UNIVERSITÀ POLITECNICA DELLE MARCHE

FACULTY OF ENGINEERING

Master Degree in Mechanical Engineering

Development and implementation of the process-relevant parameters of a model factory using Modbus

Sviluppo e implementazione dei parametri processo-rilevanti di una fabbrica modello utilizzando Modbus

Supervisors:
Prof. Maurizio Bevilacqua
Prof. Dr.-Ing. habil. Ralph Riedel

Co-supervisors:
Dipl.-Ing. Pierre Grzona
M.Sc. Martin Folz
Ing. Tobias Keller

Master thesis of:
Federico Carraturo

Academic Year 2020-2021

INDICE

1. ABSTRACT

Durante il ciclo di vita di un impianto di produzione e del suo equipaggiamento, possono subentrare diversi cambiamenti negli standard tecnici e nella strumentazione disponibile. Per affrontare questi aspetti, specialmente nell'implementare le operazioni della strumentazione, è spesso utile per le compagnie evolvere le strategie di retrofitting. Nel caso della digitalizzazione, lo stato dell'arte nella parte software e hardware cambia abbastanza spesso, con diverse tecnologie in competizione per lo scambio di informazioni tra diversi PLC (MQTT, OPC DA/UA, ecc…). Il protocollo di comunicazione Modbus fu stabilito per la prima volta negli anni '70. Negli anni recenti è stato incluso come Modbus TCP nel IEC61158 come un protocollo aperto per la comunicazione da PLC a PLC. Basandosi su questo protocollo e su un modello di fabbrica esistente, è stata svolta una ricerca, riportata nella tesi, sul se e sul come i parametri di processo possono essere trasferiti in un portale cloud, per rendere le informazioni disponibili a diversi utilizzatori e permettere la possibilità di analisi dei dati.
Il lavoro effettuato è stato declinato nei seguenti punti:

- Ricerca sul retrofitting degli impianti di produzione e dei loro sistemi informatici;
- Sviluppo di una procedura per implementare un sistema informatico basato su cloud;
- Analisi e implementazione su un sistema di controllo B&R;
- Implementazione del processamento di dati tramite un edge gateway;
- Sviluppo e implementazione di visualizzazione di dati per fattori chiave di produzione nel cloud;
- Sviluppo di scenari/casi di studio per indirizzare le informazioni a dispositivi per l'utente finale (smartphone).

Il lavoro svolto si pone nel quadro del piano Industria 4.0, con particolare focus sull'internet of things.
Il modello di fabbrica utilizzato è il 24V Fischertecknik Factory Model, che riproduce un generico impianto di manifacturing. Per essere sufficientemente completo, esso consta di 4 moduli: un magazzino con muletto automatizzato, un braccio robotico con presa sottovuoto, una stazione multiprocesso con forno e sega e un nastro trasportatore con rilevamento di colore. Il modello di fabbrica lavora con 9 modelli di semilavorato, suddivisi in 3 colori.
Il punto di partenza, all'inizio del lavoro, era uno scambio di input digitali e analogici e di output digitali tra il modello di fabbrica e un PLC, a cui erano annessi i moduli di controllo.
La flow chart dello schema logico seguito cominciava, dunque, dalla raccolta di dati booleani da parte dei sensori ottici, da cui, tramite il software Automation Studio, era possibile dedurre, ad esempio, se il sensore d'ingresso nel forno fornisse un valore vero o falso. Rimanendo nell'esempio, seguitava quindi l'elaborazione dell'informazione "il semilavorato sta entrando nel forno". Infine, un primo risultato che volevamo ottenere era un livello più alto d'informazione, del tipo "quanti semilavorati sono entrati nel forno in un'ora". In ultima analisi, volevamo automatizzare una vera e propria valutazione del processo industriale tramite KPI.
La scelta dei KPI è avvenuta analizzando la normativa europea ISO 15341, che si concentra su diversi aspetti che coinvolgono la manutenzione in una fabbrica.

In particolare, la normativa raggruppa i KPI in 8 famiglie: “manutenzione entro la gestione dell’impianto”, “Ambiente salutare e sicuro”, “Gestione della manutenzione”, “Competenza del personale”, “Tecnologia manutentiva”, “Organizzazione e supporto”, “Gestione e fornitura”, “IT e tecnologie abilitanti”. Nel vasto quadro di KPI offerto dalla normativa, è stato necessario effettuare una selezione, che prende KPI da tutte le famiglie sopra elencate, fatta eccezione per “Ambiente salutare e sicuro” e “Competenza del personale”, riguardanti il personale, non presente nel nostro caso di studio, e “Gestione e fornitura”, riguardante parametri strettamente economici.
Per consentire la comunicazione di dati tra il computer e il PLC abbiamo effettuato una configurazione dei sensori a Modbus, tramite il software Automation Studio.
Successivamente, è stato inserito un edge gateway tra il computer e il PLC, per consentire il trasferimento di dati dal PLC, che comunica tramite protocollo Modbus, e l’hub, che comunica tramite protocollo HTLM.
A questo punto, ho creato un primo elemento nell’Internet of Things, generando una “Modbus TCP Thing” all’interno dell’hub fornito dalla compagnia ELCO. In altre parole, ho generato un elemento che rappresenta i sensori fisici montati all’interno del modello di fabbrica, e mi informa se questi ultimi vengono stimolati. Esempio: il semilavorato esce dal nastro trasportatore, e per questo il sensore ottico di uscita dal nastro trasportatore viene stimolato e il suo status sul display passa da “false” a “true”.
Per ottenere un livello d’informazione più alto, come sopra anticipato, è stato necessario generare un secondo elemento di Internet of Things. Questa volta, una vera e propria Virtual Thing, un oggetto virtuale. All’interno di questo secondo oggetto, sono stati disposti i contatori (ad esempio “quanti semilavorati entrano nel forno”) e i KPI.
Per elaborare i dati della Virtual Thing e calcolare i KPI, abbiamo scritto un codice in Javascript. La prima parte del codice è inerente alla definizione delle variabili. La seconda è il calcolo vero e proprio dei KPI e la programmazione di un counter (il contatore del numero di volte che il sensore ottico del forno viene stimolato). Il calcolo è stato formulato come funzione if/else, in quanto elabora dati di tipo booleano (vero/falso). Infine, il terzo estratto del codice rappresenta i cicli while che abbiamo scritto per consentire un continuo aggiornamento in tempo reale dei dati provenienti dal sistema.
A seguito dei calcoli elaborati dal codice, è stato, infine, possibile leggere nell’hub della ELCO un nuovo quantitativo di informazioni automatizzate e in continuo aggiornamento in tempo reale.
A questo punto ci si è prospettato un nuovo obbiettivo: comunicare queste informazioni all’utente finale, ad esempio il manager dell’azienda o il responsabile alla manutenzione. Tramite un ulteriore sito internet fornito dalla ELCO, ho quindi progettato il design di una app per smartphone, che trae le informazioni processo-rilevanti dall’hub e le comunica in un modo che intende essere quanto più completo e chiaro possibile.
Tornando alla flow chart enunciata in precedenza possiamo dire, in ultima analisi, che se i primi 3 step (ottenimento di dati, elaborazione d’informazione, elaborazione più profonda d’informazione) possono essere inclusi nell’ambito dell’Era Digitale o Terza Rivoluzione Industriale, l’ultimo step fatto, ovvero la valutazione automatizzata fatta in Internet of Things e i cui risultati vengono forniti in tempo reale all’utente finale, fa parte del nuovo quadro Industria 4.0.
Ulteriori sviluppi di questo lavoro possono essere il monitoraggio di un numero più ampio di KPI, inerenti aspetti della gestione aziendale e del personale, e la progettazione di app per dispositivi differenti, come ad esempio tablet e smartwatch.

2. INTRODUCTION

During history, many changes in society and in the life of people occurred at the same time with the progress of technology. Starting from the invention of the wheel, the evolution of technology went forward, following a path of highs and lows, until the recent days. The speed of this process hasn't been constant: some historical periods have been more prolific, while other ones have represented a stop, or a step backward, and this mutable behavior lasted until the arrival of what we call industry. At the end of 18th Century, first machines were invented, exploiting the increase in volume that water encounters in moving from a liquid to a vapor state through heating.[1][2] This enhanced first forms of automation, substituting animal work, at first, and then also human, starting from the textile industry.[3] From that moment on, technology has grown in an increasingly fast way which is still ongoing. The use and spread of vapor machines had such a deep impact on society and human history to be called "First Industrial Revolution". The world "revolution" is used to express radical changes, often characterized by a violent disruption of the status quo. First Industrial Revolution gained this name after causing the mutation of society from rural to industrialized. Factories were introduced, deeply changing the working class and economy, salary replaced other forms of retribution. Nevertheless, such a change is not considered unique. After the First Industrial Revolution, a second one took place by the end of the 19th Century and the first half of 20th Century[4], this time characterized by mass production, petroleum and the introduction of electricity[5][6]. Second Industrial Revolution increased the speed of technological progress, and, more in general, the speed of life[7]. Transportation, work, mean of communication became faster, and this process laid the foundations of modern globalization. Industry kept on gradually evolving until a third, disruptive, sequence of events which took place from the second half of the 20th Century and that is called Third Industrial Revolution or Digital Revolution. This was provoked by the invention and the spread of digital computing and communication technologies, giving birth to the Age of Information. We all know the changes that it brought, from the scientific field to everyday life: spread of computer, memory systems (from the floppy disc to the USB drive), music entertainment, digital photography, internet. In other words, modern world. However, new significant changes occurred in industry in the early 2010's, marking a line with previous decades. In the 2011 Hannover Fair, a new concept was introduced[8]: Industry 4.0 or, as pointed by many, Fourth Industrial Revolution. The project was born as an initiative of the Federal Research Union of the German Government, and soon it spread worldwide.

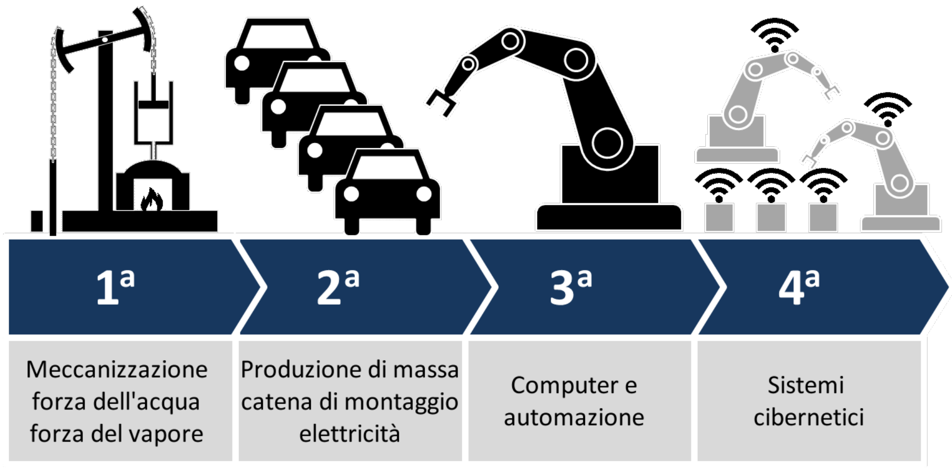

Fig. 2.1: the four industrial revolutions. The first one is strictly related to the exploitation of water expansion from liquid to vapour state; the second one is related to mass production and introduction of electricity; third one is the beginning of computers and digital age; fourth one is related to interconnected systems and big data.

Name "Industry 4.0" takes inspiration from how new software versions are named: every time that new fundamental changes are introduced, the left term number increases by one. Aim of plan 4.0 is the full digitalization of the industrial production, that consists of four principles of organizational design[9]:

networking: machines, devices, sensors and people can connect and communicate through internet of things or internet of people;

transparency of information: data from sensors improve information systems of digital factory models, to create a virtual image of the real world;

technical assistance: systems of assistance support people with the help of aggregate information, easy to visualize and understand;

decentralized decisions: cyber-security systems can take decisions independently, and they can complete tasks in the most autonomous way that is possible. Just in exceptional cases, such as malfunctioning or incoherent objectives, they move their tasks to a superior authority.[10]

In the area of industrial production, one key characteristic is the high level of personalization of product, in conditions of highly flexible production (on large scale). Moreover, automation technology required for Industry 4.0 must become more intelligent through auto-optimization, auto-configuration and auto-diagnosis, in order to help people in an increasingly complex work[11]. Thanks to these characteristics, there's a significant perspective to reduce time of production, enforce automation, make customized products and integrate data which weren't used before[12]. Another very key aspect of Industry 4.0 is the web connection of production means distributed in space, and of their related planning and control systems. Production resources work autonomously, in a knowledge-based way, enhanced by sensors, and they can configure themselves according to the situation, exchanging information. Production networks are controlled real-time, and they can adapt to a change of the limit conditions. Union of IT and production technologies in Industry 4.0 entails a number of challenges:

they have to be developed technical standards and norms to allow communication between man and machine or machine and machine, as well as in cognitive systems, for example; data are acquiring a huge importance, and many authors refer to them as "new raw material"[13], so themes of data security and data property are fundamental. In this sense, there are many legal subjects still to be solved; instruction of personnel about new technology; safe work environment.

There is still criticism about the revolutionary nature of Industry 4.0. Many authors think it's indeed an innovative plan for further digitalization of industry, but that it's too early to consider it a new industrial revolution, because we are still inside it, and we can't see all of its consequences on society, as we can do, instead, for first three industrial revolutions. History we'll show us if the previsions were true. What we can do, on our side, is to be enthusiast protagonists of a challenge that has to deal with an everyday more complex world, and with an undelayable need for a sustainable industry.

3. LITERATURE REVIEW

3.1 Procedure for retrofitting of production plants and their information systems in an Industry 4.0 context

During their life, machinery in a plant goes through consumption and failures, and, when possible, maintenance has to be done. Besides that, new machinery shows up in the market, making the old one obsolete in terms of efficiency and safety. So, time to time, renewal is necessary in order to be competitive in the market. The renewal process in technology and functionality of an industrial plant is called “retrofitting”, and, in its traditional shape, it consists of substitution of old machinery with new one. However, to be really convenient, retrofitting procedures always have to be a good compromise between cost and benefits. Moreover, in substituting old machinery, also the plant in its wholeness has to be considered, avoiding non compliances of any form, such as problems regarding available space for new machinery. Another aspect to take into account, when substituting a machine, is the increasing cost of new technology, that is not caused, as in the past decades, by the cost of their components (which are cheaper than before), but depends on how sophisticated new machine design is. For these reasons, retrofitting has a cost both in economic terms and in time, and many companies can’t deal with it. To introduce them in an Industry 4.0 context could represent a valid alternative to the substitution of old machinery, as stated by the following article by the Department of Industrial Engineering and Mathematical Science, Polytechnic University of Marche[14]:

“The costs and time to replace obsolete machines could be unsustainable for many companies, during the retrofitting duration. Making them ready to the Industry 4.0 context, may represent an alternative to the replacement. […] Industry 4.0, also called the fourth industrial revolution, is the natural consequence of the third revolution, and it is mostly based on two main factors: Internet of Things and Services (IoT) and Cyber-Physical Systems (CPS). The implementation of Industry 4.0 aims to guarantee businesses a competitive strategical advantage, organization agility, organizational efficiency and effectiveness, profitability, manufacturing innovation, maintenance costs, improved product safety and quality, improved operations, delightful customer experience and environmental and social benefits. As a result of its implementation, businesses could achieve a better result in the three dimensions of the Triple Bottom Line (TBL): economic, social and ecological. Businesses that operate with the aim of achieving these three principles can be defined as sustainable. The implementation of Industry 4.0 technologies is not simple and often presents potential problems, particularly for Small and Medium-Sized Enterprises (SME). Such problems can include limited availability of financial resources, limited knowledge and scarce technological competencies. Moreover, even if the use of technologies such as IoT and CPS improve the quality and the safety of the product and the process, another new risk, such as problems connected to cybersecurity, could emerge. SMEs, which make up 90% of European industries, are those which encounter greater difficulty in the implementation of Industry 4.0. Nonetheless, they must find a way to adapt to remain competitive against multinational industries. Whereas that the process plants often have a useful life that exceeds 20 years, in many cases, it may be very economically inconvenient to replace the old plant with a new-generation plant designed for Industry 4.0. Moreover, when a plant is replaced, it must be considered that the replacement includes an extended machine downtime. This is often unsustainable in terms of time. In this respect, it would be useful to evaluate a retrofit operation. Retrofitting improves the accuracy, energy consumption, safety level, maintainability and ease of use of an old plant to obtain optimal plant performance. In the context of Industry 4.0, we talk about smart retrofitting. In addition to the classical retrofitting features, smart retrofitting involves implementing all necessary tools and technology provided by Industry 4.0. Retrofitting is related to the need for sustainability, productivity and increased technological level. Therefore, with a smart economy and high connectivity, retrofitting allows the introduction of old plants in Industry 4.0."

3.2 KPIs

From one side, new technology allows us to make and industrial process more reliable and efficient, after getting more information from available data. From the other side, the choice of data to be analyzed is fundamental, as well. To look for the right data to watch, and trying to get new data from a system, is an essential step in understanding the goodness of a process, and a leading factor in decision-making when corrective action is needed. When analyzing the performances of a factory (specialized, in our case, in manufacturing processes) several aspects need to be considered, taking into account areas like maintenance, general efficiency, safety, ecology, research & development, sales, etc… In order to evaluate performance of our factory, we investigated the parameters which are most used nowadays. As a main reference, we have chosen ISO European Standards, for their capacity to depict modern industry and be a guideline for it, and in particular we consulted ISO 15341:2019 (Essential Performance indicators for maintenance)[15]. From that, we started building the list of KPIs (Key Performance Indicators) that better suited our case study. Key performance indicators for the maintenance function apply to any commercial facility, infrastructure, civil building or transportation system, etc.

These performance measures should be applied to:

1. the measurement of the state;
2. comparison (internal and external benchmarks);
3. diagnosis (analysis of strengths and weaknesses);
4. the identification of objectives and the setting of target objectives;
5. the planning of the improvement measures;
6. the regular measurement of change over a period of time.

ISO 15341 lists key performance indicators (KPIs) for the maintenance function and provides guidelines to establish a set of appropriate KPIs to assess and improve the effectiveness, efficiency and sustainability in the maintenance of existing assets, either commercial assets, infrastructures, operational assets, civil buildings or transport systems, etc., within the context of external and internal influencing factors. KPIs are divided into 8 categories, each one with 4 sectors, as shown in the matrix below.

SUB-FUNCTIONS; TOOLS AND METHODOLOGIES	KPIs	MAIN SECTORS			
Maintenance within the plant management	PHAi	Sustainability i = 1 to 3	Effectiveness, capacity, Integrity i = 4 to 11	Service level i = 12 to 13	Economy i = 14 to 20
Sub-function 1 Health - Safety - Environment	HSEi	Conformity with laws and regulations i = 1 to 3	Statistical records i = 4 to 12	Safe method i = 13 to 17	Prevention and improvements i = 18 to 22
Sub-function 2 Maintenance management	Mi	Strategy i = 1 to 3	Function i = 4 to 10	Technical evaluation i = 11 to 16	Continuous improvement i = 17 to 22
Sub-function 3 Staff competence	Pi	Maintenance manager i = 1 to 3	Maintenance Supervisor/Maintenance Technician i = 4 to 9	Technical maintenance specialist i = 10 to 12	Education i = 13 to 21
Sub-function 4 Maintenance technology	Ei	Capability Criticality of performance capability i = 1 to 3	Shelf life i = 4 to 9	Preventive maintenance i = 10 to 16	Technical improvements i = 17 to 19
Sub-function 5 Organisation and support	O&Si	Structure and support i = 1 to 8	Planning and control i = 9 to 22	Effectiveness, productivity i = 23 to 28	Quality i = 29 to 30

SUB-FUNCTIONS; TOOLS AND METHODOLOGIES	KPIs	MAIN SECTORS			
Sub-function 6 Management and provisioning	A&Si	Economy i = 1 to 6	Budget and control i = 7 to 19	External services i = 20 to 25	Materials and spare parts i = 26 to 29
Information and communication technologies Enabling technologies	ICTi	Management i = 1 to 6	Management and deployment i = 7 to 10	Organization and support i = 11 to 13	Engineering i = 14 to 20 TEC 18.20

Tab. 3.1: KPI matrix from ISO 15341 norm

Followingly, each group of KPI has been described.

Maintenance within asset management

Maintenance within asset management enables the optimal product life cycle management of the asset in order to sustainably achieve the stated goals of the organization. It also plays a significant role in the life cycle management of the plant, as the maintenance function ensures its integrity for the majority of its life.

Asset management emphasizes the appropriate importance of maintenance in the different phases of the product life cycle and supports maintenance management in defining the effective long-term maintenance strategy, and it determines the interrelationships between maintenance and the other asset processes to measure performance over the life of the asset in relation to the organization's vision, mission, values, and objectives.

The main KPIs referred to as PHAi, which are shared by the driver domains as indicated in Table 2, are as follows:

- Sustainability: PHA1, PHA2, PHA3.
- Capacity, effectiveness, integrity: PHA4, PHA5, PHA6, PHA7, PHA8, PHA9, PHA10, PHA11;
- Service Level: PHA12, PHA13;
- Economy: PHA14, PHA15, PHA16, PHA17, PHA18, PHA19, PHA20.

Health, Safety and Environment

"Health, Safety and Environment (HSE)" in maintenance sub-function is concerned with the implementation of policies and procedures by maintenance management using appropriate resources to prevent injury and damage and to comply with laws, regulations and company objectives.

