

Article

Open Source Control Device for Industry 4.0 Based on RAMI 4.0

Pablo F. S. Melo ^{1,*}, Eduardo P. Godoy ^{1,*}, Paolo Ferrari ² and Emiliano Sisinni ²

¹ Department of Control and Automation Engineering, São Paulo State University (Unesp), Sorocaba, SP 18087-632, Brazil; pablo.melo@unesp.br

² Department of Information Engineering, University of Brescia, 25123 Brescia, Italy; paolo.ferrari@unibs.it (P.F.); emiliano.sisinni@unibs.it (E.S.)

* Correspondence: eduardo.godoy@unesp.br

Abstract: The technical innovation of the fourth industrial revolution (Industry 4.0—I4.0) is based on the following respective conditions: horizontal and vertical integration of manufacturing systems, decentralization of computing resources and continuous digital engineering throughout the product life cycle. The reference architecture model for Industry 4.0 (RAMI 4.0) is a common model for systematizing, structuring and mapping the complex relationships and functionalities required in I4.0 applications. Despite its adoption in I4.0 projects, RAMI 4.0 is an abstract model, not an implementation guide, which hinders its current adoption and full deployment. As a result, many papers have recently studied the interactions required among the elements distributed along the three axes of RAMI 4.0 to develop a solution compatible with the model. This paper investigates RAMI 4.0 and describes our proposal for the development of an open-source control device for I4.0 applications. The control device is one of the elements in the hierarchy-level axis of RAMI 4.0. Its main contribution is the integration of open-source solutions of hardware, software, communication and programming, covering the relationships among three layers of RAMI 4.0 (assets, integration and communication). The implementation of a proof of concept of the control device is discussed. Experiments in an I4.0 scenario were used to validate the operation of the control device and demonstrated its effectiveness and robustness without interruption, failure or communication problems during the experiments.



Citation: Melo, P.F.S.; Godoy, E.P.; Ferrari, P.; Sisinni, E. Open Source Control Device for Industry 4.0 Based on RAMI 4.0. *Electronics* **2021**, *10*, 869. <https://doi.org/10.3390/electronics10070869>

Academic Editor: Hamid Reza Karimi, Cheng Siong Chin, Kalyana C. Veluvolu, Valeri Mladenov, Cecilio Angulo, Davide Astolfi, Jun Yang and Len Gelman

Received: 5 March 2021

Accepted: 5 April 2021

Published: 6 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Industry 4.0; RAMI 4.0; control device; OPC UA; Raspberry Pi; open-source

1. Introduction

The term Industry 4.0 (I4.0) or the Fourth Industrial Revolution represents the next step in the organization and control of the flow of value over the life cycle of a product [1]. The concept of I4.0 was presented as a strategy by the German government to develop cutting-edge technologies to revolutionize the organization of global value chains [2]. The I4.0 stands out for presenting the horizontal/vertical integration of manufacturing systems, digital engineering throughout the product life cycle and the decentralization of computational resources [3]. A common model within a reference architecture is necessary to support and organize the implementation of I4.0 [1,4]. Thinking of a systematic and sustainable development process within the industry, it is essential to build system architectures in a set of stable, consensual and standardized reference models [5]. As a result, in April 2015, it was presented at the Hannover Messe fair (Germany), the reference architecture model for Industry 4.0 (RAMI 4.0) [6].

The RAMI 4.0 makes the components inserted in I4.0 compatible in a three-dimensional layered model, in which technologies based on I4.0 can be integrated and developed [7]. The RAMI 4.0 structure is formed by three axes: hierarchy levels, life cycle and value stream and layers. Figure 1 represents RAMI 4.0.

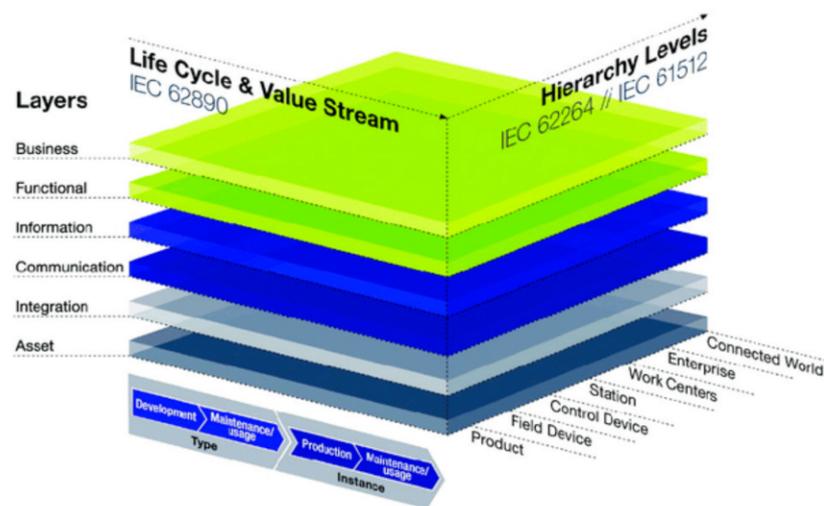


Figure 1. Three-dimensional structure of the reference architecture model for Industry 4.0.

In this way, complex relationships can be divided and developed into smaller and simpler clusters. The big challenge for the adoption of RAMI 4.0 is the implementation of compatible solutions, such as equipment, systems and standards that support the functionalities of each layer and the interactions required among the elements distributed along the three axes of RAMI 4.0 [8]. Reference [8] also emphasizes that Industry 4.0 represents a concept, not a standard, which does not specify integration tools, communication technologies and programming standards to be used. As a result, it is mandatory to integrate different technologies to provide the functionalities required by these new industrial applications, and RAMI 4.0 was defined to help with that. Even though there is much literature dealing with RAMI 4.0 related approaches based on conceptual or simulated frameworks for I4.0 applications, the presentation and discussion of experimental results and proofs-of-concept are scarce.

Currently, solutions readily available on the market covering all the aspects of RAMI 4.0 are very few, if any. For instance, the control device, which is the subject of this paper, is one element of the hierarchy levels of RAMI 4.0, comprising relationships among the following layers: *asset*, *integration* and *communication*. However, despite the idea of implementing a programmable logic controller (PLC) as an I4.0 component can be dated back to 2016 [9], it struggles to emerge out of laboratories. The control device comprises additional requirements that traditional automation controllers (e.g., PLC) cannot yet fulfill [10], such as the support to service-oriented paradigm, the interoperability between heterogeneous systems, the reconfigurability and support to plug-and-play features and the usage of information models to overcome the strict information encapsulation of controllers. The automation companies have been working on compatible RAMI 4.0 solutions, mainly in Europe. Therefore, there are on the market automation hardware, such as PLCs and software, such as Codesys, that could be used in order to develop a RAMI 4.0 compatible control device. However, these are high-cost solutions besides being proprietary and black boxes, which means that no developer customizations are allowed. Open-source technology is lately gaining increasing attention and adoption in industrial applications, such as automation systems and industrial Internet of Things (IIoT) [11–13]. These papers show that open-source hardware and software can provide adequate robustness required in industrial applications, with the advantage of open and customizable solutions.

Therefore, the main contribution of our paper is showing that it is possible to develop a control device for I4.0 compatible with RAMI 4.0 only using open-source solutions of hardware, software, communication and programming. As far as we know from our literature review, this contribution is unprecedented. Differently from previous works, as the aforementioned [9] and [10], authors focus on the exclusive use of open-source solutions for implementing such a device. Other paper original contributions are the

proposal of a specific integration of these technologies, the implementation of a proof-of-concept prototype of the control device and the reporting of experimental results related to RAMI 4.0 developments. Besides the common advantages of open-source solutions (such as reduced cost, enabling the easy update of the overall system and avoiding any constraints imposed by a proprietary one), other features and innovations of the proposed control device can be listed, such as an open and no-cost tool for programming, online and remote network programming using a web browser, built-in wireless communication and integration, open integration of the device with external solutions, and highly customizable information model using the open platform communication unified architecture (OPC UA) address space compatible with RAMI 4.0 administration shell.

