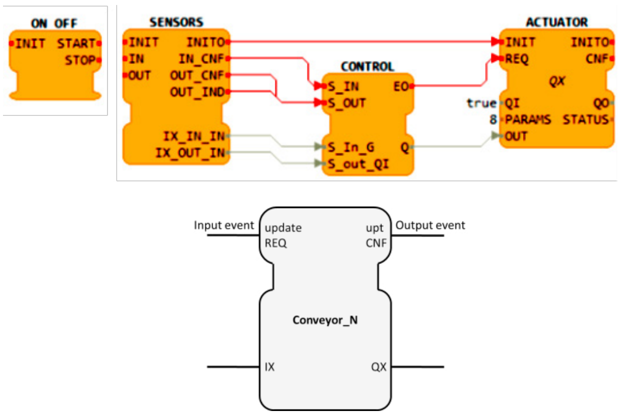


Fig. 9. IEC 61499 control model using 4diac, CFB encapsulation.

As shown in Fig. 10, the IDE of 4diac is used to verify and validate the evolution of the system. Each instance of the FB can be monitored, and all input/output events (GPIO pins) and data can be activated by pressing the simulation buttons, enabling, and disabling the workbench implementation, interaction with IEC 61131. States START and STOP initialize or stopped an "E_CYCLE" FB that control input

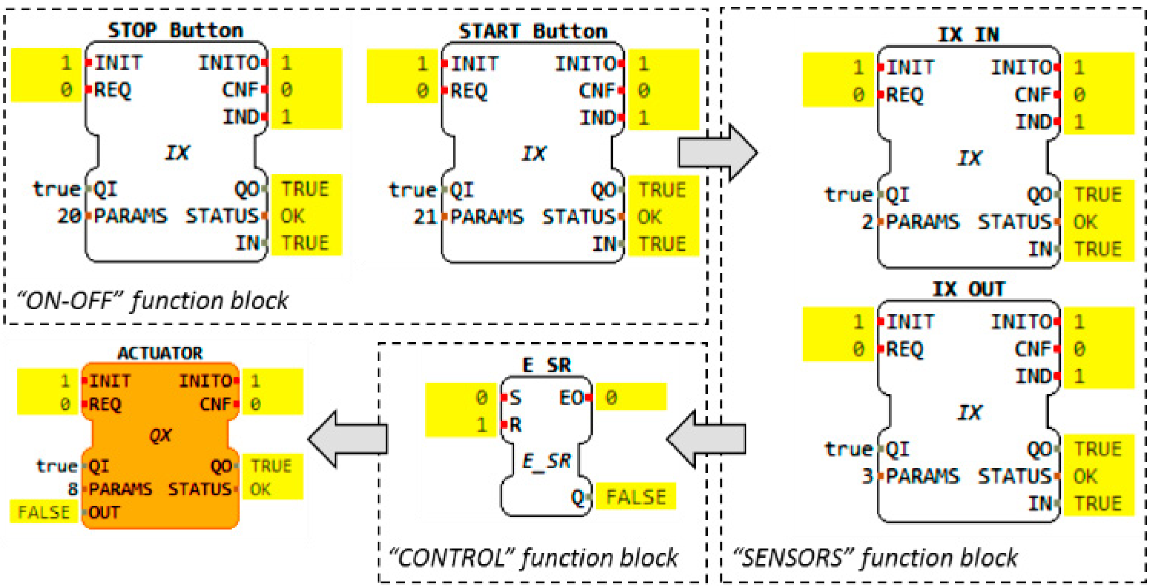


Fig. 10. Verification and validation of simulation states using 4diac.

sensors (S_In and S_Out) and, consequently, starts and stops conveyor belt FB “ACTUATOR” (Raspberry Pi GPIO pin 8 out, *QX*). The Boolean data IN equal a *TRUE* indicates that the sensors or buttons are not activated. On the other hand, in the ACTUATOR, data OUT equal a *FALSE*, indicates that motor is stopped. Data IX_IN equal to *FALSE* and QX_OUT equal to *TRUE* correspond to the activation of GPIO input of function block IX and the activation of GPIO output of function block QX. The physical part of the cyber-physical system, LED equal to ON or OFF, indicates the state of the conveyor motor (see Fig. 4).