- HSE laws, regulations and procedures are basic requirements for maintenance to carry out risk analysis and take preventive measures, maintaining the integrity of each piece of equipment in the condition that allows it to run (be operated) in a sustainable manner according to the requirements of laws and standards;
- The whole can be considered as a subsystem that requires a great deal of maintenance, as it is widely recognized as accident prone in every field of activity;
- This means that the operational availability of the plant includes compliance with HSE requirements in order to avoid unacceptable risks and unassessed hazards to people and the environment;
- Failure and Injury Risk Analysis is a tool that can be used as a preventive measure in all corrective, proactive, predictive, condition-based, preventive maintenance work or engineering improvements.

The main content and knowledge to be fulfilled are:

- Laws, policies and strategies on HSE;
- HSE responsibilities, objectives and plans;
- HSE risk analysis and RAMS assessment;
- HSE method and device for the prevention of accidents;
- Risk analysis and hazard assessment;
- HSE records and related key figures;
- Failure mode, impact and criticality analysis;
- Good practices in health - safety – environment;
- Protective equipment (collective and personal);
- Safety manual and handbook;

- Contingency Plan;
- Accident report and investigation report;
- Occupational diseases and accidents;
- Safety education and training.

It includes the HSE performance and competence of internal and external, both in-house and outsourced maintenance personnel to meet the requirement of the influencing factors.

The 4 main driver areas of HSE associated with performance for maintenance and performance metrics for maintenance called HSE1-22 are:

- Conformity with laws and regulations: HSE1, HSE2, HSE3;
- Statistical recording: HSE4, HSE5, HSE6, HSE7, HSE8, HSE9, HSE10, HSE11, HSE12;
- Maintenance safety practices: HSE13, HSE14, HSE15, HSE16, HSE17;
- Prevention enhancements: HSE18, HSE19, HSE20, HSE21, HSE22.

Maintenance Management

Maintenance management is a combination of resources, disciplines, knowledge, competences and tools to define a medium-term plan in accordance with the commercial plan of the company. It is the coordination and control of activities implemented in facilities to achieve specific objectives within the existing framework and constraints.

Maintenance management uses theories, applications and methods of management, available internal and external resources to coordinate the activities of maintenance and to achieve the most appropriate performance for maintenance. The assets are used in the existing framework, according to the vision, mission, values, policies and stated objectives of the organization.

Maintenance management must:

- comply with laws, HSE, regulations and company procedures;
- define the maintenance strategy and implement it, see ISO 55001;
- control the maintenance function and its subfunctions;
- optimize the competence of personnel, productivity of maintenance resources, effectiveness and efficiency of processes;
- improve the technical standard;
- achieve the "best added value" for plants, operating facilities and infrastructures in the optimal life cycle determined by industrial strategies and asset management;
- consider the external and internal influencing factors that are fundamental to understanding the status of the organization/site/powerhouse and its evolution towards the excellent organizational models.

The main driver areas named M and their associated KPIs are as follows:

  - Maintenance strategy: M1, M2, M3;
  - Management of the function: M4, M5, M6, M7, M8, M9, M10, M11, M12, M13;
  - Technical evaluation: M11, M12, M13, M14, M15, M16, M17, M18, M19;
  - Continuous improvement: M20, M21, M22.

The ratios M1, M2, M3, M4, M5, M6, M7, M8, M9 are qualitative, and not defined by an equation.

In order to obtain a qualitative value, it is advisable to use a "recording sheet for the evaluation of the specific degree" to evaluate the specific degree of the object in each specific reality for each indicator.

In this way, it will be possible to evaluate the position of the maintenance function in the framework of the maintenance maturity model and to estimate the level with the performance reconciliations.

Competence of staff

The competence of maintenance personnel is a priority, as maintenance is a function where manpower is a fundamental resource in terms of quantity and quality. For this reason, "competence of staff" sub-function is one of the categories of KPIs contained in the standard.

It is necessary to develop two types of competencies:

1. one refers to the specific knowledge of the business sectors: Energy, Oil and Gas, Pharmaceuticals, Food, Transportation, etc.;
2. one refers to the knowledge of any sub-function of maintenance.

The process consists of two steps to improve competence and therefore the level of professionalism.

EN 15628:2014 provides a formal outcome of an assessment and validation process of the competence of maintenance personnel, provided by an institute or school authorized by an external certification body, and determines that an individual has acquired the following:

- the required level of education;
- the required professional experience and skill on the job;
- the necessary knowledge of the maintenance discipline.

The qualification is classified by the EU recommendation as follows:

- EQR level 4-5 maintenance specialist;
- EQF level 5-6 Maintenance manager/maintenance engineer
- EQF level 6-8 Head of maintenance service/maintenance/head of maintenance function

Competency includes methods, applications and practices to achieve the best level of professionalism through training, field experience and specific knowledge of the EN 15628 maintenance discipline.

The associated KPIs, referred to as P, are:

- Maintenance manager: P1, P2, P3;
- Maintenance manager/maintenance engineer: P4, P5, P6, P7, P8, P9;
- Maintenance specialist: P10, P11, P12;
- Educational plans: P13, P14, P15, P16, P17, P18, P19, P20.

1) Education:

The result of assimilating information by learning principles and theories through programs related to technology, management, business, and engineering disciplines in school and university.

2) Experience:

It is the ability to use good maintenance methods, personal, social and/or methodological behaviour, work or study in professional and personal development.

3) Skill:

It is the ability to apply knowledge and use know-how to complete tasks and solve problems. Skills are described as cognitive (involving the use of logical, intuitive and creative thinking) or practical (involving the dexterity and use of methods, materials, tools and instruments).

4) Training

Skills and knowledge are taught or developed to oneself or others, all of which are associated with specific and useful competencies.

The training has specific goals of improvement:

-ability;
-capacity;
-productivity;
-quality;
-performances.

Maintenance Engineering

Maintenance Engineering is the discipline and processes that apply/applies competencies, skills, methods, tools and techniques for the development and support of maintenance activities to ensure that an object is able to perform its required functions throughout its life cycle in a safe, sustainable and cost-effective manner.

Maintenance engineering involves the preparation of systematic information on equipment and its components to provide sufficient and relevant information.

The main areas of maintenance engineering in relation to the influencing factors of the plant and the associated E named KPIs are:

- performance and criticality of the plants: E1, E2, E3;
- durability: E4, E5, E6, E7, E8, E9;
- preventive maintenance: E10, E11, E12, E13, E14, E15, E16;
- continuous technical improvement: E17, E18, E19.

KPIs of the "Organization and Support" sub-function is a combination of internal and external maintenance resources such as people, spare parts, tools, equipment, information, methods, processes, procedures, standards, best practices, ICT, etc. to provide the required maintenance services that deliver the best performance in terms of safety, productivity, effectiveness, quality, cost and service level.

The main tasks are:

1. determining the rules of criticality and priority;

2. establish a procedure for planning-scheduling-executing and controlling the required work;
3. design of work orders, including the necessary information: safety, technical data, accounting, scheduling, etc.;
4. use a standard computerized procedure to achieve the best possible integrated work order according to organizational structure, responsibilities, processes, etc.;
5. planning all resources, trades, support, according to specifications and good maintenance practices that are the best engineering practice to repair or restore a unit, based on available knowledge and expertise;
6. preparation of preventive costs, time and other impacts and planning of the work according to the requirements of the end user;
7. assign work to internal staff and external maintenance and monitor progress;
8. closing the work order, completing technical notes and reporting in the database;
9. measuring the quality, efficiency and effectiveness of the work carried out;
10. evaluate and provide necessary support and assistance activities to improve quality and productivity.

The main objective is to organize and support the on-site activities for maintenance within the framework of the strategy and the stated objectives of asset management, in order to achieve the optimization of maintenance resources.

The key driver areas of the organization and support and associated O&S named KPIs, are:

- Structure and Support: O&S1, O&S2, O&S3, O&S4, O&S5, O&S6, O&S7, O&S8;
- planning and control: O&S9, O&S10, O&S11; O&S12, O&S13, O&S14, O&S15, O&S16; O&S17, O&S18, O&S19, O&S20, O&S21;
- productivity and effectiveness: O&S22, O&S23, O&S24, O&S25, O&S26, O&S27, O&S28;
- quality of work: O&S29, O&S30.

Management and deployment sub-function covers three main areas:

1. comply with all maintenance economic practices and procedures with country and company financial and operational accounting regulations;
2. accounting and appropriate procedures for all maintenance resources and for budget and cost control;
3. the supply chain management for the purchase and provision of all technical support, such as equipment, tools, spare parts, materials and external maintenance, in accordance with the requirements and programs of maintenance for the execution of works.

The main contents are:

- implementation of accounting principles, accounting activities for maintenance, including the execution of works and projects, and determination of a budgeting and control system;
- support management and supervision at every step in the maintenance process with analysis and cost evaluation procedures;
- determination of the annual economic maintenance plan and budget;
- continuous updating of all administrative documents, standards, good maintenance practices, budgets, etc.;
- evaluate and determine the standard cost of maintenance resources;
- determine and apply the most economically, organizationally and technically useful KPIs and perform benchmarking activities to evaluate maintenance performance;

- definition of contracts regarding outsourcing of maintenance: e.g. contract services, full service, global services;
- provision of external maintenance, spares and materials. The main driver areas

  and associated A&S named KPIs are as follows:

  - economy: A&S1, A&S2, A&S3, A&S4, A&S5, A&S6;
  - budget and cost control: A&S7, A&S8, A&S9, A&S10, A&S11, A&S12, A&S13, A&S14, A&S15, A&S16, A&S17, A&S18, A&S19;
  - third-party services: A&S20, A&S21, A&S22, A&S23, A&S24, A&S25;
  - materials, spare parts: A&S26, A&S27, A&S28, A&S29.

Information and Communication Technology

Information and Communication Technology (ICT) is a set of interrelated sciences, criteria, methods, techniques and tools designed to improve activities related to the collection, transmission and processing of data and information in order to add value and help management and maintenance personnel to achieve the best performance.

Enabling technologies can contribute significantly to increasing competitiveness, optimizing strategies, decisions, activities, controls, in order to develop new excellent models of maintenance function, enabling new modern ways.

This subsystem of information and communication technologies includes applications to develop and achieve the best maintenance performance in the Internet of Things framework according to the vision, mission, values and stated objectives of the organization.

The key driver areas and associated ICT designated KPIs are as follows:

- management: ICT1, ICT2, ICT3, ICT4, ICT5, ICT6;
- administration and delivery: ICT7, ICT8, ICT9, ICT10;
- organization and support: ICT11, ICT12, ICT13;
- Maintenance Engineering: ICT14, ICT15, ICT16,ICT17,ICT18, ICT19, ICT20.

The metrics related to the functionality and the degree of use and integration of ICT in the main maintenance areas cannot be measured by the formulas. Consequently, it is useful to have a qualitative value by using a specific data collection sheet to evaluate the "state degree" for each metric in order to know the reality for improvement.

In the vast framework of KPIs suggested by the ISO 15341 Standard, it was necessary to select those which better could fit our analysis of the model factory. For these reasons, categories “Health-Safety Environment” (HSE) and “Staff competence” (P) have been discarded, because pertaining personnel, an element that is not present in our case study. In making our list of KPIs, “management and provisioning” (A&S) has also been excluded, because pertaining strictly economical metrics. Economy is, indeed, one key factor to be considered in managing a factory, but in our analysis it is a consequence of technical factors such as downtime/uptime ratio, and it has been analyzed through different KPIs sub-functions. After these clarification, this is the list of keys performance indicators we built.

KPI	Factors
PH6 Overall Equipment Effectiveness R1*R2*R3	Time with no failures Requested time Production time Total time Ideal time Requested time for quality compliance
PH8 Operational availability due to manteinance	Total operating time Total operating time+downtime
M11 Availability based on uptime	Operating time Required operating time
M12 Availability based on time to restore	Operating time Operating time+recovery time
M13 Production based availability	Actual production Required production or other reference level within a specified period of time
M15 Frequency of manteinance shutdown (No./Year)	Number of shutdowns Year (Week)
M19 Outcome rate (%)	Objectives achieved Assigned objectives
E5 MTBF Average operating time between failures (hours)	Total operating time Number of failures
E6 Average repair time	Total time to recovery Number of failures
E9 Downtime due to corrective manteinance	Downtime due to maintenance Total downtime due to maintenance reasons
E10 Downtime due to condition-based maintenance	Downtime due to condition based maintenance Total downtime due to maintenance reasons
O&S16 Mean time to recovery (hours)	Total time to recovery Number of failures
O&S17 Average repair time (%)	Total repair time Number of failures
ICT3 Intensity of use (%)	Number of ICT accesses or connected machines, lines, etc. Number of personnel performing maintenance work

Tab. 3.2: list of chosen KPIs

The table is divided in two columns. First column contains the names of the selected KPIs, the second one shows the two terms of the ratio necessary to calculate each KPI. As examples,

$PH8 = \frac{total\ operating\ time}{total\ operating\ time + downtime}$ and $M11 = \frac{operating\ time}{required\ operating\ time}$. Colors in the second column highlight the three most frequent factors to be monitored: operating time, number of failures and downtime. Grouping them by color helped to make it clear which were the most important factors to be analyzed, and let us calculate certain KPIs more easily. First KPI of the list is PH6 Overall Equipment Effectiveness. It is one of the KPIs that better describe the behavior of the factory in its integrity. It is the product between three ratios: R1, R2

and R3, in which: $R1 = \frac{T_2}{T_1} = \frac{available\ time}{requested\ time}$ = time loss due to maintenance reasons; $R2 = \frac{production\ time}{actual\ time}$ =time loss due to production reasons; $R3 = \frac{ideal\ time}{actual\ time\ to\ quality\ compliance}$ = time loss due to quality mismatch between materials and efficiencies of the process. The KPI is a sum of maintenance, production and quality compliance aspects. PH8 "operational availability due to maintenance" is the ratio $\frac{total\ operational\ time}{total\ operational\ time + downtime}$, and it describes how long downtime has been compared to the total operational time.

After these first two metrics, it comes the "maintenance management" subgroup of KPIs. Between them, we have chosen: M11 Availability based on uptime $= \frac{operating\ time}{required\ operating\ time}$;

M12 Availability based on time to restore $= \frac{operating\ time}{required\ operating\ time}$;

M13 Production based availability $= \frac{actual\ production}{required\ production\ over\ a\ specified\ period\ of\ time}$;

M15 Frequency of manteinance shutdown $= \frac{number\ of\ shutdowns}{year}$; M19 Outcome rate= $\frac{objects\ assigned}{objects\ achieved}$, expressed as a percentage. "Maintenance engineering" subgroup contains:
E5 MTBF Average operating time between failures $= \frac{total\ operating\ time}{number\ of\ failures}$, measured in hours;
E6 Average repair time $= \frac{total\ time\ to\ repair}{number\ of\ failures}$; E9 Downtime due to corrective manteinance $= \frac{downtime\ due\ to\ corrective\ maintenance}{total\ downtime\ due\ to\ maintenance\ reasons}$; E10 Downtime due to condition-based maintenance= $\frac{downtime\ due\ to\ condition{-}based\ mintenance}{total\ downtime\ due\ to\ maintenance\ reasons}$. These last two KPIs pertain two different types of maintenance. Corrective maintenance is a repairing process which takes place when the failure already occurred. It is meant to restore the status quo before the failure, without adding new value to the system. Condition-based maintenance is a kind of predictive maintenance, so it's a process which starts before the failure occurs, and the prediction is based upon the actual condition of the system. In the "Organization and support" group, we've selected: $O\&S\ 16 = \frac{total\ time\ to\ recovery}{number\ of\ failures}$, measured in hours; $O\&S\ 17 = \frac{total\ time\ to\ repair}{number\ of\ failures}$, calculated as percentage. Total time to recovery

includes total time to repair and delays. Between the “Information and communication technology” KPIs, we’ve chosen: ICT3 Intensity of use (%)=$\frac{Number\ of\ ICT\ accesses\ or\ connected\ machines}{Number\ of\ personnel\ performing\ maintenance\ work}$.

Beside KPIs contained in ISO 15341, mainly focused on maintenance topics, our investigation moved forward additional metrics, which involved management analysis, and which could give us a view into data that could be helpful for Research and Development. After our research, the additional KPIs are:

Allocation efficiency	Actual unit busy time
	Planned busy time
Utilization efficiency	Actual production time
	Actual unit busy time
Availability	Actual production time
	Planned busy time
Quality compliance ratio	Quality compliant workpieces
	Total amount of workpieces
Number of corrective actions in a year	
Lead time	Time to satisfy the request of a customer
Fixability of damaged workpieces	Number of fixable damaged workpieces
	Number of damaged workpieces
Station inefficiency	Number of failures of the working station
	Number of failures of the factory
Downtime due to retrofitting	Time to implement retrofitting
	Total downtime
Data acquisition capability	Data acquired
	Estimated data

Tab. 3.3: additional KPIs

The method to describe the ratios in the table 3.3, is the same that has been explained for table 3.2. Color yellow has been used for management-related KPIs, green for quality assurance and blue for research and development. Between the management-related KPIs, we can see: “Allocation efficiency”, which indicates how much time our production plant is busy in comparison with the planned busy time; “Utilization efficiency”, which describes how much time plant is actually exploited for production, related to the actual time the unit is busy; “Availability”, the ratio between the time actually dedicated to production, and the planned busy time of the plant; “Lead time”, which is, by definition, the time to satisfy the request of a customer. The quality assurance related KPI is “Quality compliance ratio”, so the ratio between the quality compliant workpieces and the total amount of workpieces.

Next chapter of the thesis will focus on PLC communication protocols and Modbus. The application of KPIs will be discussed in chapter 4.

3.3. Communication protocols between PLCs, Modbus

In industry, for control of manufacturing processes, such as assembly lines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnosis, usage of Programmable Logic Controllers is a standard. PLCs are digital computers which could also operate reliably in harsh environment and conditions, like severe vibrations, extreme temperatures and wet and dusty environment. They can range from small modular devices with tens of inputs and outputs (I/O) located in the same housing of the processor, to large modular devices with thousands of I/O mounted on racks, and which are often networked to other PLC and SCADA systems. Communication between PLCs and between PLC and PC is made possible by communication protocols, which are systems of rules that allow two or more entities of a communications system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics, synchronization of communication and possible methods to recover errors. They may be implemented by hardware, software, or combining both of them.

Communicating systems use already-defined formats for exchanging different messages. Each message has an exact meaning, intended to trigger a response, which comes from a range of possibilities that have been previously determined for that exact situation. The behavior is typically independent on how it is meant to be implemented. However, an agreement must exist about communication protocols between the parties involved.[16] To come to an agreement, a protocol may be developed into a technical standard. A programming language describes the same for computations, so there is a close analogy between protocols and programming languages. In this chapter, the main protocols which are used nowadays are treated, with a deeper look inside Modbus TCP, chosen in our case study.

The main classification that could be done for communication protocols is between fieldbus and industrial ethernet protocols, which have an increasing presence in industry[17].

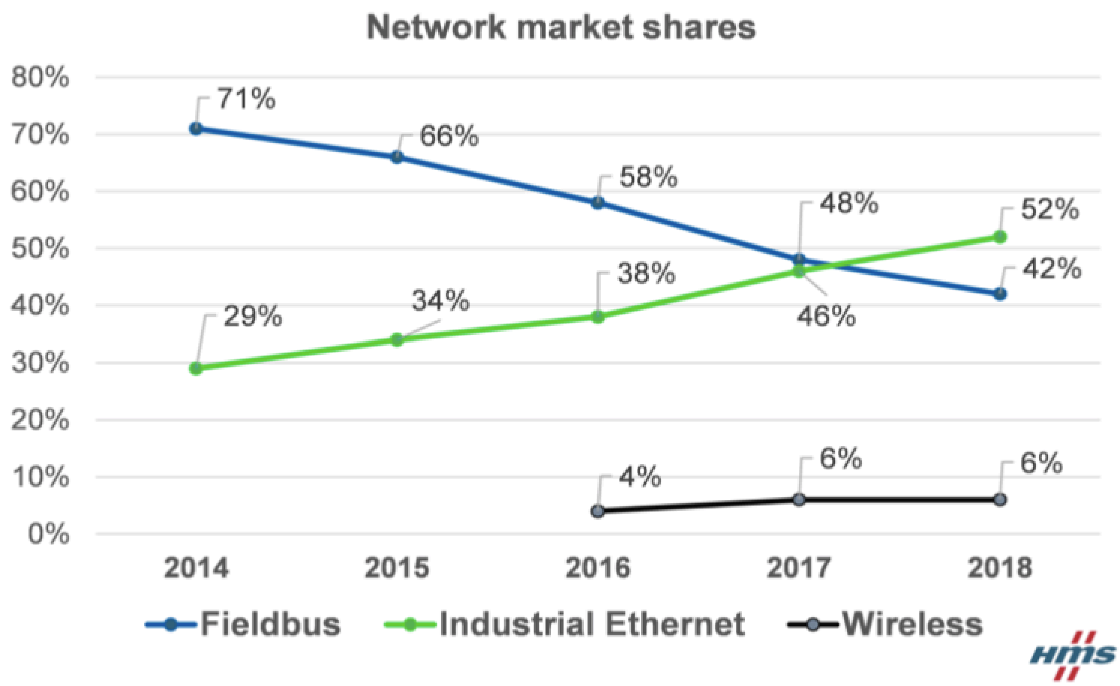

Fig. 3.1: Network market shares progression through years 2014 to 2018

The graph above represents last years trend in the usage of Industrial Ethernet over Fieldbus, while in the chart below we can see how main protocols are distributed in industry.