This article is organized as follows: Section 2 presents the main concepts and elements of RAMI 4.0 and a literature review about implementations using this model. Section 3 discusses the integration of open-source technologies to compose the control device. Section 4 describes the implementation of a proof of concept of the control device, detailing the operation of the selected technologies. Section 5 brings a survey of the results obtained from tests carried out in a real I4.0 scenario of a manufacturing process. Finally, Section 6 reports the final considerations and future works and improvement opportunities.

2. Background and Current Research about RAMI 4.0

Currently, a large part of human activities is related to different communications systems connected to the Internet. Among such systems, the following stand out: Internet of Services and Internet of Things. In addition to the traditional technological pillars of I4.0, such as the IIoT and cloud computing, the usage of advanced techniques, such as Blockchain [14] and explainable artificial intelligence (AI) [15], have been recently researched. These technologies were widely developed individually and integrated into the context of I4.0. Such integration was structured based on main standards, existing methods and approaches of the industrial scenario; this whole set being modeled in a three-dimensional structure in which each axis represents an individual aspect around I4.0 [16]. Furthermore, given this integration, it is possible for all entities in an I4.0 network to search for mutual communication throughout the life cycle of a product, regardless of any type of border between companies, states and nations [17]. Reference [8] states that all entities involved in the production chain will be able to access all the data necessary for the specific needs of each sector. For example, in the production chain, industrial machine manufacturers and developers will be able to develop their products based on a range of information regarding prior knowledge of the system components.

Through the effects of such comprehensive chains, German companies and institutions, including BITCOM, VDI/VDE, and ZVEI [17], developed and launched RAMI 4.0 [6]. This reference model allows the step-by-step migration from today's world to that of I4.0 and the definition of application domains with special requirements [18,19]. The use of RAMI 4.0 as a project standard in the context of I4.0 results in some benefits. Among these, the following stand out: the use of service-oriented architecture (SOA), the combination of information technology layers representing aspects of the physical and virtual worlds, and the division of all processes and equipment as a component of the production system, which facilitates communication between different business levels and the more robust processing of information contained in the factory environment.

Therefore, recent work focuses on the development of solutions compatible with RAMI 4.0 that use and integrate, based on the available model, different technologies within I4.0 [2]. Reference [20] addresses the steps for the migration of the ICT (information and communication Technology) infrastructure associated with control management to a digitalized industrial system, according to the requirements established by RAMI 4.0. A toolkit classified using RAMI 4.0 and developed to integrate customers in the design and manufacture of products within I4.0 is presented in [21]. In [19], an architecture compatible with RAMI 4.0 for the discovery of equipment and the operation of processes according to the product requirements is developed.

The article in [3] proposes a universal model oriented to applications to promote the easier use of RAMI 4.0 in a practical context, thus improving the applicability of the model. It is worth noting that in [3], the authors bring the topological approach to production in an illustrative way through the use of application-oriented RAMI 4.0. In this article, for example, the control device element has interactions in almost all layers, going from the *asset* layer to the *Functional* layer, not having a direct relationship only with the *business* layer since this layer is focused on business models in the I4.0 scenario. In [22], the authors highlight some stages of a product design and assembly planning in a network company concept and associate these steps for a mapping of RAMI 4.0, using *business* and *information* layers.

RAMI 4.0 recommended the open platform communication unified architecture (OPC UA) protocol [23] for the *communication* layer. Thus, several papers are already focusing their proposal on this protocol [24,25]. A configurable and adaptable human-machine-interface (HMI) for monitoring production plants based on OPC UA and AutomationML is presented in [26]. For this, the authors point out the presence of the HMI element in the *integration* layer, taking into account that the solution employs the use of various technologies and, therefore, occupies a volume in RAMI 4.0. Reference [27] describes some scenarios based on the use of the OPC UA protocol as the communication technology for flexible, transparent and adaptable production when using RAMI 4.0. It provides a list of possible building blocks that can be realized through the OPC UA, in which the *integration* layer is used to separate the structured data from the information model and the database (*information* layer). The *communication* layer is responsible for minimum functionality that a client must be able to access within the I4.0 architecture, and finally, the *information* layer is in charge of defining the graphical information models, visualizing the common information and aggregating the functionalities provided by the OPC UA server.

The fusion of physical and virtual systems is another characteristic in I4.0 applications, such as augmented, virtual and mixed reality [28,29] and digital twin [30]. These applications emphasize the need for a unified and standardized information model, and they can be mapped through RAMI 4.0 layers. Reference [31] describes the use of the asset administration shell in the *integration* layer to structure virtual components in virtual reality applications. The OPC UA address space in the *communication* layer can also be used to provide interoperability for these applications. Reference [30] presents an architecture for digital twin applications and discusses how it is mapped through RAMI 4.0.

According to the literature review, most works currently focus on migrating production systems and retrofitting or adapting equipment to be compatible with RAMI 4.0 model. Although some works implement a subset of RAMI 4.0 layers, our work differs from those and focuses on the development of a control device compatible with RAMI 4.0. This article extends our embryonic proposal for a controller interface based on RAMI 4.0 [8], discussing the development of additional modules and presenting an experimental scenario for validation and performance analysis. In addition, a discussion is presented about the opportunities and proposal to update the control device in an I4.0 (I4.°C) component of RAMI 4.0 model.

RAMI 4.0: Three-Dimensional Model

As previously stated, RAMI 4.0 is divided based on three structural axes: hierarchy levels, life cycle and value stream and layers, as shown in Figure 1. For the sake of completeness, a short description is provided in the following.

Axis of Hierarchy Levels

The axis referring to hierarchical levels concerns the connection of all elements of the productive system (IEC 62264 and IEC 61512), ranging from the product to the connected world. This axis includes information, people and machines as part of the network management that characterizes an I4.0 scenario [4]. The *hierarchy levels* axis is based on seven elements:

Product: that characteristic element or product is manufactured. In addition, it also corresponds to the ease of production and the interdependence between the various elements of the model.;

Field device: represents the functional level of an intelligent device, for example, a smart sensor or actuator. Basically, they are electronic devices used to detect and identify sensor components and technologies [4];

Control device: defined as the brain of industrial control systems and usually represented by programmable devices, such as programmable logic controller (PLC) and distributed control systems (DCS) with enhanced I4.0 functionalities;

station: the place where operators perform administrative activities to examine the entire operation in the industrial environment based on events and processes, using, for example, Supervisory control and data acquisition (SCADA);

Work centers: maintains manufacturing information, defines the state of production and supervises the renewal of raw materials for refined products;

Enterprise: usually defined in terms of enterprise resource planning (ERP) and business management systems, such as, for example, production planning, service provision, marketing, sales, financial modules, retail and other expenses;

Connected world: this component describes the group of plants and the integration with external companies, customers and component suppliers.

Axis of Life Cycle and Value Stream

I4.0 claims a great potential for improvement throughout the life cycle of products, machines and factories. The axis of the *life cycle and value stream* is responsible for organizing and standardizing these relationships. This axis was based on the IEC 62890 standard that takes care of the life cycle management for systems and products used in industrial processes [32]. Within this axis, *types* and *instances* are structured. A *type* is always created with the initial idea. This includes placing design, development and testing orders that range from the first sample to prototype production. The product *type* can be created during this step. Reference [32] informs that the products are manufactured based on this *type* created, and each product can contain an *instance* represented, for example, by a unique serial number.

