Universidade de Aveiro
Ano 2013

Departamento de Eletrónica, Telecomunicações e Informática

Tiago Francisco da Silva Martinho

Aplicação Móvel de Monitorização e Controlo para Automação Industrial

Mobile Application for Monitoring and Control in Industrial Automation

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
Ano 2013

Tiago Francisco da Silva Martinho

Aplicação Móvel de Monitorização e Controlo para Automação Industrial

Mobile Application for Monitoring and Control in Industrial Automation

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Hélder Zagalo da Universidade de Aveiro.

o júri / the jury

presidente / president

Professor Doutor Joaquim Manuel Henriques de Sousa Pinto
Professor Auxiliar da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

Professor Doutor Joaquim José de Castro Ferreira
Professor Adjunto da Universidade de Aveiro (Arguente Principal)

Professor Doutor Hélder Troca Zagalo
Professor Auxiliar da Universidade de Aveiro (Orientador)

agradecimentos / acknowledgements

Agradeço ao Professor Doutor Hélder Troca Zagalo pela orientação prestada durante a realização da dissertação, e pelo facto de desde o início do ano ter sempre defendido os meus interesses como aluno da Universidade de Aveiro.

Um grande obrigado a toda a equipa da Bresimar, por me terem recebido como um membro da equipa, e me terem dado todas as condições para o sucesso do projeto.

Para a minha família e amigos, que sempre me apoiaram no decorrer do projeto, o meu sincero obrigado.

palavras-chave Automação Industrial, PLC, HMI, Tablet PC, Aplicação Móvel

resumo

Atualmente em automação industrial, a maioria das tarefas estão automatizadas, e a intervenção humana é cada vez mais reduzida. Apesar disso as máquinas usadas continuam a precisar de manutenção. Esta é uma parte tão importante da área, que todas as indústrias têm pessoal especializado apenas para manter os sistemas de automação.

A razão é que os sistemas utilizados hoje em dia nas fábricas são cada vez mais específicos e exigentes em termos tecnológicos. Isso implica uma interação constante entre os fabricantes das máquinas e a equipa de manutenção da fábrica.

O objetivo deste estudo foi desenvolver um protótipo de prova de conceito, para servir como ponto de partida para um produto de mercado na área de automação industrial. O produto visa estreitar a comunicação entre os fabricantes e o pessoal responsável na indústria.

O foco do trabalho é uma aplicação móvel para um Tablet PC industrial, a fim de fornecer novas ferramentas para o operador industrial. Estas novas ferramentas incluem funcionalidades de suporte técnico, como a reprogramação remota de dispositivos industriais ou através de chamada de vídeo. Incluem ainda a possibilidade de aceder localmente a dispositivos industriais e funcionalidades oferecidas por dispositivos comerciais "inteligentes", como o uso da câmara.

Os principais resultados da tese foram, desenvolver um protótipo testado em ambiente industrial, bem ao descobrir que nenhum fabricante industrial fornece uma solução chave na mão com os requisitos estabelecidos.

keywords Industrial Automation, PLC, HMI, Tablet PC, Mobile Application

abstract

In today's modern industry, the majority of the tasks are getting more automated, and the human intervention reduced. Despite this fact the machines used still need maintenance. This is such an important part of the area, that all industries have specialized personnel just to maintain the automation systems.
The reason is that the systems used on nowadays factories are becoming more specific and demanding in technological terms. This translates in a constant interaction between the machine manufacturers and the factory maintenance team.
The purpose of this study was to develop a proof of concept prototype, projected to serve as a starting point for a market product in the field of industrial automation. The product aims to bridge the communication gap between manufacturers and industrial plant personnel.
The focus of the thesis is in a mobile application for an industrial Tablet PC, in order to provide new tools to the industrial operator. These new tools include technical support functionalities, like remote reprogramming of industrial devices or via video call. They additionally add the possibility to locally access to industrial devices and functionalities seen in commercial "smart" devices, like the use of the camera.
The main results of the thesis were to develop a working prototype tested in a real industrial case scenario, as well to find out that no industrial manufacturer provides a turn-key solution with the stated requirements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ADS – Automation Device Specification
AP – Access Point
API – Application Programming Interface
ARM – Advanced RISC Machine
HMI – Human Machine Interface
I/O – Input and Output
IDE – Integrated Development Environment
IP address – Internet Protocol Address
IPC – Industrial Personal computer
LAN –Local Area Network
Microsoft Windows CE – Microsoft Windows Embedded Compact
MMC – Multi Media Card
NFC – Near Field Communication
OS – Operating System
PC – Personal computer
PDF – Portable Document Format
PLC – Programmable Logic Controller
QR Code – Quick Response Code
RF – Radio Frequency
SCADA – Supervisory Control and Data Acquisition
SIL – Safety Integrity Level
SOA – Service Oriented Architecture
SSID – Service Set Identifier
TCP – Transmission Control Protocol
UC – User Control
UI – User Interface
USB – Universal Serial Bus
VNC – Virtual Network Computing
VPN – Virtual Private Networking
WLAN – Wireless Local Area Network
WPF – Windows Presentation Foundation
XAML – Extensible Application Mark-up Language
XML – Extensible Mark-up Language

Chapter 1

INTRODUCTION

This chapter contains the presentation of the thesis. An analysis of the background and its area of study are going to be made. Then the problems and objectives of the project are stated. Next, it will be presented the strategy to achieve the final objectives, in order to solve the discovered problems. Finally, the organization of the document is going to be outlined for a better understanding of the reading parts.

1.1 BACKGROUND OF THE PROJECT

Industrial automation is an area projected to develop systems to automate tasks, applied both in manufacturing and processing industries. A team of Engineers design these industrial plants, where automation systems are working 24/7, in order to create value for a company.

The creation of value can be done in different ways. One example is an assembly line, where different components are integrated to create a final product. Another example is in a chemical process, where natural resources are transformed in products.

The devices used in the automation of these tasks are the robots, the programmable logic controllers (PLC) and the human-machine interfaces (HMI). These devices work together with a variety of sensors, actuators and motors in a way that, in the end of a production line, the goal is achieved and a final product is assembled or transformed.

The robots are the machines responsible for the process in a physical way. An example is a painting robot for car doors.

The PLC is a specific purpose computer responsible for the control of the robots. This control is done by reading inputs and varying outputs regarding the process. An example is a computer with a control algorithm for the painting robot.

The HMI allows the interaction between humans and these machines. It has two main functions: first to monitor the process, for example, seeing how many car doors were painted. The second function is to control the process, for example, setting a new colour to paint.

It is typical that, the more the process is automated, and the human intervention is reduced, the greater is the efficiency and less cost to produce.

In today’s modern industry, despite the fact that the majority of the tasks are automated, the human intervention is still the backbone of a working factory. Automation systems are customized for a specific task and may require modifications along the time. The human intervention arises to validate these changes, with the monitor and control of the automation system.

Another case where human intervention is needed is in systems and equipment maintenance. This is such a crucial part of the industry world that all industries have specialized personnel only for maintenance of the systems.

The reason is that automation systems used on nowadays factories are becoming more specialized in production tasks, and more demanding in technological terms. This translates on a constant interaction between the machine manufacturers and the plant maintenance team.

It is then possible to identify a need for constant connection between engineers and clients, which could be satisfied by inserting a new system in the industrial field. This device acts as a communication tool between both parts, as well as diagnosis tool to the maintenance personnel of a given factory.

The system is intended to be a mobile device, providing a new way to provide technical support. There is not the need for the engineer to go on site as it can offer the possibility to detect malfunctions of the machines via video call, or even the functionality to reprogram the machines via remote access.

The mobile device can furthermore give the option for the operator to monitor and control the industrial process. There is a gain in time since the operator can quickly access information of interest, without the need of additional wiring or physical presence near the system. It also brings a gain in sharing the resources. In their daily rounds and quality inspections, more than one industrial operator, can monitor the same process independently.

Additionally, a mobile device can bring common functions of today's "smart" mobile systems. This can provide the ability for the operator to take a photo, add a note and send this information via email on the industrial site.

An example of this mobile device is the Tablet PC, which becomes a potential choice for this kind of product, because of its non-intrusive nature and its great market penetration, surpassing the shipments of desktop PC and Notebooks in the end of 2012 [1].

A Tablet PC is each day a more familiar device to the common user, "Tablets have set themselves apart as a new product category, leading to radically different interaction paradigms and use cases" [2].

The project will then focus in the insertion of a Tablet PC in the industrial automation area.

1.2 Motivation

The project is born from a partnership between the Department of Electronics, Telecommunications and Informatics (DETI) of the Aveiro University, and the Portuguese company, Bresimar Automação SA.

The aim of the partnership is to develop a proof of concept prototype, projected to serve as a starting point for a product to the industrial automation market.

The company Bresimar sees the integration of a mobile device in the industrial automation field as a competitive advantage.

1.3 Objectives

At present, in the industrial automation field, a set of technological tools could solve several problems, for example, remove the need for the engineer to go on site to reprogram machines, or easily replace a malfunctioning part.

Another discovered problem is that, in control quality checks, the HMI associated with the industrial process needs to be shared by the industry personnel. This interferes with the work of the industrial operator as he will not be able to do his work during the control.

One more problem revealed is that the today's industrial operator does not have access to features present in the commercial "smart" devices. There is not the option to send an email or take a photo while at the plant floor, without exposing a device to harmful conditions.

The objective of the project is to develop a proof of concept prototype with the aim of solving the stated problems. The prototype is a software application for a mobile device, to give a basis so new tools can be inserted in the industrial automation field. The new tools can go from providing technical support on site, a way to control and monitor data regarding industrial processes, and bring functions of commercial “smart” mobile devices.

The project is industry oriented, so a competitor analysis gains even more value, to assure that the investment is well applied and with reduced risks. This analysis must be made with an in-depth online search and active field comparison.

Following the state-of-the-art study, it is presented the hardware and software architecture. This phase consists on choosing the right specs and needs for the project so that the transition to the third and last part of the project is the most smooth possible: the development and validation phases.

1.4 Document Structure

In this thesis, six chapters follow the present one:

- Chapter 2 contains the state-of-the-art study, where are presented the advancements made so far, regarding the theme of the project;
- Chapter 3 contains the system requirements and system architecture, where are explained the problem tackled and the proposed solution;
- Chapter 4 contains the system development, where are presented the development choices and the implementation of the system;
- Chapter 5 contains the system validation, where are presented the tests made to the developed system;
- Chapter 6 contains the case study, where is presented a real case scenario where the developed system was inserted and tested;
- Chapter 7 contains the conclusion and future work, where is presented a summary of the project and the future work to be developed;

Chapter 2

STATE OF THE ART

This chapter contains a set of theoretical and technological solutions, related with the theme of this project. Registered patents will be presented, together with hardware and/or software solutions, provided by different companies in the industrial automation market.

At the end, a reflection is made in order to understand how the state-of-the-art study contributed to the project.

2.1 REGISTERED PATENTS

This section presents a resume of the registered patents whose ideas are related with the developed project.

2.1.1 United States Patent 7,657,492

The US patent 7,657,492 - Mobile control and monitoring system [3], granted to Siemens, relates to a HMI system with at least a mobile control and monitoring device, used for automation components in a technical unit.

Typically, HMI is a fixed component in an automation system, connected via cable. This has the motive to ensure a safe distance of operation. Safety standards, in some cases, demand direct proximity with the automation system.

The patent claims a mobile HMI device which still guarantees this safety distance. The safety is provided by measuring the distance between the operator and the technical unit. This measurement can guarantee active and warning areas to interact with the machine.

The concept enables that a mobile device substitutes the actual fixed HMI panels, with the guarantee of not compromising the security distance.

In Figure 1, it is possible to see a scheme of the invention.

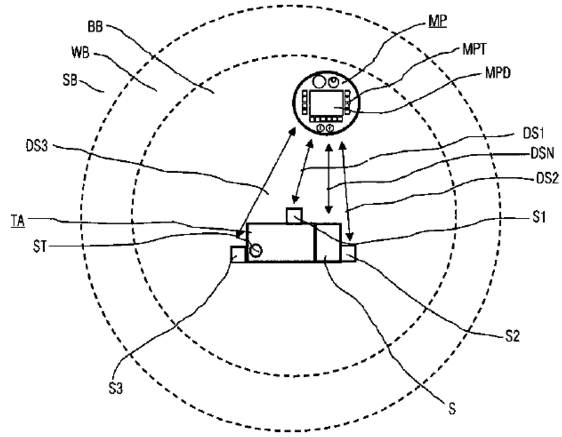

Figure 1 - US patent 7,657,492 [3]

There is a technical unit (TA) that may, for example, be part of a manufacturing or process engineering system. It is possible to see also automation components (S), which control the technical unit. In Figure 1, there is also one mobile control and monitoring device (MP). The function of MP is to control and monitor the automation system. The MP device exchanges data with the automation system in a contactless way via a radio link (DSN). The device MP can include a display (MPD), and/or a keyboard (MPT). It is possible to see also in Figure 1 the active operating area (BB), the warning area (WB), and the safety area (BB), stated by the patent.

2.1.2 United States Patent 6,167,464

The US patent 6,167,464 - Mobile human/machine interface for use with industrial control systems for controlling the operation of process executed on spatially separate machines [4], granted to Rockwell Technologies, relates to a mobile HMI for monitoring the operation of a spatially distributed control system in industrial automation.

