

Using IoT to Dynamically Test Smart Connected Devices

Carmen Cristiana Cazacu ^{1,*}, Florina Chiscop ¹, and Dragos Alexandru Cazacu ²

¹University Politehnica of Bucharest, Faculty of Industrial Engineering and Robotics, Department of Robots and Production Systems, 313, Splaiul Independentei, Bucharest, Romania

²PTC Eastern Europe SRL, Academic Team, 319, Splaiul Independentei, Bucharest, Romania

Abstract. The aim of this paper is to introduce and validate a new concept of testing smart connected devices (SCD) that have inertial measurement units (IMU) or gyroscopic and accelerometers, like smart bracelets, smart watches or even mobile phones after they have been assembled, in their way to the packaging area.

The devices will be tested dynamically, while they travel on the conveyor using the IoT (Internet of Things) and some expected values for acceleration and angle variations. Using IoT to dynamically test electronic SCD, while they travel on a conveyor after the assembly process had been completed, will reduce the time to market and will exclude the need for a testing area. Once the SCD arrives at the packing area, the system will know and sort the device that are QA (Quality Assurance) passed or failed (the devices that will not send the expected values to the system).

In order to prove this concept, we created a SCD that use an IMU connected to a Raspberry PI, boxed together. The PI is connected to ThingWorx (IoT platform that includes machine learning) and the box travel on different types of conveyor belts in order to validate this new concept.

1 Introduction & State of the Art

As most of the IoT platforms comes with analytics and machine learning, it is easier than ever to implement fast, different solutions that replace human interaction with the products and lower the processes time, eliminates any human error and that is scalable for any future products. Various types of data are available in manufacturing systems or waste management systems, and should be properly collected, encoded and transmitted to the cloud application [1, 2, 3, 4].

IoT is a new varied multi-technologies application that faces a main challenge in working complex mixed nodes network [5, 6, 7, 8].

Nowadays, Internet of Things (IoT) have been used in various fields such as: smart factories, smart homes, smart devices/cars, and smart cities [8, 9, 10, 11]. The IoT market is developing, and IoT devices, infrastructure, and applications are affecting industrial fields and daily living [12].

* Corresponding author: carmen.cazacu@upb.ro

All smart, connected devices, from home uses to manufacturing environment, share three major parts: physical elements (for example mechanical and electrical parts); smart elements (data storage, sensors, microprocessors, software, controls, an implanted operating system, and a digital user interface); and connectivity elements (ports, protocols, antennae, and networks that enable communication between the product and the product cloud, which runs on remote servers and contains the product’s external operating system) [13, 14, 15].

To create products and get them to customers, the new capabilities of smart, connected products alter every activity in this value chain: research and development (or engineering), IT, manufacturing, logistics, marketing, sales, aftersales service, human resources, procurement, waste management and finance [13,16].

Even if the smart connected devices are using or are IoT capable, as they all have a connectivity option and also software, most of them are not using IoT in the Quality Assurance (QA) processes and these processes are still made manually or in special designed stands. As for some of the devices the testing processes takes longer than the assembly processes, many smaller companies skip the QA processes and randomly test a few products from each lot. This is the reason that many smart band, smart watches and even less performant mobile devices companies sell so many fail products on the market.

Although the cost involved in developing an IoT platform is no longer a problem, most of the big companies are having this concern about using IoT to connect these devices to their system, because they are not up to date about the real cost from today. The second concern is the security of these platforms, but that is now extremely rigorous. If the device doesn’t need to connect to the IoT platform outside the production or QA area, the security doesn’t even need to be discussed, because the minimal requirement for this application may be a local connection to the IoT platform that will not allow the devices to connect outside the facility.

2 Testing smart connected devices

In this paper, we considered various existing solutions using IoT in a Smart connected devices manufacturing or processing architecture, to optimize the flow of the devices from the assembly line to the packing area [17, 18, 19, 20]. The standard, traditional process is presented in the diagram below. As you can see in the figure 1, QA process is detailed, between the assembly line exit and packing area, into 4 main activities that are run by humans (in many cases, depending of the device that is being produced, these can be up to more than 10 working posts).