Axis of Layers

The *layers*, also called information technologies (IT) layers, are related to detailing or mapping a virtual object, which is a machine or even the same process. These layers are representations used in ICT, where the properties of a complex system can be divided into layers to facilitate the implementation and visualization of new features [7]. Reference [4] highlights that this axis is responsible for defining the standards, protocols and interconnectivity for the verticalization of information, ranging from its interfaces, requirements, and accessibility. RAMI 4.0 has six IT layers:

Business: Covers abstract business models and the logic of the process, in addition to ensuring the integrity of functions along the value chain. In addition, this layer is responsible for organizing any exchange of information on industrial processes that have a commercial vision focus. In this way, it becomes possible to map all the political and economic structures of the company for eventual strategic decision making;

Functional: This layer represents the runtime manufacturing system for services and applications. In addition, the functional layer is responsible for the horizontal integration of various functions, such as rule generation, production planning and application logic.

Information: Responsible for preprocessing the events received from the *communication* layer. The information generated by the lower layers is checked for integrity and, subsequently, summarized in a new and higher quality data structure (information model). In this way, a standard interface or information model is provided for the functional and business layers to consume this data.

Communication: This layer is responsible for making the information from the *asset* and *integration* layers available to the upper layers of RAMI 4.0. It is responsible for the

communication between the elements of the I4.0 network based on standardized and uniform communication protocols, such as the OPC UA.

Integration: Responsible for providing information related to physical resources (assets) for the higher layers. This information is provided in a structured form called an administration shell (AS), which organizes the asset information. This layer contains all the elements associated with IT operations (including HMI) and logic control execution. Additionally, [4] states that the *integration* layer contains elements associated with the identification of materials and products, such as radio frequency identification (RFID) or barcode readers;

Asset: The lowest layer of RAMI 4.0. It represents all physical reality, containing objects, which are: processes, sensors, actuators, machines and even technical documentation. In this layer, products and parts are identified in relation to the execution requirements for a given machine and/or workstation.

3. The Proposed Open-Source Control Device Architecture

This paper discusses the development and the implementation of an open-source control device for Industry 4.0 applications compatible with RAMI 4.0. Considering RAMI 4.0, the developed control device, one of the items in the *hierarchy-level* axis, integrates the following elements of the *layers* axis: *asset*, *integration* and *communication*, as shown in Figure 2.

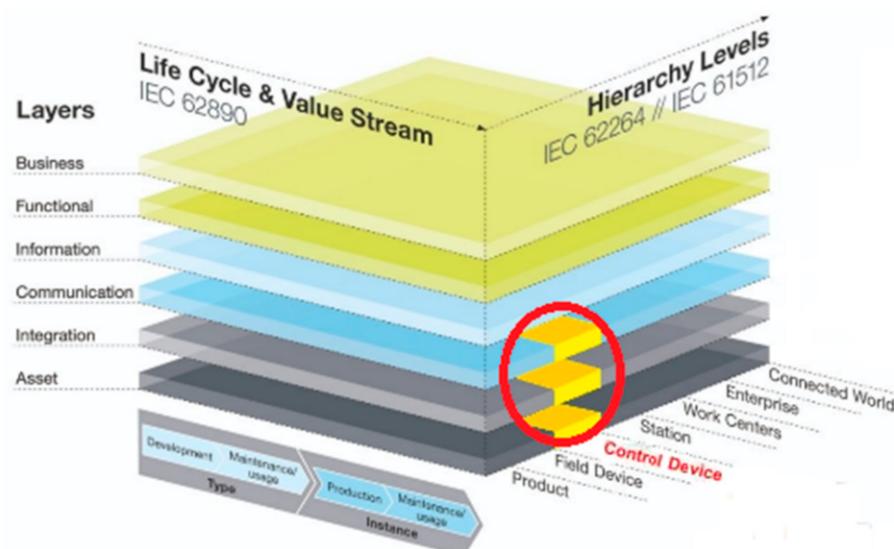


Figure 2. Relationship between the reference architecture model for Industry 4.0 (RAMI 4.0) layers and the open-source control device.

The control device is responsible for providing the programming of processes, identification of parts (e.g., through radio frequency identification—RFID), network communication through the standardized and interoperable protocol (e.g., through OPC UA, which is RAMI 4.0 recommended protocol for the *communication* layer), data management and information modeling, and supervision of equipment. All the information related to the controlled process will be available in real time for any type of platform connected to the communication infrastructure. Briefly, the open-source solutions for the proposed control device, described in the following subsections, are:

- **Software—OpenPLC project:** tasks of programming, identification of parts and network communication;
- **Software—OPC UA server project:** tasks of data management, information modeling and network integration;
- **Software—Human-machine interface (OPC UA client) project:** tasks of process supervision;

- **Hardware—single-board computer:** operational system with all processing and logic tasks;
- **Hardware—RFID system:** radio frequency identification reader and interface;
- **Hardware—IO connection board:** physical connection of industrial inputs and outputs (voltage, current) and data acquisition;

In the following, the design of the *asset*, *integration* and *communication* layers for the proposed open-source control device are better detailed.

3.1. Design of the Asset Layer

An essential resource for the development of I4.0 applications is the identification of products in relation to the activities to be performed by machines and stations. This functionality is defined in the *assets* layer. In addition, some assets can also perform a controlled process through equipment contained in the *integration* layer, such as a PLC, implemented under the IEC 61131-3 standard [33]. The assets are usually located in work units and stations of the production system [34]. In the case of the *asset* layer of the developed control device, RFID tags are used for product identification. In this way, it is possible to automate the tasks to be performed in a manufacturing process according to the need for different products, as each part contains a set of information customized for certain workstations.

3.2. Design of the Integration Layer

In the *integration* layer, all the elements associated with IT are listed, such as software for PLC programming, HMI visualization and generation of events based on the information acquired and supervision of the processes [3]. In this paper, the OpenPLC project was used [35] for the development of the *integration* layer of the proposed control device.

OpenPLC is an open-source project for integrating software and hardware for automation and process supervision applications [35] compatible with the IEC 61131-3. Complementing the OpenPLC, there are two additional parts: the HMI and the RFID reader. The HMI is used for data manipulation of the control device by operators. The RFID reader is responsible for processing the information from the tags fixed on the parts/products. Once the information is extracted from the RFID tags, they are stored in internal OpenPLC variables and made available for PLC programming. A communication driver was required and developed for the integration of the Sirit RFID reader API to the OpenPLC project.

3.3. Design of the Communication Layer

The *communication* layer of RAMI 4.0 is responsible for standardizing the communication between different networked elements based on uniform communication protocols. In view of the developed control device, the OPC UA protocol was used with the deployment of an OPC UA server and client.

The OPC UA application server developed was based on the *node-opcua* open-source project [36]. This server is responsible for providing the information of all products and parts (sensors, actuators) managed by the control device. This information is provided in a standardized form called OPC UA address space. In the case of the control device, the logical states of the inputs and outputs controlled by the OpenPLC are provided. An object containing the identified parts from the RFID system is also provided. In addition, another object provides information related to the hardware of the control device. The following information is available: percentage of free memory, percentage of CPU usage, CPU temperature, processor architecture and the operating system. Based on this information, it is possible to identify the hardware's proper capabilities before the application on a real system. It is important to highlight that another communication driver was necessary for the integration between the OPC UA server and the OpenPLC.

With the OPC UA server, any OPC UA customer (client) can access in real time any of the described information available at the control device. To provide the HMI functionality for the control device, a local OPC UA client was included. The open-source project *opcua-*

client-gui [37] (based on *freeopcua*) was chosen since it provides a graphical user interface (GUI) for easy usability, information visualization and data manipulation. This OPC UA client HMI provides to the operator information about the OPC UA server, navigation through OPC UA address space and subscription of any variables from the control device. The standardized data provided by the OPC UA server enables the data exchange between the control device and the *information*, *Functional* and *business* layers.

For the sake of clarity, Figure 3 presents a block diagram of the control device architecture and shows how the different parts are interconnected one with another.