In industrial control systems, the control program is modified as the control process evolves. Therefore is needed a way to monitor and troubleshoot the process, so it is possible to detect, for example, errors in the modifications. Installing one HMI associated with each piece of control equipment is the ideal case. However, this is an inefficient use of resources since the HMI is not needed on a consistent basis.

The invention provides a solution for this need as it specifies a total industrial control system. The objective of the system is to control the operation of a process. The process can be executed on a plurality of separate machines. This industrial control system is constituted by a data network and a plurality of interface circuits, which provide I/O signals from the machines.

The system still includes a portable HMI, with the objective of occasional monitoring and troubleshooting. The HMI obtains control system data and program information from a central processor, usually an industrial personal computer (IPC).

This portable HMI has the feature to know its location, with respect to the machines in the factory. As the user moves, the location signal will provide different values. According to the location signal, the data received from the machines will also be different. The location signal comes from the reading of a barcode, radio frequency (RF) detectors, or even manual insertion by the user through a keypad.

The user needs to authenticate himself, and this authentication will grant different levels of access.

In Figure 2, it is possible to see a simplified representation of the industrial control system (10) stated by the patent.

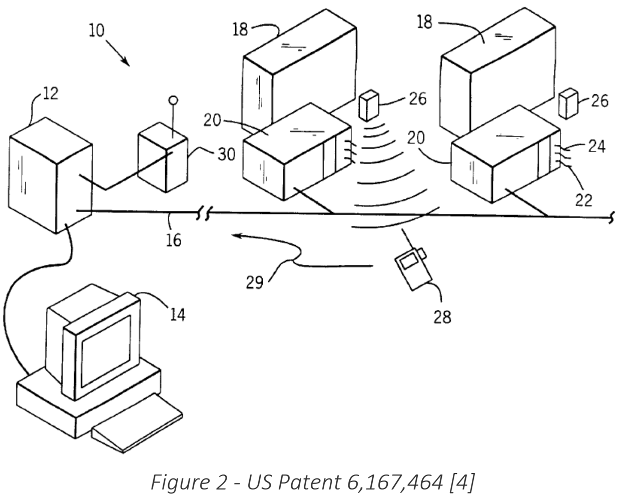

Figure 2 - US Patent 6,167,464 [4]

There is a central processor (12) and a programming terminal (14). They communicate via a data network (16), with one or more remote I/O modules (20). These I/O modules are associated with the machines of the factory (18). The modules provide electrical output signals (22) to control the machines, and receive inputs (24) from them. The machines have identifiable tags (26), which are read by the portable HMI (28). This portable HMI communicates with the central processor with a transceiver (30), via an RF link (29).

2.1.3 United States Patent 2004/0201602

The US patent 2004/0201602 - Tablet computer system for industrial process design, supervisory control, and data management [5], granted to Invensys Systems, relates to the use of a Tablet PC for data management and industrial/manufacturing process control.

In the manufacturing/industrial processes, different computer systems are used. The objectives are: to design, configure, control and manage information, related to that processes.

These computer systems are placed in a large number of locations, within the plant, as well as off site, often these computer systems are specialized machines.

One class of computer system is located within control rooms. These do not need to have protection against moisture, dust, extremely swings of temperature, etc. Their goal is to obtain information from the industrial/manufacturing control system, so they perform higher level enterprise and production management tasks.

Another class of computer system, utilized particularly to configure and manage processes, is located on the plant floor. They enable the operator to observe the processes, and at the same time check and control the state of the system. The problem is that associating one computer system with each process requires a large number of these systems. This increases the cost of the overall system since these systems require a special casing, to protect against moisture, dust, extremely swings of temperature, etc.

The patent describes the use of a Tablet PC for industrial process design, control and data management. It claims that the use of the Tablet PC provides a mobile, powerful and flexible platform with a large set of applications. This allows the computer system to be moved from station to station inside a plant.

This Tablet PC can contain a set of services and application software to help the operator. The features provided come in a way that ease the performing of various tasks, such as, design, supervisory and/or data management, as well by integrating a HMI into the tablet.

The Tablet PC provides, as well, a set of functional characteristics. These characteristics include a stylus interface, handwriting recognition, screen capture, a camera, a microphone, a barcode/RF tag reader, email and wireless networking. The wireless communication enables to connect into a process area network, in order to have access to data from different machines.

In Figure 3, it is possible to see an example of a network with the present invention incorporated.

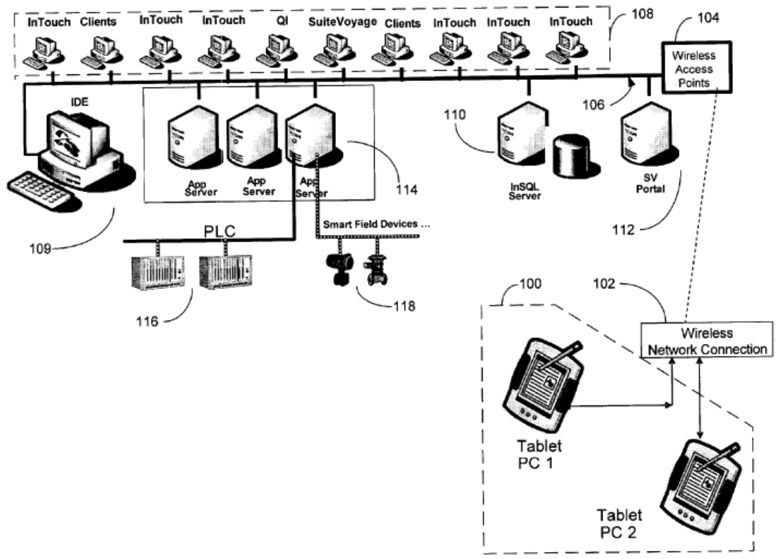

Figure 3 - US patent 2004/0201602 [5]

The Tablet PC (100), linked via a wireless connection (102) to an access point (104), on the process area network (106).

This process area network contains a group of user stations (108). The user stations execute a variety of well know applications, such as, process visualization, manufacturing information, statistical process control, etc.

An integrated development environment (IDE) station (109) supports a suite of utilities for designing and configuring a controlled industrial process. A set of servers (110, 112, 114), provide access to database support, specific applications and industrial process control applications respectively.

It is also possible to see from Figure 3 that a set of programmable logic controllers (116), receive the instructions for a set of industrial process, via this application server connected to the process area network. This same server communicates with a variety of smart field devices (118).

2.2 Hardware Solutions

This project has a strong component of software. Nevertheless, the hardware is also a very essential part since the strategy is to integrate both.

In this section, it will be indicated the currently available market solutions, including hardware and software, which converge to the idea proposed in this project.

In the following sub-sections will be present the main characteristics of each found solution. The title of the sub-section is the name of the company, along with the name of the principal solution that it offers.

2.2.1 Sintesi: Shd3

The solution Shd3 [6], from Sintesi, is a hardware and software platform, which can monitor and manage different types of machines and robots. The solution comes with the proprietary software Sintesi HMI, but can also host third party HMI software, running on Windows CE [7].

Additionally, it has a 5 inch screen with touch interface, and physical buttons, such as an emergency button. It can operate autonomously up to 6-8 hours, and communicate up to 50 meters wirelessly, from its Controller Station.

The solution comes with a USB interface for wireless communication with the PLC, with support for the most popular communication protocols in industrial automation, such as PROFIBUS-DP [8], EtherCAT [9], Modbus TCP/IP [10] and CANopen [11].

Lastly it has certified safety standards: Safety Integrity Level 2 according to the EN 62061 [12], and ISO 13849-1 [13]. These safety standards give specifications for functional and safety integrity requirements, in order to meet levels of safety and functionality tests, for example, the probability of dangerous failures per hour.

All these characteristics are presented in a device with industrial design and 750 grams of weight.

It is possible to see a picture of the device in Figure 4.

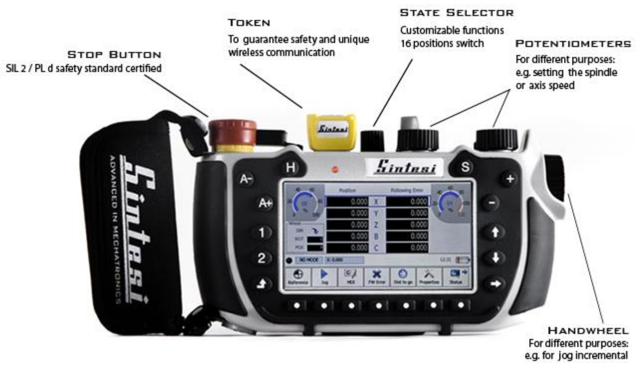

Figure 4 - Sintesi: Shd3 [14]

2.2.2 Siemens: SIMATIC Mobile Panel

The solution SIMATIC Mobile Panel [15], from Siemens, is a hardware and software platform, which offers a mobile HMI device for operating distributed or complex machines in industrial plants.

The application range can be expanded via the SIMATIC WinCC. This is a proprietary software from Siemens that enables the creation of applications to visualize industrial processes [16].

Referring specifically to the model SIMATIC Mobile Panel, 277 IWLAN Version 2 [17], which has a 7.5 inch screen with touch interface and physical buttons, such as an emergency button. It can operate autonomous up to 4 hours and has the function to hot swap the battery.

The solution comes with a USB interface, a MMC card reader to expand the memory, and support for the communication protocol PROFINET [18] via wireless local area network (WLAN).

Lastly it has a degree of protection IP 65, which guarantees a protection against dust ingress and low pressure water jets from any direction [19].

All these characteristics are presented in a device with an industrial design and 1700 grams of weight.

It is possible to see a picture of the device in Figure 5.

Figure 5 - Siemens: SIMATIC Mobile Panel [20]

2.2.3 Invensys: Wonderware Tablet Computers Series D

The solution Wonderware Tablet Computers Series D [21], from Invensys, is a HMI with extra functions, which can go from mobility to handwriting on the HMI.

The solution has a 10.4 inch screen with touch interface, which can operate autonomously up to 2-6 hours. Additionally, it offers one USB interface, one RJ45 based Ethernet port, and one RS232 port.

The Tablet PC supports the WLAN protocol IEEE 802.11n to communicate with an access point, which is connected to a process area network containing the PLC.

Lastly, it has a degree of protection IP 67, which guarantees a protection against dust ingress and short periods of immersion in water [19].

All these characteristics are presented in a device with an industrial design and 2200 grams of weight.

It is possible to see a picture of the device in Figure 6.

Figure 6 - Invensys: Wonderware Tablet Computers Series D [21]

2.2.4 Beijer Electronics: TREQ-VMx

The solution TREQ-VMx [22], from Beijer Electronics, comes with the operating system Windows Embedded CE 6.0, R3 Core or Professional, which allows to develop customized applications to monitor and control industrial processes.

The solution can operate autonomously up to 1 hour and has a 7 inch screen with touch interface. In terms of connectivity, has WLAN IEEE 802.11b/g/n support, Ethernet interface, GSM/GPRS connectivity and a USB interface.

Additionally, it offers support for the communication protocol CAN bus [23] but does not have a certified safety standard.

It is possible to see a picture of the device in Figure 7.

Figure 7 - Beijer Electronics: TREQ-VMx [24]

2.2.5 Motorola Solutions: Enterprise Tablet ET1

The solution Enterprise Tablet ET1 [25], from Motorola Solutions, can be used as a mobile HMI and comes with the operating system Android 2.3.4.

The solution has a 7 inch screen with touch interface and physical buttons. It does not have information regarding the battery life, but it has the function to hot swap the battery.

In terms of connectivity, it has WLAN IEEE 802.11b/g/n support, Bluetooth connectivity, USB interface and a MMC card reader to expand the memory. Additionally, it contains also two cameras, one in front and another in the rear.

The communication and authentication, with the PLCs are made using the wireless LAN protocol, IEEE 802.11.

Lastly, it has a degree of protection IP 54, that offers limited protection against dust ingress, and splash water, from any direction [19].

All these characteristics are presented in a device with an industrial design and 630 grams of weight.

It is possible to see a picture of the device in Figure 8.

Figure 8 - Motorola Solutions: Enterprise Tablet ET1 [26]

2.2.6 JAS Automação: Android IHM Open

The solution Android IHM Open [27], from JAS Automação, is made with a commercial Tablet Coby Kyros 1024, where is included one Android application, IHM Open. This mobile application supports Modbus [10] communication protocol, in order to read and write in PLCs of the type Phoenix 2985330 ILC 150 ETH.

The Kyros Tablet features a 10.1 inch screen with touch interface and is powered by the Android 2.2 operating system. Additionally has a built-in camera and allows to connect to wireless networks through WLAN protocol IEEE 802.11.

All these characteristics are presented in a device that does not have an industrial design.

It is possible to see a picture of the device in Figure 9.

Figure 9 - JAS Automação: Android IHM Open [28]

2.3 Software Solutions

In this section, it will be presented the software market solutions that present similarities to the idea proposed. These are software solutions designed to use on commercial mobile devices.

First it is showed the offers on native applications, followed by the multi-framework applications.

2.3.1 Native Applications

Native applications are target to a specific hardware architecture, which run a specific operating system.

In mobile operating systems, the three biggest players are Android, iOS, Windows Phone, by order of quantity of developers in each platform [2].

The study was made by taking in account what each platform has to offer. After the study, it was verified the existence of three business models, in the referred platforms.

On the first model, the application is paid but offers the use of every functionality. The user only needs to connect the PLC to a wireless router or to a Virtual Private Networking (VPN) connection [29].

In the second approach, the application is free, or has a symbolic price, but to work it needs the hardware of the same company.