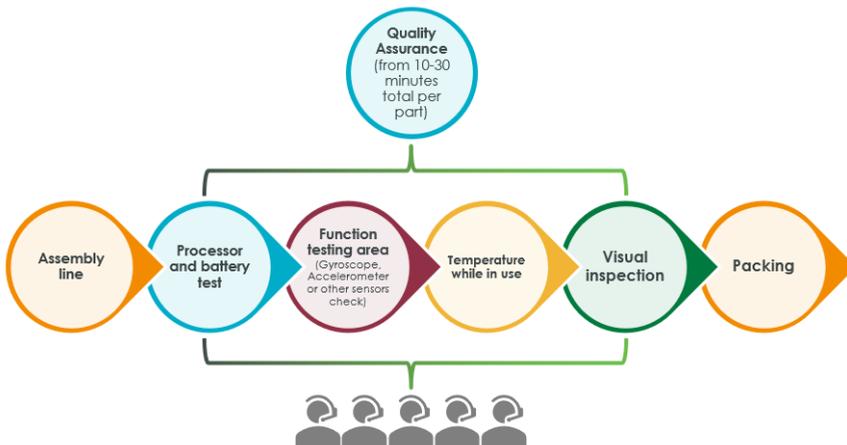


Fig. 1. The standard traditional QA process in the production flow.

There is a lot of time wasted in the QA process and depending of the flow, 4 to 10 working posts where the device is stopping for these processes. Most of these details can be read using IoT. By definition smart connected devices have a connectivity and some software that runs it. If the device gets a few more lines of code, it can also act as an edge connected device for an IoT platform or simply use representational state transfer (REST) calls to the platform, that will allow the device to search for the server, at start-up, and if the connection is validated, the device will stream its values to the platform.

As most of the companies would not like the devices to run this codes each time they start, the code can be used until the device is going into normal use state for the first time (the device is connected to the phone if we are considering a Smart Band or a Smart Watch, or the first setup was finished for a phone).

Using IoT will validate the battery (it can also monitor the wireless charging – if available – and the power consumption of the device thru the dynamic testing process), the processor, RAM memory and temperature can also be monitored, and the greatest part is that all of these, including the specific functionality can be tested dynamically while the device is traveling on a conveyor, designed to test the Gyroscope or Accelerometer, or other sensors that need to be tested, as magnetometer, altimeter, barometer etc. In figure 2 is the framework that we developed, to test the device dynamically while it travels on the conveyor.

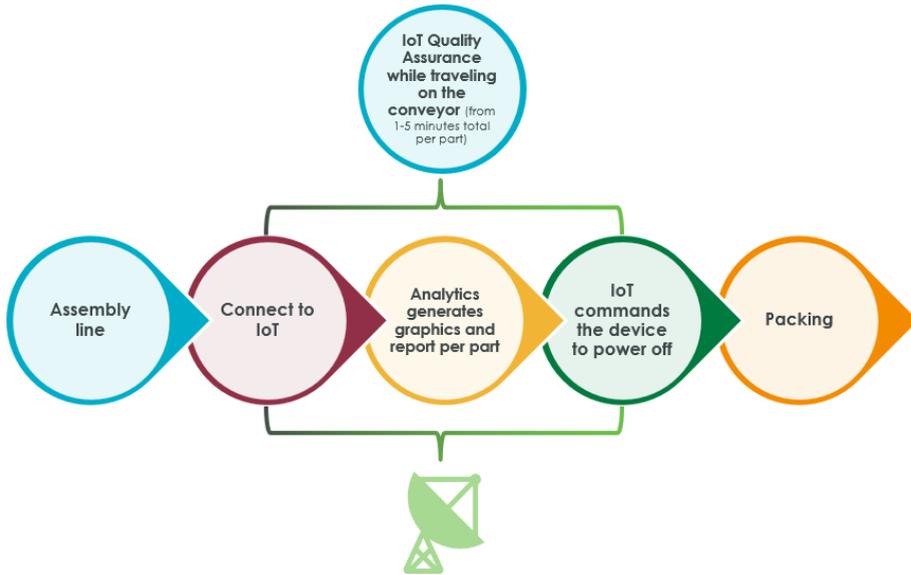


Fig. 2. The framework that is developed in this article, to test the device dynamically while it travels on the conveyor.

As this is just a framework and the intent is to fit any use case for a company that creates these devices may have, we didn't try to focus on a specific use case, and the intent of the article is to introduce this concept of dynamic QA testing.

