Realizing Seamless Integration of Sensors and Actuators into the IoT

Reinhardt Karnapke[1,2](✉) and Karsten Walther[1]

[1] Perinet GmbH, Berlin, Germany
{reinhardt.karnapke,karsten.walther}@perinet.io
[2] Brandenburg University of Technology, Cottbus-Senftenberg, Germany
karnapke@b-tu.de
https://perinet.io

Abstract. Ever since the inception of wireless sensor networks more than twenty years ago, we were promised networks consisting of dozens, hundreds, or even thousands of nodes with sensors and/or actuators that would use self-X properties in order to form a functioning network by themselves. User interaction would not be necessary, or at least be absolutely minimalistic in all cases, including the later addition of more nodes. This promise has been ubiquitous regardless of network type, be it wireless sensor network, home automation, Internet of Things or Industrial Internet of Things.

Even though these self-X properties have long been promised, planning a deployment still offers a lot of obstacles to this day. One of these obstacles manifests in the form of a break in semantics when looking top to bottom at IoT/IIoT deployments. While process management level, operations management level and corporate management level are almost everywhere IT based, sensors and actuators are often still analog, which results in a mixture of systems in between.

In this paper, we describe the problems that arise when trying to close this semantic gap in detail, explain how taking the last mile in IIoT by enabling sensors and actuators with network capabilities removes these problems, and show additional benefits that stem from our solution. Moreover, we describe the inception of the periCORE, the first module which enables the integration of sensors and actuators into Single Pair Ethernet (SPE) worldwide.

Keywords: Internet of Things · Industrial Internet of Things · Seamless integration

1 Introduction

Wireless sensor networks have been envisioned and much researched since the 1990 s. Papers about smart dust (e.g. [1,2,4]) have given rise to the assumption that within a short amount of time it would be possible to deploy hundreds or

S. Paiva et al. (Eds.): SmartCity360° 2021, LNICST 442, pp. 688–701, 2022.
https://doi.org/10.1007/978-3-031-06371-8_44

thousands of nodes within a short interval, possibly even throwing them from planes, without having to configure them manually. Instead, nodes should be organizing themselves automatically, and deliver measured valued to the user via some gateway or similar mechanism.

Even though the number of nodes is usually smaller, the same promise, nodes that can be added almost without any effort from the user, has been made in other areas of IT. Home automation products, from smart light bulbs to automatic blinds, should make the life of the inhabitants more relaxed and should be easy to (re-)configure. Adding an intelligent fridge with internet connection leads to the Internet of Things (IoT). Once more, adding a device should be easy and not involve a lot of configuration effort. However, the opposite is often true. Smart homes need to be built and configured by experts, and intelligent fridges come with a manual the size of a telephone book.

Still, sensors and actuators are being produced, shipped and installed in ever increasing numbers. This does not primarily stem from private use but largely from industrial purposes. They are used in realization of the Industrial Internet of Things (IIoT) to take over numerous tasks that are currently still done manually. However, there are a number of pitfalls that need to be mastered between the acquisition of new sensors/actuators and the existence of a continuously working system:

1. Process management level, operations management level, and corporate management level are almost everywhere IT based nowadays. Yet, they need to interact with the Fieldbus systems below in a non-IT based way, resulting in a semantical break (see Fig. 1). Therefore, the goal should be to be able to attach sensors and actuators directly to IT-based systems in order to close this semantic gap. This leads to a drastic simplification of the whole architecture, enabling a number of further improvements.
2. A large number of sensors and actuators leads to a lot of Data that needs to be transmitted from the sensors, usually to a central decider, where it can be interpreted. Thereafter, a decision must be made whether some things changed, and if such changes require actions, it must be decided which actuators should perform which concrete actions. Finally, these decisions must be transmitted to the actuators.
   Such a centralized approach does not scale well, as all actions described above increase the Data load massively and can lead to overload situations a) in the network due to the transmission of large amounts of raw Data from the sensors to the decider, b) at the centralized decider itself, which has to handle and interpret huge amounts of Data, and c) in the network during the communication of control commands, possibly augmented with Data that originated at the sensors, from the decider to the actuators.
3. For many applications, functional properties like correctness of values are not all. Rather, non-functional properties like runtime, latency, and jitter play a big role. At least the latency increases with the network size, jitter depends on the used medium access control protocol and the network load. A low