On the third model, the application is free, but all configurations or improvements are made by the user.

It will be referred an example of each one of these approaches. The others applications studied are in Appendix A - Native Applications, along with the main characteristics and requirements of each one.

mySCADA technologies: mySCADA Mobile

The mySCADA Mobile [30] application for iOS communicates directly with the PLC. It gives the ability to monitor and control data from the industrial process.

There is the possibility to configure different levels of access to different operators, as well the use of alarms in real time.

It supports devices that communicate using the following communication protocols: Ethernet/IP [31], Siemens S7 [32] and Modbus [10].

The solution also includes a project editor, where it is possible to design HMI displays that represent the process being monitored. It is possible to see a screenshot of the application in Figure 10.

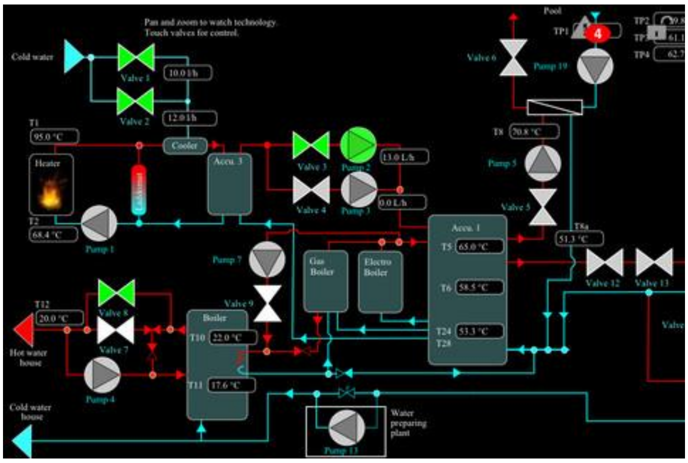

Figure 10 - mySCADA technologies: mySCADA Mobile [33]

The price of the application varies according to the number of elements and screens that are pretended to use. The number of recorded machines can be one, ten, fifty and unlimited, on the applications mySCADA One, mySCADA Lite, mySCADA 50 and mySCADA, respectively.

Opto 22: Opto iPac

The Opto iPac [34] application allows to visualize the values, digital or analogue, and the possibility to edit them.

It also allows to recognize automatically all the devices in the network, check their state, firmware version or strategy of the device.

The application is available both in iOS and Android, but it is mandatory to have Opto 22 equipment to work with it. It is possible to see a screenshot of the application in Figure 11.

Figure 11 - Opto 22: Opto iPac [35]

Hubble: Modbus Monitor

The Modbus Monitor [36] application, available for Android OS, is a Modbus Multi-Threaded client, which does the polling of any Modbus TCP/IP [10] standard equipment.

It is possible to create a list of different devices, and associate standard values with limits, so it is easier to monitor the different variables of the PLC.

The application is free, but it does not allow the editing of the PLC variables, neither has a graphical HMI for interaction with the PLC. It is possible to see a screenshot of the application in Figure 12.

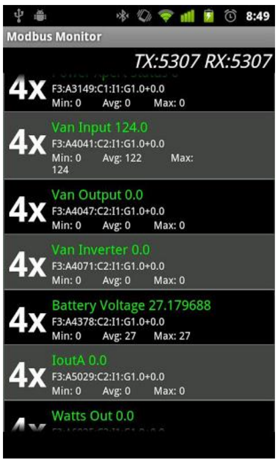

Figure 12 - Hubble: Modbus Monitor [37]

2.3.2 Multi-Framework Applications

Multi-Framework applications are developed with the objective of running independently of the hardware architecture, and operating system, of the device. The adopted philosophy is "write once, run anywhere".

Usually, they take the form of Web applications, based on a model server-client, and they are independent of the platform. They have some minimum requirements, such as, to have a browser with Java technology.

Was only found one business model, in this case the software makes all the management inside the industrial plant. Alternatively, there is an Open Source option, which gives a set of tools to develop a system solution based on Web.

Inductive Automation: Ignition HMI/SCADA

The Ignition HMI/SCADA Software [38] solution, delivered by Inductive Automation, is a Web-Based solution. It creates software abstraction levels using different module layers as shown in Figure 13. It connects to the PLC via OPC-UA [39], or classic OPC [40].

Figure 13 - Ignition Mobile: Inductive Automation [41]

Since the solution is centralized on a server, it brings new functions, like database access, for saving useful data or the program that is controlling the industrial process.

With this server, it also offers an improvement in security and takes the processing of the data to his side. So it is possible to have a solution which provides real-time control and monitoring, where every device can access to it.

This software has also a graphical editor, where it is possible to create a vision of the industrial process. With this software is given unlimited access, both to the variables of the PLC and to the machines connected into the system.

MB-HMI

The MB-HMI [42] solution is based in Web Standard technologies, HTML and JavaScript. It allows to create HMI systems, which are accessible through a Web Browser, and work on every device that has one.

The communication with the PLC is done using the Modbus TCP/IP [10] communication protocol. Then the HMIServer runs on a PC to interact with it, creating the HMI with graphical illustration of the industrial process.

To develop solutions in this platform is necessary to work with the referred Web Standard technologies. Is also needed to use the protocol Cascadas [43], an open protocol based on JSON, to communicate between the Web Browser and the server.

2.4 Analysis

It will be made an analysis and some considerations about the presented information: first, the differences about the patents and the concept of this project; second, an analysis about the hardware solutions presented by the manufacturers in the industrial automation field; and finally, an examination to the software solutions.

2.4.1 Registered Patent Analysis

Regarding US 7,657,492 patent, Siemens focused on replacing the fixed HMI panels, still ensuring a safe distance of operation. The focus of this project is not just to make of the Tablet PC a portable HMI. The idea of this project can also coexist with the fixed HMIs in the industrial plant. The goal of the project is to create a basis where it is possible to add new functions, for example, providing technical support with a video call or remote reprogramming.

In US 6,167,464 patent, Rockwell Technologies specifies a mobile HMI for occasional monitoring and troubleshooting situations. In this project, the emphasis goes to the functions offered by the mobile device. The concept of this project creates a basis where is then possible to reprogram remotely the PLC, or to incorporate functions seen in commercial Tablets, like VoIP [44], multimedia and email.

US 2004/0201602 patent has connections with the idea of this project. The fact that it provides some functions as the ones thought for this project, the access to the camera or the email. It was observed that one of the core concepts of this project was not explored in the patent, which is the possibility of providing technical support in industrial field. In this project, there is a focus on having an internet connection in the Tablet PC, which allows to do a video call, or reprogram the PLC remotely.

2.4.2 Hardware Solutions Analysis

The Shd3, from the Italian company Sintesi, has an emergency button, and a format with the adequate resistance. It possesses the necessary characteristics to be integrated in a general automation system since it has support for the most popular communication protocols. A disadvantage is that the device can only communicate with the automation systems, and so it does not have the option to connect to the internet, for example, to make a remote access to the device. It also does not have a camera interface, to get technical support from the device via a video call or take a picture to the malfunctioning machine.

The SIMATIC Mobile Panel, from the German company Siemens, is a mobile version of the actual fixed panels that the company sells. It has the disadvantage of being a solution that does not add any extra functionality, besides the mobility, regarding the fixed HMI panels.

The TREQ-VMx, from the Swedish company Beijer Electronics, is a solution presented as a mobile version of their HMIs. The first drawback is its battery life of one hour, since in an industrial environment each shift of the operator can easily surpass this duration. The solution also does not present any degree of protection, which is needed in industrial applications. It is a solution that does not present a camera interface, to get technical support from the device via a video call.

Regarding the Enterprise Tablet ET1, from the German company Motorola Solution, and the Android IHM Open, from the Brazilian company JAS Automação. They are solutions that essentially have the focus on control and monitoring industrial processes, but instead of doing that with a traditional HMI, they do it with a Tablet.

In this project a more general solution is wanted, so it is possible to take advantage of the mobile hardware chosen. The project should create a modular structure able to grown, for example, in communication protocols support. Additionally, provide remote technical support functions, by a video call, or remote reprogramming of the machines.

2.4.3 Software Solutions Analysis

Regarding the native applications, they are solutions that imply the user to buy the Tablet, install the application and set the network. Besides, they do not focus on using the other functions inherent of the Tablets, such as, communication systems and multimedia.

About the multi-framework applications, it was seen that the solution by Inductive Automation has an architecture that can be accessed by all the devices with a browser and java technology. However it should be taken in mind that this solution is a whole management system for the industrial plant, and not a single device, like the one proposed on this project. On the other side, was seen that the tools given by MB-HMI do not seem to have the flexibility, and robustness, to create a proper solution from the industrial point of view.

2.5 CONCLUSION

Concluding no one focus on creating a modular platform where the hardware and software are brought together, in order to add new tools in the industrial automation area.

The functionality to reprogram the machines remotely, only by letting the Tablet PC near the machine, is seen as innovative in this field, comparing to the solutions explored.

Lastly this project aims to create a basis so new functions can be added to industrial automation. These functions will help improve the efficiency of the operator, as well diminishes the errors committed. They include the offering of technical support on the industrial site, taking advantage of the mobility of the platform.

These three arguments give strength to advance with the project, and also a guideline of how the final product should look like.

Chapter 3

SYSTEM REQUIREMENTS AND SYSTEM ARCHITECTURE

This chapter formulates the tackled problems in the project and delivers an analysis of the proper solution.

It is a target of this chapter to define a system, in order to solve the specified problems, independently of the specific underlying hardware or software framework chosen. Therefore, the project system requirements and architecture are presented.

3.1 FORMULATION OF THE PROBLEM AND PROPOSED SOLUTION

The project focus in solving three problems in the industrial automation field.

One problem is that, occasionally, engineers go into the industrial site, inefficiently spending resources, to face problems that are easily solved by a reprogramming, or a part replacement, of the machine. Usually, this troubleshooting can be done without having physical access to the machine, only with a more accurate understanding of the situation or with remote access to the machine.

To monitor and control an industrial process, usually, only one HMI is used. This represents the second problem since the HMI is a non-shareable resource. If the responsible personnel for quality control, or other personnel of the plant, needs to monitor a process, they need to use this HMI. This will result in a decrease of productivity since the industrial operator will not be able to do his work during the control.

Lastly, today's industrial operator does not have the option to have technological functions, present in commercial "smart" devices, in the industrial floor. The operator does not have the option to, for example, send an e-mail or take a photo while in the industrial floor, without exposing a device to unsafe conditions.

The developing of an application for a mobile device is seen as a solution. This application enables the creation of new tools to use in the industrial automation field. These tools can provide solutions for the problems stated, and furthermore for problems not yet discovered.

The developed tools in the scope of this project should focus in the monitor and control of industrial processes, as well the ability to provide technical support.

The application has the requirement of being easily expandable, so the using of a modular architecture is an approach to consider. It is an aim to make possible the development of new tools independently and fit them in the application.

3.2 SYSTEM REQUIREMENTS

The importance of defining correctly the requirements of an engineering project is undeniable. The risk of not finishing the project in time is minimized, it is possible to predict errors earlier, as well to plan the stages of the project knowing clearly what is wanted on the end [45].

In this section, the requirements gathering will be presented, which include the software features, as well the users and how they operate with the software.

Then it is presented a set of use cases, functional and non-functional requirements for the developed software.

3.2.1 Requirements Gathering

The software features that the mobile system should possess, in order to solve the specified problems, were discussed in meetings with the engineers of Bresimar.

These features were based on the field experience of the engineers, as well on the evaluation of the characteristics of the commercial products shown in the state of the art study.

The requirements predicted for the system are the following:

- Provide technical support via a video call or a video call;
- Provide technical support via a remote access to a HMI and a PLC associated with an industrial process;
- Monitor and control a HMI associated with an industrial process;
- Monitor and control a PLC associated with an industrial process;
- Automatically acquire the communications parameters of the devices to monitor and control;
- Provide the use of multimedia functions;
- Provide a way to record notes;
- Provide access to an email client;
- Provide a way to record events;
- Provide access to a contacts list;

The users of the software application are the industrial plant personnel, which use the functions stated.

To organize all these functions, in a way that the user can navigate through them, the idea of creating a dedicated application on top of an operating system arose. An environment will be created where is presented to the user a set of icons, where each icon is a module that represents a software feature. This is similar to the concept found in mobile operating system, where the functions are separated through applications.

The software application should be a main application, which displays a home screen containing a set of icons. This application does all the management needed and loads the different functions.

3.2.2 Division of Requirements in Modules

The modules are part of the software that implements the different functions used by the industrial personnel. They are initialized and disabled when the user clicks the different icons. These modules have direct interaction with the user and provide tools to use in industrial automation.

The modules predicted in the scope of the final product are the next:

- **Technical Support**, allows to provide technical support via a voice and video call, or via remote access to a HMI and PLC associated with an industrial process;
- **Access the HMI**, allows to monitor and control a HMI associated with an industrial process;
- **Access the PLC**, allows to monitor and control an PLC associated with an industrial process;
- **Camera**, allows to have access to a camera to take pictures or make videos;
- **Multimedia**, allows to visualize multimedia present on the system;

- **Notes**, provides a way to take notes and to visualize them;
- **Email**, provides access to an email client;
- **Events**, provides a way to record events;
- **Contacts**, provides access to a contacts list;
- **Lock**, allows to put the system in a low-power mode when it is not in use;
- **Settings**, allows to do a set of configurations to the system;

Regarding these modules, is possible to make a sorting according to their importance in solving the previously stated problems. The following list does the sorting in three levels of priorities:

Maximum Priority:

- Technical Support;
- Access the HMI;
- Access the PLC;

Medium Priority:

- Camera;
- Multimedia;
- Email;
- Lock;
- Settings;

Minimum Priority:

- Notes;
- Events;
- Contacts;

The priorities assignment is based on the core of the project, to monitor and control industrial processes, as well to provide a way to give technical support. In order to do that, is needed a tool to communicate with the client, other tool to have remote access to the industrial process, and another tool to access the industrial process locally. The maximum priority modules fulfil these objectives.