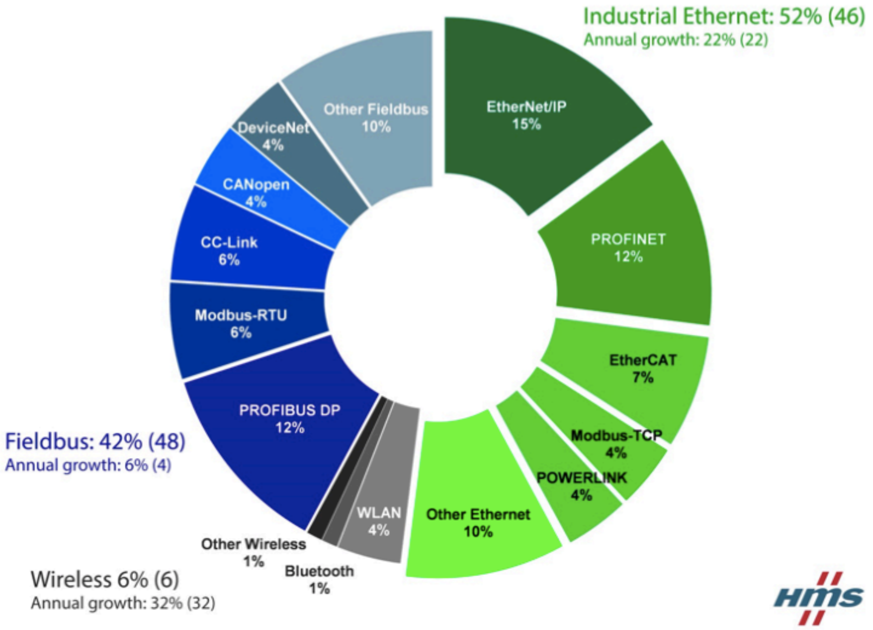

Fig. 3.2: Distribution of communication protocols in industry

According to HMS Industrial Networks, the most common industrial fieldbus protocols are:

- Profibus DP
- Can/Can open
- CC-Link
- Modbus RTU

Profibus (PROcess fieldBUS) is a fieldbus promoted by Siemens. Profibus technology was born as a German standard in compliance with DIN 19245 (1991) norm, and from 1996 it is compliant to the EN 50170 standard. Profibus DP (Device Peripheral) was born in 1994, and it's optimized for cheap high-performance connections. This PROFIBUS version (that is the most common one), is mainly addressed to the communication between control systems and I/O devices. Communication functions are made available to the user by user interface (the functions are mapped on level 2 by a Direct Data Link Mapper (DDLM) and, on an higher level, they are managed by Profiles).

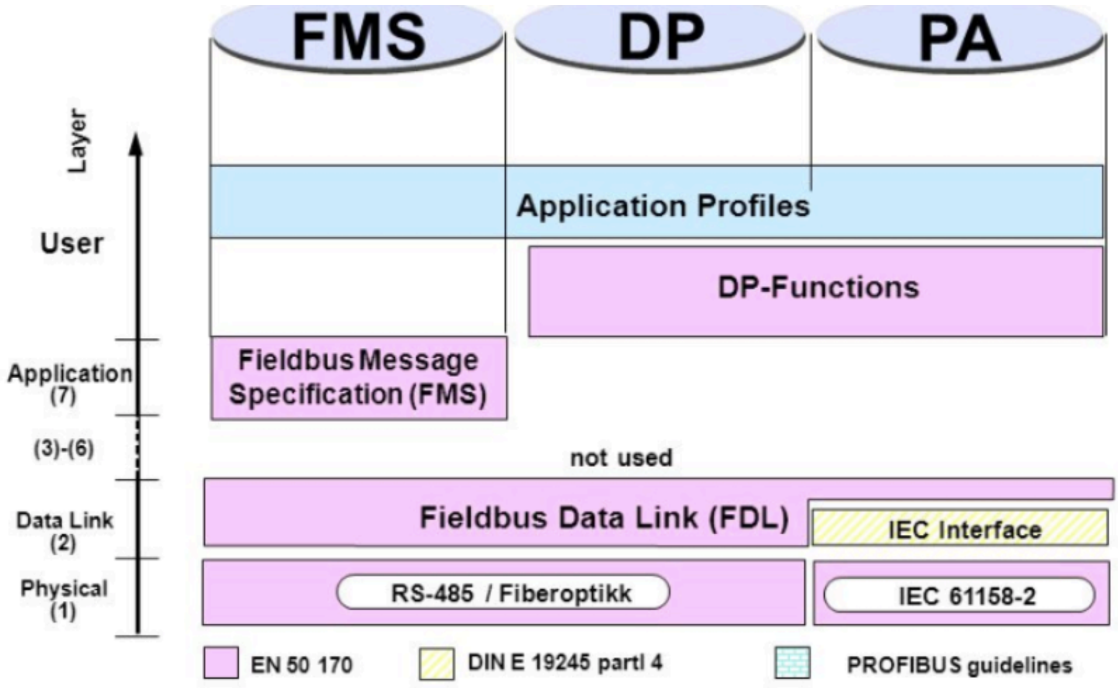

Fig. 3.3: The graph highlights how different aspects of profibus are regulated by norms, on different layers. Also, three different versions of Profibus are represented: FMS (Fieldbus Message Specification), DP (Device Peripheral) and PA (Process automation).

CAN (Controller Area Network) was developed by Robert Bosch in 1985, and was born to allow the communication between the many electronic devices inside a vehicle (automotive field). The access to Bus doesn't follow a pre-determined scheme: it's not deterministic, access is random. It is not based on the physical addresses of the nodes, but on the content of a communication which is marked by an Identifier, which assigns the level of priority to the message. CAN has an event-driven communication type (access to Bus only if required by the application) and a producer/consumer communication type. It has an arbitration system to control accesses to Bus (CSMA/NBA, /Non-destructive Bitwise Arbitration). It uses NRZ/bit stuffing coding, a mask to filter messages and the protocol directly manages error status of the nodes (fault confinement). 5 different frames are defined, and according to the bit number of identifiers, there are two kinds of CAN: 11 bit identifier (2^{11}) is the standard version (CAN 2.0A), while 29 bit identifier (2^{29}) is the extended version (CAN 2.0B). In the table below, the communication model is shown.

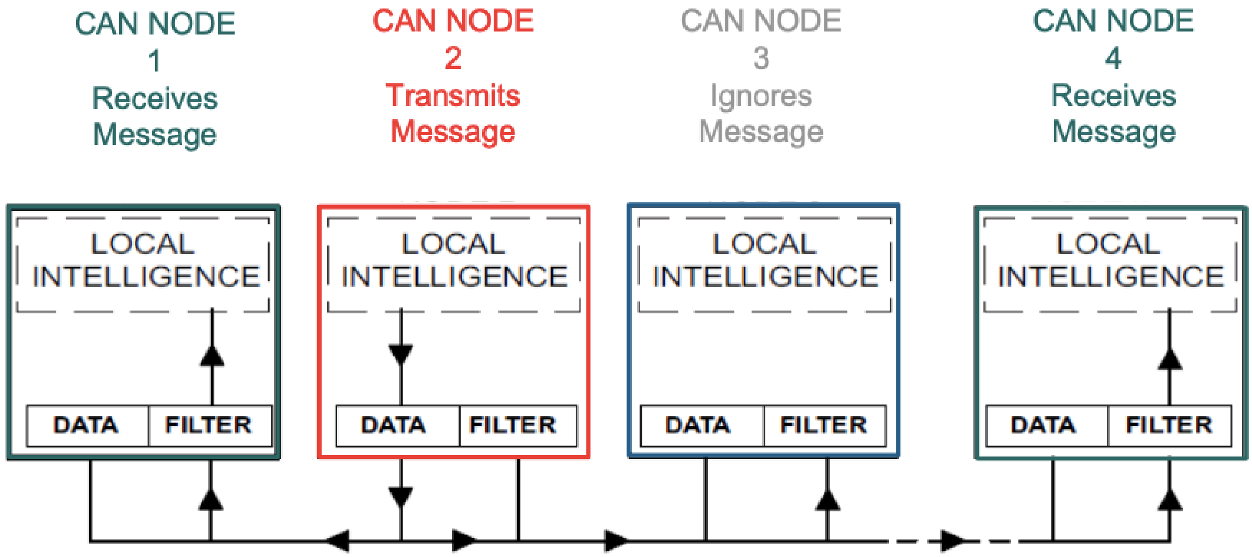

Fig. 3.4: CAN Open communication model

CAN open has the following characteristics and targets:

- to allow a deterministic behaviour of the net, using a Client/Server model;
- to create a clear and short standard to ease implementation and maintenance;
- to use existing international standards as much as possible;
- to use the smallest possible amount of communication objects (COB-ID 11 bit CAN identifiers) and to standardize them (creating a list);
- to cover the large amount of devices used in automation;
- to create the Object Dictionary (OD), a database in which they are defined objects, times, data and ways that a device will use to communicate. OD is divided into sections (profiles), each one dedicated to a precise sector. Every object is addressed using a 16 bit index and a 8 bit sub-index. Access to the Object Dictionary is made via communication objects and their relative services.

Index	Object
0000_h	not used
$0001_h – 001F_h$	Static data types
$0020_h – 003F_h$	Complex data types
$0040_h – 005F_h$	Manufacturer-specific complex data types
$0060_h – 025F_h$	Device profile specific data types
$0260_h – 03FF_h$	reserved
$0400_h – 0FFF_h$	reserved
$1000_h – 1FFF_h$	Communication profile area
$2000_h – 5FFF_h$	Manufacturer-specific profile area
$6000_h – 67FF_h$	Standardized profile area 1st logical device
$6800_h – 6FFF_h$	Standardized profile area 2nd logical device
$7000_h – 77FF_h$	Standardized profile area 3rd logical device
$7800_h – 7FFF_h$	Standardized profile area 4th logical device
$8000_h – 87FF_h$	Standardized profile area 5th logical device
$8800_h – 8FFF_h$	Standardized profile area 6th logical device
$9000_h – 97FF_h$	Standardized profile area 7th logical device
$9800_h – 9FFF_h$	Standardized profile area 8th logical device
$A000_h – AFFF_h$	Standardized network variable area
$B000_h – BFFF_h$	Standardized system variable area
$C000_h – FFFF_h$	reserved

Tab. 3.4: Object dictionary

Modbus is a communication protocol created in 1979 by Modicon (today in the Schneider Electric group). It was born as a serial protocol, and later it became also a TCP/IP protocol. Modbus allows a Client/Server or Master/Slave communication (query/response). It has become a standard in industrial communication. Nowadays it is one of the most diffused communication protocols, also in the energy management field.

The main reasons for such a wide usage are:

1) it's been specifically designed for a query/response communication;
2) it's an open and royalty-free protocol;
3) easy installation and maintenance.

There are two different types of modbus:

-serial bus RS232/RS485, which is an example of fieldbus;

-bus TCP/IP or UDP, an industrial ethernet protocol.

The one used in our case study is TCP/IP.

Modbus is based on a master/slave communication, where the Master decides the basis of times. Master device starts a query, to whom Slaves react with a response: this is why we refer to it as a query/response methodology of access. Communication can be meant as a data exchange directly between memory allocations of microcontrollers of the nodes (big shared memory distributed in the net, through the many nodes from which the net is formed).

Function code in the Master query tells the Slave the type of action to make. Slave response provides a structured message, too, in which function field is a simple "echo" of the message received by Master, while data depict information read in the memory registers of the node microcontroller. Only one Master can be connected and send queries on the BUS. Many Slaves (up to 247), though, can be connected to the same BUS and respond to the Master queries. No Slave can communicate with another Slave. Available addresses are 256, and they are divided in this way:

- 0 Broadcast;
- 1 – 247 individual Slave addresses;
- 248 – 255 Reserved (255=0xFF=1 byte)

Master hasn't a specific address, only Slaves have one, unique for the whole bus. Scheduling of the access to the nodes is preconfigured, but there's the possibility to change different schedules while the system is working. Duration of the query, response and broadcast phases depends on the communication characteristics (frame lenght, communication speed, characterics of the line, etc…), while duration of waiting time and preparation depends on the time for the Slave to elaborate them.

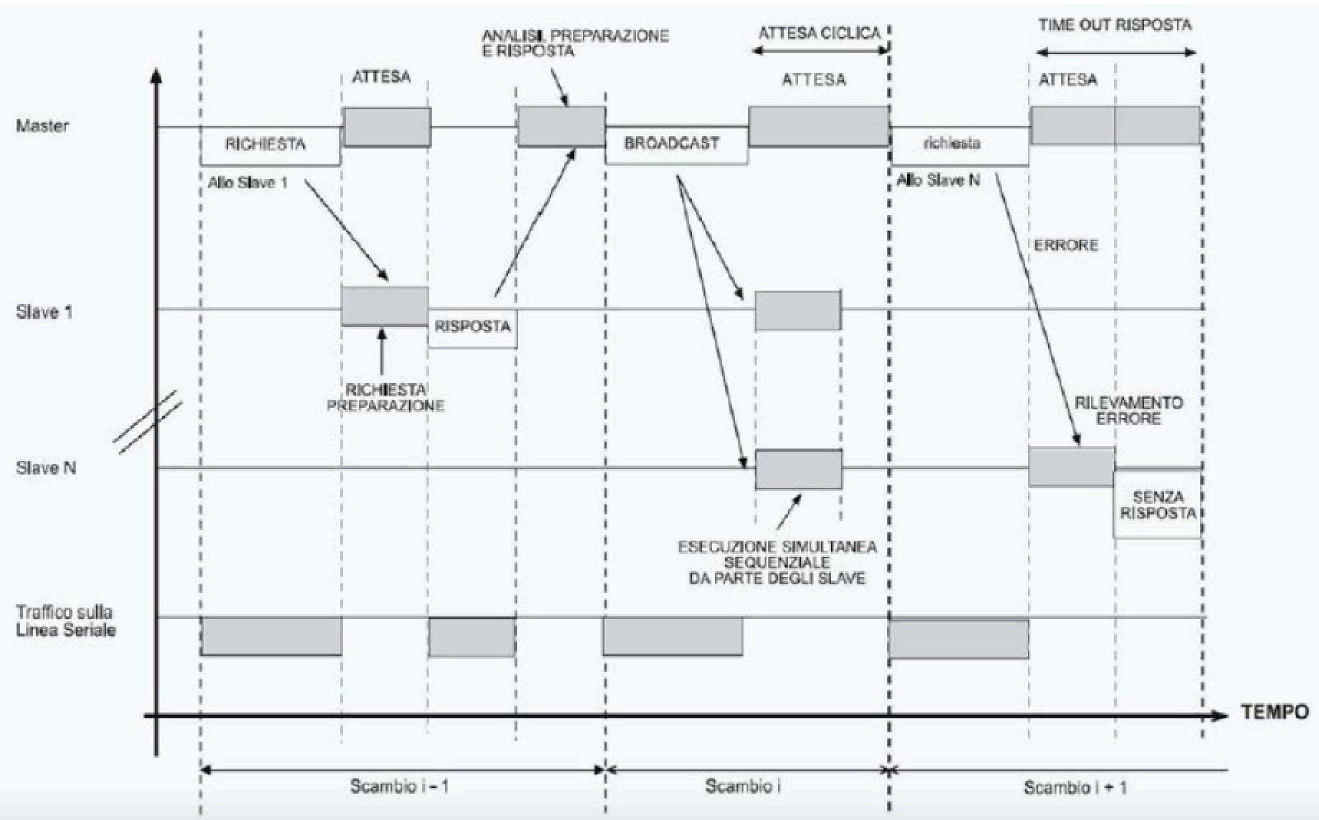

Fig. 3.5: Query/Response duration

Modbus defines a base level of the protocol called PDU (Protocol Data Unit) which doesn't depend on the BUS type used. It's valid both for the Serial Bus and for the TCP/IP.

PDU is made by the following two fundamental fields (maximum dimension of PDU is 253 bytes):

- Function code;
- Data

Protocol mapping inside serial BUS add fields to PDU:

- Address;
- Function code;
- Data;
- CRC/LRC control

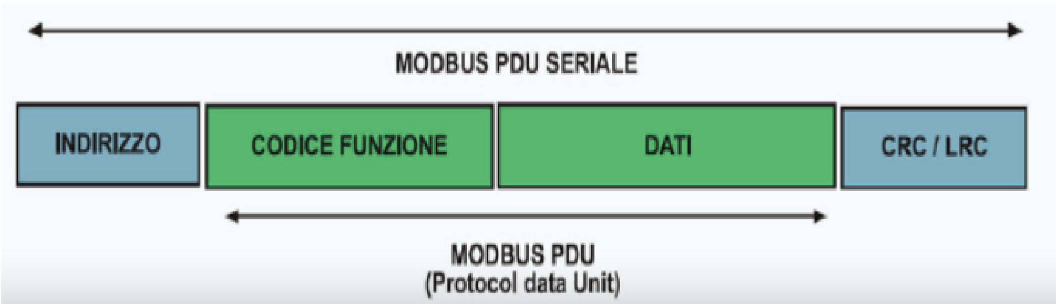

Fig. 3.6: Fields added to PDU in the context of Serial Bus

Function codes are master queries that allow to access the variables inside the Slaves.

Public Function Codes:

- are predetermined;
- guaranteed and unique;
- validated by the organization modbus.org;
- tested to conformity

Function Codes user defined:

- can be implemented without the permission of the modbus.org organization.

				Codici Funzioni		
				Codice	Sottocodice	HEX
Accesso ai Dati	Accesso a Bit	Ingressi Fisico ai Singoli Bit	Leggi Ingressi Discreti	02		02
		Lettura/Scrittura Bits Interni o Fisici	Leggi Bits	01		01
			Scrivi Singolo Bit	05		05
			Scrivi Bit Multipli	15		0F
	Accesso a 16 Bits (World)	Ingresso Fisico ai Registri (World)	Leggi Registro Ingressi	04		04
		Lettura/Scrittura Registri (WORLD) Interni o Fisici	Leggi Registri Interni	03		03
			Scrivi Singolo Registro	06		06
			Scrivi Registri Multipli	16		10
			Leggi/Scrivi Registri Multipli	23		17
			Maschera Registro Di Scrittura	22		16
			Leggi Coda FIFO	24		18
	Accesso ai Records		Leggi Record File	20	6	14
			Scrivi Record File	21	6	15
Diagnostica			Legge lo Stato (Eccezione)	07		07
			Dignostica	08	00-18	
			Acquisisce il Contatore degli Eventi della Seriale	11		0B
			Acquisisce il Log degli Eventi della Seriale	12		0C
			Legge le Caratteristiche dello Slave ID	17		11

Tab. 3.5: Function codes

Modbus configuration of the B&R sensors connected to the model factory is treated in chapter 4.2.1.

3.4. Internet of things

Internet of Things is a term referred to the extension of internet to the world of objects and material places. A sight into this subject could be useful in the respect of our work, which aims to include the model factory in an internet of things network.
Introduced by Kevin Ashton, cofounder and executive director of Auto-ID Center (research cooperative located in the Massatchussets Institute of Technology), during a presentation at Procter & Gamble[18] in 1999, the concept was, lately, developed by the research agency Gartner[19]. Internet of Things represents a possible evolution of the internet web: objects ("things") make themselves recognizable and acquire intelligence communicating data about themselves, and accessing information from others. Few practical examples from the wide world of applications that IoT can have are: alarm clocks ringing earlier in case of traffic, sport shoes communicating performance statistics to runners, medicine flasks alerting familiars of the patient in case he forgets to take the

medicine. Every object can acquire an active role thanks to internet connection[20]. As "things" or "objects" we can more expressly intend categories as: devices, pieces of machinery, plants and systems, materials and "touchable" products and goods. Connected objects, which are the base for the internet of things, are defined "smart objects", and they are characterized by some properties or functionalities. The most important are identification, connection, localization, capacity to elaborate data and capacity to interact with the environment[21]. Internet of Things objective is to make it possible, for electronic world, to make a "map" of the real world, giving an electronic identity to things and places. Objects and locations with Radio Frequency Identification (Rfid) or QR Codes communicate data to the web or to mobile devices like smartphones. Fields of applicability are many: industry (such as manufacturing processes), logistics, mobility, energetic efficiency, remote assistance and environmental care.
Talking about expectations of future development, Gartner predicts that after 2021 more than 26 million things will be globally connected. Market value is estimated to be 80 billion dollars[22]. The main part of this market is represented by Smart Metering applications (smart gas counters which are installed in houses). In the next future a further acceleration in the market is predicted, mainly regarding areas of Smart Metering, Smart Car, Smart Home and Industrial IoT . Expectation is that Internet of Things will change our lifestyle in a radical way. Smart objects, with decision-making capability, will let energy savings both in an individual level (smart home) and in macroscopic level (smart city and smart grid). Additional fields interested by IoT development are synthetically listed below:
-agriculture;
-intelligent transportation systems;
-avionics;
-automotive;
-biomedical engineering;
-robotics;
-wireless sensor networks;
-smart grid;
-smart city;
-embedded systems;
-telemetry;
-telematics;
-zootechnics.

Integration with internet implies the use of univocal IP addresses. IPv4 allows to have 4,3 billions univocal addresses. This is why IoT devices developers are adopting the IPv6 standard, which allows to reach 2^{128} (around $3,4 \times 10^{38}$) addresses.