The interaction between these models allows the visualization of the evolution of the physical system through simultaneous monitoring, in an IEC 61131 (HMI/PLC) and IEC 61499 (4DIAC-IDE) environment.

## 5. Conclusions and future work

This paper presents a case study for a modular CPS composed of an individual cyber-physical conveyor. The cyber-physical conveyor is composed by a conveyor belt, physical part, and a logical control part, implemented through a Siemens SIMATIC TIA Portal S7-1200 PLC and a Raspberry Pi board. This paper illustrates an approach to using RPi’s boards as a low-cost CPS device implementation and is also another possible starting point for comparing the IEC 61131 implementation with IEC 61499 approach. On the other hand, the feasibility of implementing a cyber-physical methodology for the control, implementation and visualization of projects using low-cost devices (for example, Raspberry Pi) under IEC 61499 standard is confirmed. This feasibility of implementation translates, essentially, in the ease of reusing the model of computational (MoC) component (FB as a software element) as a representation of the physical element (conveyor). This ease of implementation of this type of systems is an important aid in the adoption of CPS concepts under the Industry 4.0 paradigm.

So, this paper is one of several possible approaches for modelling cyber-physical system. This is a simple example of the potentialities of applying the CPS to the industrial environment considering a global model of the plant and the control. The simulation and validation of the CPS using LT and SiL was obtained using a workbench and a virtual platform (physical conveyor by FESTO and a Raspberry Pi whit a breadboard, under IEC 61499) respectively. It can be concluded that the use of the proposed methodology (using a low-cost device and IEC 61499 under *Eclipse 4diac* software), for modeling CPS, can be used. On the other hand, it is shown that the major advantage of this approach lies in the fact that the models of the plant and the controller do not need to be modified, since they interact with each other, through the inputs and outputs of the PLC and the RPi.

In the future we will develop a more complex system, with more parts and interaction with other stations, allowing us to apply modeling and simulation to CPS in a distributed environment.

## CRediT author statement

Adriano A. Santos: Conceptualization, Methodology, Software; Writing- Original draft. António Ferreira da Silva: Software, Validation, Reviewing and Editing.

## Acknowledgements

We acknowledge the financial support of CIDEM, R&D unit funded by FCT – Portuguese Foundation for the Development of Science and Technology, Ministry of Science, Technology and Higher Education, under the Project UID/EMS/0615/2019, and it was supported by FCT, through INEGI and LAETA, under project UIDB/50022/2020.

## References

[1] W. Dai, V. N. Dubinin and V. Vyatkin, "Migration From PLC to IEC 61499 Using Semantic Web Technologies," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 3, pp. 277-291, March 2014. http://doi.org/10.1109/TSMCC.2013.2264671.

[2] International Electrotechnical Commission, “International Standard IEC 61131-3, *Programmable Logic Controllers - Part 3*: *Programming Languages*”, 2013. https://webstore.iec.ch/publication/4552.

[3] A. A. Santos and M. de Sousa, "Replication Strategies for Distributed IEC 61499 Applications," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, 2018, pp. 2225-2230. http://doi.org/10.1109/IECON.2018.8592737.

[4] P.K. Verma *et al.*, “Machine-to-Machine (M2M) communications: A survey”, in *Journal of Network and Computer Applications,* vol. 66, May 2016, Pages 83-105. https://doi.org/10.1016/j.jnca.2016.02.016.

[5] G. I. Kurcheeva, G. A. Klochkov and S. S. Firsova, "Machine-to-Machine Communication: Development of a Subscriber Tracking System for a Given Area," in *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, Novosibirsk, 2018, pp. 465-468. http:/doi.org/10.1109/APEIE.2018.8545867.

[6] Hugh Boyes, Bil Hallaq, Joe Cunningham, Tim Watson, “The industrial internet of things (IIoT): An analysis framework,” in *Computers in Industry*, vol. 101, 2018, pp. 1-12, ISSN 0166-3615. https://doi.org/10.1016/j.compind.2018.04.015.

[7] International Electrotechnical Commission, “International Standard IEC 61499. *Function Blocks-Part 1, Architecture*”, 2012. https://webstore.iec.ch/publication/5506.

[8] V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review," in *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, 2011, pp. 768-781, doi: 10.1109/TII.2011.2166785.