The modules with medium priority, do not belong to the core function of the project, they are extra tools to provide technical support and to customize the software.

The modules with minimum priority add extra functions in the use of the application, but they are not essential to its core function.

Besides these modules, there is a part of the software responsible for the activity management of the application, which creates the graphical user interface and loads the different modules of the software. The name of this software part is **Modules Manager.**

3.2.3 Use Cases

It will now be described a series of use cases, to define the interactions the users will have with the maximum priority modules, as well with the main program that manages these modules, the **Modules Manager**.

Table 1 - UC-1: Load a module

Name	**UC-1:** Load a module
Summary	A module is loaded by the user.
Rationale	For the user to access the different functions of the software it is needed to load different modules.

Preconditions	The software is loaded.
Basic Course of Events	1. The user clicks the icon of the module that he wants to start, on the home screen. 2. The software responds by creating a panel with the icons on the left side and a space for the module to run on the right side.
Alternative Paths	1. If a module is already loaded the user starts another by clicking on the icon in the left panel created.
Post conditions	A module is loaded.

Table 2 - UC-2: Unload a module

Name	**UC-2:** Unload a module
Summary	A module is unloaded by the user.
Rationale	For the user to access the different functions of the software it is needed to load different modules, but the user may also have the option to unload the current module, when it is no longer needed.
Preconditions	The software is loaded, and a module was previously loaded.
Basic Course of Events	1. The user clicks the icon with the Home or Back icon. 2. The software responds by unloading the current module and returning to the home screen.
Alternative Paths	1. If the user clicks on another icon in the left panel created the current module is also unloaded, and another module is loaded.
Post conditions	The current module is unloaded.

Table 3 - UC-3: Allow Remote Access

Name	**UC-3:** Allow Remote Access
Summary	A connection to allow remote access to the Tablet PC is established.
Rationale	When giving technical support, the need to access the Tablet PC remotely may arise. This happens because the engineer giving technical support may need to reprogram or configure the machine remotely.
Preconditions	The **Technical Support Module** is loaded, and an Internet connection is available.
Basic Course of Events	1. The user clicks the button with the description “Allow remote access”. 2. The software responds by starting a desktop sharing application. 3. The user interacts confirming the desktop sharing application and allowing remote access to the Tablet PC.
Alternative Paths	1. This use case is also available during a Voice Call. 2. The user may decide to abort the operation at any time during steps 1, 2 or 3. This interruption is made by: returning to the home page or initiating a module, including the **Technical Support Module**.
Post conditions	A desktop sharing application is started.

Table 4 - UC-4: Make a Video call

Name	**UC-4:** Make a Video call
Summary	A video call is done by selecting a contact from a list.
Rationale	The user can receive technical support by making a video call. This video call is useful to give a point of view of the situation to the person providing technical support. If the machine is not working properly or for troubleshooting cases. The person providing support will have a perspective of the situation without the need to go on site.

Preconditions	The **Technical Support Module** is loaded, an Internet connection is available, and a contact is online.
Basic Course of Events	1. The user selects a contact from the list 2. The user clicks the button with the description "Video Call". 3. A video call is established.
Alternative Paths	1. If step 2 is done previously than step 1 a message appears. 2. Step 3 is jumped if the contact is not available. 3. The user may decide to abort the operation at any time during steps 1, 2, or 3. This interruption is made by returning to the home page or initiating a module, including the **Technical Support Module**.
Post conditions	A video call is established

Table 5 - UC-5: Change the Camera in use during Video Call

Name	**UC-5:** Change the Camera in use during a Video call
Summary	The user has the option, during a video call, to change the camera in use.
Rationale	The user can receive technical support by making a video call. This video call is useful to give a point of view of the situation to the person providing technical support. If the machine is not working properly or for troubleshooting cases. The person providing support will have a perspective of the situation without the need to go on site, but the user may be able to change this perspective, for example, to show the video captured by an IP Camera in a physically non-accessible point of the machine.
Preconditions	The **Technical Support Module** is loaded, and a video call is established.
Basic Course of Events	1. The user clicks the button with the description "Change Camera in use". 2. The software presents a list of the available cameras. 3. The user selects the camera from the list. 4. The software changes the camera in use.
Alternative Paths	1. The user may decide to abort the operation at any time during steps 1, 2 or 3. This interruption is made by clicking in the surrounding area of the **Technical Support Module**
Post conditions	The camera in use for the video call is changed

Table 6 - UC-6: Take a screenshot during Video Call

Name	**UC-6:** Take a screenshot during Video call
Summary	The user has the option, during a video call, to take a screenshot of image provided by the camera in use.
Rationale	The user can receive technical support by making a video call. This video call is useful to give a point of view of the situation to the person providing technical support. During this support, the user may have the need to capture the image provided by a camera in use for future reference. Thus, the option to take a screenshot during a video call is provided.
Preconditions	The **Technical Support Module** is loaded, and a video call is established.
Basic Course of Events	1. The user clicks the button with the description "Take Screenshot". 2. The software saves the image in a predefined folder of the Tablet PC.
Post conditions	A screenshot of image provided by the camera in use is saved.

Table 7 - UC-7: Make a Voice Call

Name	**UC-7:** Make a Voice call
Summary	A voice call is done by selecting a contact from a list.

Rationale	The user can receive technical support by making a voice call. This voice call is useful, for example, to give the parameters to allow remote access to the Tablet PC. Other use is to have a voice communication in an industrial environment.
Preconditions	The **Technical Support Module** is loaded.
Basic Course of Events	1. The user selects a contact from the list 2. The user clicks the button with the description "Voice Call". 3. A voice call is established.
Alternative Paths	1. If step 2 is done previously than step 1 a message appears. 2. Step 3 is jumped if the contact is not available. 3. The user may decide to abort the operation at any time during steps 1, 2, or 3. This interruption is made by returning to the home page or initiating a module, including the **Technical Support Module**.
Post conditions	A voice call is established.

Table 8 - UC-8: Access to a HMI with a QR code

Name	**UC-8:** Access to a HMI with a QR code
Summary	A HMI is accessed by reading and decoding of a QR code.
Rationale	The access to a HMI requires specific parameters for establishing a connection. In order to simplify the work of the user, these parameters are encoded in a QR code. Thus, the access to the HMI is done automatically by reading and decoding of a QR Code.
Preconditions	The **Access the HMI Module** is loaded.
Basic Course of Events	1. The user clicks the button with the image of a QR code, under the descriptive title of the action. 2. The software responds by initiating the rear camera of the Tablet PC and displaying information for the user to decode the QR code. 3. The software decodes the information in the QR code, once it detects one. 4. The software asks the user to store the information read. 5. The user inserts the name of the machine, which information is stored. 6. The HMI is accessed.
Alternative Paths	1. In step 4, if the software fails to decode a QR code or the information on it is invalid, it returns to immediately before step 1. 2. If the information was already stored steps 4 and 5 are jumped. 3. Steps 5 can be jumped if the user selects so. 4. Step 6 is jumped if the software cannot access the HMI device. 5. The user may decide to abort the operation at any time during steps 1, 2, 3, 4, 5, or 6. This interruption is made by returning to the home page or initiating a module, including the **Access the HMI Module**.
Post conditions	Information regarding a machine is stored in the memory of the program for future access. An access to a HMI is made.

Table 9 - UC-9: Access to a HMI by Memory

Name	**UC-9:** Access to a HMI by Memory
Summary	A HMI is accessed by choosing from a list of previously stored information of a machine.
Rationale	The access to a HMI requires specific parameters for establishing a connection. In order to simplify the work of the user, these parameters are

	encoded on a QR code, but the user may want not to have to read a QR code each time he accesses a HMI. Therefore, the option to access a HMI by selecting from a list of devices is also provided.
Preconditions	The **Access the HMI Module** is loaded, and information of a machine was previously stored.
Basic Course of Events	1. The user selects a HMI device from the list 2. The user clicks the button with "Access the HMI" description. 3. The HMI is accessed.
Alternative Paths	1. If step 2 is done previously than step 1 a message appears 2. If the software cannot access the HMI device, step 3 is jumped and a message appears. 3. The user may decide to abort the operation at any time during steps 1, 2, or 3. This interruption is made by returning to the home page or initiating a module, including the **Access the HMI Module**.
Post conditions	An access to a HMI is made.

Table 10 - UC-10: Connection to a PLC with a QR code

Name	**UC-10:** Connection to a PLC with a QR code
Summary	A connection to a PLC is done by reading and decoding of a QR code.
Rationale	The access to a PLC can be done in two ways. Accessing the variables directly or using the PLC proprietary software. It requires specific parameters for establishing a connection. In order to simplify the work of the user, these parameters are encoded in a QR code. Thus, the connection to the PLC is done automatically, by reading and decoding of a QR Code.
Preconditions	The **Access the PLC Module** is loaded.
Basic Course of Events	1. The user clicks the button with the image of a QR code, under the descriptive title of the action. 2. The software responds by initiating the rear camera of the Tablet PC and displaying information for the user to decode the QR code. 3. The software decodes the information in the QR code, once it detects one. 4. The software asks the user to store the information read. 5. The user inserts the name of the machine, which information is stored. 6. A connection to a PLC is established.
Alternative Paths	1. In step 4, if the software fails to decode a QR code or the information on it is invalid, it returns to immediately before step 1. 2. If the information was already stored, steps 4 and 5 are jumped. 3. Step 5 can be jumped if the user selects so. 4. Step 6 is jumped if the software cannot connect to the PLC. 5. The user may decide to abort the operation at any time during steps 1, 2, 3, 4, 5 or 6. This interruption is made by returning to the home page or initiating a module, including the **Access the PLC Module**.
Post conditions	Information regarding a machine is stored in the memory of the program for future access. A connection to a PLC is established.

Table 11 - UC-11: Connection to a PLC by Memory

Name	**UC-11:** Connection to a PLC by Memory
Summary	A connection to a PLC is done by choosing from a list of previously stored information of a machine.

Rationale	The access to a PLC can be done in two ways. Accessing the variables directly or using the PLC proprietary software. It requires specific parameters for establishing a connection. In order to simplify the work of the user, these parameters are encoded on a QR code, but the user may want not to have to read a QR code each time he connects to a PLC. Therefore, the option to connect to a PLC by selecting from a list of devices is also provided.
Preconditions	The **Access the PLC Module** is loaded, and information of a machine was previously stored.
Basic Course of Events	1. The user selects a machine from the list 2. The user clicks the button with the description "Access the PLC". 3. A connection to a PLC is established.
Alternative Paths	1. If step 2 is done previously than step 1 a message appears 2. Step 3 is jumped if the software cannot connect to the PLC. 3. The user may decide to abort the operation at any time during steps 1, 2, or 3. This interruption is made by returning to the home page or initiating a module, including the **Access the PLC Module**.
Post conditions	A connection to a PLC is established.

Table 12 - UC-12: Accessing PLC with proprietary software

Name	**UC-12:** Accessing PLC with proprietary software
Summary	The user has the option to access the PLC using a proprietary software.
Rationale	The access to a PLC can be done in two ways. Accessing the variables of the PLC directly or using the proprietary software of the PLC. Accessing with the proprietary software of the PLC allows the user do some configurations and reprogram the machine. This function is useful when providing technical support via a remote access to the Tablet PC.
Preconditions	The **Access the PLC Module** is loaded, and a connection to a PLC is established.
Basic Course of Events	1. The user clicks the button with the description "Access using the proprietary software of the PLC". 2. The application launches the corresponding proprietary software of the PLC.
Alternative Paths	1. The user may decide to abort the operation at any time during steps 1 or 2. This interruption is made by returning to the home page or initiating a module, including the **Access the PLC Module**.
Post conditions	The proprietary software of the PLC is launched.

Table 13 - UC-13: Accessing directly to the PLC variables

Name	**UC-13:** Accessing directly to the PLC variables
Summary	The user has the option to access directly to the variables of the PLC.
Rationale	The access to a PLC can be done in two ways. Accessing the variables directly or using the PLC proprietary software. Accessing to the variables of the PLC directly allows the user to monitor and control the industrial process. This monitor and control can be done with the creation of a HMI by the user. This function is useful when using the Tablet PC for monitor and control on the industrial site without the need of a dedicated HMI.
Preconditions	The **Access the PLC Module** is loaded, and a connection to a PLC is established.
Basic Course of Events	1. The user clicks the button with "Access directly to the variables of the PLC" description. 2. The software displays a list containing the PLC variables.