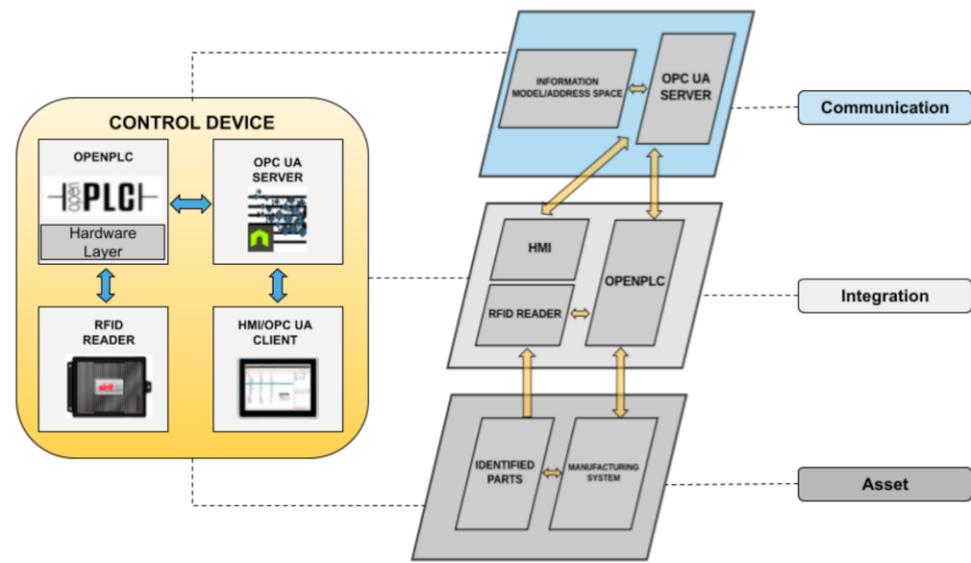


Figure 3. Architecture of the control device for I4.0.

In the *asset* layer, there are the identification of parts and the manufacturing system. The *integration* layer is comprised of the RFID reader, the OpenPLC project and the HMI. Finally, the *communication* layer is based on the integration of an OPC UA client–server data exchange.

4. Proof-of-Concept Implementation of the Control Device

Based on the control device architecture described in the previous section, a proof-of-concept prototype has been realized as described in the following. Considering the need for open-source hardware, the control device consists of a single-board computer (SBC), specifically the Raspberry Pi (RPi) Model 3B+, acting as the host device of the system. The choice for the RPi platform is due to its widespread availability, open-source resources, adequate processing capacity and memory availability besides the native Wi-Fi connectivity in addition to the wired ethernet. The connection of inputs and outputs to the RPi is done using an expansion board. In particular, the UniPi1.1 open-source IO connection board was chosen, as this is not only fully compatible with the RPi but is also supported by the OpenPLC project. Considering the need for open-source software of the control device, the Node.js platform running on a Raspberry Pi OS (Linux distribution for RPi) was used. Once the main hardware and software elements of the control device are defined, it is necessary to detail the software implementations related to RAMI 4.0 layers.

4.1. Interface of Integration and Communication Layers

The interface between the *integration* and *communication* layers of the control device uses a driver developed to enable the communication between the OpenPLC and the OPC UA server. The driver implemented for the proof-of-concept prototype transparently maps all variables of the controlled process and products (RFID tags) on the OPC UA server, using a built-in Modbus TCP/IP functionality included in OpenPLC. The OpenPLC has a

Modbus server that provides access to its internal variables. In addition, the driver may in the future provide a way to connect the control device to any remote IO device that supports the Modbus TCP/IP protocol.

Concerning the OPC UA server of the control device, the development was based on the stack and the SDK provided by the *node-opcua* project. The address space, which is the most important item of the OPC UA protocol, is used to customize the object model that the server will make available to clients. It standardizes and organizes the information in any OPC UA server. The control device has an internal function that creates the server address space with the necessary variables to define the required information model following RAMI 4.0 administration shell.

Thus, as shown in Figure 4, the control device OPC UA server provides an address space (information model) with five main objects: *server*, *RPi hardware*, *RFID tags*, *discrete outputs*, *discrete inputs* and *internal registers*. It is important to clarify that the visualization (icons, windows frames) of the information model in Figure 4 depends on the software used (OPC UA client). The control device OPC UA server programmatically defines the structure (folders, objects) and the contents of the model.

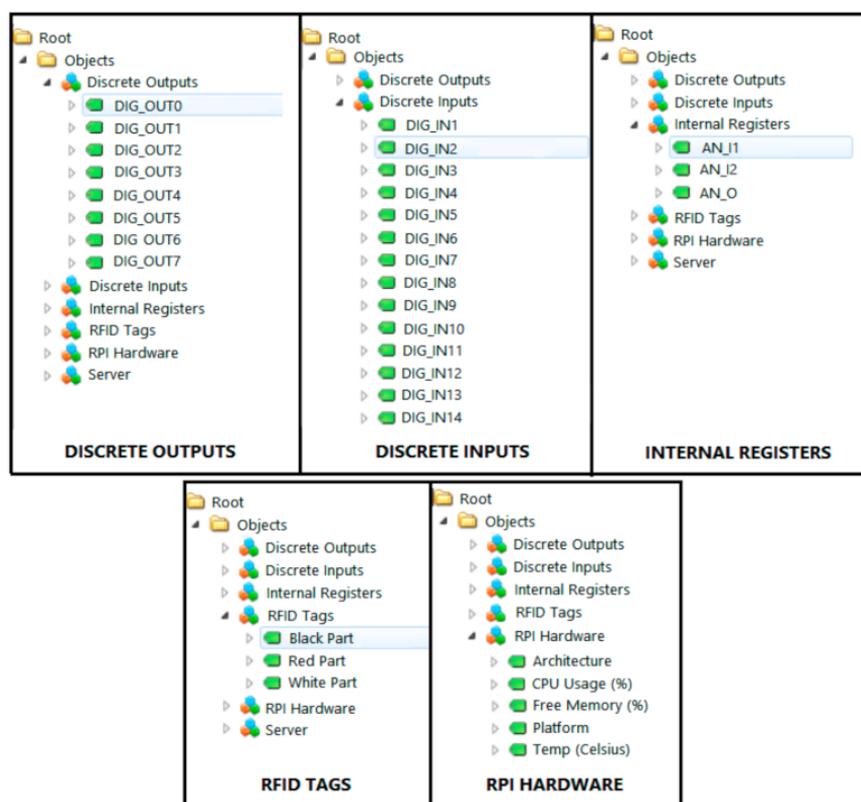


Figure 4. Control device address space of the open platform communication unified architecture (OPC UA) server.

The *server* object in Figure 4 contains general information about the OPC UA server, such as release date and info, product name, software version, current date, server start date, current server status, among others. The *RPi Hardware* object exposes operational information about the hardware, such as percentages of CPU usage and free memory and the CPU temperature (°C), among others. The *RFID Tags* object contains the information related to the parts/products identification using RFID. In accordance with the RFID reader used and configuration, variables are created containing the parts identified. In our case, the control device provides information about the tag (*TagId*—identification number and, *TagData*—*data*) and the identification of the antenna used (usually, an RFID reader may have up to 4 antennas). The last three objects are related to information about the inputs

and outputs of the control device (available on the UniPi expansion board used). It is important to note that all of these objects created in the server address space used *FolderType* as a reference. This structure allows information to be viewed in a hierarchical manner, improving interoperability between applications. For each of the objects created, classes of Variable Nodes were defined, representing within the proposal, data attributes of these objects. In the case of Discrete Outputs and Discrete Inputs objects, the *DataType* defined server was Boolean. For the Internal Register and RFID Tags, the defined type was Double.