[9] Zoitl A, Lewis R., Modelling Control Systems Using IEC 61499, 2nd Edition. Institution of Engineering and Technology, 2014. https://digital-library.theiet.org/content/books/ce/pbce095e.

[10] J. Yan and V. V. Vyatkin, "Distributed execution and cyber-physical design of Baggage Handling automation with IEC 61499," in *2011 9th IEEE International Conference on Industrial Informatics*, Caparica, Lisbon, 2011, pp. 573-578. http://doi.org/10.1109/INDIN.2011.6034942.

[11] A. Veichtlbauer, M. Parfant, O. Langthaler, F. P. Andrén and T. Strasser, "Evaluating XMPP communication in IEC 61499-based distributed energy applications," *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, 2016, pp. 1-8, doi: 10.1109/ETFA.2016.7733744.

[12] A. Malik, P. S. Roop, N. Allen and T. Steger, "Emulation of Cyber-Physical Systems Using IEC-61499," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, 2018, pp. 380-389, doi: 10.1109/TII.2017.2724206.

[13] S. Yoon, J. Lee, Y. Kim, S. Kim and H. Lim, "Fast Controller Switching for Fault-Tolerant Cyber-Physical Systems on Software-Defined Networks," in *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)*, Christchurch, 2017, pp. 211-212. http://doi.org/10.1109/PRDC.2017.35.

[14] Jay Lee, Behrad Bagheri, Hung-An Kao, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems,” in *Manufacturing Letters*, vol. 3, 2015, pp. 18-23, https://doi.org/10.1016/j.mfglet.2014.12.001.

[15] Nuno Canadas, José Machado, Filomena Soares, Carlos Barros, Leonilde Varela, “Simulation of cyber physical systems behaviour using timed plant models,” in *Mechatronics*, vol. 54, 2018, pp. 175-185. https://doi.org/10.1016/j.mechatronics.2017.10.009.

[16] García, C.A.; Castellanos, E.X.; García, M.V, “UML-Based Cyber-Physical Production Systems on Low-Cost Devices under IEC-61499,” in *Machines (MDPI AG)*, vol. 6, no. 2:22, 2018. https://doi.org/10.3390/machines6020022.

[17] P. Lindgren, J. Eriksson, M. Lindner, A. Lindner, D. Pereira and L. M. Pinho, "End-to-End Response Time of IEC 61499 Distributed Applications Over Switched Ethernet," in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 287-297, Feb. 2017. http://doi.org/0.1109/TII.2016.2626463.

[18] André Platzer, Logical Foundations of Cyber-Physical Systems. Springer, 2018. https://doi.org/10.1007/978-3-319-63588-0.

[19] C. Sunder *et al.*, "Usability and Interoperability of IEC 61499 based distributed automation systems," in *2006 4th IEEE International Conference on Industrial Informatics*, Singapore, 2006, pp. 31-37. http://doi.org/10.1109/INDIN.2006.275713.

[20] Framework for Distributed Industrial Automation and Control (4DIAC). (Accessed on 15 September 2020). Available online at: https://www.eclipse.org/4diac/.

[21] FESTO-Project kit Conveyor/Buffer. (Accessed on 3 October 2020). Available Online at: https://www.festo-didactic.com/int-en/services/printed-media/technical-documentation/project-kit-conveyor-buffer.htm.

[22] Radek Kočí and Vladimír Janoušek, “Validation of Specification Models Based on Petri Nets,” in *The Twelfth International Conference on Software Engineering Advances (ICSEA)*, Athens, 2017, pp. 232-237. https://www.thinkmind.org/articles/icsea_2017_11_10_10157.pdf.

[23] G. Kunz, J. Machado, E. Perondi and V. Vyatkin, "A Formal Methodology for Accomplishing IEC 61850 Real-Time Communication Requirements," in *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6582-6590, Aug. 2017. https://doi.org/10.1109/TIE.2017.2682042.

[24] European Standard IEC 60848. GRAFCET specification language for sequential function charts. Edition 3.0, IEC, 2013. https://webstore.iec.ch/publication/3684

[25] Barbosa J., Leitão P., Teixeira J., “Empowering a Cyber-Physical System for a Modular Conveyor System with Self-organization. In: Borangiu T., Trentesaux D., Thomas A., Cardin O. (eds) Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence, vol 762. Springer, Cham., 2018. https://doi.org/10.1007/978-3-319-73751-5_12.