Even if the process is similar for mainly any production line that manufactures the SCD that we address, the architecture will be different. For a smart band the architecture will be quite simple and the main difference from a classic QA will be the conveyor belt and the IoT platform used to receive and analyse data.

For a mobile device, the architecture needs to be different, it needs multiple custom assets, like the support for each mobile in some cases, custom 3D modelled physical shapes to read in the path to packing (to verify the face recon and camera accuracy), different luminosity areas and many other tools.

For any of these architectures, the framework will be the same and the Real Data Set will be generated by using Machine learning (like the Thing Watcher in ThingWorx or any other similar tool) on certified devices, that are used to define the normal behaviour of the devices.

In this case, the product, once is exiting the assembly line, can go over a wireless charger or any other tool that can initialize it. After the device is on, it will run the edge (or use REST calls) and check for the IoT connection, that will be available. Again, if the device doesn't find any response from the IoT platform the code will stop, and no data will be sent. The device, once connected to the IoT server, will send all its values continuously at a refresh rate set. All the values are being compared by the analytics of the IoT platform used with the expected ones. If the values are close enough and, in some cases, (temperature, processor load etc) not over a certain limit, the product passed QA and can be packed. Also, the IoT will store the records about the device.

To make the sensor readings more relevant this method will use conveyors, designed to stimulate the sensors. There will be small gaps between the conveyors or many direction changes of the conveyor on all axis to make the device read accelerations and direction changes, to check the gyroscope and accelerometer. There may be some variable magnetic field to stimulate magnetometer, a "fog room" could easily excite the humidity sensor and hot area or hot/cold blowers for temperature readings etc. In figure 3 we designed a conveyor belt custom made to test a specific smart band (Xiaomi Mi Band 5), but in a similar way any other device can be transported for dynamical test.

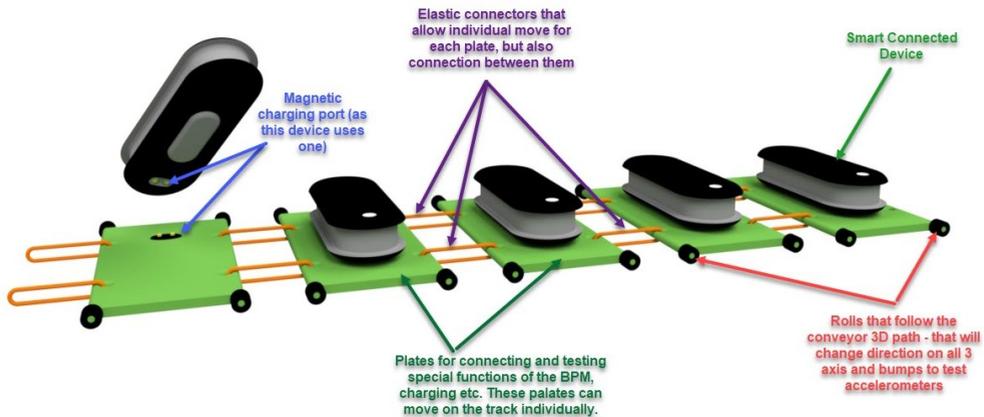


Fig. 3. A conveyor belt custom made to test a specific smart band.

This custom conveyor belt has fix magnetic charger that will attract, orient, and fix each band to each of its plates. Once fixed, the SCD will start charging and each of them will boot and connect to the IoT platform. As each plate is connected to its neighbours with elastic connections (can also be magnetic or any other type that will allow the plates to move in all directions while keeping their order), each plate holding a SCD will travel along the path.

The path needs to be designed so all the main functionality of the bands can be tested (the path can have up and downs, corners even turn the plate upside down). Axial vibrations, small falls and other needed physical situations can be added during the path.

To prove this concept, we used a Raspberry PI, with an inertial measurement unit (IMU), both placed in a 3D printed case that allows ventilation. This device is presented in figure 4. The testing was not carried on a real conveyor belt, we just simulated manually the same kind of impulses as a travel conveyor with corners and gaps. For connectivity it is used a network cable, but also a USB wireless card. The Raspberry PI will run an edge that will allow it to

connect to ThingWorx (a top IoT platform with analytics) and send the readings at a specific rate.

The edge service was custom made for this application, but it is not the purpose of this paper, as ThingWorx allows also Rest API and other connectivity protocols.