Finally, security mechanisms of the OPC UA protocol were included at the control device server. These mechanisms include the device Security Type Configuration (based on the policies None, Basic, Basic128Rsa15, and Basic256), the device authentication and certification, and communication encryption. The information available in the OPC UA server address space of the control device is updated within 100 ms, using the interface driver described previously. Figure 4 presents in a systematic way the variables created for the different types of objects within the address space of the OPC UA server of the control device.

4.2. Interface of Integration and Asset Layers

The relationship between the *asset* and *integration* layers in the control device occurs in two ways: parts identification (radio frequency identification—RFID system) and controlled process (manufacturing system and OpenPLC). RFID tags attached to the parts were used for the identification. The information about the parts identified in the asset layer using the RFID reader must be available at the OpenPLC. Using this part information, the OpenPLC can perform customized actions at the workstation following the part. The interface between the *integration* and *asset* layers of the controller Interface uses a driver developed to enable the communication between the OpenPLC and RFID readers. This driver implements the reader API (ethernet TCP/IP) and transparently maps the part information (RFID tags) to the OpenPLC workspace. The driver was developed within the OpenPLC hardware layer and compiled with the project. It configures the RFID reader in active reading mode, based on asynchronous events. Given the occurrence of arrival and/or departure of RFID tags in any of the reader antennas, events are transmitted. These events are processed, and the information from *TagId*, *TagData* and *Antenna* are stored in internal OpenPLC variables.

4.3. Interaction of the Control Device with Other RAMI 4.0 Elements

Even though the developed prototype is focused on the control device operation validation, it comprised interactions between elements of the hierarchy levels and the layers axes of RAMI 4.0, as shown in Figure 5. The compatibility of the control device with RAMI 4.0 was described detailing the interactions using numbered arrows.

The arrow 1—*product* refers to the parts with RFID identification, generating information about the product (*TagID* and *TagData*) for RAMI 4.0 *communication* layer. The arrow 2—*field device* within the represents the manufacturing system data from the RFID system. In this case, it is possible to identify which workstation is being used in the process, as the information of the antenna (*Antennald*) is also available for the *communication* layer. The interaction that occurs in the arrow 3—*control device* concerns the operation of Open-PLC, being able to receive the information from the station and control the manufacturing process (ladder program following the IEC 61131-3 standard), as well as assigning this information to the *communication* layer. At the controlled station, whenever a registered product is identified by the RFID system, the *integration* layer obtains this information and executes a specific logic (instructions) for each identified part. Arrow 4—*communication structure* represents the display of organized information regarding the identification of product parts, identification of the manufacturing system, hardware performance information and logic states of the sensors/actuators through the address space of the OPC UA server. Finally, the arrow 5—*station* lists the use by users/operators (OPC UA clients)

of all information available (server), which can be provided through HMIs and used for supervision tasks.

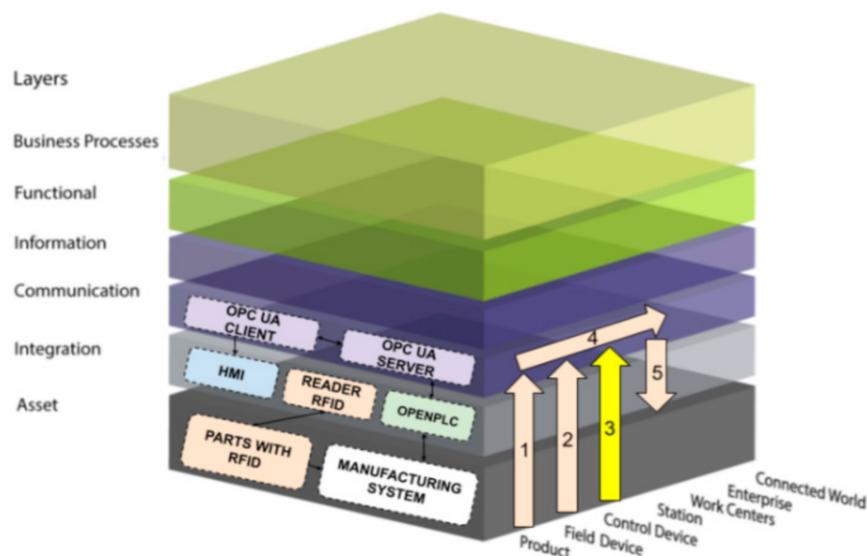


Figure 5. Interactions among elements of RAMI 4.0 axes in the experiment of the control device validation.

5. Proof-of-Concept Experimental Results

5.1. Experimental Setup

A laboratory experiment in an I4.0 scenario was performed using the previously described proof-of-concept for validating the development of the open-source control device based on RAMI 4.0. Industrial equipment related to a modular production system by FESTO MPS (sorting station) was used in an application of customized realization of tasks in machines and workstations in accordance with the type of product provided by the identification of parts. The idea is to process and sort different parts (different colors) at the workstation using the information from the RFID tags included in each part (<https://youtu.be/1miDnc7bpCY> (accessed on 21 March 2021)). The control device is responsible for all logic and control tasks. The connection diagram of the experiment is presented in Figure 6. A picture showing the equipment and the production system used in the experiment is presented in Figure 7.

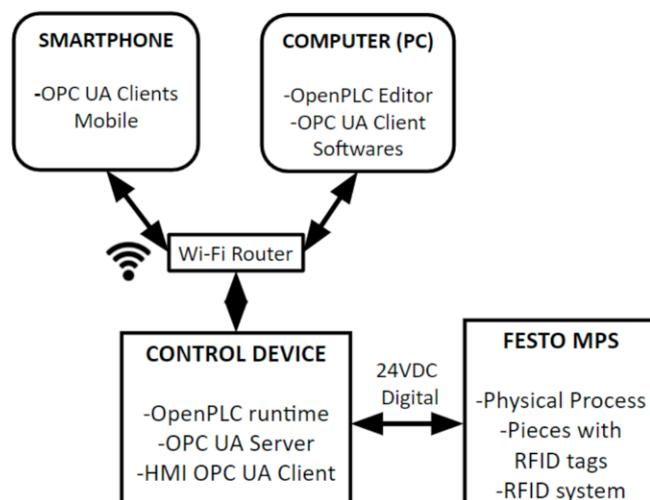


Figure 6. Connection diagram of the experiment.

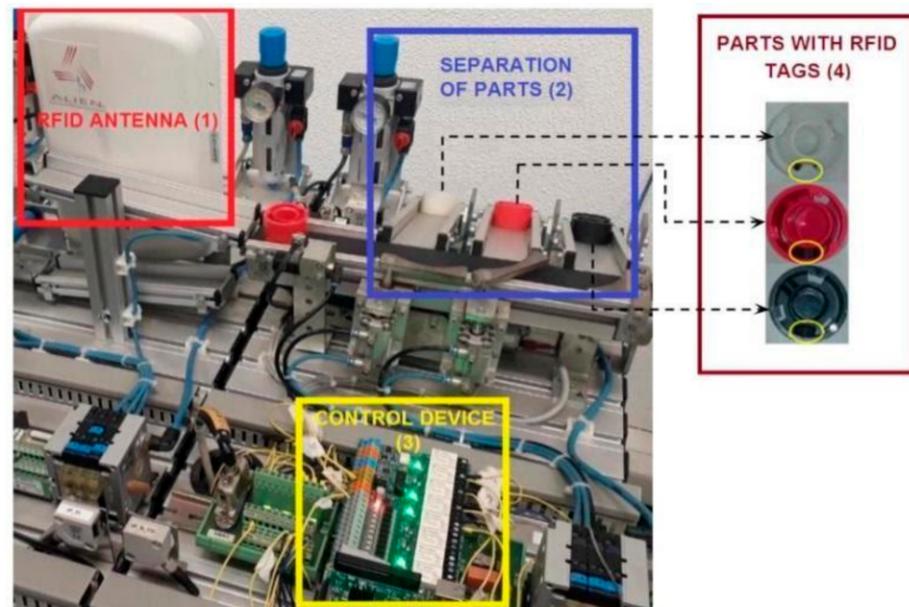


Figure 7. Details of the experiment: stations, parts and control device.