However, there are also risks associated to Internet of Things, in terms of safety and privacy, depending on the fact that increasingly huge centralized systems are proportionally more vulnerable and manipulable; majority of companies is accordingly working on solutions that are not based anymore on centralized system, but distributed, like blockchain, which could make devices safer. Another aspect to be considered, is the importance not to have undesirable or hidden components inside devices: to guarantee this, trusted computing protocols have been created to monitor products after their exit from the factory.
Talking about privacy, Peter-Paul Verbeek, professor of Philosophy of Technology at the University of Twente (Netherlands), writes that technology already has a strong influence on our moral, which has an impact on our actions. Moreover, he advices not to consider technology simply as an object, and he recommends to consider it as an active agent, instead.[23] Justin Brookman, of Centre for Democracy and Technology (CDT), showed worries about privacy of costumers, telling that some people from the IoT market give priority to data acquisition and big data analytics, and that they just later care about safety.[24] Tim O'Reilly, founder of O'Reilly Media, writes that "IoT is

an extension of human being. Applications are way different when there are sensor and data which influence decision-making."[25] In many articles, WIRED magazine warn about loss of privacy, and about our unconsciousness about the matter.[26] American Civil Liberties Union (ACLU) revealed worries about the possibility that IoT could threaten everyone's control on his own life, declaring that we could become increasingly more transparent to corporations which could have illegitimate intensions.
Researchers Perera, Ranjan, Wang, Khan and Zomaya in a report for EEE IT Professional Magazine[27], identify some problems that every stakeholder (app producers and developers, but also customers), in the field of IoT, needs to face, and they examined the responsibility for every part involved to ensure privacy. Key themes in their report are:

- customer consent: customers need to give consent to data acquisition, despite the fact that they have limited time and reduced technological knowledge;
- freedom of choice: both privacy protection and related norms have to promote liberty of choice;
- anonymity: IoT platforms could increase attention to anonymity using, for example, TOR or similar technologies not to profile too extensively the customer depending on his Things behaviour.

Furthemore, safety is another concern. According to the Business Insider Intelligence Survey of the last four month period of 2014, 39% of the interviewed investors think that safety is the most urgent priority in adopting Internet of Things[28]. Joseph Steinberg, responsible for the cybersecurity section of the magazine Forbes, warned in an article published in January 2014 about the usage of many home IoT devices, such as televisions, kitchen devices, cameras and thermostats, which could spy people in their homes[29]. Even computer systems in cars could be hacked[30].

3.5. Cloud-based information systems

This chapter regards another subject which is closely related to our work: cloud based information systems. Those are increasingly diffused in industry, and their definition has been given in recent years. In 2011, National Istitute of Standards and Technology (NIST), the agency for technology and standardization of the United States of America Department of Commerce, released a special publication aimed to define Cloud computing, with the purpose to serve as a mean for broad comparisons of cloud services and deployment strategies, and to provide a baseline for discussion from what is cloud computing to how to best use cloud computing. The document is reported.

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics**,** three service models, and four deployment models.

Essential Characteristics:

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user- specific application configuration settings.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

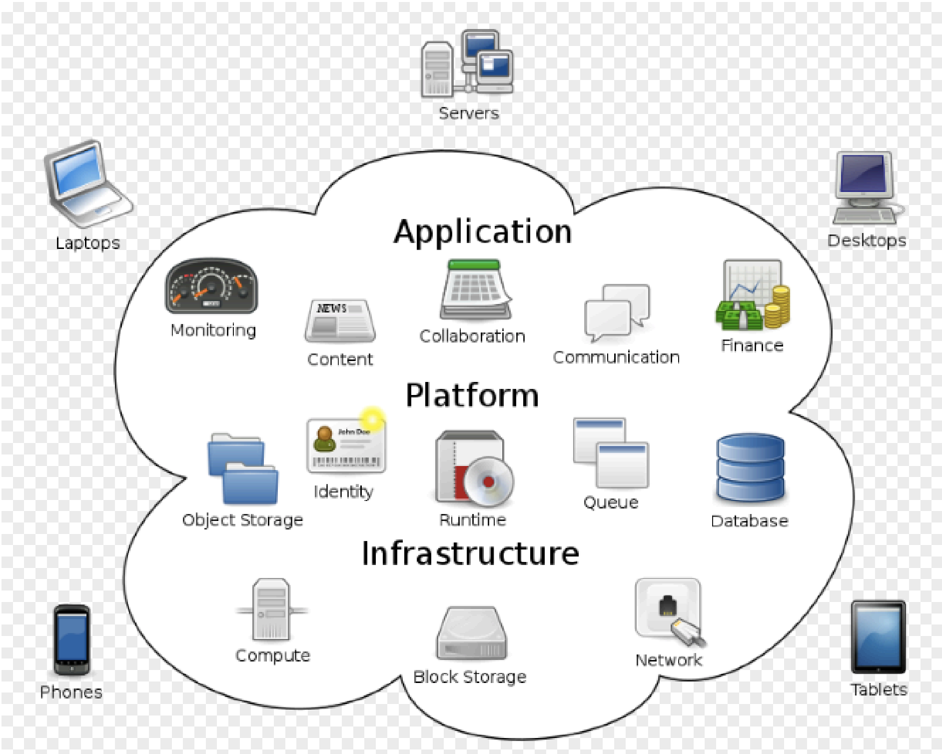

Fig. 3.7: Cloud computing logic scheme

Deployment Models:

Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds)."[31]

Architecture of cloud computing is made by one or more physical servers, generally highly reliable (groups of servers) and physically located in the data center of the service supplier. Service supplier shows interfaces to list and manage its services; administrator client uses these interfaces to select the requested service (e.g. a complete virtual server or just archive) and to manage it (configuration, activation, deactivation). Final client uses the service configured by the administrator client; physical characterics of implementation (real server, location of data center) are not relevant.

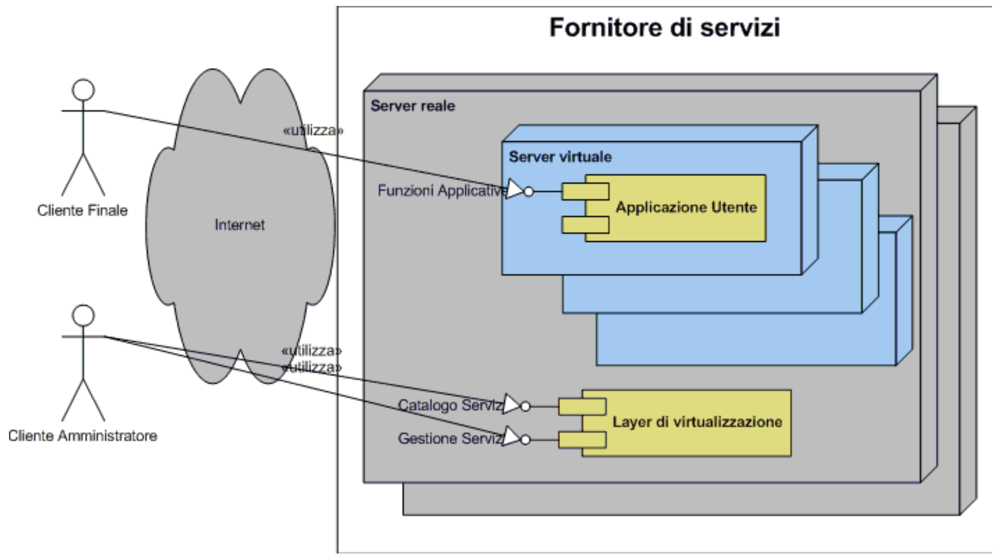

Fig. 3.8: Cloud computing architecture

There are many sensitive aspects related to cloud technology, mostly regarding the volatility of information stored, cryptography and approach to IT security. Many of these aspects could be unknown to final users because, for their nature, they would require advanced informatics knowledge. Cloud computing systems are criticized mainly for users exposure to the following risks:

1) IT safety and customers privacy:

- using a cloud computing system to memorize personal or sensitive data, exposes customer to potential problems in privacy violation. Personal data are saved in Server Farms of companies which, often, are located in a different country from the customer. Cloud provider could illegitimately access to personal data for marketing research and customer profiling[32];
- with wireless connection, safety risk increases and it becomes easier to be victim of informatic piracy, due to more vulnerability of no-wire nets. In case of embezzlement or illegal appropriation of personal data, damage could be very high for the customer, who could even have difficulties to legally persecute the supplier or obtain a refound;
- when talking about companies, data collected in external databases are under risk of industrial espionage

2) International risks in economy and politics:

- these kind of issues can occur when public data are collected and stored in private databases, located in a different country from the "cloud" customers. Intellectual property and personal data are increasingly saved as digital data in private digital databases and partially accessible. No assurance is given to customers for a future access;

- the centralization of cloud databases in some countries could increase the digital gap between those countries and the others.

More safety and guaranties are present when the service supplier is in the same country of the customer, both applying the same privacy and safety rules.

3) Continuity of supplied service:

   - delegating to an external service data management and elaboration, user is strongly limited when suppliers are out of service. A misfunction would also hit a big number of people at the same time (services are shared). Even if the best cloud computing services use redundant architectures (architectures with more means to execute the same function) and qualified personnel to avoid malfunctions and reduce the probability of failure, they don't delete the problem.

4) Difficulty in moving data when service supplier changes:

   - there isn't a standard between service suppliers, and so a supplier change is complicated. This would be problematic in case of bankrupt, tariff changes or litigation with the service supplier.

Implementation of a cloud based information system in our case study is presented in the next chapter.

4. CASE STUDY: RETROFITTING OF A PRODUCTION PLANT IN INDUSTRY 4.0 AND DESIGN OF AN APP FOR DATA COMMUNICATION TO A CUSTOMER

This chapter describes the practical part of the work done during my internship in the Chemniz University of Technology in Summer Semester 2021. After the literature research about KPIs has been done, we applied these metrics to a model factory, which will be described in the next paragraph. In the model factory, optical sensors are mounted, sending data to a Programmable Logic Controller (PLC), receiving them as analogic or digital input through B&R control modules. PLC itself is also capable to receive instruction from a computer, and sending them to the model factory in order to start the process, or actuate a single part of it. Instructions and data are available in Automation Studio Software. This could be defined as the starting point of our work. After that, we configured a selected list of sensors to Modbus, and we added an edge gateway to our setup. Thanks to this, we sent data to an Internet of Things (IoT) Hub, as described in paragraph 4.2. In the Hub, every sensor value was the Property of a Modbus TCP type of thing. We created also a Virtual thing containing KPIs and time/start counters inside the Hub, and we wrote a code in JavaScript language in order to give to the hub instructions of how to obtain results in real time. Last part of our work, described in paragraph

4.3, shows how I designed an app for smartphone in order to communicate this data to a final customer.

4.1.1 Introduction to Fischertechnik Model Factory

24V Fischertechnik model factory is a training model which reproduces a machining line. The system is composed by 4 modules linked together, which are an high-bay warehouse with an automated rack feeder, a robot arm with vacuum gripper, a multiprocessing station (which includes models of an oven and of a saw) and a sorting line with conveyor belt and different exits with detection of color. As reported in the user manual: “The model has four 24 Volt printed circuit boards and can be controlled via any conventional PLC. This way you can create a completely unique program and with the aid of the assignment plan directly control the inputs and outputs. However, the individual programs must be matched to each other, so that it does not lead to a collision.”

The process actuated by the model is the following. Initially, 9 workpieces are stored in the warehouse. Every minute, the rack feeder takes one workpiece and bring it in the location in which it will be collected by the robot arm. Workpieces are divided in 3 colors, and every column of the warehouse is filled with one color. When one column is emptied, the rack feeder moves to the next one, until every workpiece will be taken. At the end of the first station, robot arm grabs the workpiece and, combining a rotation with a translation movement, it brings it to the multiprocessing station. Robot arm has a suction cup in its extremity, which allows to grab workpieces under the effect of vacuum. As mentioned before, the multiprocessing station is composed by the model of an oven, followed by a saw. Initially the workpiece enters the oven, and for a period of time a red light is on, representing the heating process. After that, a short conveyor belt brings the workpiece to the second part of the station, where the saw is activated. Lately, the workpiece is moved by the main conveyor belt, at the end of which an automatism pushes it into a different collecting point based on workpiece’s color. Finally, workpieces are brought back to their positions by the robot arm and the rack feeder.

The following 4 paragraphs are made by main parts of the user manual, describing in detail each station.

4.1.2 Vacuum gripper robot arm

The Society of German Engineers (VDI) defines industrial robots in VDI guideline 2860 as follows:

“Industrial robots are universal handling systems with several axes whose motions, with respect to movement sequence and paths or angles, are freely programmable (i.e. with no mechanical or human intervention) or sensor- guided. They can be equipped with grippers, tools or other means of production and can perform handling and/or production tasks.”

The 3D vacuum gripper robot is therefore an industrial robot that can be used for handling tasks. A workpiece can be picked up with the help of the vacuum gripper robot and moved within a workspace. This workspace is the result of the kinematic arrangement of the robot, and it defines the area that can be reached by the robot's effector. In the case of the vacuum gripper robot, the suction cup of the effector and the workspace correspond to a hollow cylinder whose vertical axis coincides with the robot's axis of rotation.

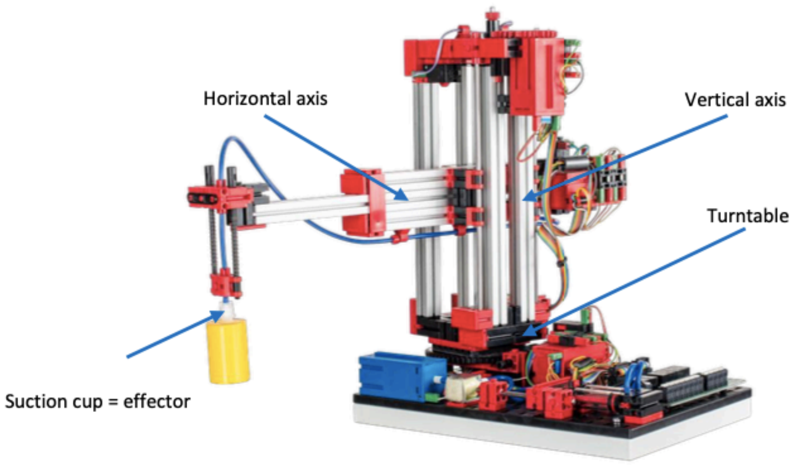

Fig. 4.1: Vacuum Gripper robot arm

The geometry of the workspace is the result of the kinematic setup shown in Figure 4 and comprises one rotary axis and two linear axes.

The typical job for this type of robot can be broken down into the following work steps:

- Positioning the vacuum gripper at the workpiece location
- Picking up the workpiece

- Transporting the workpiece within the workspace
- Setting down the workpiece

Positioning the vacuum gripper or transporting the workpiece can be defined as a point-to-point motion or as a continuous path. The individual axes can be controlled sequentially and/or parallel. This is significantly influenced by the obstacles or predefined intermediate stations present in the workspace.

It is practical to first integrate a reference run in the program in order to establish the absolute position or the absolute angle. To do this, the three axes of the robot are moved to their reference positions and then their positions or angles are set to zero. Now the position of the workpiece can be approached and the workpiece picked up.

The following steps can now be carried out sequentially:

- The gripper robot moves to the alternate position.
- Set the workpiece down.
- The gripper robot pauses at this position.
- Pick up the workpiece again.

For the position control the pulse count of the encoder and the direction of rotation of the motor is combined and can thus, since this is a monotonous movement, approach positions or angles precisely. During this the three axes can be controlled parallel, as long as there is no obstacle present in the workspace.

For this purpose the following measurement and set point values are required:

- Target position or target angle
- Actual position or actual angle
- State of reference switch
- Motor direction of rotation
- Measured encoder pulse

During the suction process of a workpiece the suction cup must first be lowered, in order to create an airtight connection between the workpiece and the suction cup. Then a vacuum must be created in order to temporarily fasten the workpiece on the suction cup. Now the suction cup can be lifted with the workpiece. The function for setting down the workpiece can also be divided into three sections. First the suction cup is lowered, then the air is removed from the cylinder, eliminating the vacuum, and finally the suction cup is raised again.

In the factory simulation the vacuum gripper robot (VSG) is the interface to the other models. Here the vacuum gripper robot should pick up the workpieces from the storage locations of the sorting line with detection and transport them to the "Conveyor system with identification" of the automated high-bay warehouse (HRL). The VSG should first pick up the workpieces from the first storage location (white), until the light barrier located there indicates that there is no more workpiece in the storage location. After this the other workpieces should be picked up in the same manner. It should now place the workpieces in the ready standing workpiece carrier on the conveyor system with identification. If all 9 workpieces (3 white, 3 red, 3 blue) are stored in the high-bay warehouse, these should be taken out of storage sequentially and brought to the multi processing station. For this the VSG should remove the workpieces from the standing ready workpiece carries, transported to the "oven" of the multi processing station and there placed on the extended oven slider. After the workpieces in the sorting line have been sorted according to color, the vacuum gripper robot should transport these back to the high-bay warehouse.

4.1.3 Automated high-bay warehouse (HRL)

A high-bay warehouse is a space-saving storage area for storing and retrieving goods. In most cases high-bay warehouses are designed as pallet rack storage systems. This standardization provides for a high level of automation and connection to an ERP (Enterprise Resource Planning) system. High-bay warehouses are characterized by superior space utilization and high initial capital costs.

Storing and retrieving goods is handled by rack feeders that move in a lane between two rows of racks. This area is part of the receiving station, where identification of goods also takes place. Using conveyor systems, such as chain, roller or vertical conveyors, the goods arrive and are transferred to the rack feeders. If the rack feeders are automated, no one is allowed to enter this area. In the case of the automated high-bay warehouse, the goods are provided on a conveyor belt. The goods are identified by a barcode, which is ready by the trail sensor.

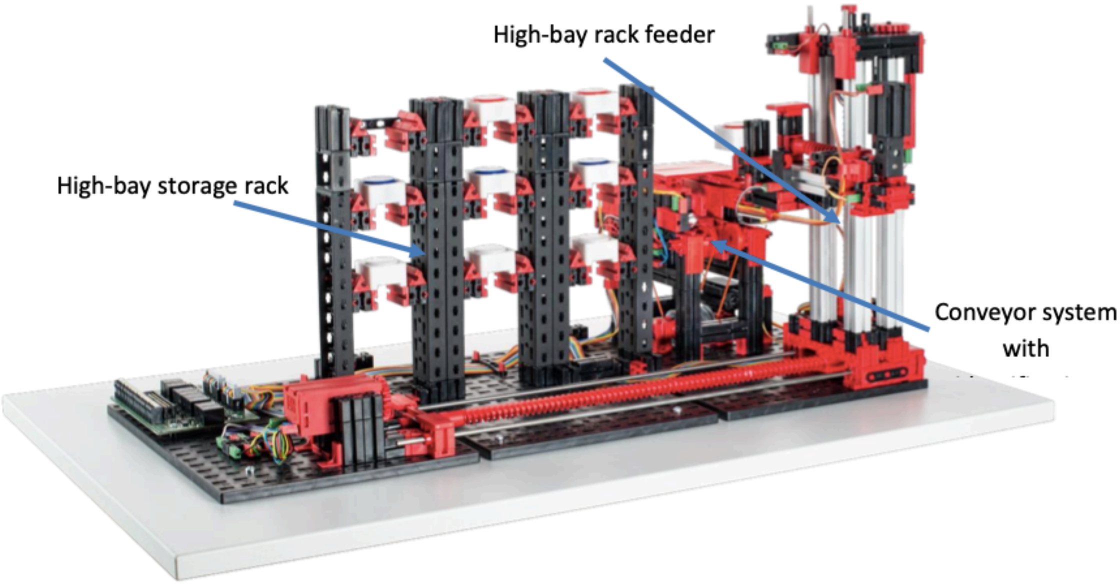

Fig. 4.2: High bay warehouse

Goods are frequently stored based on the dynamic warehousing principle. There is no fixed arrangement between storage position and goods, so the goods to be stored are placed in any free spot. This promises path efficiency. The warehouse management system saves the position of the stored goods, making them available. A (partly) automated identification of goods, which is usually done using FRID chips or barcodes at a central location called the identification site, and standardization of storage areas (same external dimensions, same permitted unit weights) are indispensable.

The ABC strategy in which the warehouse is divided into three zones at varying distances from the storage/retrieval area, is used to further streamline the pathways. Frequently required goods are placed in the A zone, which is directly next to the storage/retrieval area. Rarely needed goods are correspondingly stored in the C zone, which is far away from the storage/retrieval area.

With the automated high-bay warehouse you can demonstrate both the dynamic and the static storage. In the case of static warehousing, for instance, each row is assigned a color. For instance, the top row is assigned the color white, the middle row is assigned red and the bottom row is assigned blue. The individual colored rows are filled from the position closest to the pre-loading zone to the position farthest away from the pre-loading zone.

Regardless of whether you want to use the static or dynamic warehousing, it is practical to first integrate a reference run of the high-bay rack feeder. To do this move the vertical and horizontal axes to their reference positions and then set their positions to zero.

For the factory simulation the static warehouses is suitable, since the workpiece carrier is already in the high-bay warehouse and the workpieces are sorted from the sorting line. If the rack management is now designed so that the high-bay warehouse is filled in sequence, the workpieces are automatically stored sorted by color, since the VSG picks up the sorted workpieces from the storage locations of the sorting line. Thus the white workpieces are stored in the top row, the red workpieces in the center row and the blue workpieces in bottom row. For this no signals of the track sensor are required and simplifies the program.