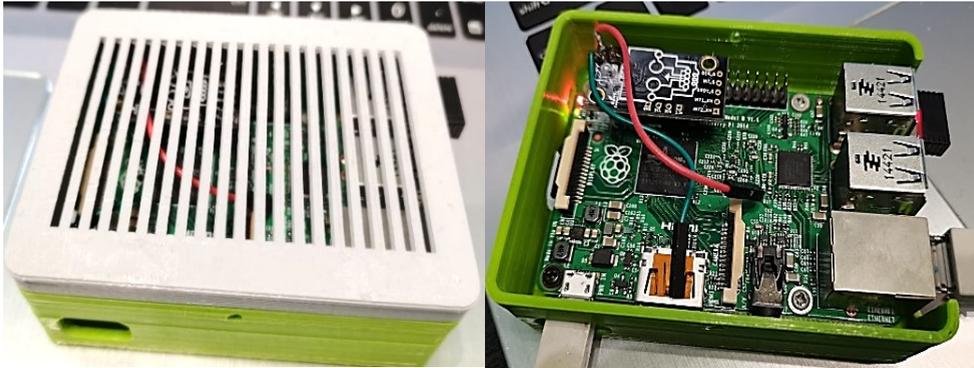


Fig. 4. The device used to test smart belt – The Raspberry PI with an inertial measurement unit (IMU) connected to ThingWorx (a top IoT platform with analytics).

The Raspberry PI has installed on it an edge (designed for this application) that streams the data to the IoT platform, allows us to connect to a wireless or LAN network and to set the frequencies of the data. The IMU unit is installed directly on the board, and it allows the device to read any movement of the assembly. Even if the IMU is reading angles and accelerations, we decided to use only the rotation on each axis, for this paper. The accelerations could be read, sent and analysed in the exact same way, but in order to have a clear simple mashup, these values were not shown.

The Raspberry PI also has a Wireless USB installed to allow it to connect to any network. All the connectivity and data are managed from the edge service, but as ThingWorx permits almost any connectivity protocol, these functions and data can be managed also on the Raspberry (if for instance the IoT connection will be using Rest Calls).

For this use case it was checked the rotation on the 3 axes, altitude, pressure, temperature, but any other data can be analysed in the same way. It was also monitored the Raspberry PI values of the processor, memory, and disk usage.

ThingWorx will get these values at a rate of 0.5 seconds and will use the custom defined services and events to show in a simple mashup the live values, graphics, and alerts for different scenarios, defined for each failure. ThingWorx has also machine learning that can be trained to understand the normal behaviour of a smart connected device on its way to packing, on the conveyor, and trigger an alert if any of the following items will have different values then the expected ones (out of range values, NaN reading error, rotations or accelerations that don't change). As these devices use already known and calibrated sensors, the fails will be easy to be detected. The value doesn't change means a sensor is not reading the value, the value changes, but it is too low or too high, compared to the normal working device can also be set if needed.

In ThingWorx we had to script some services and trigger some events in order to manage the data received from the prototype device. This is how the mashup, that is shown to the users, will present relevant, important, and easy to understand information.

In figures 5 and 6, are captured two screenshots to show the values changing as a conveyor curved trajectory was simulated, by moving the real device, by hand. The device is running as expected, as the values change on all 3 axes and all the parameters are in the normal range (defined by the use case).