The equipment used in the experiment is wirelessly connected to a 2.4 GHz Wi-Fi network. The developed control device has built-in support for Wi-Fi communication and executes the OpenPLC runtime, the OPC UA server with the information model and the HMI (which is an internal OPC UA client) for local monitoring and supervision. The control device was connected to the Festo Sorting station IO (sensors/actuators) using 24 Vdc digital signals (discrete inputs and outputs). During the experiment, other OPC clients were connected to the control device for monitoring. These clients were executed on a computer (PC) and in a smartphone.

The computer also executes the OpenPLC editor, which is used only during the logical programming part. The logical programming of the FESTO station manufacturing process was created using an IEC 61131-3 ladder language using the OpenPLC editor. Subsequently, the generated file referring to the programming logic is loaded to the OpenPLC runtime (at the control device) remotely through the network. At the end of this process, the control device can perform the required logic control for the process automation and control.

The previously highlighted programming is performed physically at the Festo workstation, presented by Figure 7, in which (1) represents the RFID antenna, (2) separation of parts, (3) the developed control device and (4) Parts with RFID tag.

In Figure 7, the control device controls the station operation according to the requirements of the application. In this way, the parts (white, red and black colors) at that station are identified using the RFID system that has the antenna attached to the station's belt (1). According to the identified part (color), the control device controls the station for sorting the parts in different stocks (2). It is important to note that, in its original configuration, the classification station has specific sensors and a PLC responsible for the I/O interface and programming of the logic that controls the operation of the station. However, this equipment was not used in this paper, as the control device (3) was responsible for managing all functions within the operation of the station and the classification of the part using RFID.

5.2. Operational Results

Operational data acquired during the experiment described in Section 5.1 were collected to evaluate the control device. Considering the nature of the control device operation in I4.0 scenarios, multiple connections from OPC UA clients are expected to happen. These connections to the control device OPC UA server were done during its operation for controlling the Festo sorting station as described before. Up to five OPC UA clients were connected simultaneously to the control device using a Wi-Fi connection. It is important

to clarify that the communication between the client/server is asynchronous in OPC UA. Usually, this communication happens in two ways: during the connection and after the connection is established for data communication. During the OPC UA client/server connection, the client uses the *browse* service to communicate with the server and receive its address space (information model). For data communication, the client uses the *subscribe* or *read/write* services to communicate with the server. With the *read/write*, the client polls the server each time the data are needed. With the *subscribe*, the server automatically sends the data to the client every time its value changes.

Figure 8 shows a complete view of the experimental setup, in which five clients (numbered from 1 to 5) were connected to the OPC UA server of the control device (A).

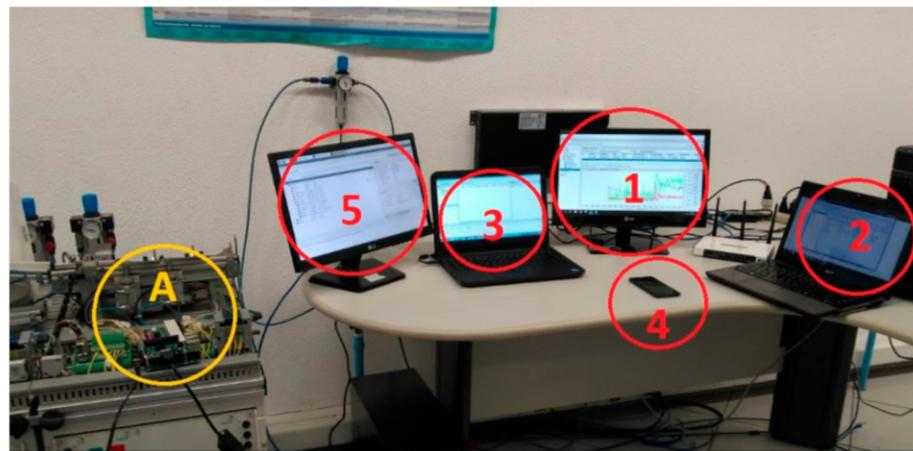


Figure 8. External OPC UA clients (1 to 4) and internal OPC UA client (5) connected to the OPC UA server.

The experiment was designed in the following way: a new client connection to the control device's OPC UA server was established every two minutes (de facto increasing the server load as time goes by). Each connection used a different OPC UA security policy and type to verify the control device security compatibility and evaluate the impact on its operation. Table 1 details the connection sequence, the type of OPC UA client, the security mode, the security policy and the time at which the connection with the control device server takes place.

Table 1. Sequence of client connections to the control device OPC UA server.

Client ID	Client OPC UA	Security Type	Security Policy	First Connection Time (min)
1	Prosys software	None	None	0
2	Prosys software	Sign	Basic128Rsa15	2
3	UAExpert software	Sign	Basic256	4
4	OPC UA mobile	Sign/encrypt	Basic256Sha256	6
5	Control device internal (HMI OPC UA client)	None	None	8

It is worth mentioning that the five clients connected to the control device for receiving the values of three variables from the "RPi hardware" object: CPU usage (%), free memory (%) and temperature (°C). The graph of Figure 9 shows changes in these acquired data of the control device during the experiment and the connection of the users (clients).

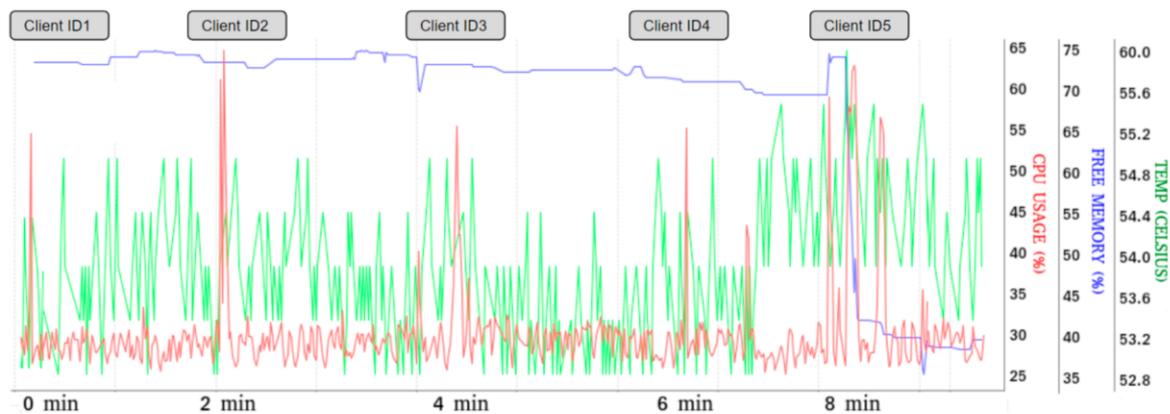


Figure 9. Control device operational data during the experiment with multiple user connections (clients).

The control device operation was robust, with no interruption or failure, neither communication problems, nor OPC UA disconnection during the experiments. The control device was kept in operation for 12 h within the experiment. However, as the results of the monitored variables maintained the same patterns and values, it was chosen to show in Figure 9 only the first 10 min. Considering that the production cycle of the experiment (time interval between consecutive input of pieces in the system) is around 15 s, a great number of pieces were processed, or production cycles were executed by the control device. In addition, it is worth mentioning that the logical results of the control device or the correct execution of programming tasks and control of the manufacturing process were validated during the experiment.