While the vacuum gripper robot transports a workpiece from the storage locations to the HRL, the rack feeder simultaneously picks up an empty workpiece carrier from the high-bay warehouse and places on the conveyor belt of the "conveyor system with identification". The conveyor belt should now transport the workpiece carriers to the other end of the conveyor belt. When the VSG has placed the workpiece in the workpiece carrier, the workpiece carrier including the workpiece should be conveyed by the track sensor and place on the extension of the rack feeder. Then the rack feeder should store the workpiece on the corresponding storage location. To remove from storage the rack feeder should remove the loaded workpiece carriers and transport to the conveyor system with identification. From there the vacuum gripper robot can remove the workpiece again.

However, if a dynamic warehousing is desired, the signal of the track sensor must be implemented. In addition the barcodes shown in Figure 8 must be placed on the workpiece carriers, so that it can be differentiated between the three colors (white, red and blue).

The workpiece is identified by the automated high-bay warehouse using a simple barcode. The workpiece carriers have a code on them, which is assigned the color white, red or blue. This code is analyzed by a trail sensor. Here the track sensor registered light/dark differences and now these must be assigned a color.

The time interval is limited through the two light barriers before and after the identification unit. Since undesirable reflections can occur on the edges of the workpiece carriers, these must be dismissed in order to avoid false interpretations. This can be dealt with if the width of the light areas (reflective points) or the number of sequential time increments are interpreted as light. So then, for example, the light areas which include more than five sequential time steps can be evaluated as marking and those which have less than five sequential time steps as reflection. This thus defined minimum width limits the number of patterns to be distinguished which can be used to identify the workpiece, but it is sufficient for coding the three colors.

The time interval is limited through the two light barriers before and after the identification unit. Since undesirable reflections can occur on the edges of the workpiece carriers, these must be dismissed in order to avoid false interpretations. This can be dealt with if the width of the light areas (reflective points) or the number of sequential time increments are interpreted as light. So then, for example, the light areas which include more than five sequential time steps can be evaluated as marking and those which have less than five sequential time steps as reflection. This thus defined minimum width limits the number of patterns to be distinguished which can be used to identify the workpiece, but it is sufficient for coding the three colors.

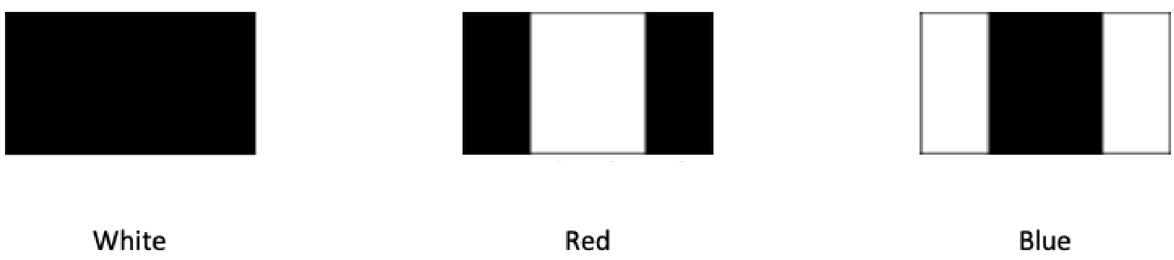

Fig. 4.3: marks applied on the side of the workpiece carrier coding colors

Figure 4.3 shows the assignment between the codes used and the respective colors. These marks are applied to the workpiece carrier side facing the trail sensor, thus allowing assignment of a workpiece carrier to a colored workpiece.

4.1.4 Multi-processing station with oven

In the case of the multi processing station with oven, the workpiece automatically runs through several stations that simulate different processes. These processes use different conveyor systems, such as a conveyor belt, a turntable and a vacuum gripper robot. Processing begins with the oven. The processing starts as soon as the vacuum gripper robot places the workpiece on the oven feeder. The light barrier is interrupted when this happens, thus opening the oven door and drawing in the oven feeder. At the same time, the vacuum gripper is called, which brings the workpiece to the turntable after the firing process. Following the firing process, the door of the oven should be opened again and the oven feeder move outward again. The already positioned gripper robot should pick up the workpiece as with the VSG, transport it to the turntable and set it down there. Provisions are made that the turntable positions the workpiece under the saw, waits there for the duration of processing and then moves to the position on the conveyor belt. There the pneumatic actuated ejector should push the workpiece onto the conveyor belt, which conveys the workpiece to a light barrier and then transfers it to the sorting line with detection. Crossing the light barrier should cause the turntable to return to its starting position and the conveyor belt to come to a delayed stop.

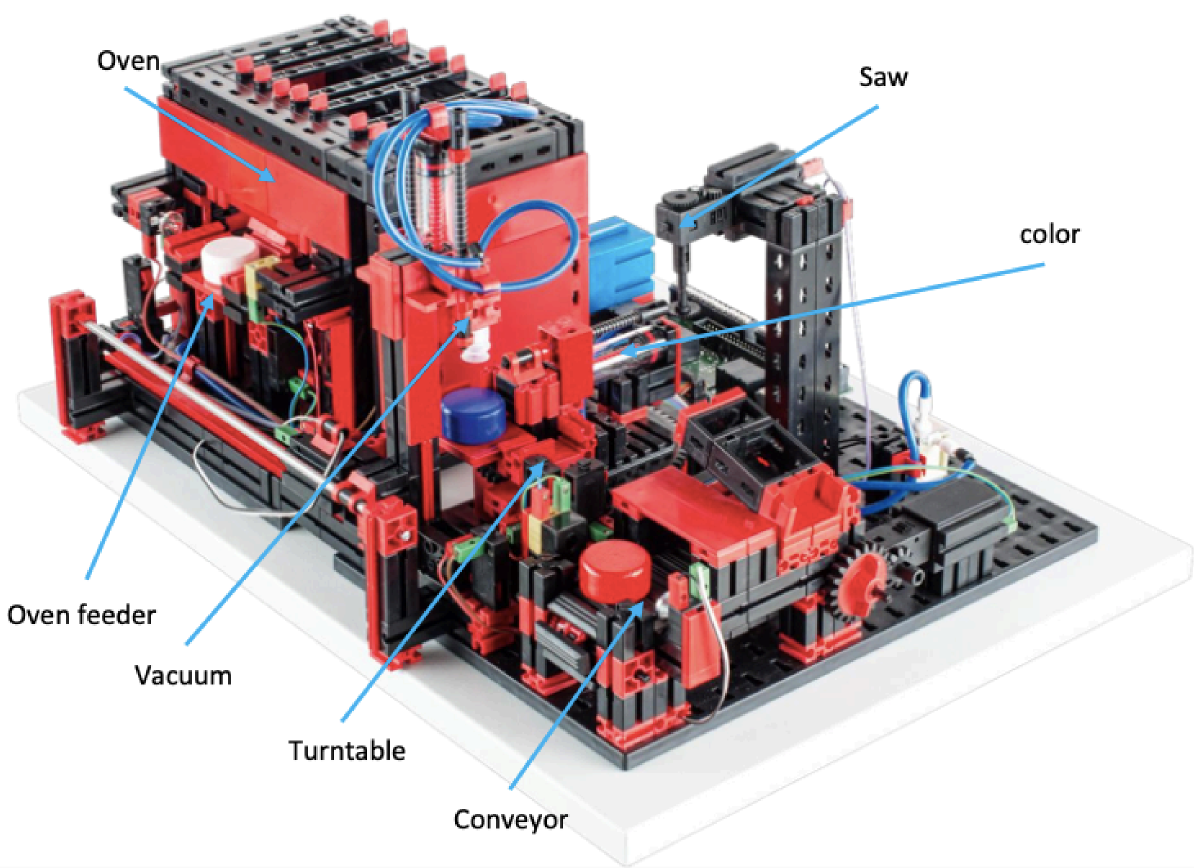

Fig. 4.4: Multiprocessing station with oven

The program sequence can be controlled due to the many inputs and outputs present. Therefore it is practical here to divide the program into three units: oven, vacuum gripper robot and turntable. The particular processes should communicate with each other and thereby ensure that no collisions occur.

4.1.5 Sorting line with detection of color

The sorting line with detection is used for the automated separation of different colored building blocks. In this process, a conveyor belt conveys geometrically identical, yet different colored components to a color sensor, where they are separated according to their color. The conveyor belt is powered by an S motor and the transport route is measured with the help of a pulse switch. The ejection of workpieces is handled by pneumatic cylinders, which are assigned to the appropriate storage locations and are actuated by solenoid valves. Several light barriers control the flow of the workpieces and whether the workpieces are in the storage locations.

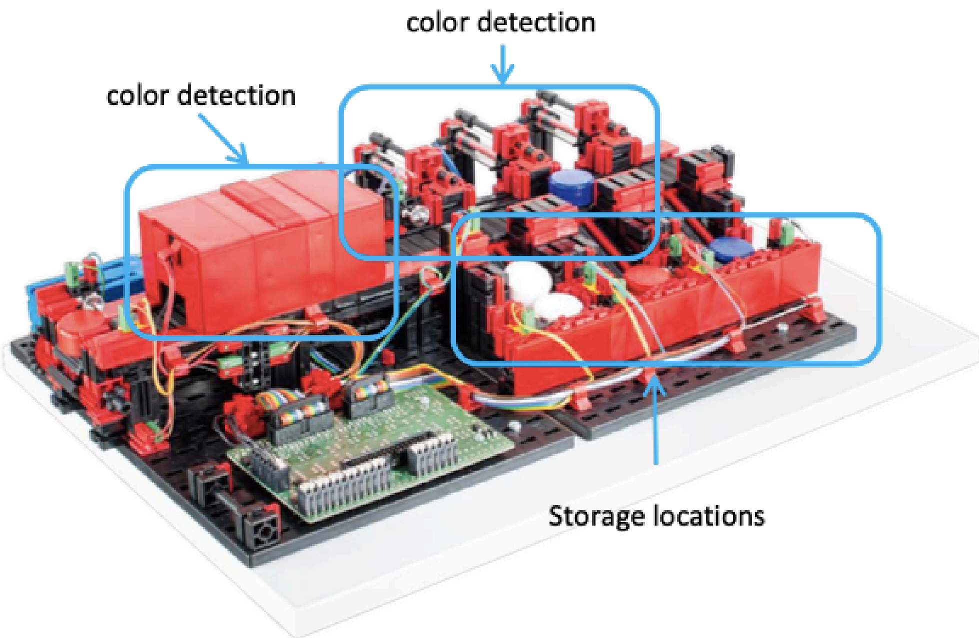

Fig. 4.5: sorting line with detection of color

During this process, color detection is handled by an optical color sensor, which emits a red light and can detect their color based on a surface reflection. Technically speaking the color sensor is therefore a reflective sensor which indicates how well a surface reflects light. The sensor's measured value is therefore not proportional to the wavelength of the measured color and even the assignment of color coordinates or color spaces (e.g. RGB or CMYK) is not possible. In addition to the object's color, ambient light, the surface of the object and the distance of the object from the sensor influence the quality of the reflection. For this reason, it is imperative that the color sensor is protected from ambient light and the surface of the objects are similar. In addition, it is important that the sensor is installed perpendicularly to the object's surface. Threshold values that limit the measured values of individual colors make a distinction between the colored workpieces. Since the value ranges of different color sensors differ, these limit values must absolutely be determined.

The process should be started and the conveyor belt switched on as soon as a workpiece is transferred from the processing station to the conveyor belt of the sorting line and in the process interrupts the light barrier. For the color detection the workpiece runs through a darkened sluice, in

which a color sensor is installed. During this time interval the color should be measured and the workpiece assigned. During this the measured value should be compared with two limit values to assign the workpiece the colors white, red or blue. While the first limit value (for example "limit1") can be used to distinguish between white and red, the second limit value (for example "limit2") can be used to distinguish between red and blue. These limit values must be determined with the aid of tests. Ejection can be controlled with the help of the light barrier located before the first ejector. Depending on the color value detected, the corresponding pneumatic cylinder can be triggered with a delay after the light barrier is halted by the workpiece. This is where the pulse switch comes in, which senses the rotation of the gear wheel driving the conveyor belt. Unlike a time-dependent delay, this approach can withstand disruptions in the conveyor belt speed. The ejected workpieces are fed through three chutes to the particular storage locations. During this the storage location, which is found closest to the detection is assigned the color white, the center the color red and the furthest away the color blue. The storage locations are equipped with light barriers that detect whether the storage location is filled or not. However, the light barriers cannot tell how many workpieces are in the storage location.

From this storage location the vacuum gripper robot can now again pick up the workpiece and transport it to the high-bay warehouse to store it there again.

4.2 Implementation of a cloud-based information system on a B&R plant control system

The Fischertechnik model factory is controlled by an industrial PLC, which exchanges inputs and outputs with a computer through a B&R system. In this chapter is presented how we proceeded to implement a cloud-based information system, starting from the existing layout.

4.2.1 Modbus configuration of B&R controls

B&R Industrial Automation GmbH is an Austrian automation and process control technology company. It was founded in 1979 by Erwin Bernecker and Josef Rainer, and its headquarter is in the state of Upper Austria. The company produces systems for machine and factory control, HMI and motion control. In addition to complete setups, B&R also offers single components. The product range is focused on machinery and equipment manufacturing, and the company is also active in the field of factory and process automation. Due to its characteristics, a B&R control system was followingly chosen for our model factory. The following picture was taken from above, and, from the sight, it's possible to see the assembly of the B&R system mounted on the factory model platform.

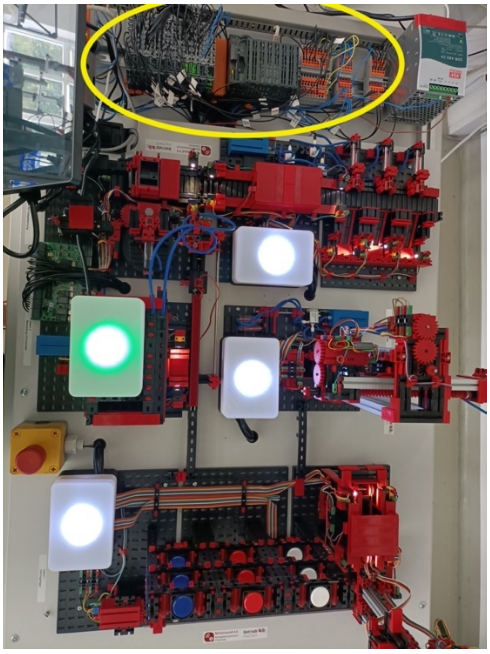

Fig. 4.6: A yellow balloon highlights the B&R control system in a sight from above

The logic behind control of the system is based on PLC receiving data from sensors as digital and analogic inputs and communicating them to the computer, and, reversely, sending the computer instructions as digital outputs which will be actuated by controls. Different types of B&R modules are also related to different types of inputs/outputs. In our application, we have 1 analogic input module, 2 digital input modules, 1 digital input/output module, 3 digital counter modules and 3 digital output modules. The hardware structure that we've just seen is managed by Automation Studio, a circuit design, simulation and project documentation software for fluid power systems and electrical projects conceived by "Famic Technologies Inc.". Automation studio offers a variety of tools and views that allow to manage an industrial system. First of all, after our B&R system has been configured, in the "Physical view" we have the list of all the B&R modules, and if we are in the "system designer" mode it's also possible to have a photographic view of them. From this view, it's also possible to access options and information regarding PLC ports.

Fig.4.7: B&R Modules connected to the Model Factory, digital view from the software Automation Studio. From left to right: PLC, Analog Input module, 2 Digital Input modules, Digital Input/Output Module, 3 Digital Counter Module and 3 digital output modules

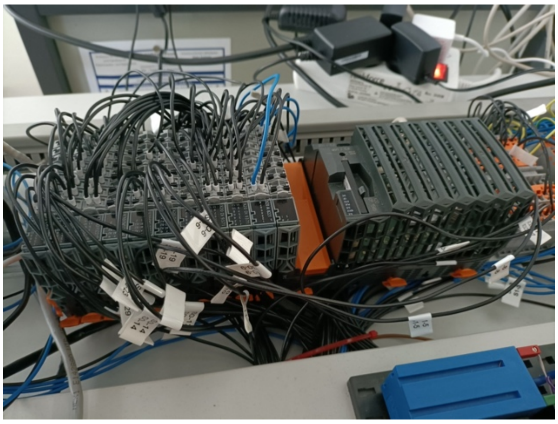

Fig. 4.8: real B&R controls

From the "Physical view" list, it's possible to open the "I/O mapping" of each module, that is a window showing the channel list of inputs and outputs, the name of the channel program and the data type related to each channel (e.g. "int", "bool", etc…). Previously, a list was made containing a correspondence of each channel name with its related sensor, and saved as an Excel file.

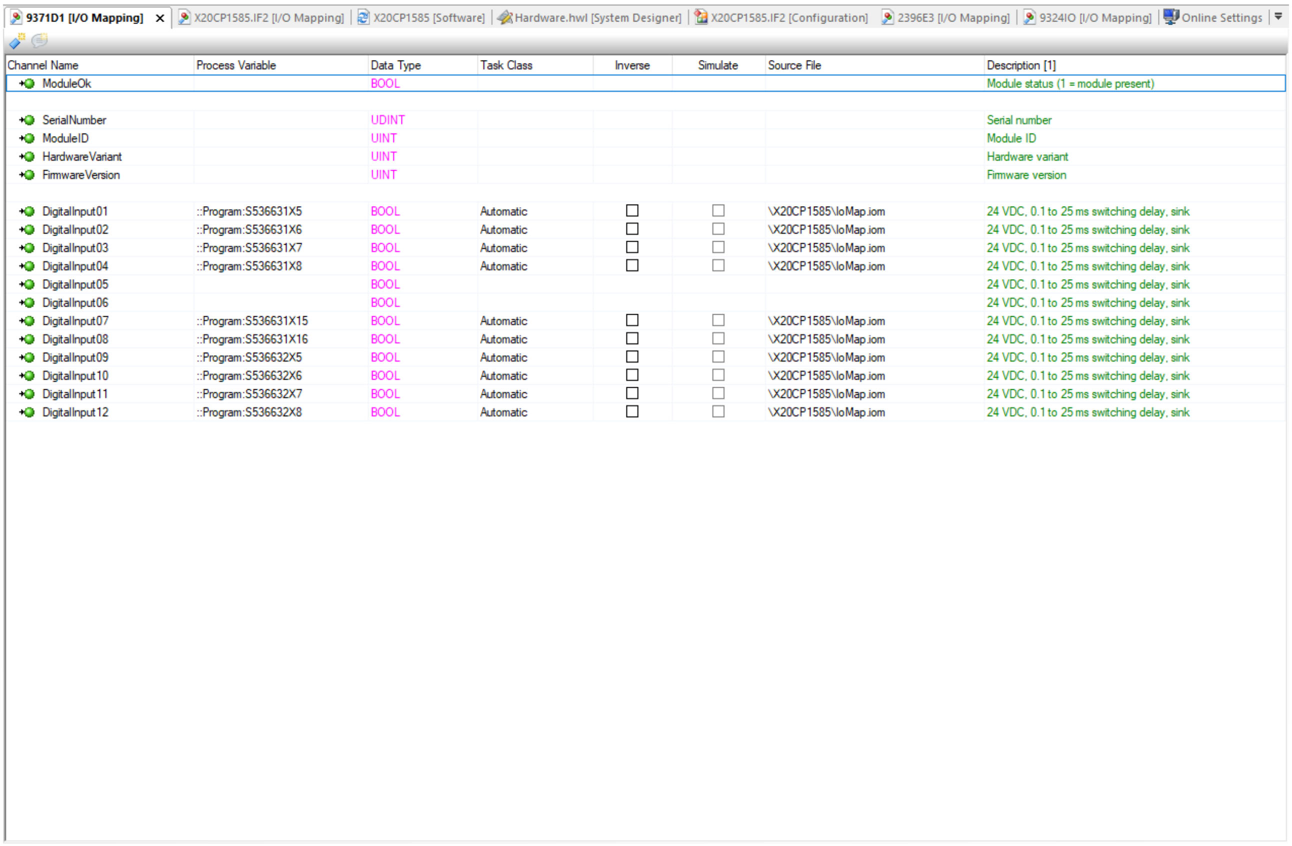

Fig. 4.9: I/O Mapping window of 9371D1 (Discrete Input Module)

If we activate "monitor mode", in the "I/O mapping" it becomes possible to visualize real data coming from the model factory while it's running.

Up to this point, we can get information from the model factory, just relying on the data coming from each single channel of the "I/O mapping" view. What we aim to do, as a further step, is to connect the model factory and the B&R automation to an IoT Hub, in order to convert a fully "wired" system into a cloud-based information system. To accomplish that, the first step is to make the Modbus configuration of the sensors.

At first, we have to establish which sensors have to be monitored. In order to select which sensors have to be chosen between the multitude, we have a look into the selected KPIs, and we create a table with all the required sensors.