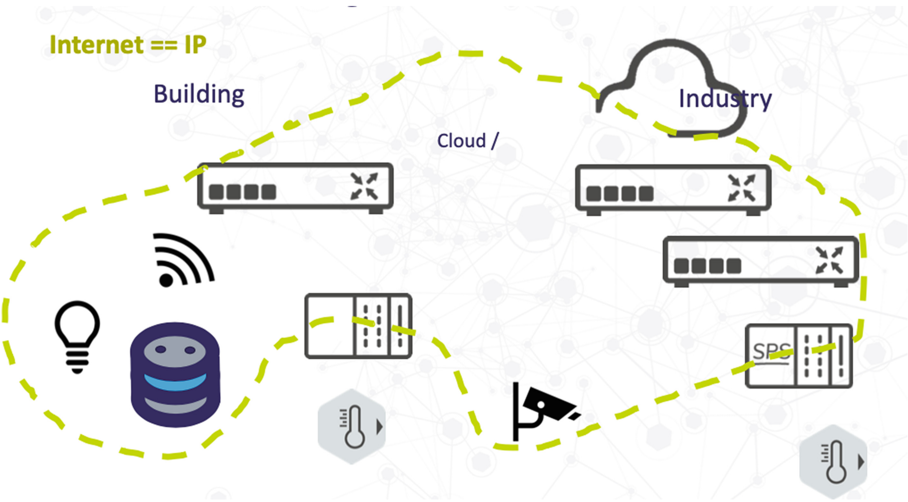

Fig. 1. Current state of connectivity. While upper layers are IT based, sensors and actuators are still analog and often connected through Fieldbusses.

latency despite a large network requires expensive high speed networks and specialized real-time capable medium access control protocols.

4. The fact that a lot of sensors supply only analogous Data or bitfields and can only be addressed by the port of a machine they are attached to leads to a complicated installment procedure and high deployment overhead. Therefore, these installments usually need to be carried out by experts.

In this paper we describe a way to simplify these four points, in order to allow users an easy entry into IoT and IIoT or an easy retrofitting. We started by closing the semantic gap between IT-based and non-IT-based systems with the introduction of network capabilities into the sensors and actuators. This is achieved though the design of the periCORE, the worlds first module for integration of sensors and actuators into Single Pair Ethernet (SPE) However, on one hand this was not enough, on the other hand it opened more possibilities than those we initially aimed for, resulting in quite some follow-up work.

The remainder of this paper is structured as follows: Sect. 2 looks at the problems described above one by one and produces a theoretical solution for each of them. Section 3, in contrast, focuses on the new challenges that arise when trying to apply the theoretical solutions to the real world. In Sect. 4, we show the hard- and software we designed at Perinet, taking into account the insights from both the theoretical as well as the practical challenges, the hardware and software. We finish with a conclusion and future work in Sect. 6.

2 Seamless Integration in Theory

Our main idea was to take the term "Internet of Things" literally. We are talking about networking, meaning the connection of devices where every part of the route, down to the last sensor, is made network-capable. This is a deliberate distancing from traditional, bus based sensors, moving on to network based ones. This results in a role switch for the sensors, as they turn from purely passive devices that get read from time to time using polling into smart communication partners that can decide on their own if and when communication is necessary, and even initiate this communication themselves. This solves a number of the problems mentioned above directly:

