Using De-characterized TV BOX in Industry 4.0 Applications

Guilherme Henrique Pinheiro & Eduardo Paciência Godoy
Scientific Inititiation Program PIBIC Junior CNPQ/Reitoria UNESP
Universidade Estadual de São Paulo (UNESP) – Campus Sorocaba
Sorocaba, São Paulo, Brasil
guilhermehenriquepinheiro2022@gmail.com; eduardo.godoy@unesp.br

Vitor Mendes Caldana
Professor, Eletronics Department
Instituto Federal de São Paulo (IFSP) – Campus Sorocaba
Sorocaba, São Paulo, Brasil
vitor.caldana@ifsp.edu.br

Abstract

With the advances of Industry 4.0 that seeks to automate the production lines as a hole to improve its efficiency, the use of computational resources is important. Raspberry Pi is a common hardware used for the needs of this automation. TV Box are embedded computers confiscated by the government due to piracy laws, and with de-characterization and reprograming they can be used to replace Raspberry Pi or similar hardware. Their use does not only allow for reuse of hardware that was supposed to be discarded, but can serve as an integrative and zero-cost solution for universities to implement Industry 4.0 solutions. To that end this research proposes the use de-characterized TV Box to control and exchange information used in a Node-RED and OPC UA platform in a centralized PC into several different stations, decentralizing the need for computational resources and creating flexibility for future implementations of the existing Digital Twin project inside the laboratory.

Keywords
Industry 4.0, TV Box, Raspberry Pi, Linux, Node-RED

1. Introduction

The term Industry 4.0 was introduced in Germany in 2011. A concept that came to automate and robotize all of the production line in favor of optimization and modernization of labor for a greater productivity and efficiency. From the production point of view, Industry 4.0 is a concept that integrates the physical parts of the process (such as machines, various devices, sensors and etc.) to cybernetic parts (such as advanced software, digital twins and etc.) through integration and driven by the pillar technologies of Industry 4.0 that can be used to predict, control, maintain and integrate production processes (Bajic et al., 2021)

The driven Technologies of industry 4.0 are: Cyber-Physycal systems (CPS), Internet of Things (IoT), Big Data Analysis (BDA), Cloud Computing, Fog and Border Computing, Augmented and Virtual Reality (AR/VR), Robotics, Cyber Security and Additive Manufacturing (AM) (Rikalovic et al., 2022).

TV Box, as display in **Erro! Fonte de referência não encontrada.**, are embedded computers that are confiscated by the Federal Government of Brazil since they are used for Piracy purposes, by unblocking paid content from subscribed TV channels, and displaying the for free. To contribute to the environmental management, ecoefficiency and social responsibility the government implement the program "*Receita Cidadã*"[1] that donates the hardware to universities so they can be de-characterized from the "TV" function and the hardware used for other purposes, making them similar to Raspberry Pi but with zero-cost.

Since Industry 4.0 is a concept that involves several technologies capable of optimizing and modernize labor, with driven technologies that can require great computational power, the governmental project supplies an important hardware for that end, allowing for Fog and Border computing to be implemented as well as the decentralization of computational resources.

---

[1] https://www.gov.br/receitafederal/pt-br/acesso-a-informacao/acoes-e-programas/cidadania-fiscal/novos-destinos/tv-pirata

Figure 1: TV Box model BTV E10 (The authors)

This paper is organized as follows: section 2 is dedicated to the literature review; in section 3 we display the methods and materials; in section 4 we present the de-characterization process of TV Box followed by results and discussion in section 5. Section 6 is dedicated to the conclusion and future work.

1.1. Objectives

The purpose of this paper is to create a method to use the existing hardware in the TV Box that once was discarded and destroyed by the government into a hardware similar to a Raspberry Pi that can be used in application for Industry 4.0.

This will be done by developing a study to transform the pirate operating systems in a micro controlled device that can be implement in Industry to control and automate machines and process replacing paid solutions such as Raspberry Pi. Due to the needs of the greater project of the laboratory, we will aim to install a Linux based operating system on the hardware that can run a Node-RED server. The final objective will be to remotely control and access several devices that will be networked to test the performance of the computation power for real-time control of the production process, replacing the need for centralized control.

2. Literature Review

In this section a brief literature review in undertaken to explain the main concepts and tools used during this Project. EducaBox project was used as initial option to build the necessary knowledge on de-characterizing the TV Box and after the tests described in the results and discussion session and the challenges listed in section 5.1 the project shifted to the Armbian Linux image. Raspberry Pi was used as a basis for the end goal of the de-characterized TV Box.