	3. The user selects which are the variables he wants to monitor and control. 4. The software creates a HMI with the chosen variables. 5. The user can interact with this HMI by monitoring their value and editing them.
Alternative Paths	1. The user may decide to abort the operation at any time during steps 1, 2, 3, 4 or 5. This interruption is made by returning to the home page or initiating a module, including the **Access the PLC Module**.
Post conditions	A HMI is created with the variables chosen by the user

Table 14 - UC-14: Erase Information of a Machine

Name	**UC-14:** Erase Information of a Machine
Summary	Selected information of a machine is erased.
Rationale	While using the **Access the PLC Module**, or the **Access the HMI Module**, the user may want to reorganize the list of the machines previously saved, for example in the case of the configuration have changed. Therefore, there must exist the option to erase one information of a machine, as well to erase all the machines information.
Preconditions	The **Access the PLC Module**, or the **Access the HMI Module**, is loaded, and at least one information of a machine was already stored.
Basic Course of Events	1. The user selects a machine from the list. 2. The user clicks the button with the description “Erase Information of a machine”. 3. The software asks for a confirmation of the action. 4. The software erases the information stored.
Alternative Paths	1. If in step 2 the user clicks and hold the button “Erase Information of a machine” description, the information erased will not regard one machine but all machines stored. 2. If step 2 is done previously than step 1 a message appears. 3. Step 4 is jumped if the user does not confirm the operation. 4. The user may decide to abort the operation at any time during steps 1, 2, 3 or 4. This interruption is made by returning to the home page or initiating a module.
Post conditions	The information of a machine is erased.

3.2.4 Functional Requirements

It will now be described a set functional requirements, to define the behaviour of the maximum priority modules and also the main program that manages these modules.

Table 15 - FR-1: Full Screen on the OS launch

Name	**FR-1:** Full Screen on the OS launch
Summary	The application is launched at full screen on the launch of the operating system.
Rationale	In order to minimize the run-time errors, or misuse of the tool provided, it is needed to prevent the user to access the operating system.
Requirements	When the user starts the Tablet PC the software is launched in full screen mode.

Table 16 - FR-2: Status Bar

Name	**FR-2:** Status Bar
Summary	A status bar with useful information is presented to the user.

Rationale	The user does not have access to the operating system, it is then needed to give information inside the application.
Requirements	Present a status bar with the following information: • Date; • Time; • Wireless Network Signal; • Cellular Network Signal; • Battery life; • Supervisor Name;

Table 17 - FR-3: Home Screen

Name	**FR-3:** Home Screen
Summary	Creates a set of icons with the different modules charged on the project.
Rationale	For the user to choose what modules to start a set of icons are displayed in the home screen.
Requirements	The set of icons created have the following characteristics: • The image and name of the icon is set previously; • It is not supported creation of folders; • It is not possible to have more than one page to show icons; • The page has a limitation to define of icons to present; • It is not possible to change the position of the icons in run-time;
References	**UC-1:** Load a module.

Table 18 - FR-4: Run Screen

Name	**FR-4:** Run Screen
Summary	When the user starts a module is created a panel with the icons of the modules on the left side, and a blank space for the module to run on the right side.
Rationale	For the user to access the software feature the modules have to be loaded and reserved space for them to run.
Requirements	• Create a panel with the icons of the modules on the left side; • Create a blank space for the module to run on the right side; • Present on the panel the module running, with a visual indication on the icon; • Clicking on the button of another module, the current module is stopped and the clicked one is started; • Clicking on the button Home the user comes back to the home page;
References	UC-1: Load a module. UC-2: Unload a module.

Table 19 - FR-5: Multilingual Support

Name	**FR-5:** Multilingual Support
Summary	The user can change the language of the software.
Rationale	The user may have the need to change the language of the software to its mother language. The reason is to improve the longevity and scalability of the project. It is not desirable to limit the application to only one language.
Requirements	• The application has to come with two changeable languages, Portuguese and English. • The application has support to a language file so that new languages can be simply inserted

Table 20 - FR-6: Resource Management

Name	**FR-6:** Resource Management
Summary	The **Modules Manager** saves information used by the modules.
Rationale	There is information transversal to the different modules. The **Modules Manager** will then store and provide access to this information.
Requirements	• Stores information about the languages of the software; • Stores the information about the themes of the software; • Stores the images used by different modules;

Table 21 - FR-7: Internet Access

Name	**FR-7:** Internet Access
Summary	An internet access connection is needed to provide the functions of the **Technical Support Module**.
Rationale	For having remote access, making a video call, or voice call with the Tablet PC it is needed an internet connection.
Requirements	It is needed an internet access connection while at the same time use the Wi-Fi connection to communicate with the devices. • A 3G connection has to be configured on the Tablet PC to have an internet access while at the same time have Wi-Fi communication;
References	UC-3: Allow Remote Access. UC-4: Make a Video call. UC-7: Make a Voice Call.

Table 22 - FR-8: Sharing Desktop Application

Name	**FR-8** Sharing Desktop Application
Summary	For allowing remote access to the Tablet PC is used a Sharing Desktop Application.
Rationale	For having remote access to the Tablet PC it has to be used a Sharing Desktop Application. This software allows to have remote access to the Tablet PC.
Requirements	For having remote access using a proprietary software, is needed to: • The software to launch a Sharing Desktop Application
References	UC-3: Allow Remote Access.

Table 23 - FR-9: Video Call

Name	**FR-9:** Video Call
Summary	For making a video call with the Tablet PC is used the proprietary software "Skype" [46] or similar.
Rationale	For making a video call with the Tablet PC is used the proprietary software Skype or similar. This allows to make a video call from the Tablet PC without back-end.
Requirements	For making a video call with the proprietary software, is needed to: • Install the proprietary software in the Tablet PC. • The software to launch an external software.
References	UC-4: Make a Video call.

Table 24 - FR-10: Multitasking

Name	**FR-10:** Multitasking
Summary	A call or a remote access are not interrupted when changing the loaded module.
Rationale	When giving technical support the need to have access remote and use other software may arise. For example, if the person giving remote access needs to access the proprietary software of the PLC

Requirements	A call, voice or video, or a remote access, are not interrupted with the changing of the loaded module.
References	UC-3: Allow Remote Access. UC-4: Make a Video call. UC-7: Make a Voice call.

Table 25 - FR-11: Access to multiple cameras

Name	**FR-11:** Access to multiple cameras
Summary	The user can change the camera in use when making a video call.
Rationale	The user may be able to change the camera in use, for example, to show the video captured by an IP Camera in a non-accessible physical point of the machine.
Requirements	Have access to both camera of the Tablet PC and to the IP Camera connected to the wireless local area network.
References	UC-4: Make a Video call. UC-5: Change the Camera in use during Video call.

Table 26 - FR-12: Take a screenshot from the camera in use

Name	**FR-12:** Take a screenshot from the camera in use
Summary	The user can take a screenshot of the image provided by the camera in use when making a video call.
Rationale	The user may have the need to capture the image provided by a camera in use for future reference. Thus the option to take a screenshot during a video call is provided.
Requirements	Have access to the camera in use or to the functionality of taking a screenshot of the image provided.
References	UC-4: Make a Video call. UC-6: Take a screenshot during Video call.

Table 27 - FR-13: Camera interface

Name	**FR-13:** Camera interface
Summary	Is provided access and the option to take a picture from the camera of the Tablet PC.
Rationale	A user has the option to access or to record a machine, by reading and decoding a QR Code. To read this QR Code is needed to capture a picture from the camera interface.
Requirements	When a user is trying to read a QR code the software must give the image provided by the camera interface.
References	UC-8: Access a HMI with a QR code UC-10: Connection to a PLC with a QR code

Table 28 - FR-14: Decoding of a QR Code

Name	**FR-14:** Decoding of a QR Code
Summary	To access and to record a machine is needed to decode the information in a QR Code.
Rationale	A user has the option to access and to record a machine by the reading and decoding of a QR Code. This QR code contains information about the machine chosen.
Requirements	An image from the camera of the Tablet PC is obtained and then processed to decode the QR Code.
References	UC-8: Access a HMI with a QR code. UC-10: Connection to a PLC with a QR code. FR-13: Camera interface.

Table 29 - FR-15: Decryption of information

Name	**FR-15:** Decryption of information
Summary	The information encoded on the QR Code is decrypted.
Rationale	The information encoded on the QR Code has to be encrypted in order to improve the security, for this reason is needed a way to encrypt and decrypt information.
Requirements	The information encoded in the QR Code is encrypted using an algorithm and a password. The software has to decrypt the information in order to use it.
References	UC-8: Access a HMI with a QR code. UC-10: Connection to a PLC with a QR code. FR-13: Camera interface. FR-14: Decoding of a QR Code.

Table 30 - FR-16: Persistence of information

Name	**FR-16:** Persistence of information
Summary	The information decoded from a QR Code is stored.
Rationale	A user has the option to access the devices by selecting from a list of devices. The information previously decoded from a QR Code needs to be stored.
Requirements	Information is persisted when reading and decoding of a QR Code is done.
References	UC-8: Access a HMI with a QR code. UC-9: Access a HMI by Memory. UC-10: Connection to a PLC with a QR code. UC-11: Connection to a PLC by Memory. FR-13: Camera interface. FR-14: Decoding of a QR Code. FR-15: Decryption of information.

Table 31 - FR-17: Connection to a Network

Name	**FR-17:** Connection to a Network
Summary	The Tablet PC connects to the network where the device the user wants to access is connected.
Rationale	Since the user does not have access to the operating system, and is needed to change the network the Tablet PC is currently connected to, a way to control the connection to wireless networks is needed.
Requirements	• If the Tablet PC is not on the same network of the device, it has to connect to that network; • If the Tablet PC is connected to the same network of the device, it has to check the availability of the device;
References	UC-8: Access a HMI with a QR code. UC-9: Access a HMI by Memory. UC-10: Connection to a PLC with a QR code. UC-11: Connection to a PLC by Memory.

Table 32 - FR-18: VNC Access

Name	**FR-18:** VNC Access
Summary	Establishes remote connection via VNC to a HMI.
Rationale	To access a HMI is used a VNC client. It is then needed a manner to establish a connection of this type.
Requirements	• A VNC connection is established using the parameters encoded in the QR code.

	• With the connection is possible to visualize the screen of the HMI, as well to interact with it.
References	UC-8: Access a HMI with a QR code. UC-9: Access a HMI by Memory.

Table 33 - FR-19: Proprietary Software

Name	**FR-19:** Proprietary Software
Summary	The PLC is accessed using a proprietary software.
Rationale	To reprogram or configure the PLC often it is needed to use its proprietary software. This function is needed when accessing the Tablet PC remotely, in order to provide technical support.
Requirements	For accessing the PLC using a proprietary software it is needed: • To install the proprietary software in the Tablet PC; • The application to launch an external software;
References	UC-3: Allow Remote Access. UC-10: Connection to a PLC with a QR code. UC-11: Connection to a PLC by Memory. UC-12: Accessing PLC with proprietary software.

Table 34 - FR-20: Communication Protocol

Name	**FR-20:** Communication Protocol
Summary	The PLC variables are accessed directly using a specific communication protocol.
Rationale	For monitor and control the PLC variables directly it is needed to support the PLC communication protocol.
Requirements	• Support for different communication protocols;
References	UC-10: Connection to a PLC with a QR code. UC-11: Connection to a PLC by Memory. UC-13: Accessing directly to the PLC variables.

3.2.5 Non-Functional Requirements

The application of the project needs a modern graphical interface, so that it provides an ease of use when working with the provided tools. The user should know how to access and use the tools on the first time he uses the product.

This is an arguable topic, what for certain users is intuitive, for others it is not, and so the graphical part cannot be considered to be a requirement. Is possible to see, as example, how contradictory can be the feedback in terms of usability of the operating system Windows 8, in [47].

What can be made, along the project, is to have always focus on the user of the product. Focusing that the persons using the product do not have a strong informatics background knowledge, but still they are getting familiar with commercial "smart" devices, like Tablets and Smartphones.

Another important topic to follow is the different guidelines, and concepts, in the creation of applications for different mobile platforms. Mobile platforms, like iOS, Android, Windows Phone, that despite the fact they target the development of applications for their framework, they still provide useful design principles, like the ones found in [48] and [49].

3.3 Software UI Mockup

An UI Mockup was designed to create user interfaces so it is possible to show how the software will look and feel, without having to code one line.

The main objective was to predict user interfaces of the software, and to discuss them with the team of engineers in Bresimar.

The presentation of the mockup is divided into the **Modules Manager** and each module of the mobile application. It will be presented the developed mockup as well a description of the interaction and functions of the software part. In the **Modules Manager**, and the Modules with maximum priority, is also made a link with the previously defined use cases and functional requirements.

3.3.1 Modules Manager Mockup

The **Modules Manager** is the most vital part of the software. It provides a support basis for all the system. The **Modules Manager** is responsible to launch the software and host the different modules developed.

The operation of the **Modules Manager** is the following: when the Tablet PC is turned on, the software starts automatically and presents to the user a set of icons with a status bar on top. This is named the Home Screen, it is possible to see an example of the graphical user interface in Figure 14.

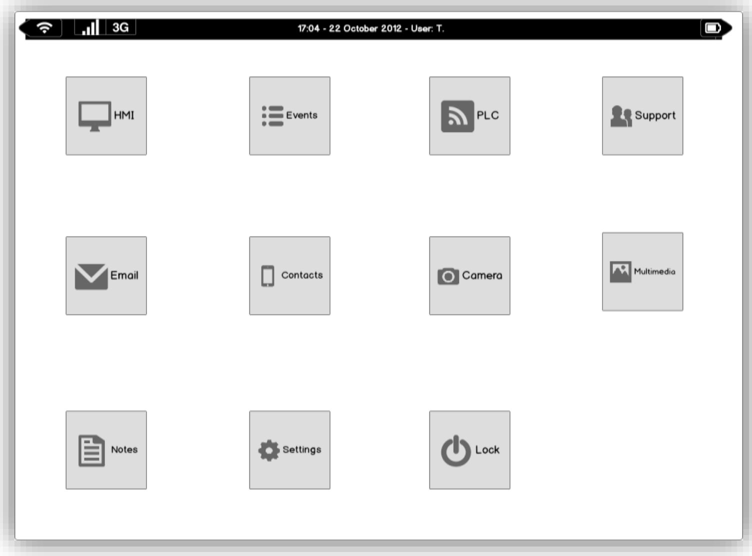

Figure 14 - Modules Manager: Home Screen Mockup

The icons in the Figure 14 represent the different modules charged on the software. They implement the different functions provided to the user. The status bar on top shows useful information to the user.