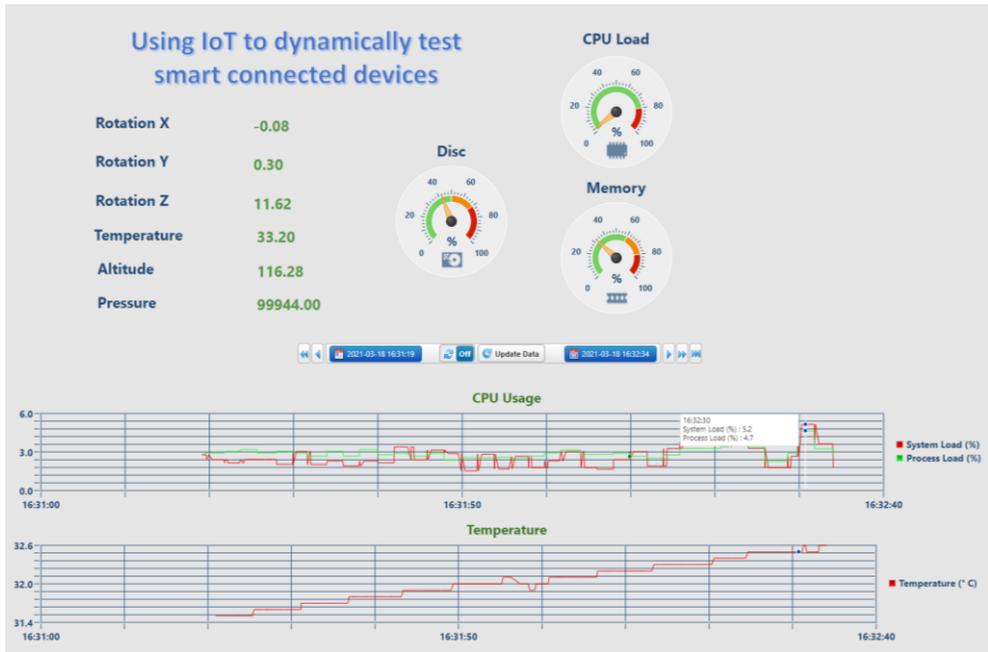


Fig. 5. A screenshot from the ThingWorx mashup with the values registered during the smart belt tests.

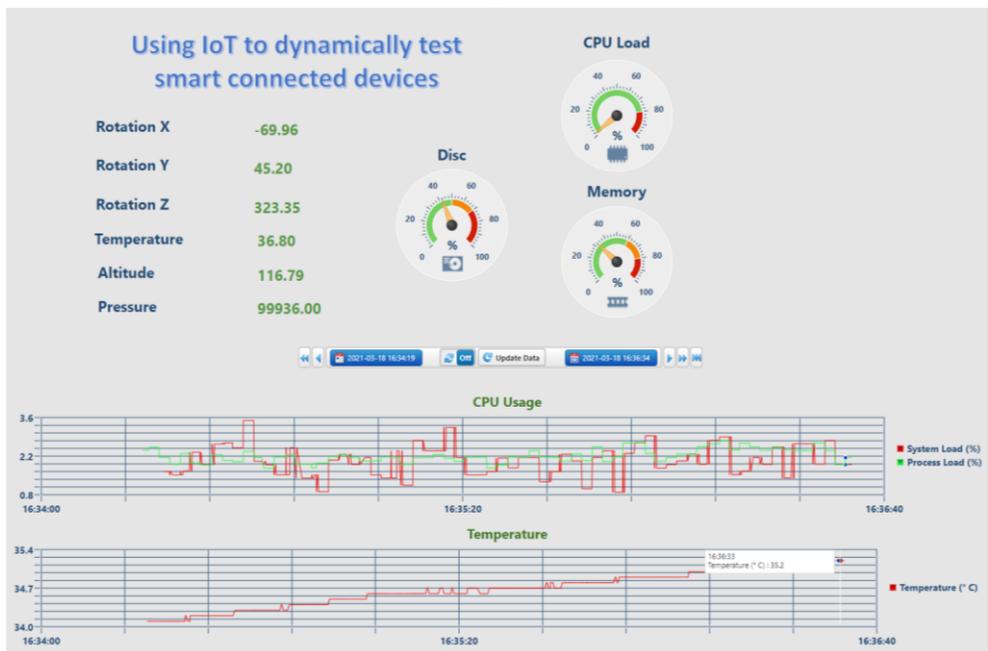


Fig. 6. A screenshot from the ThingWorx mashup that show us that values are changing during the smart belt tests.

If the device doesn't send any of the data, or the data remains constant, ThingWorx was requested to show a red mark on that value. In figure 7 this situation was simulated on the physical device by stopping rotation on X.