2.1. Embedded Systems

A system is classified as embedded when it is dedicated to a single task and interacts continuously with the environment around it through sensors and actuators (Ball, 2002). The denomination "embedded" comes from the fact that the systems are designed to be independent of an outside power source such as an outlet or a generator (Chase, 2007).

All embedded systems have a processing unit, an integrated circuit soldered into the main circuit board. It has the capability to process information from an internal software being run on the unit, thus the software will be embedded into the processing unit. All embedded software are known as firmware (Ball, 2002)

2.2. Raspberry Pi

Raspberry Pi is a series of microcomputers from a single multi-platform board, or reduced size with integrated components. It works, primarily, as a standard computer running operating systems such as Linux where it is extremely useful to run multiple activities for various objectives such as control and automation of complex processes. This ability to run multiple tasks simultaneously is very useful in the industry (Mathe et al., 2024).

Apart from its industrial use, Raspberry PI also has applications in teaching (programming), residential automation, electronics projects, servers and even media centers.

2.3. Armbian

Armbian started as a personal hobby project in 2013 in GitHub as a simple script to create a Linux Debian image. There was no name or dedicated website to it. All the communication of the project were made by interaction and comments on personal blogs, e-mails and third-party forums. The work on Debian for Cubieboard, the first supported project, gained attention since the early days. Today Armbian is a highly optimized operational systems, specialized in Single Board Computers (SBC). It offers an ultra-light base ideal for servers and Internet of Things (IoT) projects, as well as providing a complete desktop interface including support to 3D models and video (Armbian, 2024).

To properly start the image, a Device Tree Blob (DTB) file is required. It is a binary file that describes the device´s hardware, allowing the operational system to recognize and configure all the components of the system, such as processor, memory, USB Ports, Wi-Fi antenna amongst others. (Armbian, 2025)

2.4. EducaBox

EducaBox is a Project based on the governments project "*Receita Cidadã*" before mentioned. It aims to transform the hardware into educational computers. By achieving this end EducaBox hopes to widen the access to knowledge, especially in regions where the technological infrastructure can be limited. Students, professors and technicians have the ability to explore educational resources, interactive contents and engage in involving learning experiences, all from the recovered hardware from the TV Box (EducaBox, 2025).

3. Method and Materials

The method displayed in the flowchart of Figure 2 was developed to de-characterize the TV Box, removing the pirate software and installing an Armbiam Linux OS (or similar) and modifying it to run Node-RED. To achieve that, the correct and compatible DTB must be located to interact with the hardware.
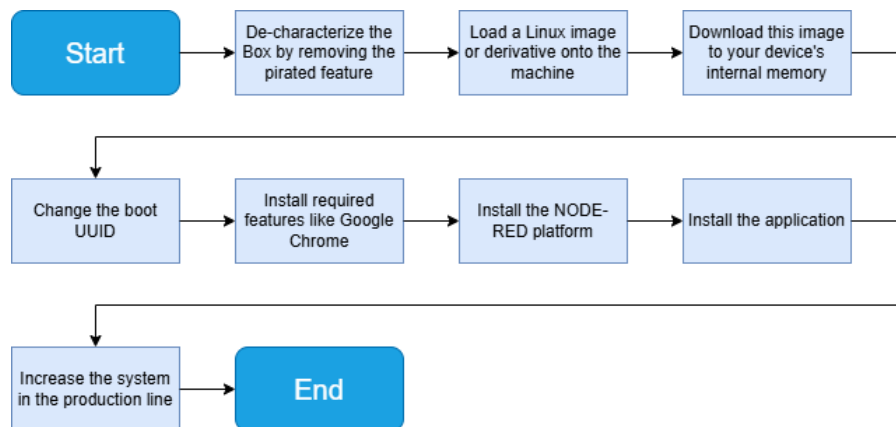

Figure 2: De-characterization flowchart

3.1. TV BOX

The specifications for the TV BOX BTV E10 that is available to perform the studies is listed in Table 1 below.