KPI	Factors	Sensors
PH6 Overall Equipment Effectiveness R1*R2*R3	Time with no failures	const
	Requested time	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss)
	Production time	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
	Total time	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss)

	Ideal time	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss)
	Requested time for quality compliance	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss)
PH8 Operational availability due to manteinance	Total operating time	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
	Total operating time+downtime	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
M11 Availability based on uptime	Operating time	"
	Required operating time	
M12 Availability based on time to restore	Operating time	"
	Operating time+recovery time	"+timer recovery time
M13 Production based availability	Actual production	Ventil Auswurf (blau/rot/weiss)
	Required production or other reference level within a specified period of time	Const
M15 Frequency of manteinance shutdown (No./Year)	Number of shutdowns	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
	Year (Week)	const
M19 Outcome rate (%)	Objectives achieved	const
	Assigned objectives	const
E5 MTBF Average operating time between failures (hours)	Total operating time	"
	Number of failures	=number of shutdowns
E6 Average repair time	Total time to recovery	Timer recovery time
	Number of failures	=number of shutdowns
O&S16 Mean time to recovery = E6	Total time to recovery	"
	Number of failures	=number of shutdowns
O&S17 Average repair time = E6	Total repair time	
	Number of failures	=number of shutdowns
Allocation efficiency	Actual unit busy time	= operational time+downtime
	Planned busy time	Const
Utilization efficiency	Actual production time	= operational time
	Actual unit busy time	= operational time+downtime
Availability	Actual production time	= operational time
	Planned busy time	Const
Quality compliance ratio	Quality compliant workpieces	Ventil Auswurf (blau/rot/weiss)
	Total amount of workpieces	Ventil Auswurf (blau/rot/weiss)
Number of corrective actions in a year		Counter button for the maintenance actions

Lead time	Time to satisfy the request of a customer	Total operational time due the chosen no. of workpieces
Station inefficiency	Number of failures of the working station	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
	Number of failures of the factory	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
Downtime due to retrofitting	Time to implement retrofitting	Timer
	Total downtime	Motor vertical hoch (HRL), Ventil Auswurf (blau/rot/weiss), Motor drehen gegen uhrzeigersinn (Sauggreifer), Leuchte Ofen
Data acquisition capability	Data acquired	Amount of information acquired from the factory
	Estimated data	R&D

Tab. 4.1: List of KPIs with related sensors

In the first column, it's possible to read the name of KPIs. In the second one, following the method showed in the Chapter 3.2, there are the factors of the ratio that has to be calculated. Finally, in the third column, we can read the list of sensors to be monitored. "Const" stands for "constant", and indicates a parameter which have to be theoretically chosen, and doesn't depend on the sensor measurements. Colors helped to orient ourselves and have a clearer view. Even if every KPI refers to maintenance, I could identify some of them which can also be referred to other aspects, and, specifically, in the first column, green is related to quality assurance, yellow to economic management and blue to research and development. Colors in second column have the same meaning of tab.

After this, a table with only sensors was written, grouping them by station, adding a brief note when helpful, and containing the address to be used in order to make the Modbus configuration.

Station	Sensor	Notes	Address
Sauggreifer	Motor drehen gegen uhrzeigersinn	Rotation motor anticlockwise	F322O3 (Key 6)
Sauggreifer	Motor horizontal vorwarts	Motor horizontal foreward	F322O3 (Key 4)
Sauggreifer	Encoder horizontal impulse 1		2396E1 (Key 2)
Sauggreifer	Encoder vertikal impulse 1		2396E1 (Key 1)
Sauggreifer	Encoder drehen impulse 1	Rotation impulse 1	2395E2 (Key 1)

HRL	Motor vertikal hoch	Vertical motor up	F322O1 (Key 6)
HRL	Motor ausleger vorwarts	Engine outtrigger	F322O1 (Key 7)
HRL	Referenztaster horizontal		9371D1 (Key 1)
HRL	Referenztaster vertikal		9371D1 (Key 2)
HRL	Referenztaster ausleger vorne	Front arm	9371D1 (Key 7)
Ofen	Motor säge		F322O1 (Key 12)
Ofen	Leuchte ofen		F322O2 (Key 1)
Ofen	Referenzschalter Drehkranz	Position sauger	9371D1 (Key 9)
Ofen	Referenzschalter Drehkranz	Position Förderband	9371D1 (Key 10)
Ofen	Lichtschranke Ende Förderband		9371D1 (Key 11)
Sortier	Lichtschranke Eingang	Light barrier entrance	9371D2 (Key 7)
Sortier	Lichtschranke nach Farbsensor	Light barrier after color sensor	9371D2 (Key 8)
Sortier	Ventil Auswurf	White	F322O2
Sortier		Blue	
Sortier		Red	

Tab. 4.2: List of B&R sensors which have been configured to Modbus and connected to the IoTHub via a Gateway

Once we have cleared which sensors to monitor, and their addresses, we move to the configuration itself. Communication between PLC and gateway is made through the PLC Ethernet port, so, in order to configure Modbus, we have to use the Ethernet configuration window in Automation Studio.

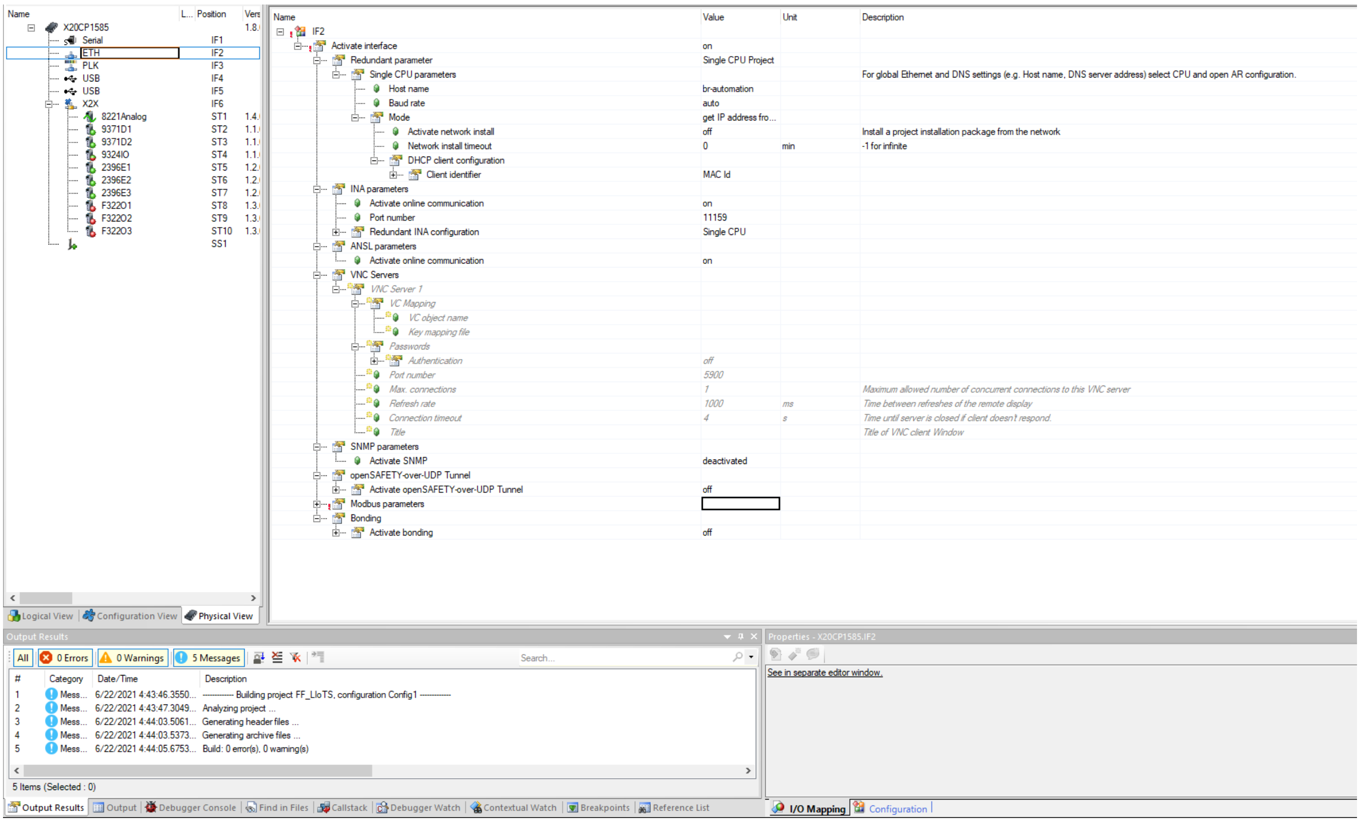

Fig. 4.10: Ethernet configuration window

From the window, we enable “Activate Modbus communication” button, and “Use as modbus slave”, because, in Modbus logic applied to our case, PLC is the slave and IoTHub is the master. Modbus slave has 4 addressing types: “Coils”, “Discrete inputs”, “Input registers” and “Holding registers”.

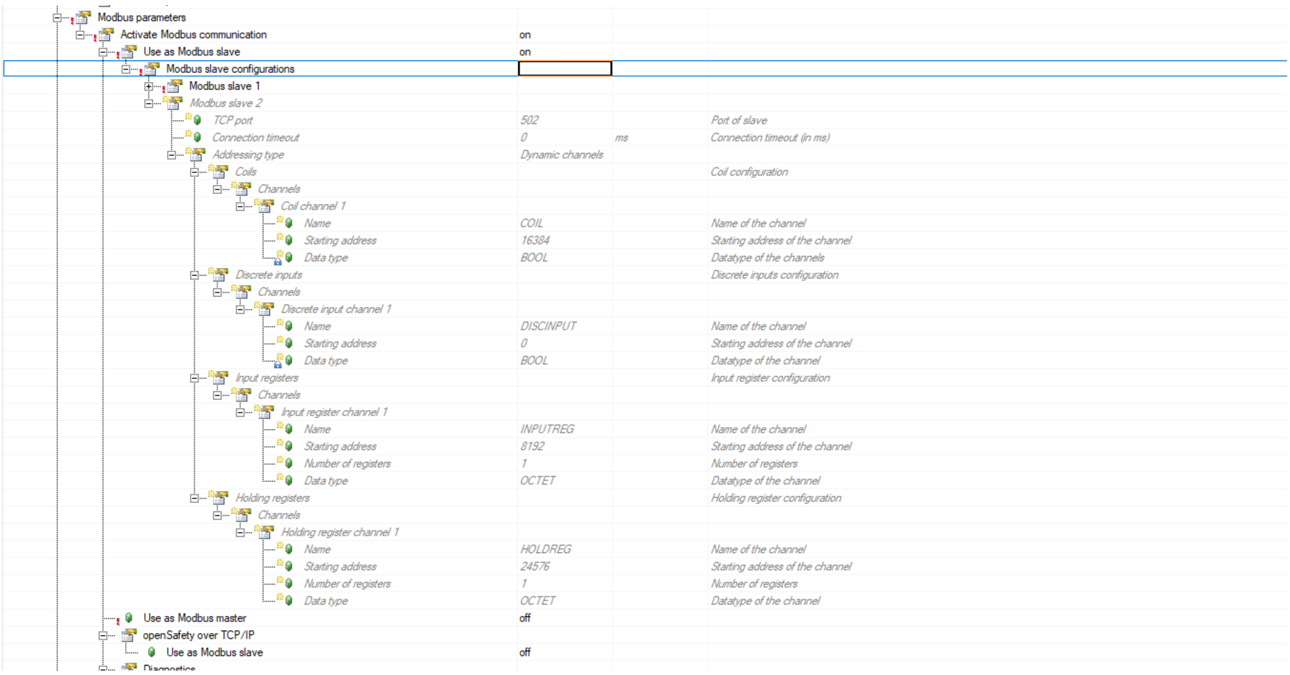

Fig. 4.11: Modbus slave configuration

What we are going to do, is to create a new dynamic node for each sensor that we want to monitor. For every sensor, we give to the channel a name and a starting address (we've set addresses from 0 to 20). The dynamic nodes created are "Discrete Input", because they are related, as possible to read in the Automation Studio guide (image xx), to digital input ("Digitaler Eingange").

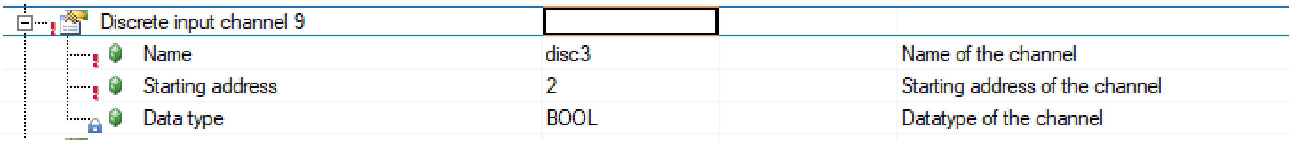

Fig. 4.12: Creation of a new dynamic node

Funktions Code	Interne Bezeichnung	Protokollspezifische Bezeichnung
1	Lesen mehrerer digitaler Ausgänge	Read Coils
2	Lesen mehrerer digitaler Eingänge	Read Discrete Inputs
3	Lesen mehrerer analoger Ausgänge	Read Holding Registers
4	Lesen mehrerer analoger Eingänge	Read Input Register
5	Schreiben eines digitalen Ausgangs	Write Single Coil
6	Schreiben eines analogen Ausgangs	Write Single Register
15	Schreiben mehrerer digitaler Ausgänge	Write Multiple Coils
16	Schreiben mehrerer analoger Ausgänge	Write Multiple Registers
23	Lesen und Schreiben mehrerer analoger Ausgänge	Read/Write Multiple Registers

Tab. 4.2: Screenshot from the Automation Studio guide. It is possible to see that the protocol "read discrete inputs" is related to digital input.

After that, we select "build the configuration", and later we transfer the configured program to the PLC.

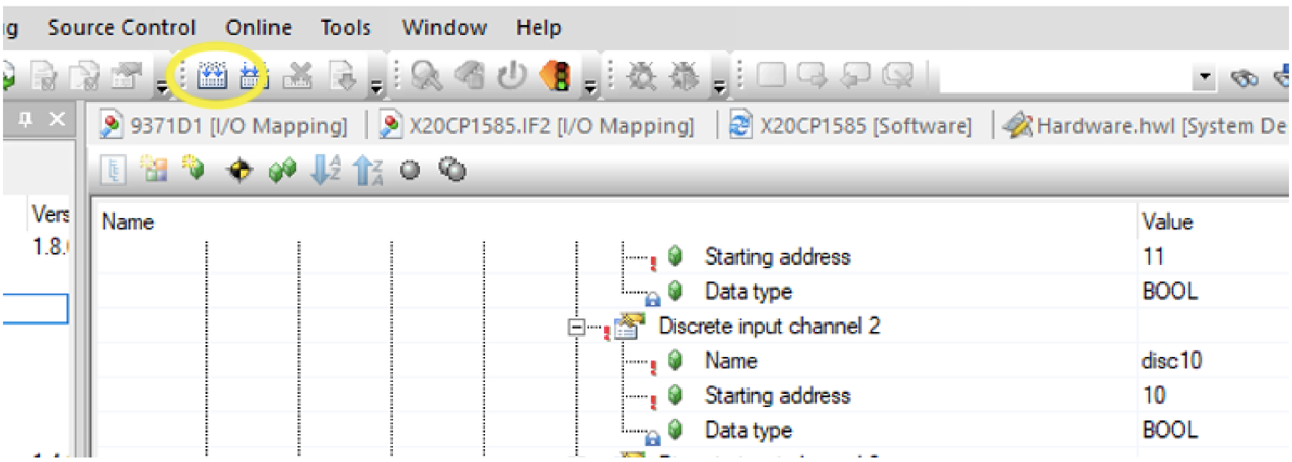

Fig. 4.13: Building the configuration

At this point, we go to the Ethernet I/O Mapping window and we select the program for our sensor.

disc3	::Program:S536633X20	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
discIn01	::Program:S536631X6	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc12	::Program:S536633X10	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc13	::Program:S536633X7	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc15	::Program:S536631X5	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc16	::Program:S536631X6	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc17	::Program:S536630X6	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc14	::Program:S536633X6	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc18	::Program:S536632X5	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc19	::Program:S536632X6	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc20	::Program:S536632X7	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc21	::Program:S536631X15	BOOL	Automatic	☐	☐	\X20CP1585\IoMap.iom
disc23	...	BOOL				

Tab. 4.3: Ethernet I/O Window. Selecting a program for the dynamic node that we created

It is possible to know the code identifying each program by reading it in the I/O Mapping window of the module to which the sensor is associated.

Once the program has been selected, it is possible, again, to build the configuration and transfer it to the PLC.

4.2.2 Implementation of the IoT hub

The IoT Hub was provided by Elco company, and it's a website which gives to a developer the possibility to use many functions to monitor and manage data flow coming from PLC. The main sections of IoTHub are "Dashboard", "Things", "Agents", "Workflows" and "App Designer". Dashboard is a window in which data coming from the "things" can be visualized through graphs and charts, giving the developer the chance to design them. Things are the core of the Hub. They are the representation of real or virtual objects we can interact with. In the help section of the website, the title of the paragraph dedicated to things is very representative: "Everything is a thing". They can be a device like a RFID scanner, a controller like a PLC from a supported vendor or standardized protocol (like OPC UA), a database, a file or any other collection of properties. A Thing must have a (unique) title and a type. In our case, we created two things of interest: a "Modbus TCP" type of thing, which is the representation of the physical control system mounted in the factory, and a "Virtual" thing, which is going to give us the calculation results for our KPIs, plus a time counter and a number-of-starts counter which are propaedeutic to the calculations.
In creating the Thing, first step was to specify the Thing type.

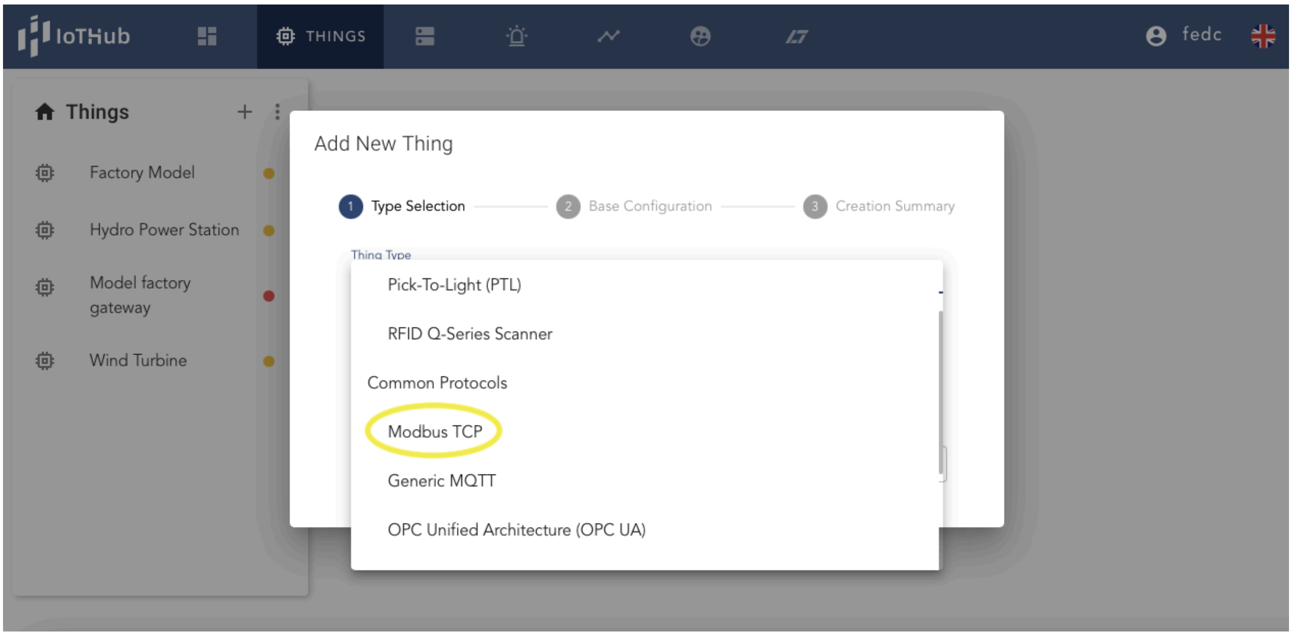

Fig. 4.14: Add a new Thing window from IoTHub, first step: Type Selection

As a second step, we type the name of the Thing and a description, and we select the Endianness.

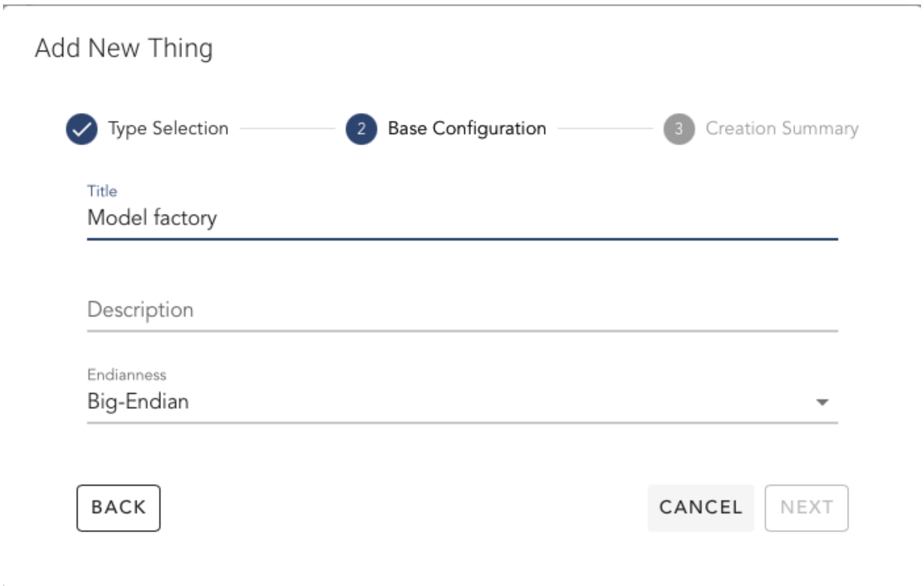

Fig. 4.15: Add a new thing second step: Base Configuration

As last, we check the creation summary and we install.