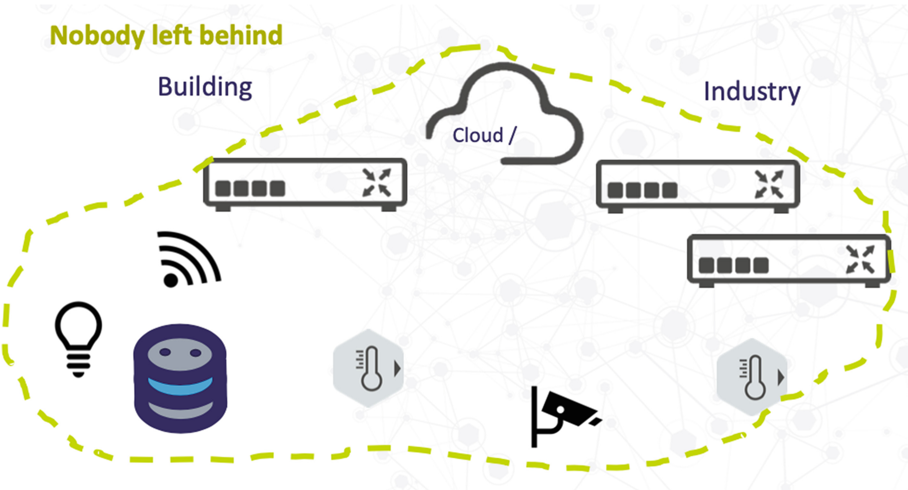

Fig. 2. Envisioned future connectivity: network connectivity down to the last sensor and actuator

1. The network capabilities of the sensors enable a direct communication from process management level, operations management level and corporate management level to them. The semantic gap is removed and no more translations are necessary. Due to the fact that each sensor has its own IP-Address, they can be directly addressed from any level using HTTP (see Fig. 2). It is no longer necessary to know to which port on which machine they have been connected, and to communicate with this machine first. This also enables a rethinking in the design process for future IIoT installments: Instead of starting with the focus on the Fieldbus (bottom up), systems can now be designed top down.
2. As the application does not change, the amount of Data that is produced by the sensors and must be processed does not change. However, this Data no

longer needs to be transmitted through the whole network in their raw, large form. Adding not only network capabilities but also a little processing power to the sensors means that local preprocessing of the Data can be realized directly in the sensors, which not only enables a reduction in size but also local decisions whether the Data needs to be transmitted at all. Instead of periodic polling and data transmission to the central decider, we are now able to realize an alarm functionality, meaning that sensors only transmit their (already preprocessed) Data if changes occurred and, for example, a threshold was exceeded.

This results in an avoidance of the overload situation in the network that stemmed from the communication between sensors and central decider. Also, the local decisions relieve the central decider and remove the danger of overloading it. Moreover, the maintenance overhead is reduced, as there is no need to maintain a (widely) distributed system anymore.

As for the actuators, the same principle can be applied to them, in order to reduce the network load due to communication between decider and actuators, reducing the danger of overloading the network even further. Moreover, the preprocessing and Data reductions prevent the accumulation of expensive Data-Lakes, as only relevant information is transmitted by the sensors. In fact, it is even possible to move the decisions, which actions are necessary, completely to the sensors and actuators, removing the need for a central decider. Of course, for monitoring reasons, it might still be desirable to collect some information about the state of sensors, actuators, and network at a central location from time to time in order to enable easy access for humans. This is, however, no longer required for the general operating of the system.

3. When communication only takes place locally, the latency does not increase with an increase in network size, as most communication will stay within certain boundaries. This means that standard components can be used for the network, and there is no need to install expensive high speed networking hardware. Real-time capable medium access control protocols might still be necessary, but their realization is much easier if they are only necessary on a local basis with only the few necessary participants.

   This focus on local communication can be seen in detail in the following: A sensor decides autonomously how its measured values need to be interpreted. Instead of transmitting the values as it was done before, it already evaluates them and decides, whether they merit a reaction. If so, the sensor addresses the corresponding actuator directly. However, this is not a static one-to-one connection where a sensor is assigned an actuator beforehand. Instead, the smart communication partners, in this case the sensor itself, can choose the actuator that is interested in values that lie within a certain range and can react accordingly. This removes the detour via the central decider.

   Even more, not only the sensors are equipped with intelligence, so are the actuators. A single measured value, even one that has already been digitalized and boxed within certain boundaries, can have different origins. Often, only the combination of multiple values measured by different sensors can bring clarification. Being able to detect such connections has previously been the advantage of the centralized approach, as the central decider could fuse multiple received values to form its decision. In our approach we also want to be able to perform this fusion, albeit at a different location. This can be realized by coupling multiple sensors, as a sensor might be interested in the measured values of other sensors to finish the evaluation of its own values. Alternatively, it can be realized in actuators. An intelligent actuator can declare that it is interested in the values of multiple sensors and fuse these to decide whether it needs to act.