When the user clicks an icon, the according module is started. The **Modules Manager** creates a space on the right side for the module to run, and a panel on the left side with the icons of the modules. This is named the Run Screen, and it is possible to see an example of the graphical user interface in Figure 15.

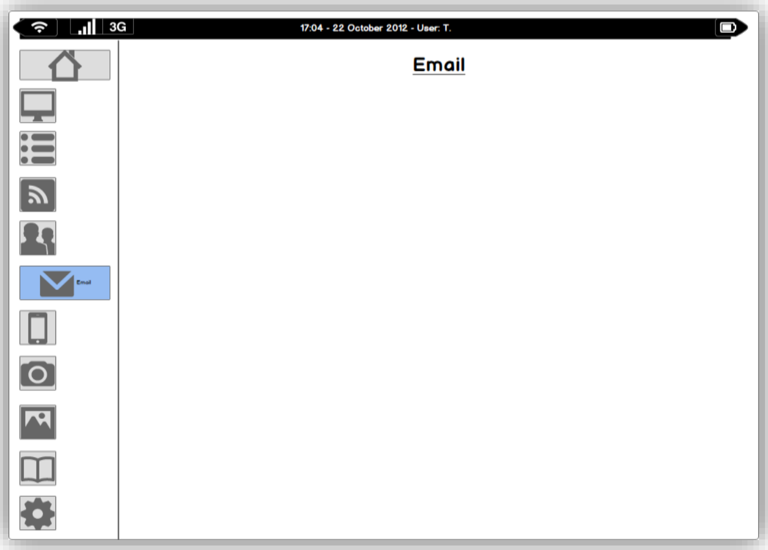

Figure 15 - Modules Manager: Run Screen Mockup

The **Modules Manager** is responsible by four main functions, already specified in the Functional Requirements.

First, to start the software and create the graphical user interface. More specifically, initiate in full screen mode and create a set of icons with the different modules charged on the project.

Second place, to manage the different modules through their different lifecycles, the **Modules Manager** is responsible for starting/stopping the different modules.

Third, to manage the resources shared among the modules, for example, the language of the system.

Finally the **Modules Manager** has the function to present different useful information in the form of a status bar. The information presented is the wireless network signal, cellular network signal, date, time, user name, and battery life.

The Mockup presented in the Figure 14 and Figure 15, relate to the Use Cases: - UC-1: Load a module, - UC-2: Unload a module, and to the Functional Requirements: - FR-1: Full Screen on the OS launch, - FR-2: Status Bar, - FR-3: Home Screen, - FR-4: Run Screen, - FR-5: Multilingual Support, - FR-6: Resource Management.

3.3.2 Technical Support Module Mockup

Using the **Technical Support Module** is possible to allow remote access to the Tablet PC, as well to make a video call or voice call.

When the module is loaded, it presents a list of contacts and three buttons to implement these functions. It is possible to see an example of the graphical user interface in Figure 16.

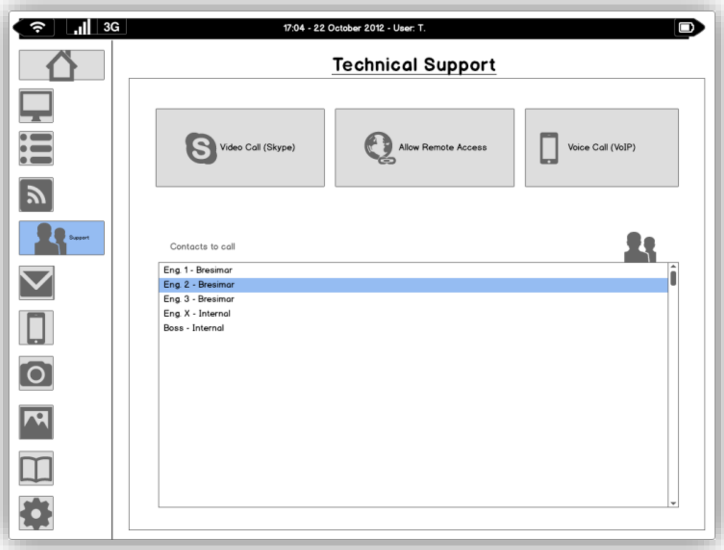

Figure 16 - Technical Support: Initial Screen Mockup

If the user selects the "Allow Remote Access" option, the software starts a Desktop Sharing application. It is possible to see an example of the graphical user interface in Figure 17.

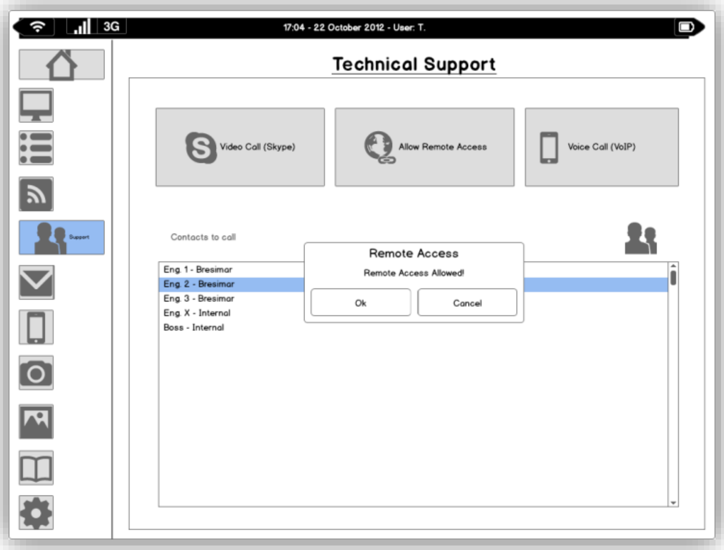

Figure 17 - Technical Support: Allow Remote Access Screen Mockup

The user also has the option to make a video call. It is possible, during the video call, to take a picture from a selected camera. In the Figure 18 it is shown an example of the graphical user interface during a call.

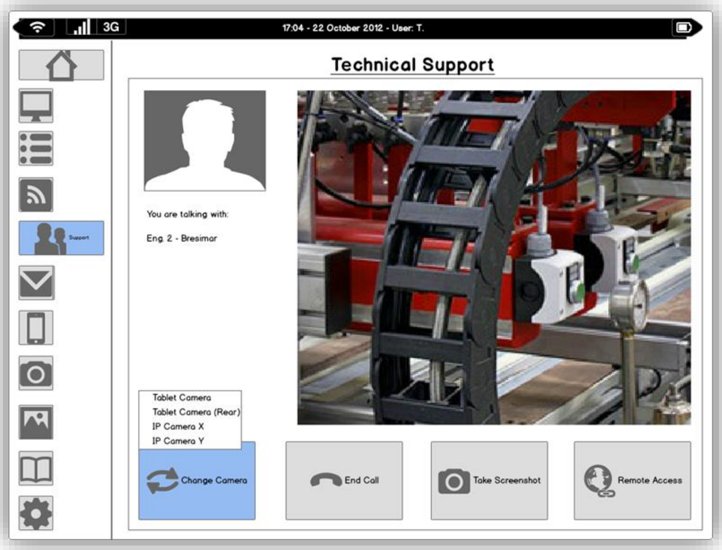

Figure 18 - Technical Support: Call Screen Mockup

If the call is only of voice the two buttons, "Change camera" and "Take Screenshot", as well the image of the camera, will not appear.

While making a call it is also possible to grant remote access. It is possible to see in Figure 19 an example of the graphical user interface.

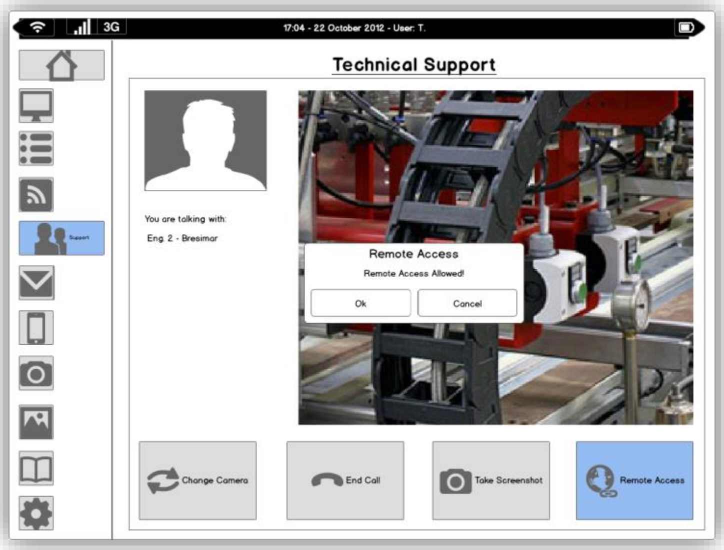

Figure 19 - Technical Support: Allow Remote Access in Call Screen Mockup

The functions presented in this module have to remain active while accessing other modules. That is, while the user makes a call, or granting remote access, the user is still able to access the PLC or the HMI.

The Mockup presented in the Figure 16, Figure 17, Figure 18 and Figure 19, relate to the Use Cases: - UC-3: Allow Remote Access, - UC-4: Make a Video call, - UC-5: Change the Camera in use during Video Call, - UC-6: Take a screenshot during Video Call, - UC-7: Make a Voice Call, and to the Functional Requirements: - FR-7: Internet Access, - FR-8: Sharing Desktop Application, - FR-9: Video Call, - FR-8: - FR-10: Multitasking, - FR-11: Access to multiple cameras, - FR-12: Take a screenshot from the camera in use**Error! Reference source not found.**.

3.3.3 Access the HMI Module Mockup

The **Access the HMI Module** allows the operator to access remotely to a HMI panel.

When the **Access the HMI Module** is loaded a window is presented with two titles that describe two options to gain access to the HMI. One of the options is by the reading of a QR code, the other one is by choosing from a list of previously stored machines. It is possible to see an example of the graphical user interface in Figure 20, where is presented a button with the image of a QR code associated with the first option, as well two buttons associated with the second option.

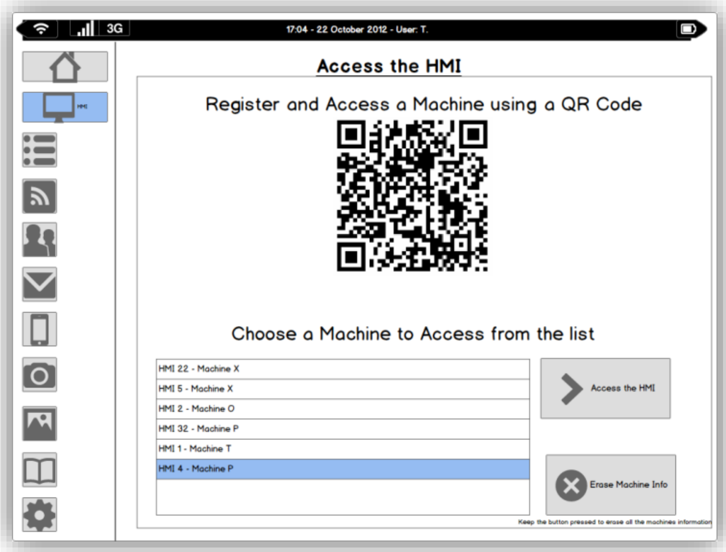

Figure 20 - Access the HMI: Initial Screen Mockup

When the HMI is accessed it is presented a window with a button on top to return to the selection window, and under a screen that contains the remote HMI screen. It is possible to see an example of the graphical user interface in Figure 21.

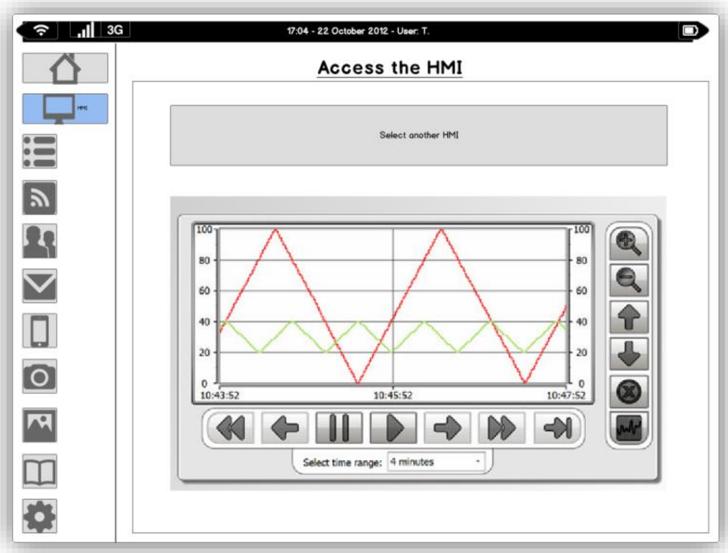

Figure 21 - Access the HMI: Access Screen Mockup

The Mockup presented in the Figure 20 and Figure 21, relate to the Use Cases: - UC-8: Access to a HMI with a QR code,- UC-9: Access to a HMI by Memory,- UC-14: Erase , and to the Functional

Requirements: - FR-13: Camera interface, - FR-14: Decoding of a QR Code, - FR-15: Decryption of information, - FR-16: Persistence of information, - FR-17: Connection to a Network, - FR-18: VNC Access.