The operational results in Figure 9 show normal hardware usage of the control device during the experiment with the control of the Festo station and with the connections of several users (clients). There were no significant temperature changes during the control device operation, and the temperature was in a suitable interval. The CPU usage was approximately 30%. Some short-time peaks in CPU usage happened when each external client (ID1 to 4) connected to the control device. The change in the percentage of free memory with the connection of the external clients was small and up to 5% (from 75 to 70%). The external clients were kept connected (after their connections) during the whole experiment, showing that there was no significant change in the CPU usage and free system memory related to the connection and communication of external clients with the control device. In addition, this result shows that the control device would be able to support many simultaneous connections of users (OPC UA clients).

The execution of the internal HMI, which is an OPC UA client (ID-5), had a significant impact on the hardware usage of the control device. It was expected as the internal HMI client execution consumes resources of the control device hardware for processing and video interface (visualization). A relevant decrease in the percentage of free system memory, as well as a relevant increase in CPU usage, can be seen in Figure 9 with the execution and connection of the internal HMI client (ID-5). The CPU usage peak oscillated but maintained a high value of around 60%, while the internal HMI was kept running. The usage only returned to normal values around 30% after the internal HMI was turned off.

Finally, the results of the experiment also showed that the use of different types and security policies supported by the control device do not significantly impact its performance and hardware usage. In a nutshell, the logical and operational results validated the control device development and shown the availability of capacity for future deployments and improvements.

6. Discussion and Final Remarks

The great challenge in developing a RAMI 4.0 compatible solution is the comprehension of the required relationships among technologies, equipment and systems related to RAMI 4.0 axes for each type of I4.0 application. As RAMI 4.0 is abstract, different

perspectives and interpretations may be feasible according to the intended solution. The developed control device was conceived as a proof-of-concept, focusing on the integration of open-source solutions and covering the relationships among three layers of RAMI 4.0: *assets, integration and communication*.

The control device integrated a set of different technologies, such as the OpenPLC project, the RFID system and the OPC UA (server/client) projects, in a one package solution. To do that, it used built-in communication and integration functionalities available in these projects as well as it was necessary to develop additional drivers. For example, the industrial IO connection board used (UniPi) was selected because it is already supported in the OpenPLC project. However, in case of using other not supported IO board, a driver would need to be developed for integration of the IO data on the Open PLC. It was the case of the RFID system. As the OpenPLC project does not support any RFID reader, an API of the reader used was developed and added to the core of the OpenPLC project. As a result, the data from the RFID tags became available at the OpenPLC. This fact can be seen as the main disadvantage of the control device, as it limits its application flexibility or at least would require any integration rework in case of using different solutions from the ones described.

Reference [38] describes that an I4.0 component is the combination of objects from both the physical and the digital world. The physical world is composed of elements of the *asset* and *integration* layers of RAMI 4.0. The digital world is composed of the upper layers, from the *communication* up to the *business*, depending on the element. Considering the control device element of RAMI 4.0, the digital world would include interactions up to the *information* layer. The majority of the required functionalities related to the *asset* and *integration* layers (physical world) and to the *communication* layer (digital world) were included in the developed control device.

Analogies between the information structure of the developed control device with the administration shell (AS) of the I4.0 component in [38] can be cited. The OPC UA address space of the control device contains the definition of physical objects that are accessed by different types of clients. These objects are basically modeled in terms of views, attributes, variables, methods and relationships with other objects. Therefore, both methodologies structurally have a similar meaning in the specification of concepts, such as assets x nodes; property of assets x attributes and variables of objects, submodels x object; resource manager x address space.

7. Conclusions

This paper presented the development of an open-source control device for Industry 4.0 (I4.0) based on RAMI 4.0. The control device comprised elements of the *communication, integration and asset* layers of RAMI 4.0. This development serves as a proof of concept of a RAMI 4.0 compatible open-source control device and can be adapted for a manufacturing system aiming to fulfill the requirements of I4.0 applications.

The architecture of the control device is modular and integrates different components and open-source technologies to fulfill the development requirements of the included RAMI 4.0 layers. The OpenPLC project was used in the *integration* layer for IEC 61131-3 compatible logic control of manufacturing processes. Additional support for parts identification using RFID required at the *asset* layer was included using the hardware abstraction layer of the project. The *node-opcua* project was used for the implementation of the OPC UA protocol, which is RAMI 4.0 recommended protocol for the *communication* layer. The developed address space of the control device relates to RAMI 4.0 administration shell and provides information related to the controlled process in a standardized and fully interoperable manner.

Future work will focus on updating the control device towards an I4.0 component of RAMI 4.0. Even with the current similarities between the control device address space with the Administration of RAMI 4.0, a standardized model for the *information* layer, with

additional information from the assets, such as manuals, configuration and communication parameters, maintenance history and functional profile, is required for the update.

Author Contributions: Conceptualization, E.P.G.; data curation, P.F.S.M. and E.S.; formal analysis, E.S.; funding acquisition, E.P.G.; investigation, P.F.S.M. and P.F.; methodology, P.F.; project administration, E.P.G.; software, P.F.S.M.; validation, P.F.; writing—original draft, P.F.S.M. and E.P.G.; writing—review and editing, P.F. and E.S. All authors have read and agreed to the published version of the manuscript.

Funding: Research supported by grant 2018/19984-4, São Paulo Research Foundation (FAPESP).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Dorst, W.; Glohr, T.; Han, F.; Knafla, U.; Loewen, R.; Rosen, T.; Schiemann, F.; Winterhalter, C. Implementation Strategy Industrie 4.0: Report on the results of the Industrie 4.0 Platform. January 2016. Available online: <https://www.bitkom.org/Bitkom/Publikationen/Implementation-Strategy-Industrie-40-Report-on-the-results-of-the-Industrie-40-Platform.html> (accessed on 21 March 2021).
- Schwab, K. The fourth Industrial revolution. World Economic. In World Economic Forum. 2016. Available online: <https://www.weforum.org/about/the-fourth-industrial-revolution-by-klaus-schwab> (accessed on 21 March 2021).
- Wang, Y.; Towara, T.; Anderl, R. Topological Approach for Mapping Technologies in Reference Architectural Model Industrie 4.0 (RAMI 4.0). In Proceedings of the World Congress on Engineering and Computer Science, vol II WCECS, San Francisco, CA, USA, 25–27 October 2017.
- Adolphs, P.; Bedenbender, H.; Dirzus, D.; Ehlich, M.; Epple, U.; Hankel, M.; Heilde, R.; Hoffmeister, M.; Huhle, H.; Karcher, B.; et al. Reference Architecture Model Industrie 4.0 (RAMI 4.0). Status Report. VDI/VDE and ZVEI. July 2015. Available online: https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/5305_Publikation_GMA_Status_Report_ZVEI_Reference_Architecture_Model.pdf (accessed on 21 March 2021).
- Bangemann, T.; Bauer, C.; Bedenbender, H.; Diesner, M.; Epple, U.; Elmas, F.; Friedrich, J.; Goldschmidt, T.; Gobe, F.; Gruner, S.; et al. Industrie 4.0—Technical Assets: Basic Terminology Concepts, Life Cycles and Administration Models. Status Report. VDI/VDE and ZVEI. March 2016. Available online: <https://www.vdi.de/ueber-uns/presse/publikationen/details/industrie-40-technical-assets-basic-terminology-concepts-life-cycles-and-administration-models-english-version> (accessed on 21 March 2021).
- Adolph, L.; Ammon, E.; Becker, J.; Bedenbender, H.; Bellinghausen, V.; Börkircher, M.; Braunmandl, A. German Standardization Roadmap Industrie 4.0- Version 4. DIN/DKE. March 2020. Available online: <https://www.din.de/en/innovation-and-research/industry-4-0/german-standardization-roadmap-on-industry-4-0-77392> (accessed on 21 March 2021).
- Hankel, M. Industrie 4.0: The Reference Model Industrie 4.0 (RAMI4.0), ZVEI. April 2015. Available online: https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/ZVEI-Industrie-40-RAMI-40-English.pdf (accessed on 21 March 2021).
- Melo, P.F.S.; Godoy, E.P. Controller Interface for Industry 4.0 based on RAMI 4.0 and OPC UA. In Proceedings of the 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), Naples, Italy, 4–6 June 2019; pp. 229–234.
- Langmann, R.; Rojas-Peña, L.F. A PLC as an “Industry 4.0 Component”. In Proceedings of the 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), Madrid, Spain, 24–26 February 2016; pp. 10–15.
- Langmann, R.; Stiller, M. The PLC as a Smart Service in Industry 4.0 Production Systems. *Appl. Sci.* **2019**, *9*, 3815. [CrossRef]
- Minchala, L.I.; Peralta, J.; Mata-Quevedo, P.; Rojas, J. An Approach to Industrial Automation Based on Low-Cost Embedded Platforms and Open Software. *Appl. Sci.* **2020**, *10*, 4696. [CrossRef]
- González, I.; Calderón, A.J.; Portalo, J.M. Innovative Multi-Layered Architecture for Heterogeneous Automation and Monitoring Systems: Application Case of a Photovoltaic Smart Microgrid. *Sustainability* **2021**, *13*, 2234. [CrossRef]
- Ali, A.S.; Coté, C.; Heidarinejad, M.; Stephens, B. Elemental: An Open-Source Wireless Hardware and Software Platform for Building Energy and Indoor Environmental Monitoring and Control. *Sensors* **2019**, *19*, 4017. [CrossRef] [PubMed]
- Shin, D.; Bianco, W.T. In Blockchain We Trust: Does Blockchain Itself Generate Trust? *Soc. Sci. Q.* **2020**, *101*, 2522–2538. [CrossRef]
- Shin, D. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI. *Int. J. Hum. Comput. Stud.* **2021**, *146*, 102551. [CrossRef]
- Contrenas, J.D.; Garcia, J.I.; Pastrana, J.D. Developing of Industry 4.0 Applications. *Int. J. Online Eng. (IJOE)* **2017**, *13*, 10.
- Marcon, P.; Zezulka, F.; Bradac, Z.; Arm, J.; Benesl, T.; Vesely, I. Communication Systems for Industry 4.0 and the IIoT. *IFAC-PapersOnLine* **2018**, *51*, 150–155.
- Hoffmeister, M. Industrie 4.0: The Industrie 4.0 Component. ZVEI. April 2015. Available online: https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/ZVEI-Industrie-40-Component-English.pdf (accessed on 21 March 2021).