4. The fact that each device has a unique identity that is independent of the port it is attached to and even the place it is located enables an easy installation and configuration that no longer needs to be carried out by experts. Users that already have experience with the internet are used to opening their browser and configuring new devices there, meaning the users do not need to learn to work with a new configuration tool (even without internet connection the devices can be configured in the browser). As all devices are network capable in our solution, they can be simply connected to the local network. Once they are connected, they announce their presence and can be configured using the browser the user is used to. Also, due to the network capability of the sensors, they can be easily replaced or reconfigured at runtime. Moreover, the fact that all parts of the network are now IP-based enables a number of improvements for security. Standardized end-to-end encryption protocols like TLS [5] can easily be included, and now also encompass the "last mile", the connection to the sensors.

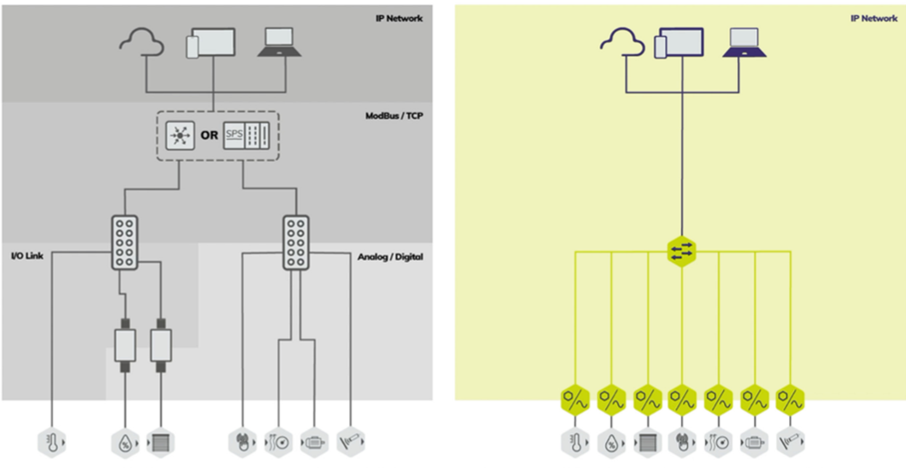

Fig. 3. Example of current setups (l) and future setup (r)

Figure 3 shows a comparison between a typical setup as might have been realized in the past (left) and the setup that we want to achieve with IP-based sensors or analog sensors retro-fitted with network capabilities. Whereas until now a lot of different interfaces and analog/digital converters are needed, a simple switch should suffice in the future when IP-based communication is realized down to the last sensor/actuator.

3 Seamless Integration in Practice

As wonderful, logical and simple as this idea seems, there is one question that you always need to ask when you want to build a new company based on a seemingly simple idea: Why has no one done that before? The answer, in our case, was that there were a number of obstacles that had to be overcome, concerning the technology as well as at the usability.

3.1 Challenges: Hardware

Starting from the bottom, one of the main challenges was the missing availability of standardized cabling and connectors for IP based sensors and actuators. This changed with the appearance of the Single Pair Ethernet (SPE) standard IEC 63171–6 which was proposed to feature the T1 interface from Harting in 2016 and was released on January 23rd, 2020 [3].

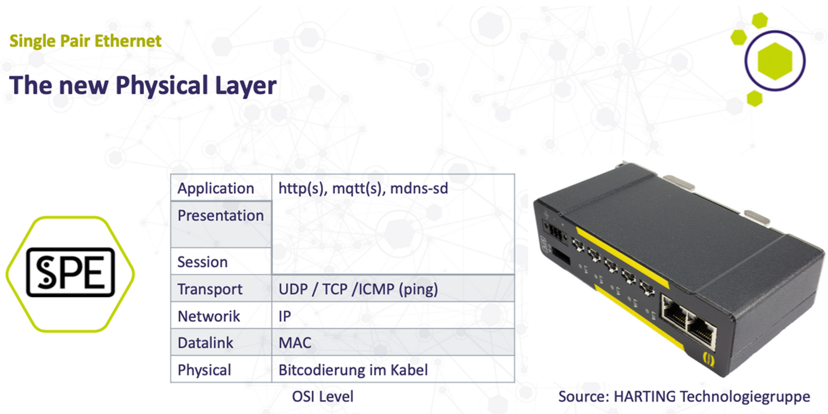

Fig. 4. An SPE switch from Harting Technology Group

Figure 4 shows an SPE switch from Harting Technology Group, which can be used to connect up to five SPE cables with a classic LAN.