3.3.4 Access the PLC Module Mockup

The **Access the PLC Module** allows to access a PLC with the Tablet PC. The PLC is connected to an access point that creates a wireless LAN, then the Tablet PC connects to this WLAN in order to communicate with the PLC. The communication is done using the protocol of the PLC.

The communications protocols used in industrial automation can be proprietary or non-proprietary [50]. A proprietary protocol is created and used by an individual or organization, it is a closed protocol. Usually every industrial automation manufacturer develops his own protocol. A non-proprietary, or open protocol, is maintained by an independent organization, and it is freely to be used by any person, an example is the Modbus protocol [51].

When the **Access the PLC Module** is loaded a window is presented, with two titles that describe two options to gain access to the PLC. One of the options is by the reading of a QR code, the other one is by choosing from a list of stored devices. It is possible to see an example of the graphical user interface in Figure 22, where is presented a button with the image of a QR code associated with the first option, as well two buttons associated with the second option.

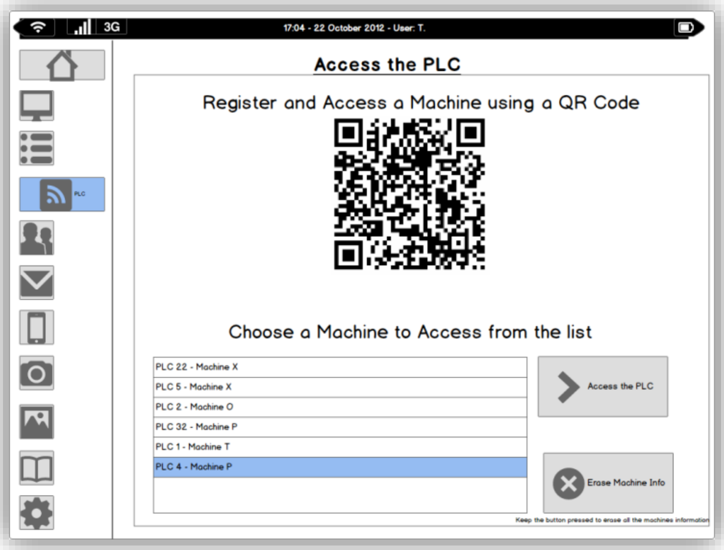

Figure 22 - Access the PLC: Initial Screen Mockup

When the communication with the PLC is established a window is presented, with two options to access the PLC, accessing the variables of the PLC directly, or accessing through the proprietary software of the PLC. It is possible to see an example of the graphical user interface in Figure 23.

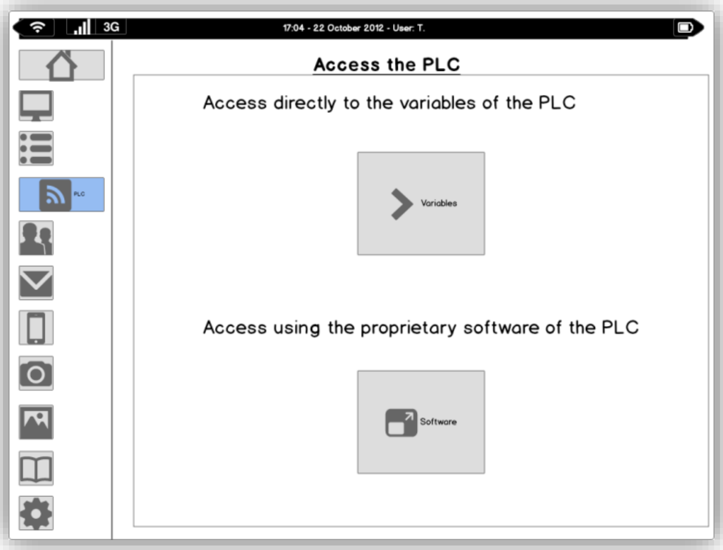

Figure 23 - Access the PLC: Access Screen Mockup

If the user selects the option to access using the proprietary software of the PLC, he will start the software associated with that particular PLC. With the proprietary software it is possible to reconfigure and to reprogram the PLC. This function is to be used by specialized people when they are doing a remote access to the Tablet PC. It is possible to see an example of the graphical user interface in Figure 24, where the user started the proprietary software to edit the machine code.

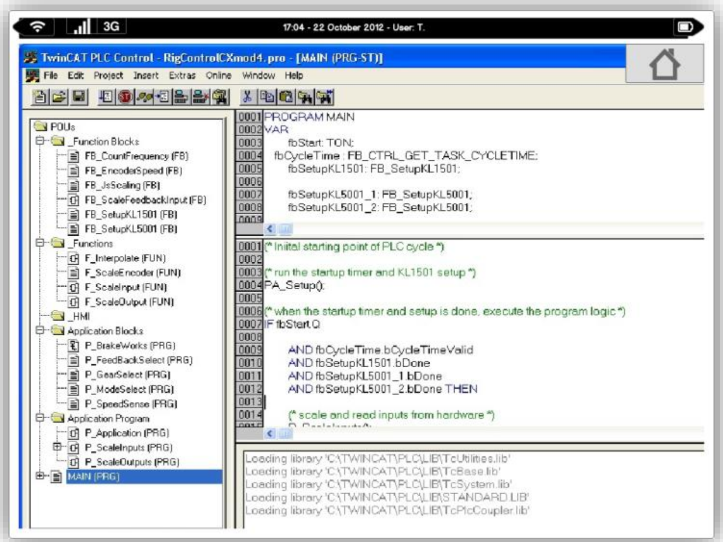

Figure 24 - Access the PLC: Access by Software Screen Mockup

If the user selects the option to access the variables directly, he has the option to create a HMI to monitor the PLC. It is possible to see an example of the graphical user interface in Figure 25, where the user has the option to choose certain variables to monitor, from a list containing all the variables of the PLC. He also has the option to load a previously created HMI.

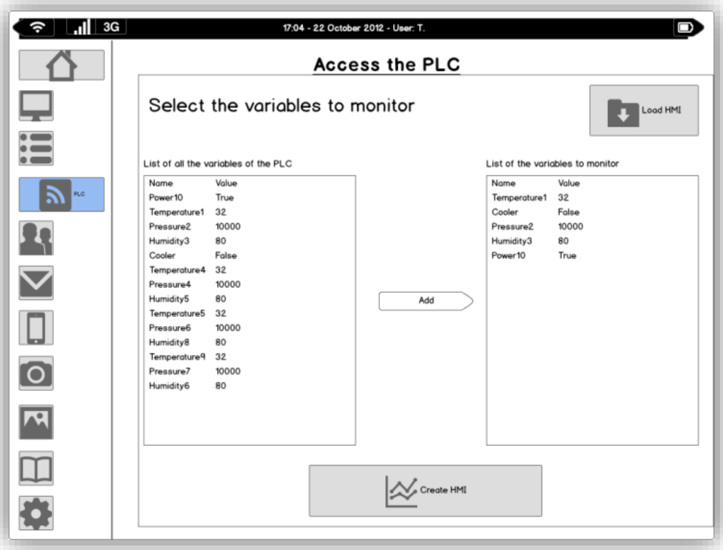

Figure 25 - Access the PLC: Select Variables Screen Mockup

The digital variables of the HMI created are represented by buttons, and the analogue variables by textboxes. In here, it is possible to edit their values, by pressing the button (for the digital inputs), or by writing a new value on the text box (for the analogue inputs). The option to save the created HMI for future access is also included. It is possible to see an example of the graphical user interface in Figure 26.

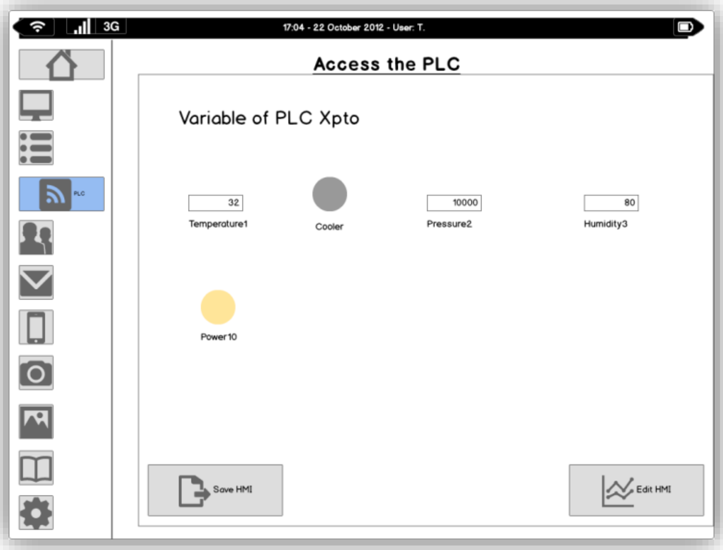

Figure 26 - Access the PLC: HMI Screen Mockup

When the user selects the option to edit the HMI, he has the option to edit properties associated with each variable. This editing can go from changing the variable associated, changing the action associated, or even changing the position of the control. It is possible to see an example of the graphical user interface in Figure 27.

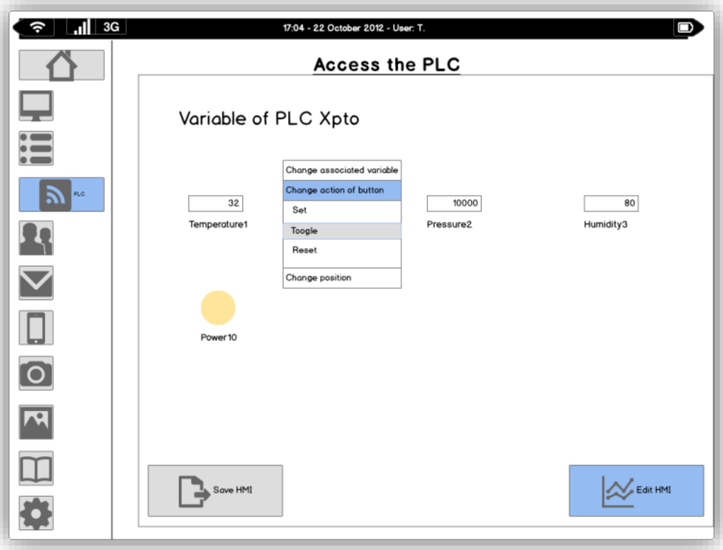

Figure 27 - Access the PLC: Edit HMI Screen Mockup

The final objective for this module, is to have a basis to add support for different communication protocols, so the system can grow in terms of applicability.

The Mockup presented in the Figure 22, Figure 23, Figure 24, Figure 25, Figure 26 and Figure 27, relate to the Use Cases: - UC-10: Connection to a PLC with a QR code, - UC-11: Connection to a PLC by Memory, - UC-12: Accessing PLC with proprietary software, - UC-13: Accessing directly to the PLC variables, - UC-14: Erase , and to the Functional Requirements: - FR-13: Camera interface, - FR-14: Decoding of a QR Code, - FR-15: Decryption of information, - FR-16: Persistence of information, - FR-17: Connection to a Network, - FR-19: Proprietary Software, - FR-20: Communication Protocol.

3.3.5 Camera Module Mockup

The **Camera Module** allows the user to have access to a camera, in order to take a picture, or to make a video.

It is possible to see an example of the graphical user interface in Figure 28. There are three buttons that allow to take a picture, make a video, or change the camera in use.

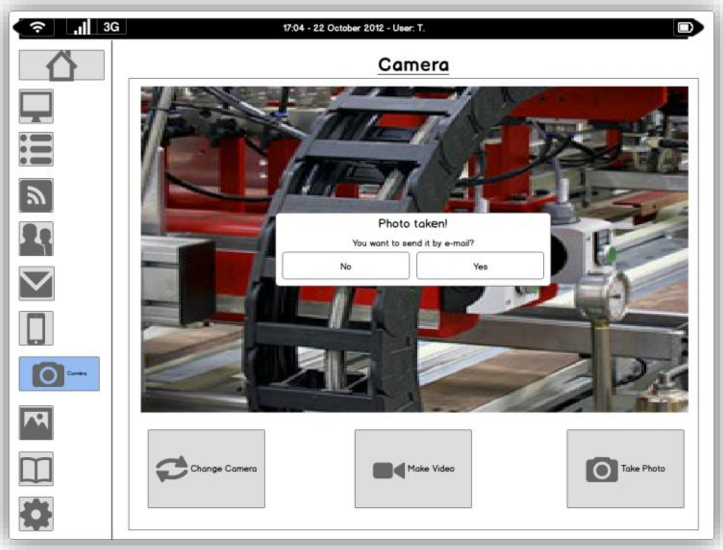

Figure 28 - Camera: Initial Screen Mockup

3.3.6 Multimedia Module Mockup

The **Multimedia Module** presents the photos and videos contained in the memory of the Tablet PC.

It is possible to see an example of the graphical user interface in Figure 29. In here it is possible to see two buttons to navigate through the gallery, a button to erase, a button to send through email and a button to editing, these last two only available for photos.