The key element to connect data sources like devices to the IoTHub are Agents. An Agent is a standalone application, which supports one or more protocols and can connect to many devices, controllers or other data sources, each represented by a thing. It communicates with the IoTHub via a gateway with a dedicated port number. Each agent must have a predefined avatar in the IoTHub. It is identified by a unique id. With this id, the agent application can be registered at the IoTHub. For our application, also the gateway has been provided by the Elco company. The peculiarity of the gateway in comparison with a common router is that the Gateway can communicate with

Modbus protocol language, being a translator for the PLC, which "speaks" Modbus, to the HTML speaking IoT Hub.

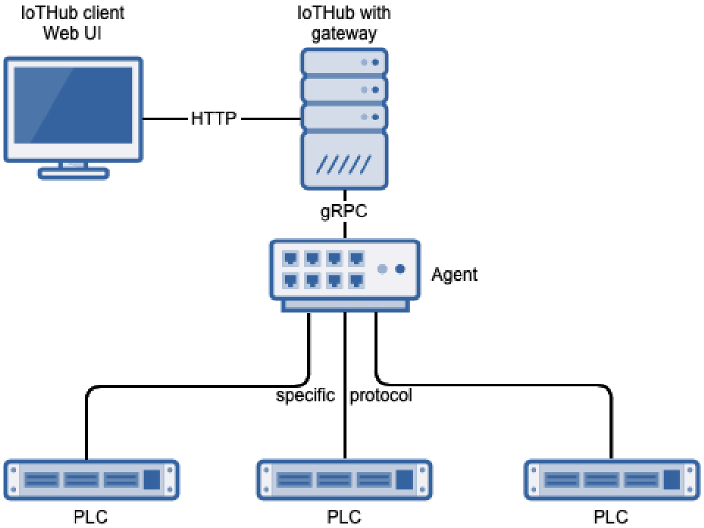

Fig. 4.16: Agent architecture

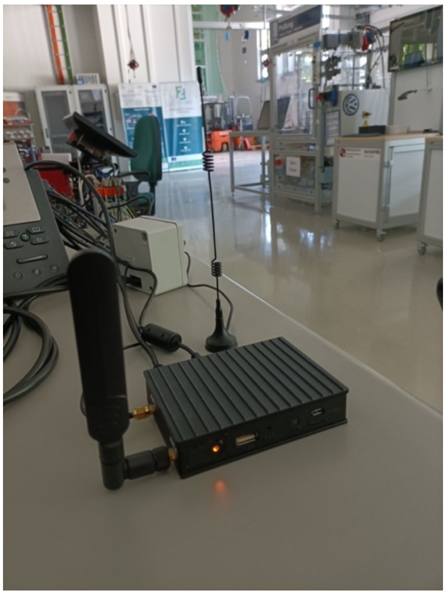

Fig. 4.17: gateway

In other words, PLC communicates via ModBus protocol, and the edge gateway moves data to the cloud via HTML protocol.

When creating a new Agent, we name it and we select "Modbus TCP" as Agent type.

Fig. 4.18: IoTHub add Agent window

In the new generated Agent window, it is possible to read its token, which is its unique identifier code, and the ID of the Thing related, with its properties.

In order to monitor the sensors via the IoTHub, it is now possible to add properties from the Model Factory Thing window. As a first step, we set the name of the sensor as the title for the Property, then we set "Boolean" as a type (sensor value could be "True" or "False"), "Discrete Input" in the "Area" field and the same starting address we selected in Automation studio in "Address" field.

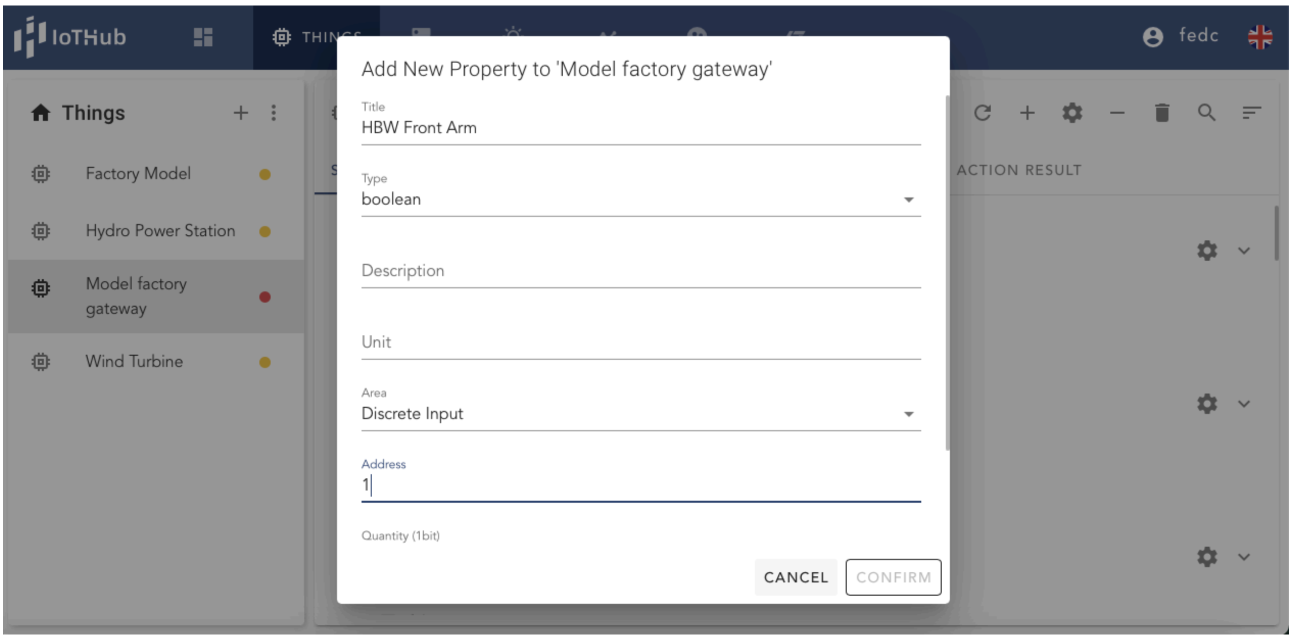

Fig. 4.19: Add a new Property window in the IoTHub

Once the property is created, we have to connect the Thing in the Actions window.

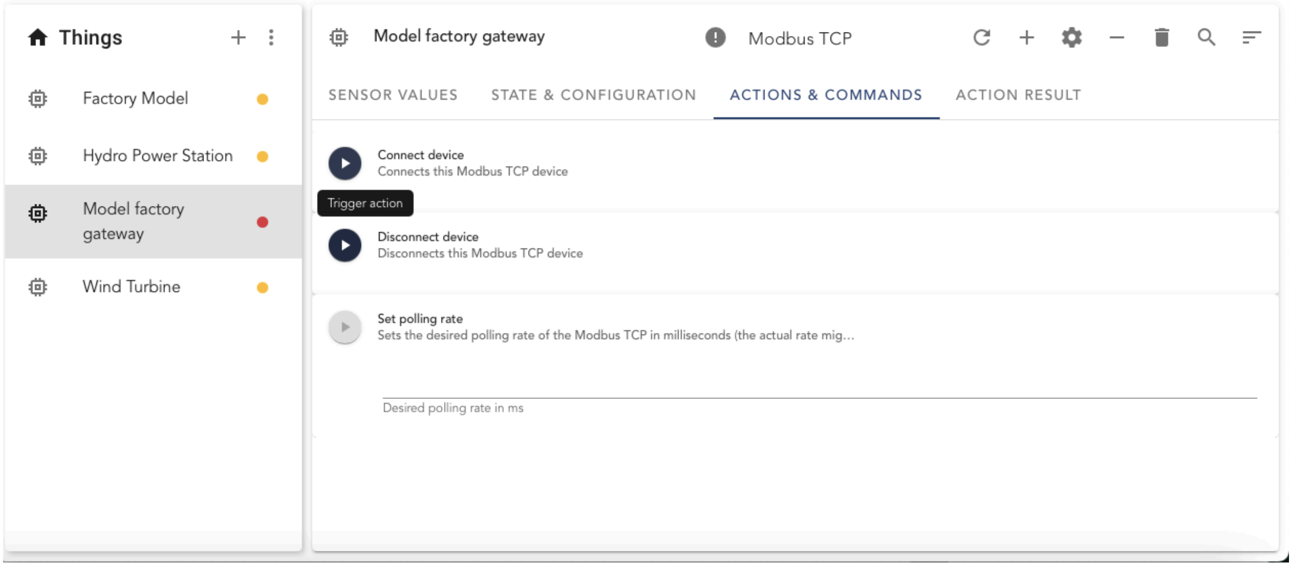

Fig. 4.20: Thing "Actions & Commands" window

A green led will appear in case the Thing will be connected. At this point, created Properties will look as shown in the picture below.

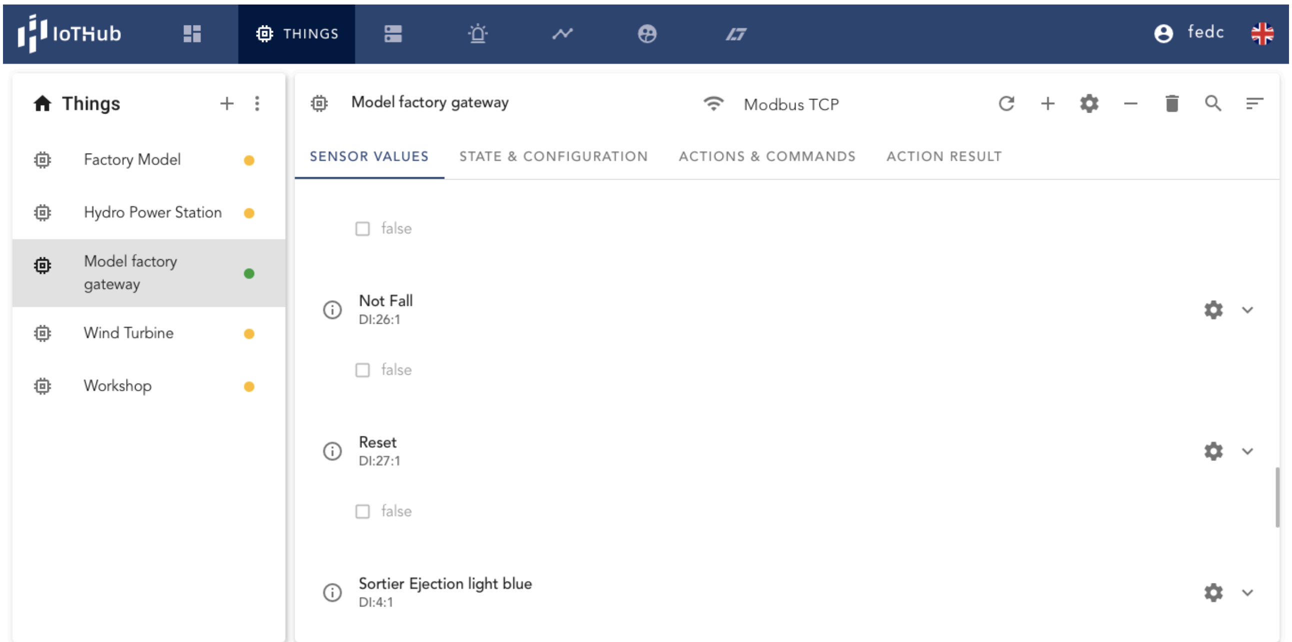

Fig. 4.21: Connected Thing "Sensor values" window

It's now possible to check the functionality of the property starting the factory from Automation Studio.

While the system is running, it is possible to see the property to switch from "false" to "true" when the sensor is triggered by the passing of a workpiece.

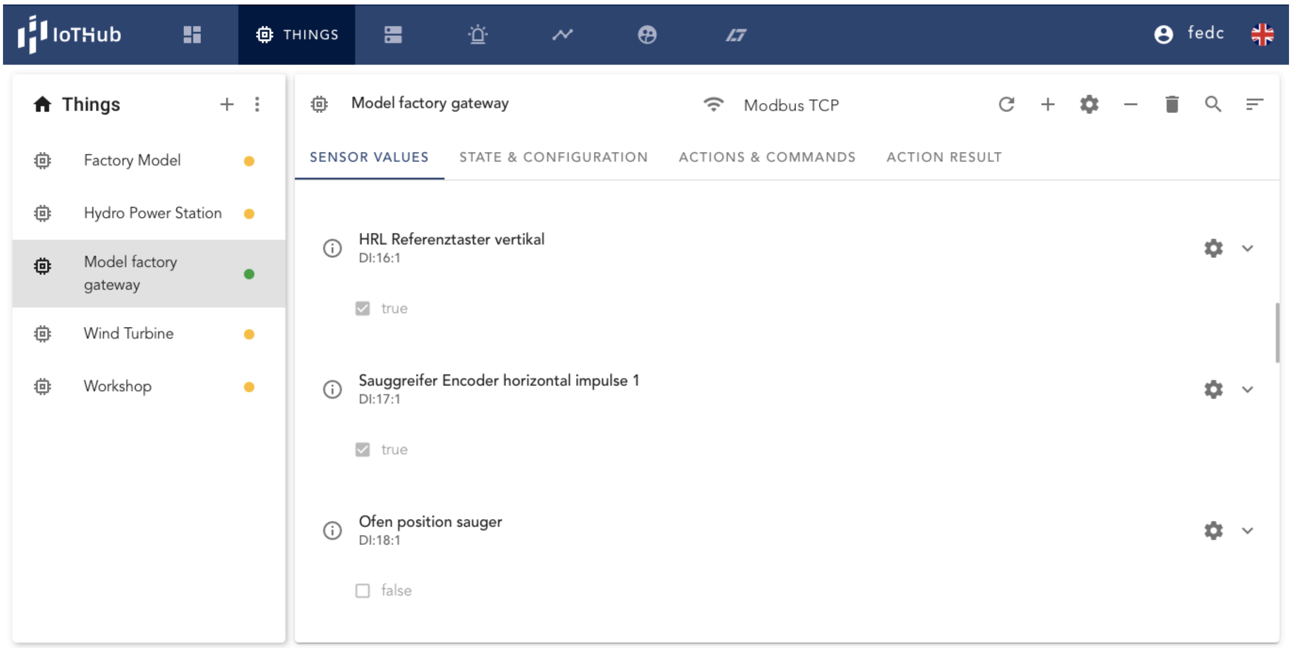

Fig. 4.22: triggered sensors from the sensor values IoTHub window

After this, first steps in making a transition from a wire-based to a cloud-based system have been done. Boolean data communicating the status of a sensor are now available both in Automation Studio software (installed in the same pc where PLC is connected) and on internet, in the IoT hub. Exploiting the cloud-based computing, we want to make a further step, and obtain a deeper level of information. In fact, at the moment, we just know if a sensor is triggered or not. We now want to know how many times the sensor is triggered during a cycle, and, from this, we want to calculate a certain KPI for the part of the factory involved. To do so, our next step is to create a second Thing in the IoT Hub, but, this time, a virtual thing. Our aim is to add one property, in the virtual thing, for each KPI. From the initial list of KPIs, a further selection has been done, complying with time and coding restraints. The coding part will be shown next. The final list of KPIs is summarized below.

M11 Availability based on uptime	$$M11 = \frac{operating\ time}{required\ operating\ time}$$
M12 Availability based on time to restore	$$M12 = \frac{operating\ time}{operating\ time + recovery\ time}$$
M13 Production based availability	$$M13 = \frac{actual\ production}{required\ production\ within\ a\ specified\ period\ of\ time}$$
PH8 Operational availability due to maintenance	$$PH8 = \frac{total\ operating\ time}{required\ operating\ time + downtime}$$
E5 MTBF Mean time to failure	$$E5 = \frac{total\ operating\ time}{number\ of\ failures}$$
E6 Average recovery time	$$E6 = \frac{total\ time\ to\ recovery}{number\ of\ failures}$$
O&S 17 Average repair time	$$O\&S17 = \frac{total\ time\ to\ repair}{number\ of\ failures}$$

Tab.4.4: list of chosen KPIs

Before it could be possible to get these results, it was necessary to write a code containing the calculations. IoT Hub also contains a window called "Workflows", in which it was possible to write a code to use data and obtain new results. A screen of the workflow window is presented below.

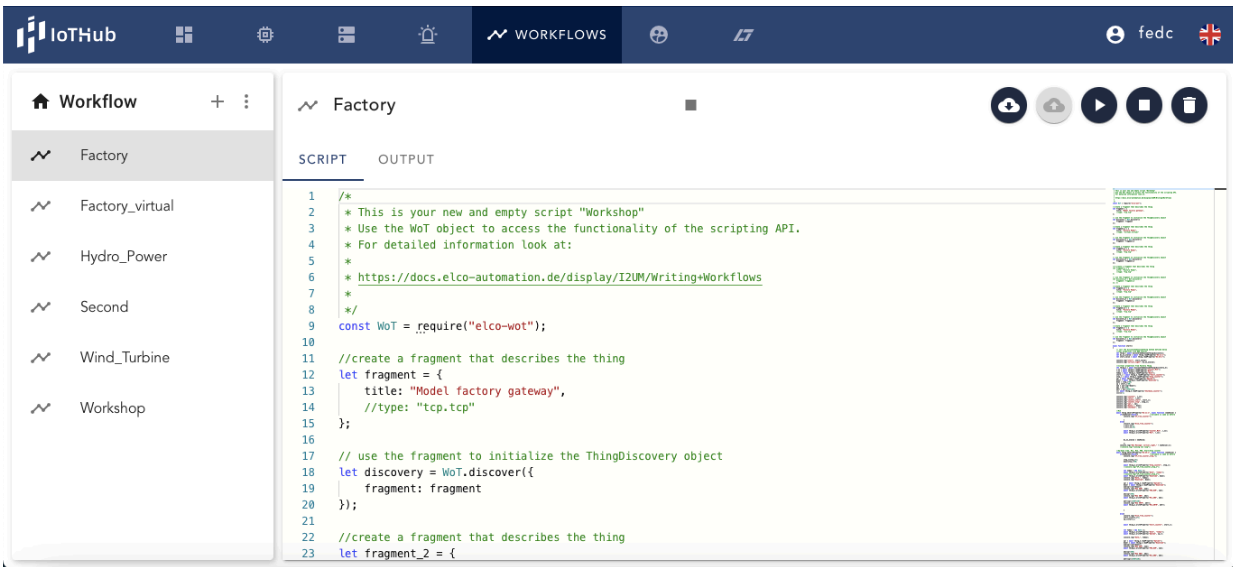

Fig. 4.23: screenshot of the "workflow" window in IoT hub.