Building upon IEC 63171–6, we started the design of our hardware. At the same time, software had to be taken into account and the development started in parallel because, as mentioned before, usability is also a main issue for IoT and IIoT devices.

3.2 Challenges: Software

In the following, we describe a few design decisions and challenges our software needed to be adapted to.

User Interface. Most of the potential users will not be IT experts, therefore they will not want to work with some strange setup program. However, almost everybody is used to use a web browser nowadays. If all configurations and all presentations of results could be done in the browser, user acceptance is expected to be high. Therefore, it was decided to use a browser based approach as described above. Browsers differ, though, and services that run on one browser might not be runable on another, resulting in exhaustive testing for different browsers.

Device Discovery. Even though IPv6 theoretically offers enough IP addresses for all embedded devices worldwide, it is not viable to print the IP of every node on its outside. Partly because the nodes shall be encased, partly because users should be spared the effort of copying long IPv6 addresses from all nodes into their computers manually. Therefore, we need device discovery mechanisms. Whether we can use DHCP is strongly dependent on the local configurations at the deployment site. For routers, there is the default IPv4 127.0.0.1 which can always be used but which default value should be used for sensors and actuators? As it should be only visible within the local network, a link local value would be good. However, for these to work a local name translation is required.

Name Translation. One of the major problems with the name translation we ran into is that it is not limited to local. This results in high delays. In combination with standard browser timeouts, this led to cases where our hardware was correctly connected but the browser did not show it because the discovery time was higher than the default browser timeout.

We also saw that IPv4 is still often preferred to IPv6 in browsers. A simple example which illustrates this can be found when using the Chrome browser. Chrome tries to connect to the internet using IPv6 when starting and if that initial connection fails, it switches to IPv4. While this is of course the correct behavior to enable the connection to the internet that is usually sought by the users, Chrome remains completely in IPv4 mode for the rest of its operation, oblivious to any IPv6 elements that might be connected or try to connect later. If you have only an IPv4 connection to the internet, you can not use IPv6 in your local network. This means that our sensors will not be discovered even though they are connected correctly, as Chrome does not support their link local addresses.

Another interesting observation that can be made here is that there are standards actually solving these problems, but most browsers do not fully implement these standards.

Zeroconf [6] already tried to address and solve all of these problems 20 years ago. However, the IETF working group dedicated to this problem was closed in 2004 due to their inability to reach a consensus.

4 Perinet

All of the ideas, challenges and features above described make for a nice academic thought experiment. However, it is even much more useful when completely realized and ready to roll out. For this reason, the Perinet GmbH was founded in 2018. Building upon the knowledge of the then upcoming Standard IEC 63171–6, Perinet already startet designing additional hardware and software. By now, we have designed a number of different modules with which the final gap in IoT can be closed. These include the periCORE, periNODE, periSWITCH, periSTART periLINE and periMICA.

4.1 The periCORE SPE Communication Module

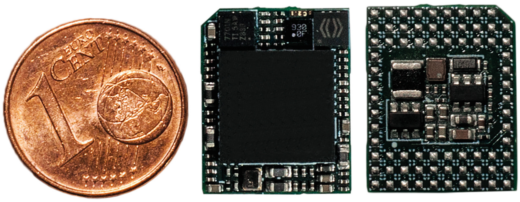

Fig. 5. The periCORE SPE communication module. Cent coin for scale

The periCORE (Fig. 5) is the central piece of the innovation and the main product of Perinet. It has about the size of a euro cent coin and can be used to make sensors and actuators directly network-capable. However, this would only be possible for the manufacturers of sensors and actuators.

4.2 The periNODE Smart Adapter

Even if sensors and actuators would be manufactured with network-capability in the future, a lot of factories, automated buildings etc. already exist with analogous sensors and actuators. As we know that these would need support for transitioning to completely IP-based solutions without replacing all existing sensors and actuators, the periNODE smart adapter was developed (Fig. 6). It is an adapter that can be attached to bus-based sensors and actuators to make them network-capable. This way, existing deployments can be retrofitted and extended to be completely IT-based. This would lead to a realization as was already depicted in Fig. 3, with IP communication all through the network, down to the periNODE.