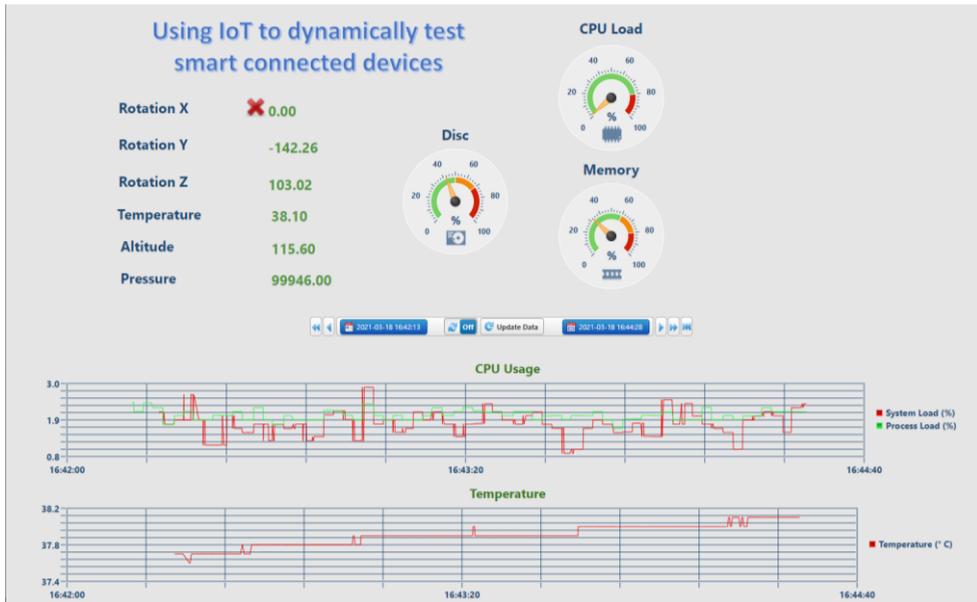


Fig. 7. A screenshot from the ThingWorx mashup with the error in case that device doesn't send any of the data (stopping rotation on X).

Alerts were added for the rest of the values and an over-temperature value is shown in figure 8 (the vent gaps were covered with a paper to simulate this issue on the real testing device).

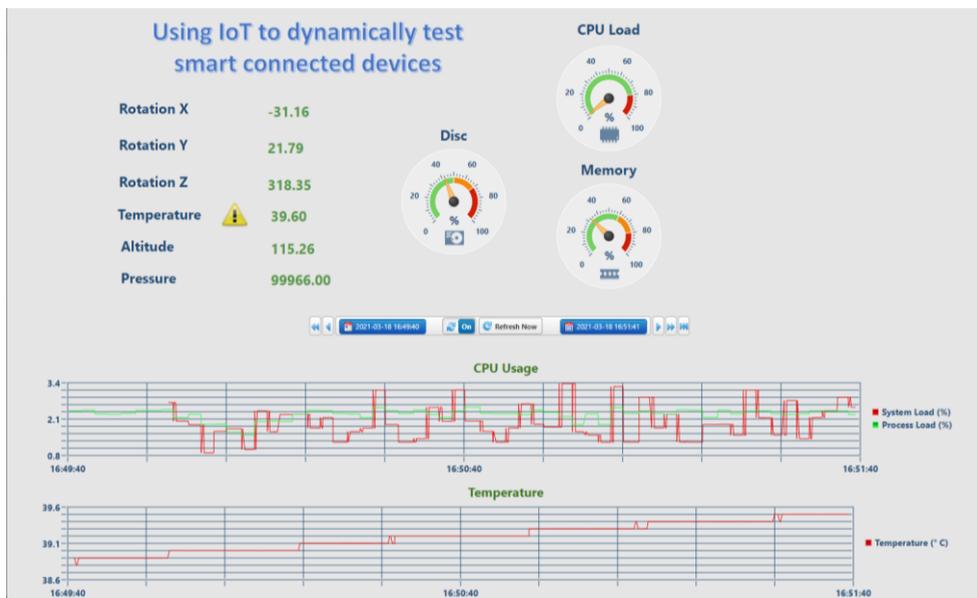


Fig. 8. A screenshot from the ThingWorx mashup with the alerts for over-temperature value.

So, this way, we managed to have a fully functional IoT system that can test smart connected devices as they travel on a conveyor belt from the assembly line to the packing area. Of course we didn't spoke here about the edge connection, how each device will be identified by the platform, how long the travel needs to be, as we just presented here the

concept and all these custom information need to be tested and adapted on each device and to each production line.

3 Conclusions

By using IoT in the quality assurance process of smart connected devices that are supposed to read and record movements and other physical parameters, such as smart bands, smart watches or mobile devices, the production time can be lowered. Also, a company may reduce the human interactions with the devices and replace human operated stands that do all the testing with a conveyor and the IoT. This will lower the production total costs of the product, as replacing the human workforce with a smart dynamic testing environment will reduce the cost with workforce (usually an important cost in any production flow).