Language used is Javascript. So, what we did, was to "explain" to the IoT Hub which parameters to use from the Modbus TCP Thing and how to combine them in order to calculate KPIs. We can identify three main parts in the code. The first part is the definition of the variables used (time counters, workpiece counters, failure counters, first definition of KPIs). Second part is made by three if/else functions, which contain the operations to calculate KPIs. If/else functions were chosen to deal with Boolean data, coming from sensors. So in case we had a true type data, we "trigger" one operation, if we had a false type data, we use another operation. Finally, the last part of the code is occupied by while cycles, which have the scope to let us get continuously updated results, and monitor the factory real-time. The full code is reported below.

```
/*
 * This is your new and empty script "Workshop"
 * Use the WoT object to access the functionality of the scripting API.
 * For detailed information look at:
 *
 * https://docs.elco-automation.de/display/I2UM/Writing+Workflows
 *
 */
const WoT = require("elco-wot");

//create a fragment that describes the thing
let fragment = {
    title: "Model factory gateway",
    //type: "tcp.tcp"
};

// use the fragment to initialize the ThingDiscovery object
let discovery = WoT.discover({
    fragment: fragment
});

//create a fragment that describes the thing
let fragment_2 = {
    title: "Factory Model",
    //type: "virtual.virtual"
};

// use the fragment to initialize the ThingDiscovery object
let discovery_2 = WoT.discover({
    fragment: fragment_2
});

//create a fragment that describes the thing
let fragment_3 = {
    title: "Factory Model",
    //type: "tcp.tcp"
};

// use the fragment to initialize the ThingDiscovery object
let discovery_3 = WoT.discover({
    fragment: fragment_3
});

//create a fragment that describes the thing
let fragment_5 = {
    title: "Factory Model",
    //type: "tcp.tcp"
};

// use the fragment to initialize the ThingDiscovery object
let discovery_5 = WoT.discover({
    fragment: fragment_5
});
```

```
//create a fragment that describes the thing
let fragment_6 = {
    title: "Factory Model",
    //type: "tcp.tcp"
};

// use the fragment to initialize the ThingDiscovery object
let discovery_6 = WoT.discover({
    fragment: fragment_6
});

//create a fragment that describes the thing
let fragment_7 = {
    title: "Factory Model",
    //type: "tcp.tcp"
};

// use the fragment to initialize the ThingDiscovery object
let discovery_7 = WoT.discover({
    fragment: fragment_7
});

async function start()
{
    // call the discoverAndConsumeThing method defined below
    //real properties from B&R Control
    let thing = await discoverAndConsumeThing(discovery);
    let di_11_ovalue = await thing.readProperty("DI:11:1");
    let start_value = await thing.readProperty("DI:25:1");

    console.log("start", start_value);
    console.log("sortier_light", di_11_ovalue);

    //virtual properties from Factory Thing
    let thing_2 = await discoverAndConsumeThing(discovery_2);
    c_11 = await thing_2.readProperty("Counter_M13");
    r_11 = await thing_2.readProperty("M13");
    today = await thing_2.readProperty("Date");
    start_c = await thing_2.readProperty("Start_counter");
    stop_c = await thing_2.readProperty("Stop_counter");
    upr = await thing_2.readProperty("Uptime");
    downr = await thing_2.readProperty("Downtime");
    down = stop_c+80;
    up = start_c-80;
    oam = upr/(upr+downr);
    abu = upr/540;
    abtr = upr/(540+120);
    cs= await thing_2.readProperty("Shutdown_counter");
    css=cs-1;

    console.log("counter", c_11);
    console.log("ratio", r_11);
```

```
console.log("counter_start", start_c);
console.log("counter_stop", stop_c);
console.log("upt_", up);
console.log("downt_", down);
console.log("shutdown", cs);

//M13
await thing.observeProperty("DI:11:1", async function (newValue) {
    if(newValue==false){            //variable is same as before
        console.log("IF_tree_counter");

        }
    else{
        console.log("Else_Tree_counter");
        c_11=c_11+1;
        r_11=c_11/18;

        await thing_2.writeProperty("Counter_M13", c_11);
        await thing_2.writeProperty("M13", r_11);



        di_11_ovalue = newValue;

        };
    console.log("New Message: sortier_light_" + newValue);});
    //console.log("running C13 loop");

//Up-down time, M11, M12, PH8, Start/Stop counter
await thing.observeProperty("DI:25:1", async function (newValue) {
    if(newValue==false){            //variable is same as before
        console.log("IF_tree_counter_stop_");

        stop_c=stop_c+1;
        down=stop_c+80;

        await thing_2.writeProperty("Stop_counter", stop_c);
        //console.log("IF_tree_counter_stop_1");

        let today = new Date ();
        await thing_2.writeProperty("Date", "today");
        //console.log("IF_tree_counter_stop_2");
        await thing_2.writeProperty("Downtime", down);
        console.log("Date_", today);
        console.log("Downtime", down);

        upr = await thing_2.readProperty("Uptime");
        downr = await thing_2.readProperty("Downtime");
        oam=upr/(upr+downr);
        console.log("New OAM", oam);
        await thing_2.writeProperty("PH8_OAM", oam);

        abu=upr/540;
```

```
        console.log("New ABU", abu);
        await thing_2.writeProperty("M11_ABU", abu);

        abtr=upr/(540+120);
        console.log("New ABTR", abtr);
        await thing_2.writeProperty("M12_ABTR", abtr);


        }

    else{
        console.log("Else_Tree_counter");
        start_c=start_c+1;
        up_c=start_c;


        await thing_2.writeProperty("Start_counter", start_c);


        let today = new Date ();
        await thing_2.writeProperty("Date", "today");
        await thing_2.writeProperty("Uptime", up_c);

        console.log("Date_", today);

        upr = await thing_2.readProperty("Uptime");
        downr = await thing_2.readProperty("Downtime");
        oam=upr/(upr+downr);
        console.log("New OAM", oam);
        await thing_2.writeProperty("PH8_OAM", oam);

        abu=upr/540;
        console.log("New ABU", abu);
        await thing_2.writeProperty("M11_ABU", abu);

        abtr=upr/(540+120);
        console.log("New ABTR", abtr);
        await thing_2.writeProperty("M12_ABTR", abtr);

        start_value = newValue;

        };
    console.log("New Message: start_value_" + newValue);});
    //console.log("running C13 loop");




//Shutdowns, E5, E6, O&S17
 await thing.observeProperty("DI:25:1", async function (newValue) {
    if(newValue==start_value){              //variable is same as before
        console.log("IF_tree_counter_stop");
```

```
            }

        else{
            console.log("Else_Tree_counter");

            cs=cs+1;
             await thing_2.writeProperty("Shutdown_counter", cs);
            console.log("Message_shutdown", cs);


            mtbf = upr/cs;
            console.log("Message_mtbf", mtbf);
            await thing_2.writeProperty("E5_MTBF", mtbf);

            art = downr/cs;
            console.log("Message_art", art);
            await thing_2.writeProperty("E6_ART", art);

            os_art = (downr-80)/cs;
            console.log("Message_os_art", os_art);
            await thing_2.writeProperty("O&S_ART", os_art);

            start_value = newValue;

            };
        console.log("New Message: shutdown_counter" + newValue);});
        //console.log("running C13 loop");
}


async function start_2()
{
let thing_3 = await discoverAndConsumeThing(discovery_3);
c_11 = await thing_3.readProperty("Counter_M13");
var i=1;

while (i == 1) {
await thing_3.writeProperty("Counter_M13",c_11);}

}

async function start_6()

{
let thing_5 = await discoverAndConsumeThing(discovery_5);
cs = await thing_5.readProperty("Shutdown_counter");

var x=1;

while (x == 1) {
await thing_5.writeProperty("Shutdown_counter",cs);}
```

```
}

async function start_5()
{
    let thing_8 = await discoverAndConsumeThing(discovery_6);
    start_c = await thing_8.readProperty("Start_counter");
    var h=1;

    while (h == 1) {
        await thing_8.writeProperty("Start_counter", start_c);}

}

async function start_7()
{
    let thing_7 = await discoverAndConsumeThing(discovery_7);
    stop_c = await thing_7.readProperty("Stop_counter");
    var l=1;

    while (l == 1) {
        await thing_7.writeProperty("Stop_counter", stop_c);}

}


start();
start_2();
start_5();
start_6();
start_7();




// this method implements the actual thing discovery and consumption
async function discoverAndConsumeThing(discovery)
{

    // start the discovery process
    discovery.start();

    // while there are new results:
    while (!discovery.done) {

        // initialize an empty thing description
        let thingDescription = {};

        // try to get the next thing description and handle errors if they occur
        try {
            thingDescription = await discovery.next();
        } catch (err) {
            if (err.message === "not found") {
                // No thing matching the fragment could be found
                discovery.stop();
```

```
            console.log("Thing was not found.");
        } else {
            // Some other error occured, print it to the console
            discovery.stop();
            console.log(err.message);
        }
        // return on an error
        return;
    }

    // initialize an empty consumed thing
    let consumedThing = {};

    //consume the thing description and handle errors if they occur
    try {
        consumedThing = await WoT.consume(thingDescription);
    } catch (err) {
        // Some other error occured. Print it to the console and return
        console.log(err);
        return;
    }

    console.log("Discovered thing: " + consumedThing.getThingDescription().title);

    // Here you can observe, read and write the properties of the consumedThing
    return consumedThing;
  }
}
```

Thanks to this code, it's possible to have our results in the Virtual Thing, and so it's possible to read the values of the KPIs. A screenshot of the virtual thing in the IoT Hub is presented below.

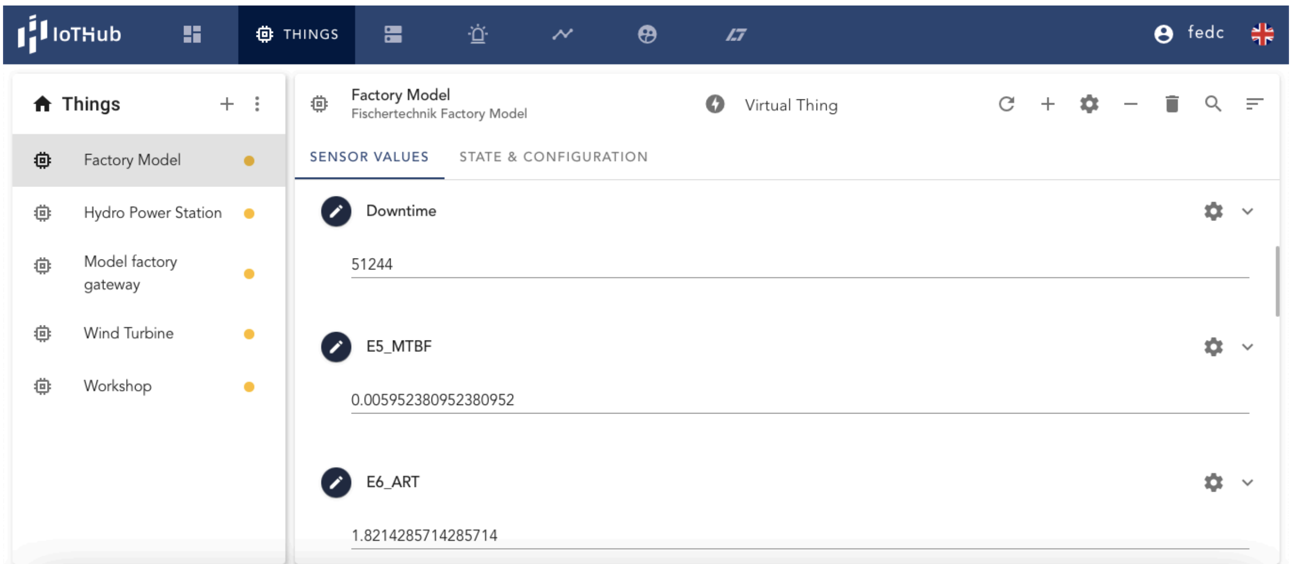

Fig. 4.24: screenshot from the Virtual Thing window of the IoT Hub. From this excerpt, it is possible to read the values on Downtime (in seconds), E5 KPI (Mean Time to Failure) and E6 KPI (Average Recovery Time)

4.3 Development and implementation of a data visualization for production key figures in user specific end device (Smartphone)

The work done so far allows us to handle a bigger amount of information in real time. In our initial situation we could read, on the Automation Studio software interface, data coming from the PLC, and these data just informed us on if the optical sensors mounted in the model factory were triggered or not. After creating the Modbus TCP Thing in the IoT hub, we were able to move the same data to the internet, and to use them in a workflow in order to make new operations and write a Virtual Thing, in which to visualize real time counters and KPIs. New data were readable from IoT hub, which is an interface closer to the developer than to the customer. So our final purpose was to design an interface with easy access via an ordinary device: an app for smartphone. First of all, we targeted as ideal customer a general manager of the factory, or a maintenance manager: somebody inside the company who is responsible for maintenance and efficiency of the plant. After that, I used the function App Designer of the ELCO IoTHub to design an app which could be an interface for the client to read real time data from the factory.

App Designer offers the possibility to communicate information via a text format, which can be used to give titles and literal descriptions of data, but also provides graphic tools such as animated bars. Designer is free to choose colors and spaces in the screen. The unit to set spaces and dimensions is the pixel. After creating the graphic design, we specify the Thing in the Hub from which we want to take data.

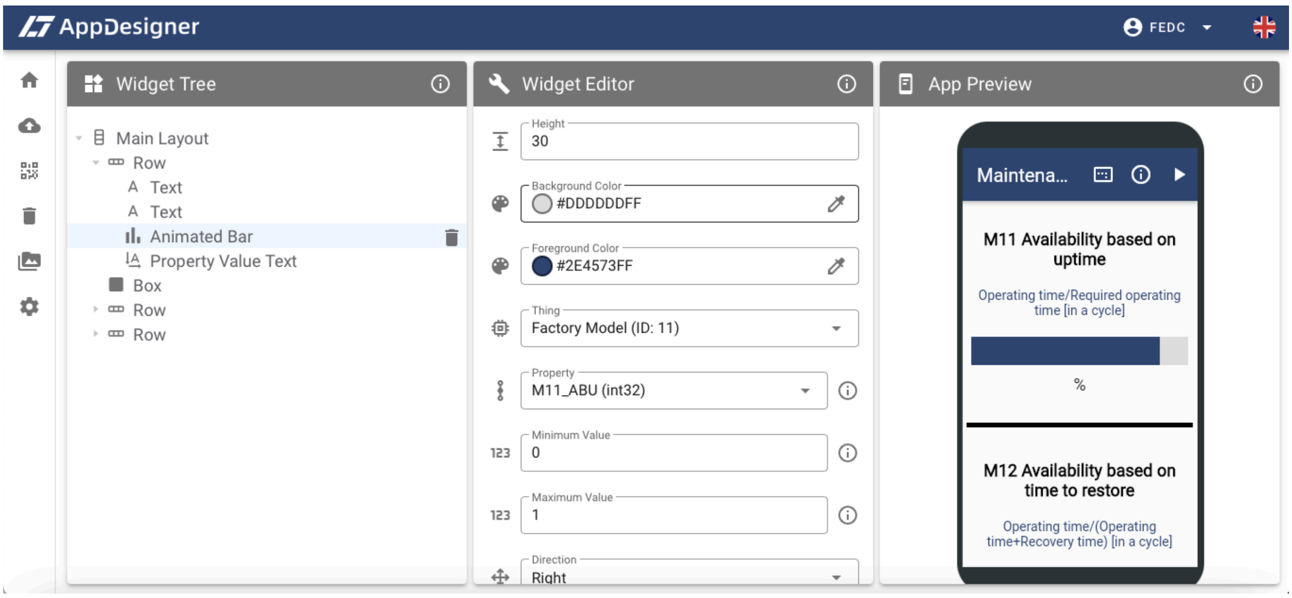

Fig. 4.25: screen of App Designer. In the window, selection of dimensions and spaces of the animated bar, color codes for the bar and the background, and source of data ("Thing" and "Property")

The idea behind design choices is to create a window as clear and meaningful as possible. To do so, I divided KPIs in two windows. In the first window, I presented Maintenance Management KPIs (M11, M12, M13). In the second one KPIs regarding Physical Asset, Maintenance Engineering and Operations&System. Below, it's possible to see a screen of the first window.

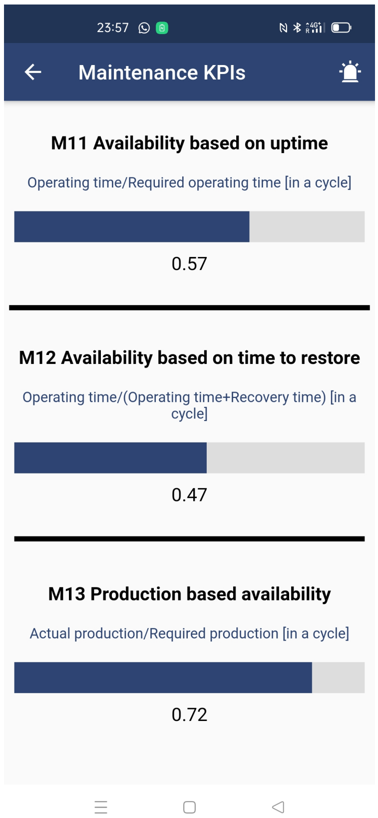

Fig. 4.26: screen of Maintenace KPIs related window of the App Created.

As possible to see from the screen, the main title on a blue background describes the group of KPIs (Maintenance KPIs). Then, each KPI has its own space, containing KPI code from norm (M11, M12, M13), name, a brief literal description, an animated bar which changes in function of real time data, and a numeric value which gives us the exact value of the KPI.

5. CONCLUSIONS AND FURTHER DEVELOPMENTS

Let's make a summarize of the work done. First of all, we had the intent to study a system to help companies in monitoring their manufacturing processes, in order to let managers and other figures to better understand the process and keep it efficient, safe and ecologic. When the plant needed to be updated, we wanted to give our customer a good mean to make retrofitting.

In the first instance, we had an offline system: data coming from optical sensors were collected by the PLC and communicated to the computer, and instructions given by the computer were transmitted to PLC and then actuated by factory motors and controls. The whole system worked with boolean data and provided a basic level of information. We added to the system an edge gateway, which could enable communication between PLC and an IoT Hub, and we configured sensors to Modbus. After we had the availability of real time data in the Hub, we wrote a code inside the hub itself in order to calculate new values: counters and KPIs. In the end, I designed a Smartphone App to let the customer visualize results.

Further steps can be done starting from our work, moving towards two main directions: KPI section and app section. On KPI side, more KPIs from ISO 15341 can be chosen to be monitored. Moreover, economic metrics could be added to the list, evaluating the cost of the energy used by industrial process. Also KPIs for the Research and Development section of a company can be displayed, looking more into detail the origin and the position of failures in the factory. Even personalized KPIs can be displayed and monitored. Moreover, after a first usage of the App, also a customer satisfaction feedback can be looked at, in order to improve communication of useful information. Finally, it can be also evaluated the effectiveness of KPIs themselves. From the app side, it could be also developed an app for tablet, in order to give the customer more detailed and graphic information, and a smartwatch, to let him now essential statements immediately.

As conclusion, we can say that where we started was part of the Digital Age, or Third Industrial Revolution. But where we ended, moving from basic information to more detailed information to evaluation of the process, communicating data online in a transparent way directly to the customer, puts its roots in guidelines and horizon of Industry 4.0.

1. *"Industrial History of European Countries".* European Route of Industrial Heritage. Council of Europe. Retrieved 2 June 2021.
2. *Landes, David S. (1969). The Unbound Prometheus.* Press Syndicate of the University of Cambridge. ISBN 978-0-521-09418-4.
3. *Roe, Joseph Wickham (1916), English and American Tool Builders, New Haven, Connecticut: Yale University Press, LCCN 16011753*. Reprinted by McGraw-Hill, New York and London, 1926 (LCCN 27-24075); and by Lindsay Publications, Inc., Bradley, Illinois, (ISBN 978-0-917914-73-7).
4. *Storia e storeografia*, A. Desideri, Casa Editrice G. D'Anna, Messina – Firenze, 1978, Vol. III, pag. 5
5. *Maxwell, James Clerk (1911). "Faraday, Michael" . In Chisholm, Hugh (ed.). Encyclopædia Britannica.* ***10*** *(11th ed.). Cambridge University Press. p. 173.*
6. Description of lightbulb experiment in *The Times*, 29 December 1881
7. *Storia contemporanea, Vol.I L'Ottocento,*G. Barraclough op. cit. in Tommaso Detti, Giovanni Gozzini, Bruno Mondadori editore 2002, pag. 4
8. *Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution.* H. Kagermann, W.-D. Lukas, W. Wahlster. Memento des Originalsvom 4. März 2013
9. *Design Principles for Industrie 4.0 Scenarios*. M. Hermann, T. Pentek, B. Otto. In: *2016 49th Hawaii International Conference on System Sciences (HICSS).* 1. Januar 2016
10. Norbert Gronau, Marcus Grum, Benedict Bender: *Determining the optimal level of autonomy in cyber-physical production systems*. In: *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*. IEEE, Poitiers, France 2016
11. *Intelligente Assistenzsysteme zur Beherrschung der Systemkomlexität in der Automation*, Jürgen Jasperneite, Oliver Niggemann. In: ATP edition – Automatisierungstechnische Peaxis, 9/2012, Oldenbourg Verlag, München, September 2012
12. *Industry 4.0 – Potentials for Creating Smart Products: Empirical Research Results*, Rainer Schmidt, Michael Möhring, Ralf-Christian Härting, Christopher Reichstein, Pascal Neumaier. In: *Business Information Systems* (*Lecture Notes in Business Information* Processing) Nr. 208. Springer International Publishing, 2015.
13. *Daten sind die Rohstoffe des 21. Jahrhunderts*, Angela Merkel. In: faz.net. 12/03/2016
14. *Retrofitting a Process Plant in an Industry 4.0 Perspective for Improving Safety and Maintenance Performance*. Fabio Di Carlo, Giovanni Mazzuto, Maurizio Bevilacqua and Filippo Emanuele Ciarapica, 12 January 2021
15. ISO 15341:2019 (Essential Performance indicators for maintenance)
16. *Protocol*, Encyclopædia Britannica, retrieved 24 September 2012
17. *Introduzione a Reti Industriali*, prof. Stefano Maggi, Politecnico di Milano, December 2020
18. *That 'Internet of Things' Thing*, Kevin Ashton, in RFID Journal
19. *Supranet,* P. Magrassi, in "Dizionario dell'economia digitale", a cura di V. Di Bari, Sole 24Ore Pirola, Milano, 2002
20. *Internet of Things: gli oggetti intelligenti prima di ogni „cosa"*, Angela Tumino, 24/04/2019
21. *The net shapes up to get physical*, Sean Dodson,"The Guardian", 16/10/2008
22. *Mercato IoT: quanto vale l'Internet of Things in Italia*, Angela Tumino, 11/04/2019
23. *Moralizing technology: understanding and designing the morality of things*, Peter-Paul Verbeek, The university of Chicago press
24. *At Newark Airport, The Lights Are On, and They're Watching You*, Diane Cardwell, *The New York Times,* 17/02/2014
25. "Tim O'Reilly Explains the Internet of Things", Quentin Hardy, The New York Times Bits. The New York Times. 04/02/2015

26. *Say goodbye to Privacy*, Geoff Webb. WIRED 05/02/2015
27. *Privacy of Big Data in the Internet of Things Era*, Charit Perera, Rajiv Ranjan, Lizhe Wang, Samee Khan, Albert Zomaya. *IEEE IT Professional Magazine*. PrePrint (Internet of Anything). 2015
28. *We asked Executives about the Internet of Things and their answers reveal that security remains a huge concern*. Business Insider, January 2015
29. *These devices may be spying on you (even in your home)*. Joseph Steinberg, Forbes, 27/01/2014
30. *Hackers remotely kill a Jeep on the Highway-with me in it*, Andy Greenberg, Wired, 21/07/2015
31. *The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology*. Special Publication 800-145. Peter Mell, Timothy Grance. National Institute for standards and Technology, US Department of Commerce, September 2011
32. *Processi di outsourcing informatico e cloud computing: la gestione dei dati personali ed aziendali*, in „Dir. informaz. Informatica, 2010, pag. 673-696