Fig. 6. The periNODE smart adapter. Euro coin for scale

4.3 The periSWITCH Multi-port Switch, periSTART Media Converter and periLINE

Now that we have the possibility to retrofit existing sensors and actuators or build new ones with networking capabilities already included, we still need a network that connects them. For this reason, we designed the periSWITCH, periSTART, and periLINE.

As networking needs switches, the periSWITCH (Fig. 7, left side) was designed and is offered by Perinet. It is a multiport SPE switch equipped with either three or eight ports.

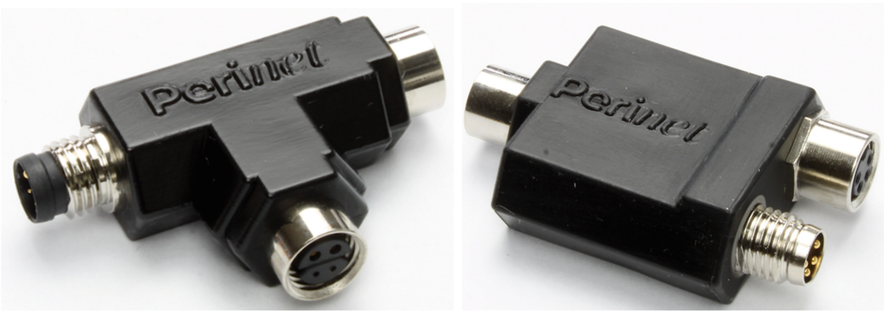

Fig. 7. A periSWITCH multi-port switch with three ports (l) and a periSTART media converter (r)

In order to convert between different media, the periSTART (Fig. 7, right side) can be used. It can be attached directly to a periSWITCH or periLINE.

The periLINE (Fig. 8) is a hybrid cable (power and ethernet) that can be used to span distances. The figure shows an example where it is used to connect a periNODE that has been used to retrofit a distance measuring sensor with a periSTART media converter.

Fig. 8. A periLINE connecting a periSTART with a periNODE that has been attached to an analog distance measuring sensor for retrofitting network capabilities

4.4 The periMICA Modular Edge Computer

The periMICA (Modular Industrial Computing Architecture) (Fig. 9) is an extremely compact and robust industrial edge computer with a modular hardware and software design for gathering and preprocessing data locally, close to the source of data, i.e. sensors and actuators. It comes in different variants and two protection classes - IP20 and IP67. Customization options are available both on the hardware as well as on the software side. The figure shows the backsides of two different versions: A minimalistic version with only two interfaces (I/O and PoE) and a version with multiple interfaces, including two times USB.

4.5 Basic Setup

Figure 10 shows a basic setup of the components described above. Two sensors, temperature and distance, are each retrofitted with a periNODE. The periNODEs are connected to a three-port periSWITCH by periLINEs, which in turn is connected to a periMICA. On the periMICA, calculations that are too complex to be performed directly on the sensors or actuators and the periNODEs connected to them can be carried out. Also, the periMICA can store values for later inspection. The periMICA is connected to the local network. As most local networks are not using SPE, this is done with a normal LAN cable.

Fig. 9. Two different versions of periMICA: A minimalistic version with only the two interfaces for I/O and PoE (top) and a second version with maximized different interfaces (USB, ETH etc.)

Therefore, Perinet offers all elements needed to switch from bus based systems to network based ones. As far as we know, there is no comparable solution offered by any competitor at the moment.

5 Example Project

While we of course tested our designs ourselves, we also have taken up contacts within the industry and realized a first reference project.

This first reference project is currently being realized in a plant for the electroplating of plug contacts. In this type of finishing, monitoring of the prevailing PH value is necessary for quality assurance purposes. Permanent manual monitoring of the PH value is very expensive, which is why this shall be done fully automated in the future.