Table 1: TX BOX BTVE10 Specifications

Property	Value
Codename	BTV E10
Supplier	BTV
Model	E10
Motherboard	BTV E10-LPDDR4 V.10 201-0308

DTB Board	g12a-u212_2g
CPU	Amlogic S905X2
Family	Cortex-A53
Processing Speed	100 – 1800 MHz
GPU	Mali-G31 MC1
Wifi	RTL8189FTV
RAM Memory	2GB
Storage Capacity	8GB
Display Resolution	1080X1920

4. TV BOX De-characterization procedure

Based on the flowchart presented in Figure 2 we present the procedure created by the researchers to de-characterize the TV Box and insert an EducaBox images, that as described in section 2.4, was the initial image used as it was already developed with full resources. Further development and the challenges listed in section 5.1 made necessary to change the image to an Armbian Linux OS and development is still being undertaken with this new approach.

4.1. Saving and Configuring the Image into the Memory Device

After downloading the desired image, it is necessary to store it into a flash-drive or memory card. For the purpose of this paper the Balena Etcher[2] software was used, but it can be replaced by any similar one. The process is straightforward as shown in Figure 3.

Figure 3: Saving the Image (The Authors)

Once the process is finished, the memory device needs to be removed and re-inserted cancelling the format notices from Windows. The memory device needs to be configured to the correct DTB file. To do that, find the file *"armbianEnv.txt"* and remove the "#" sign on the file that matches the correct DTB for your hardware. In the case of this paper, we use the TB BOX BTV E10, and the correct file is *"meson-g12a-sei510.dtb"*. The details of this procedure are displayed in Figure 4 below.

It may be necessary to update the rootdev information for the universally unique identifier (UUID) on the memory device in case it was not automatically changed. To achieve this in a Linux device, open the Terminal and with the memory device connected type the command line *"sudo blkid"*. This command will return the UUID of all the partitions of the device, including removable memory. eMMC and SDCards often have the UUID starting with [mmcblk1, mmclk2] and Flash drives and external HD [sda, sdb, ..., sdx]. The detail of the line placement can be seen in Figure 8 in section 4.3.

---

[2] https://etcher.balena.io/

```
#------------------------------------------------------------
# Exemplo DTBs para TVs Boxes
#------------------------------------------------------------
# Este é padrão para os processadores Amlogic
# Exemplo - [NÃO DESCOMENTE ESSAS LINHAS]
# Amlogic S905X      -> meson-gxl-s905x-p212.dtb
# Amlogic S905W      -> meson-gxl-s905w-p281.dtb
# Amlogic S905X2     -> meson-g12a-sei510.dtb
# Amlogic S905X3     -> meson-sm1-sei610.dtb
# Descomente a linha da sua Box ou Dtb Abaixo:
#------------------------------------------------------------
# S905X
#------------------------------------------------------------
# MY BOX TV | BTVBX | TIGRE 2
#fdtfile=amlogic/meson-gxl-s905x-p212.dtb
#------------------------------------------------------------
# S905X2
#------------------------------------------------------------
# BTVE10
fdtfile=amlogic/meson-g12a-sei510.dtb
#------------------------------------------------------------
# S905X3
#------------------------------------------------------------
# BTV11 | HTV7 | PLAY TV |
#fdtfile=amlogic/meson-sm1-sei610.dtb
# Para ativar algumas placas de Rede/Wifi
#fdtfile=amlogic/meson-sm1-sei610-ethfix.dtb
#------------------------------------------------------------
# S905W
#------------------------------------------------------------
# EXTREMO | HTV6+ | IMOVE | IMOVE PRO |
#fdtfile=amlogic/meson-gxl-s905w-p281.dtb
```

Figure 4: armbianEnv.txt file (The Authors).

4.2. Loading the new image in the TV BOX

To start the desired image, insert the memory device in the TV BOX, turn de device upside-down to match Figure 5 and with a sharp thin object press the "Reset" button while holding the "Update" button pressed until the new image starts to be loaded, as shown in Figure 6 . It is important to notice that other models of TV Box may have a different procedure to start a new image, with some including a hidden button on the AV input.