19. Pisching, M.A.; Pessoa, M.I.A.O.; Junqueira, F.; Filho, D.J.S.; Miyagi, P.E. An architecture based on RAMI4.0 to discover equipment to process operations required by products. *Comput. Ind. Eng. J.* **2017**, *125*, 574–591. [[CrossRef](#)]
20. Schulte, D.; Colombo, W. RAMI 4.0 Based Digitalization of an Industrial Plate Extruder System: Technical and Infrastructural Challenges. In Proceedings of the IECON-43 rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017.
21. Mourtzis, D.; Gargallis, A.; Zogopoulos, V. Modelling of Customer Oriented Applications in Product Lifecycle using RAMI 4.0. In *International Conference on Changeable, Agile, Reconfigurable and Virtual Production*; Elsevier: Amsterdam, The Netherlands, 2019.
22. Ojanpera, M.Y.; Sierla, S.; Papakonstantino, N.; Vyatkin, V. Adapting an agile manufacturing concept to the reference architecture model industry 4.0: A survey and case study. *J. Ind. Inf. Integr.* **2019**, *15*, 147–160.
23. González, I.; Calderón, A.J.; Figueiredo, J.; Sousa, J.M.C. A Literature Survey on Open Platform Communications (OPC) Applied to Advanced Industrial Environments. *Electronics* **2019**, *8*, 510. [[CrossRef](#)]
24. Ferrari, P.; Flammini, A.; Rinaldi, S.; Sisinni, E.; Maffei, D.; Malara, M. Impact of Quality of Service on Cloud Based Industrial IoT Applications with OPC UA. *Electronics* **2018**, *7*, 109. [[CrossRef](#)]
25. Rocha, M.S.; Serpa Sestito, G.; Dias, A.L.; Turcato, A.C.; Brandao, D.; Ferrari, P. On the performance of OPC UA and MQTT for data exchange between industrial plants and cloud servers. *Acta IMEKO* **2019**, *8*, 80–87. [[CrossRef](#)]
26. Fleischmann, H.; Brossog, M.; Beck, M.; Franke, J. Automated Generation of Human-Machine Interfaces in Electric Drives Manufacturing. In Proceedings of the 7th International Electric Drives Production Conference and Exhibition, Wuerzburg, Germany, 5–6 December 2017.
27. Schleißen, M.; Gilani, S.S.; Bischoff, T.; Pfrommer, J. OPC UA & Industrie 4.0- Enabling Technology with High Diversity and Variability. *Procedia CIRP* **2016**, *57*, 315–320.
28. Damiani, L.; Demartini, M.; Guizzi, G.; Revetria, R.; Tonelli, F. Augmented and virtual reality applications in industrial systems: A qualitative review towards the industry 4.0 era. *IFAC-PapersOnLine* **2018**, *51*, 624–630. [[CrossRef](#)]
29. Tang, Y.; Au, K.; Leung, Y. Comprehending products with mixed reality: Geometric relationships and creativity. *Int. J. Eng. Bus. Manag.* **2018**, *10*, 1847979018809599. [[CrossRef](#)]
30. Aheleroff, S.; Xu, X.; Zhong, R.Y.; Lu, Y. Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model. *Adv. Eng. Inform.* **2021**, *47*, 101225. [[CrossRef](#)]
31. Adwernat, S.; Wolf, M.; Gerhard, D. An Ontology-Based Concept to Support Information Exchange for Virtual Reality Design Reviews. In *Product Lifecycle Management Enabling Smart X. IFIP Advances in Information and Communication Technology*; Nyffenegger, F., Ríos, J., Rivest, L., Bouras, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 594.
32. Zaheer, M.A. RAMI 4.0 (Part 2): Product Lifecycle and Value Streams. IoT Zone Article. November 2017. Available online: <https://dzone.com/articles/part-2-rami-40-startup-of-smart-electronic-industr> (accessed on 21 March 2021).
33. Ludder, A.; Schleißen, M.; Schmidt, N.; Pfrommer, J. One step toward an Industry Component. In Proceedings of the 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, China, 20–23 August 2017; pp. 1268–1273.
34. Hell, K.; Kristofer, R.; Hillmann, A.; Luder, H.; Ropke, J.; Zawisza, J.; Schmidt, A. Demands on Virtual Representation of Physical Industrie 4.0 Components. In Proceedings of the 2nd INCOSE Italy Conference on Systems Engineering, Turin, Italy, 14–16 November 2016; Volume 1728.
35. Alves, T.R.; Buratto, M.; Souza, F.M.; Rodrigues, T.V. OpenPLC: An Open Source Alternative to Automation. In Proceedings of the IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, USA, 10–13 October 2014; pp. 585–589.
36. Rossignon, E. Node-Opcua: An Implementation of an OPC UA Stack Fully Written in JavaScript and Node.js. Available online: <https://github.com/node-opcua/node-opcua/> (accessed on 21 March 2021).
37. Overn, A. opcua-client-gui. Available online: <https://github.com/akselov/opcua-client-gui> (accessed on 21 March 2021).
38. Fuchs, J.; Schmidt, J.; Franke, J.; Rehman, K.; Sauer, M.; Karnouskos, S. I4.0-compliant integration of assets utilizing the Asset Administration Shell. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1243–1247.