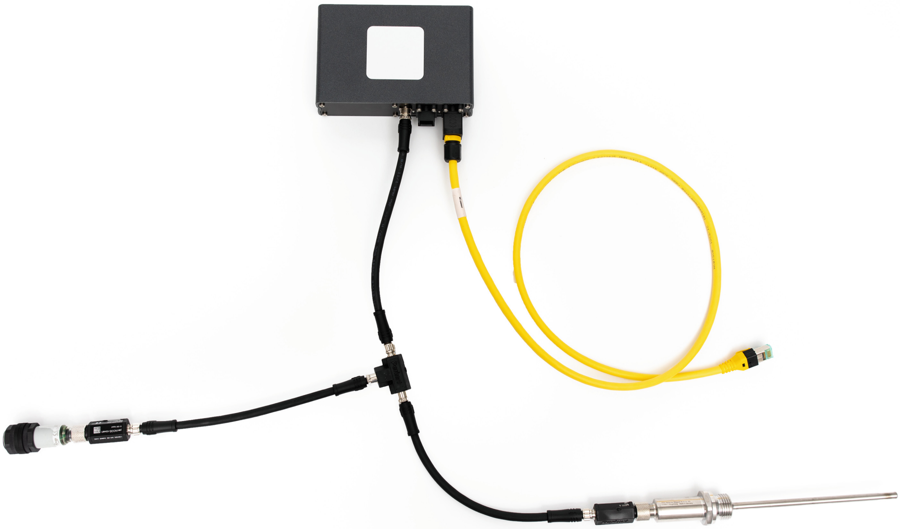

Fig. 10. Complete Setup

In a first step, a PH-value probe, which was not network capable until now, was connected to the network by means of a periNODE. The data from the probe is transferred to a MICA, where it is analyzed and stored. The analyzation/preprocessing also includes the comparison of the measured values with thresholds, and the raising of an alarm if one of them is exceeded. Thus, workers only need to intervene when they receive a notification that there is a deviation from the desired value range.

6 Conclusion and Future Work

In this paper, we have described our vision for the seamless integration of sensors and actuators into the (Industrial) Internet of Things. This vision can only be realized using IP-based technology end-to-end. However, there are a number of snares and pitfalls that need to be avoided in order to have this vision become reality. The new IEC 63171–6 standard is a big step in this direction, and Perinet supplies compatible nodes, switches, and cables.

Realizing the user interface in a browser enables end users to work in a familiar environment. However, the fact that most browsers are still focussed on connections to the Internet leads to problems when trying to realize local networks. Therefore, the fact that most browsers do not completely follow the standards resulted in a lot of extra work for us. For the future, we would wish for browser versions that are more focussed on local networks or at least do not throw any bricks in our way. If browsers completely implemented existing standards, a lot of problems would be solved.

Currently, we use a link local address to assure network connectivity between sensor and the operating system on a user's computer. However, the user might or might not see the device in his/her browser, depending on the browser variant. Typing the sensors address into the browser manually would solve this problem but require, from our point of view, too much work from the user. We are currently investigating how to solve this problem even if browsers remain non-compliant to standards. Moreover, we are looking into approaches to increase the intelligence of sensors and actuators even further.

References

1. Chatzigiannakis, I., Nikoletseas, S., Spirakis, P.: Smart dust protocols for local detection and propagation. In: POMC 2002: Proceedings of the Second ACM International Workshop on Principles of Mobile Computing, pp. 9–16. ACM Press, New York (2002). https://doi.org/10.1145/584490.584493
2. Chatzigiannakis, I., Nikoletseas, S., Spirakis, P.G.: Efficient and robust protocols for local detection and propagation in smart dust networks. Mob. Netw. Appl. **10**(1–2), 133–149 (2005). https://doi.org/10.1145/1046430.1046441
3. https://www.vde-verlag.de/iec-normen/248360/iec-63171-6-2020.html
4. Kahn, J.M., Katz, R.H., Pister, K.S.J.: Next century challenges: mobile networking for "smart dust". In: MobiCom 1999: Proceedings of the 5th annual ACM/IEEE International Conference On Mobile Computing And Networking, pp. 271–278. ACM Press, New York (1999). https://doi.org/10.1145/313451.313558
5. https://tools.ietf.org/html/rfc8446.html
6. https://datatracker.ietf.org/wg/zeroconf/charter/