This process won't generate any security breaches, as it can use a performant IoT platform, like ThingWorx, or simply allow connections only for the devices that are inside the factory and that are not connected to any other device or network. We used the app key generated in ThingWorx to secure connection between the Raspberry PI and the ThingWorx platform.

By using an edge connection on the first initialization, the customers will not be impacted in any way of this extra code, as this can be set to run only on first initialization or until the device is setup for the first time. This way the device will not use the battery resources to search for the IoT platform once the customer started to use it.

Our future plan is to add AR in this process, so using computer vision and the IoT data to be able to identify and remove from the conveyor the devices that are fails, so only the pass devices will get to the packing area.

IoT could be a great tool for getting feedback from the products, once sold, allowing us to learn about the way they are being used, by the real end customers, but we have this in attention for future research. By allowing the smart connected devices to connect from time to time to the IoT platform (for example when it is charging and has an internet connection available) and send relevant data back to the seller, many issues can be diagnosed and prevented (preventing the failure for example). Also, if the manufacture plant knows the exact usage of the product, it can add or improve or exclude some of the functionalities to optimize the cost, user experience and client satisfaction.

References

1. Y. Liu, L. Wang, X.V. Wang, X. Xu, L. Zhang, *Int. J. Prod. Res.*, **57(15-16)**, 4854 (2019)
2. C.E. Cotet, G.C. Deac, N.C. Deac, C.L. Popa, *Sustainability*, **12(5)**, 1839 (2020)
3. C.L. Popa, G. Carutasu, C.E. Cotet, N.L. Carutasu, T. Dobrescu, *Sustainability*, **9(11)**, 2064 (2017)
4. D. Popescu, D. Anania, C.E. Cotet, C. Amza, *Int. J. Simul. Model*, **12(2)**, 82 (2013)
5. R. A. Sadek, *An Agile, Egypt. Inform. J.*, **42(2)**, 13 (2018)
6. E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, *IEEE Trans. Ind. Inform.*, **14(11)**, 4724 (2018)
7. W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, *IEEE Access J.*, **6**, 6900 (2017)
8. R.S. Alonso, I. Sittón-Candanedo, R. Casado-Vara, J. Prieto, J.M. Corchado, *Sustainability*, **12(14)**, 5706 (2020)
9. Y. Laili, S. Lin, D. Tang, *Robot. Cim-Int. Manuf.*, **61**, 101850 (2020)
10. J. Leng, P. Jiang, *J. Intell. Manuf.*, **30(3)**, 979 (2019)
11. C. Wu, A.N. Toosi, R. Buyya, K. Ramamohanarao, *IEEE Trans. Cloud Comput*, **9(1)**, 182 (2021)
12. J. Jeon, J. H. Park, Y.-S. Jeong, *IEEE Access J.*, **8**, 96899 (2020)

13. M. E. Porter, J. E. Heppelmann, Harvard Business Review - How Smart, Connected Products Are Transforming Companies, (2015)
14. T. Qu, S.P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, G. Q. Huang, The Int. J. Adv. Manuf. Technol., **84**, 147-164, (2016)
15. C.E. Cotet, C.L. Popa, G. Enciu, A. Popescu, T. Dobrescu, Int. J. Simul. Model, **15(1)**, 5-16, (2016)
16. C.L. Popa, C.E. Coteț, D. Popescu, M. Solea, S.G. Șașcîm (Dumitrescu), T. Dobrescu, Waste Manag. Res., **36(7)**, 653-660, (2018)
17. M. Fleischmann, K. Kloos, M. Nouri, R. Pibernik, *Eur. J. Oper. Res.*, **286**, 250-266, (2020)
18. D. Berthelot, C. Raffel, A. Roy, I. Goodfellow, ICLR 2019, *Understanding and improving interpolation in autoencoders via an adversarial regularizer*, (New Orleans, USA, 2019)
19. Y. Ai, M. Peng, K. Zhang, Digit. Commun. Netw, **4(2)**, 77-86, (2018)
20. I. Sittón-Candanedo, R.S. Alonso, J.M. Corchado, S. Rodríguez-González, R. Casado-Vara, Future Gener. Comput. Syst., **99**, 278-294, (2019)