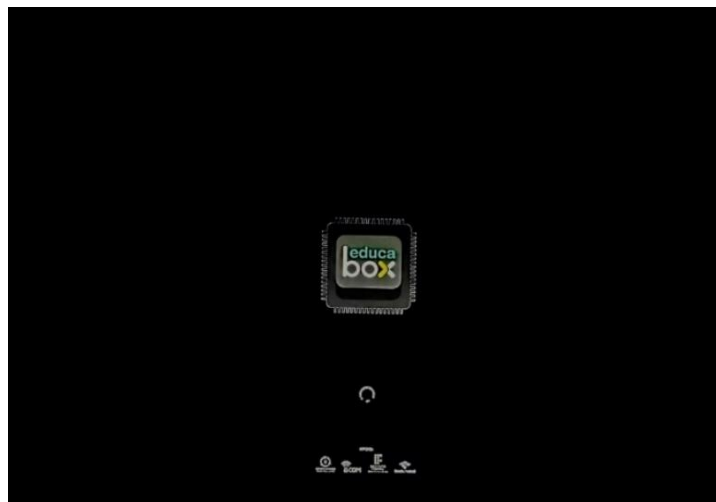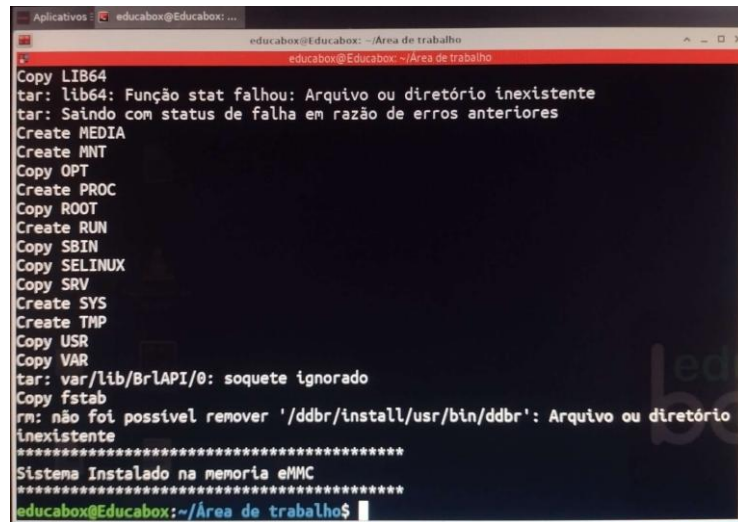**Figure 5**: Reset and Update buttons (The Authors)

Figure 6: EducaBox Botting (The Authors).

4.3. Transferring the image to the internal memory

After the image is installed, we have access to the de-characterized TV BOX, however it still needs the memory device to boot. In order to transfer the image to the internal eMMC, the Linux terminal must be opened and the command "*sudo /boot/install-aml.sh*" typed. After the necessary procedure the image will be transferred and confirmation is displayed such as in Figure 7.

Figure 7: Internal Memory Transfer completed (The Authors)

Once the process is finished, rootdev must be updated to refer to internal eMMC UUID. To do that we must access the "*armbianEnv.txt*" file in the "*boot*" folder. The file is displayed in Figure 8 with the detail of the UUID path. This is the same procedure as listed in section 4.1.

Figure 8: Changing rootdev (The Authors)

4.4. Updating and maintain the system

When the image is installed, it is very likely that it will need updates or have some broke packages. The lack of updates can result in error executing some commands and affect the hardware performance. To perform the update/fix procedure the following commands are used:

- *"sudo apt update && sudo apt upgrade –y"* – for upgrades
- *"sudo apt --fix-broken install –y"* – to fix dependencies
- *"sudo apt install –f"* – to install missing libraries
- *"sudo apt autoremove && sudo apt autoclean"* to search and clean broken packages

4.5. Modifying the Image
After the image is installed and updated, in the case of the EducaBox image, some programs and resources needed to be removed since they were not going to be used. In the case of the Armbian image Google Chrome web browser was installed. In both cases Node-RED server needed to be installed.

To remove programs from the image, the command line that needs to be used is "*sudo apt remove --purge (package name)*". In order to find the package name of the program to be uninstalled, two methods are possible: a) *"apt list --installed"* will return all the installed programs and you can get the package name of multiple programs at the same time or b) *"apt list --installed | grep (app_name)"* that will return a single line of result if you are targeting a specific program.

To clean the system from packages and dependencies that are not used, the command *"sudo apt autoremove"* is used and to make sure no configuration files are left the command *"sudo apt clean"* is needed.

To install resources, such as the web browser Google Chrome, you will need *"wget"*, so first type on the terminal *"wget install"*. After that you must first download the installation package from the internet and soon after it starts pause or cancel it, copying the installation link from the browser. Then you use the command "*wget (copied link)"*. After the download is complete, the terminal will display an address, that you will need to copy to insert into the command *"sudo apt install (copied address)"*. After the installation is complete you will use the same name on the terminal to run the program. Some programs may require administrative privileges to run, in those cases the command line must be begin with *"sudo"*.

5. Results and Discussions
After the procedure that was developed above in section 4, it was possible to achieve the de-characterization of the TV Box and the proper installation of the EducaBox image. The image from before the procedure can be seen below in Figure 9, with the pirate system still installed as received by the university from the "*Receita Cidadã*" program.
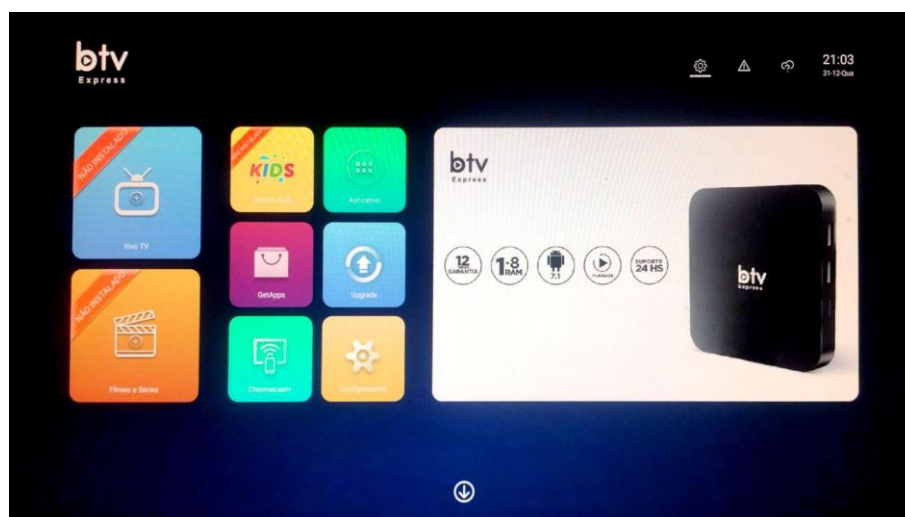

Figure 9: Screenshot before the procedure (The Authors)

After the procedure was Applied, it was possible to install the EducaBox image on the TV Box as seen in Figure 10 below. It was possible to configure and install the necessary programs to run the Node-RED server. The image would run the firmware, connect to the internet via the Wi-Fi adapter and run programs correctly, however it was

not possible to complete the processes to transfer the image to the internal memory, with all modifications remaining on the external memory device as it was required to boot up the hardware.

Figure 10: Screenshot after the de-characterization process (The authors)

At the end of the setup and adjustment procedures, all the unnecessary apps were removed to achieve the optimum system performance and maintain internet access, vital to the needs of the future use of the hardware on the Digital Twin project. However, even with the correct UUID boot procedure being executed to transfer the image to the internal eMMC memory, the system would not boot up without the external memory device.

A proposition was made where the issue should be with the image size and extra coding, native to the EducaBox project and its end use. For that purpose, a new image was then used, and with a very similar process to the one described in section 4, an Armbian image was installed. Since the image is lighter, it was possible to complete all the steps of the configuration and setup procedure, including the transfer to the eMMC internal memory. However, some features such as the Wi-Fi adaptor are still not working properly, and the hardware is using a cabled connection, which is not ideal for remote locations but usable for now.

5.1. Challenges

During the execution of the research project the biggest challenge was to start the program directly from the internal memory. All attempts to do so by changing the UUID address was unsuccessful. Even though the configuration was correct on the file, the system would still seek the external memory device to boot.

An attempt was made to change the UUID in the external memory device before it was transferred to the TV Box, however this information was changed once the image was installed. Another attempt was made by manually "dragging and dropping" all files and folders, including the boot folders and files however the system would still look for the external memory device. Attempts were also made to change the "*rootdev*" through the Armbian-config, however the option to change the UUID was not displayed with this method. Another procedure of installation of the image, via *"Armbian-install"* was tested, but this procedure presented failures and errors in recognizing USB devices and even the main memory.

Since some of the issues could be related to the use of the wring DTB file, that would not correctly relate to the hardware specifications, some other possibilities for DTB files were used, but all yielded a worse end result than the image mention in section 4.1.

Since we had several TV Box units to test, we moved to another component. This is where we found out that even though these TV Box where all model BTV E10 the hardware between them would vary significantly. We were able to notice different processors, GPUs and base operational system. This made it impossible to create a "one-fits-all" procedure to de-characterize the hardware and a more comprehensive one was needed that would incorporate the different hardware configurations that are common, thus the added configuration for the UUID and *rootdev* files that is presented in section 4.

6. Conclusion and future work

This research set out to provide a suitable replacement to computing systems such as Raspberry Pi promoting the use of hardware that was supposed to be discarded and destroyed by the Federal Government. The existing project allowed us to achieve that goal, with a working Armbian pure image being tested now with the Node-RED server for the main Digital Twin project of the laboratory.

The procedure created was detailed enough, and broad enough, to cover all the different hardware changes that the same model of TV Box can present and a working image with good performance was delivered. The shift to the Armbian image was fundamental to install the image inside the internal memory and now we have a Node-RED server running with the same program as the PC. Some details still need to be attended as detailed in the discussions, but this does not affect the de-characterization process and usability of the new hardware in the scenario intended of Industry 4.0 automation. The main delay that did not allow for a full decentralized structure with individual units were the issues related in section 5.1.

Research on the area will continue, focusing on solving the small issues that still exist in the image. The networking of several TV Box is also another point of interest for the future, to allow for Fog and Edge Computing.

Acknowledgements

References

Armbian. (2024). *Armbian Documentation*. https://docs.armbian.com/

Armbian. (2025). *Device tree overlays—Armbian Documentation*. https://docs.armbian.com/User-Guide_Armbian_overlays/

Bajic, B., Rikalovic, A., Suzic, N., & Piuri, V. (2021). Industry 4.0 Implementation Challenges and Opportunities: A Managerial Perspective. *IEEE Systems Journal*, *15*(1), 546–559. IEEE Systems Journal. https://doi.org/10.1109/JSYST.2020.3023041

Ball, S. (2002). *Embedded Microprocessor Systems: Real World Design*. Elsevier.

Chase, O. (2007). Sistemas Embarcados. *SBA Jovem*, *10*(11), 13.

EducaBox. (2025). *educabox—Overview*. GitHub. https://github.com/educabox

Mathe, S. E., Kondaveeti, H. K., Vappangi, S., Vanambathina, S. D., & Kumaravelu, N. K. (2024). A comprehensive review on applications of Raspberry Pi. *Computer Science Review*, *52*, 100636. https://doi.org/10.1016/j.cosrev.2024.100636

Rikalovic, A., Suzic, N., Bajic, B., & Piuri, V. (2022). Industry 4.0 Implementation Challenges and Opportunities: A Technological Perspective. *IEEE Systems Journal*, *16*(2), 2797–2810. IEEE Systems Journal. https://doi.org/10.1109/JSYST.2021.3101673

Biographies

Guilherme Henrique Pinheiro is currently a student of the technical course of Electronics integrated to High Scholl at the IFSP Campus Sorocaba, started in 2022 to be concluded at the end of 2025. Since 2024 holds a scholarship of the Scientific Initiation program PIBIC Junior CNPQ/Reitoria Unesp at the Sorocaba Campus of UNESP.

Vitor Mendes Caldana began his career with a technician course in Electronics from Liceu de Artes e Ofícios in 1999, followed by an undergraduate degree in Electronic Engineering from Universidade Presbiteriana Mackenzie in 2004. In 2016, finished his M.Sc. in Industrial Engineering. Since 2023 is a student at UNESP to obtain his Ph.D. in Electronic Engineering in the Industry 4.0 field. In 2014 began his teaching career in FIEB as a substitute teacher, followed by an associate professor position for the Technical Course of Electronics. In 2016 became a full-time professor by joining IFSP, moving to the Sorocaba Campus to implement the Electronics High-School Technical Course. In 2018 started the Research Group in Industry 4.0 at IFSP and has been its leader since. Between 2019 and 2020, along with his colleagues, designed and implemented the first Post-Graduate Program in Industry 4.0 of IFSP at the Sorocaba Campus. He is currently involved in research projects in Industry 4.0. https://lattes.cnpq.br/8361188962318020.

Eduardo Paciencia Godoy received the B.Eng. degree in Control and Automation Engineering at Itajubá Federal University (MG-Brazil) in 2003 and the M.Sc. and Ph.D. degrees in Mechanical Engineering at University of São

---

Paulo (SP-Brazil) in 2007 and 2011. Currently he is an Associate Professor of Unesp (SP-Brazil) with research interests in IoT and Industry 4.0. http://lattes.cnpq.br/0072632067545698.