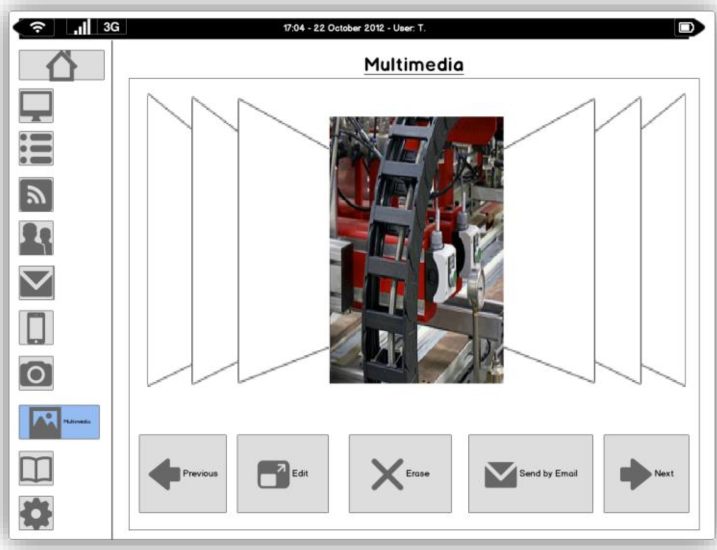

Figure 29 - Multimedia: Initial Screen Mockup

3.3.7 Email Module Mockup

The **Email Module** allows the user to see check an inbox of received email, as well the possibility to write and send an email. It is possible to see an example of the graphical user interface in Figure 30.

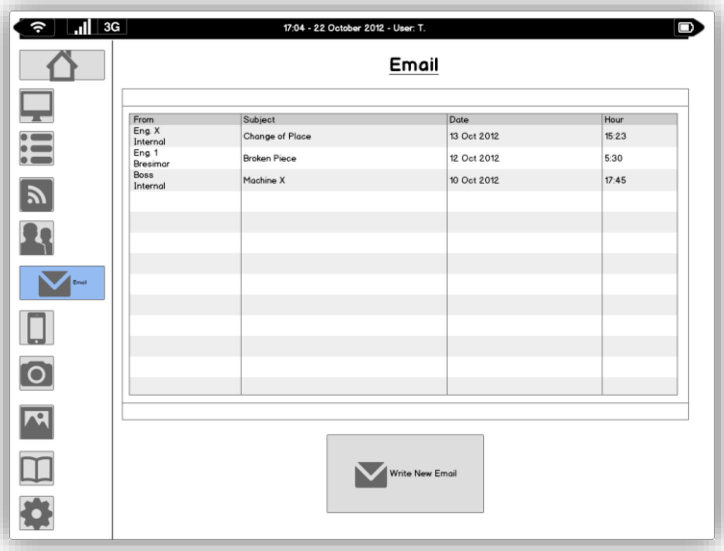

Figure 30 - Email: Initial Screen Mockup

If the user selects an email to read it the information of the recipient, the subject and the body are presented. It also shows the option to answer to email or to go back to the inbox. It is possible to see an example of the graphical user interface in Figure 31.

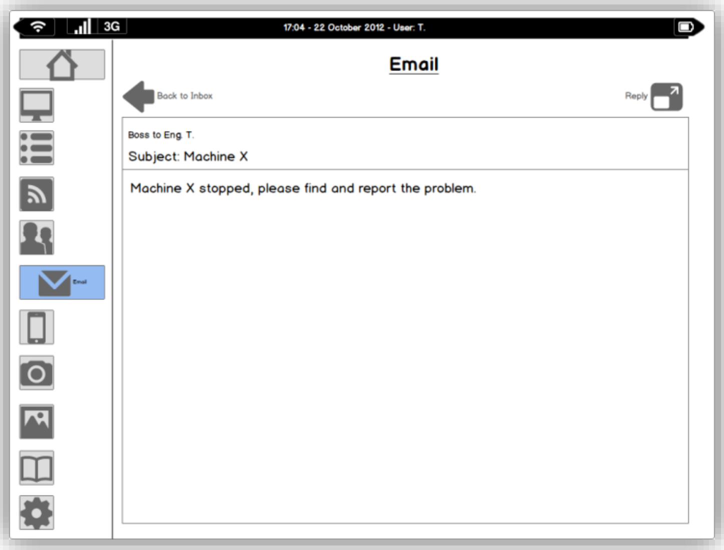

Figure 31 - Email: Read Screen Mockup

When is selected the option to write a new email the module allows to: write the message, select the contacts to send, attach a file, send and cancel. The email will be sent from the account of the operator using the Tablet PC. The files to attach in the email are photos in the memory of the Tablet PC. It is possible to see an example of the graphical user interface in Figure 32.

Figure 32 - Email: Write Screen Mockup

3.3.8 Lock Module Mockup

The **Lock Module** has the function to suspend the Tablet PC, in order to save energy, and to secure the Tablet PC. To enter again in the software is necessary to enter the user credentials. It is possible to see an example of the graphical user interface in Figure 33.

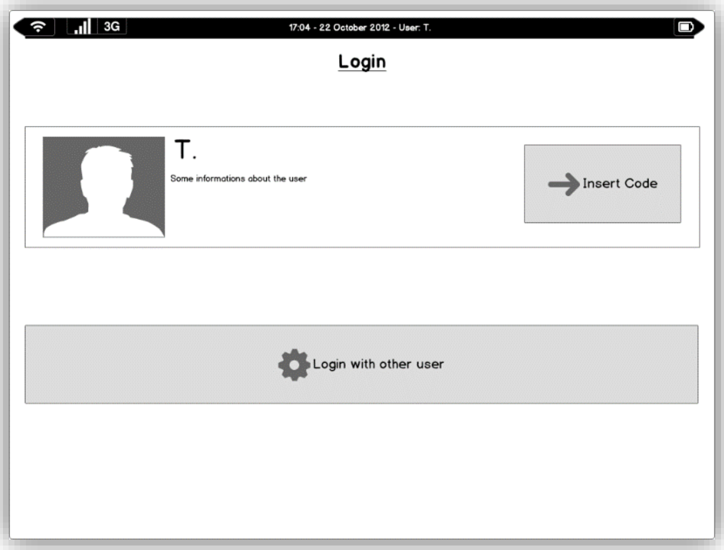

Figure 33 - Lock: Initial Screen Mockup

3.3.9 Settings Module Mockup

The **Settings Module** presents a list of options for the user to configure the software. It is possible to see an example of the graphical user interface in Figure 34.

Figure 34 - Settings: Initial Screen Mockup

This module allows to change the current language of the system.

The configurations of the 3G module of the Tablet PC include: the activation or deactivation of the module, as well the configuration of the APN in use.

In the email and Skype configurations it is possible to add or remove an account associated with the current user.

There is also the possibility to synchronize files with a USB pen, to manage users and others settings not yet predicted.

3.3.10 Notes Module Mockup

The **Notes Module** presents a list of the existing notes, the possibility to add new notes, to erase existing notes, attach images and send notes by email. It is possible to see an example of the graphical user interface in Figure 35.

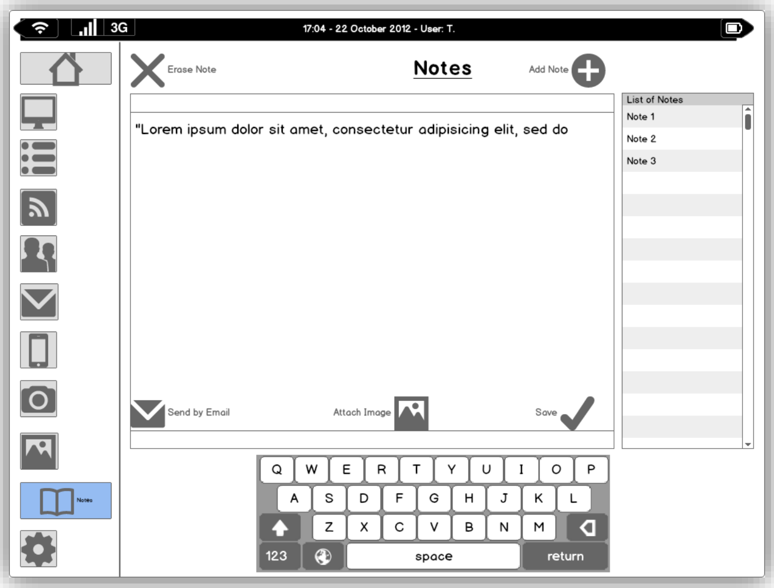

Figure 35 - Notes: Initial Screen Mockup

3.3.11 Events Module Mockup

The **Events Module** allows the user the possibility to record events that happen during his shift.

It is presented a list of recorded events, and two buttons, one to add a new event and another to erase the event. It is just necessary to write the name of the event and the other information are acquired automatically.

It is possible to see an example of the graphical user interface in Figure 36.

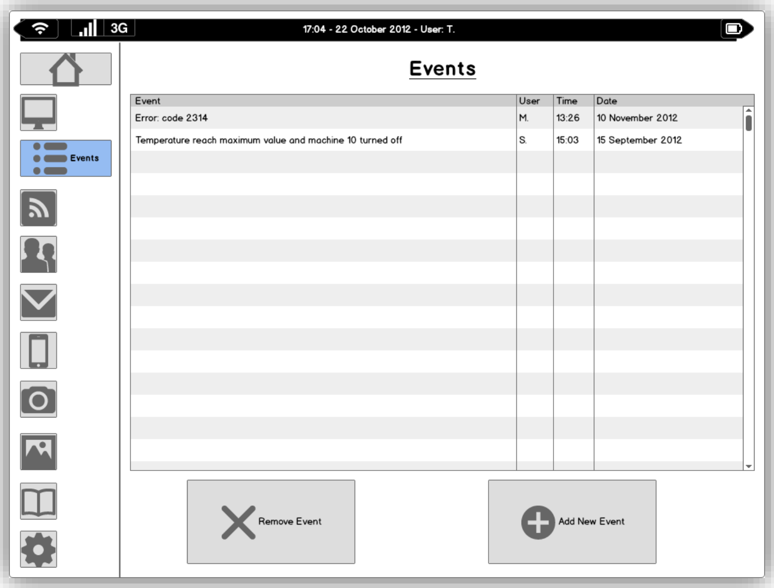

Figure 36 - Events: Initial Screen Mockup

3.3.12 Contacts Module Mockup

The **Contacts Module** shows a list with all the contacts, and gives possibility to add or to remove one. It is possible to see an example of the graphical user interface in Figure 37, where it is shown the list of contacts and two buttons that implement the mentioned functions.

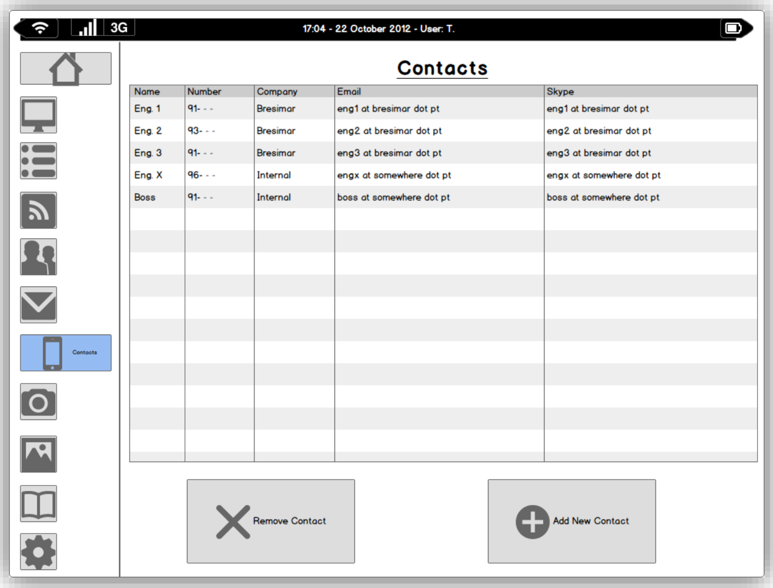

Figure 37 - Contacts: Initial Screen Mockup

The information stored consist on: name, number, email, and Skype name. This information is shared with different modules, for example, the **Technical Support Module** or **Email Module**.

3.4 System Architecture

This section presents an overview of the system architecture, describing both hardware and software architecture. The architecture presented is independent of specific underlying hardware or software framework chosen and it provides an overview of the system, without thinking in implementation details.

3.4.1 Hardware Architecture

The objective of this subsection is to define the hardware architecture, enumerating the different components, defining how they are interconnected and organized to create the final system.

The different components in use are: an embedded PC, a HMI panel, a Tablet PC, an access point and an IP camera.

The embedded PC is the PLC that controls the machine involved in the industrial process. The embedded PC has a specific architecture for industrial control. It contains I/O modules for interaction with the machine, as well a PC technology for the control algorithm. The PC technology underlying has a robust casing and a flash storage, in order to protect against physical shock, moisture, dust, extremely swings of temperature, etc.

The HMI panel enables the monitor and control of the process. The panel is fixed, usually associated with one embedded PC and one industrial process. It runs a customized software for interaction, which represents graphically the industrial process associated.

The Tablet PC is the hardware that contains the application for user interaction. It has an industrial casing and adequate robustness for an industrial environment.

The access point is the device that enables the wireless communication between the embedded PC and the Tablet PC. This communication is done using the protocol of the embedded PC over the WLAN created. The access point also provides the backing for new devices to be included in an industrial plant, for example, an IP camera.

The IP Camera serves as an example of how new devices can be inserted to provide new functions. In this case, the functionality is a live video of a machine where the operator does not have physical access.

In the Figure 38, a scheme of the architecture is provided. It is possible to see the referred devices, and how they are interconnected.

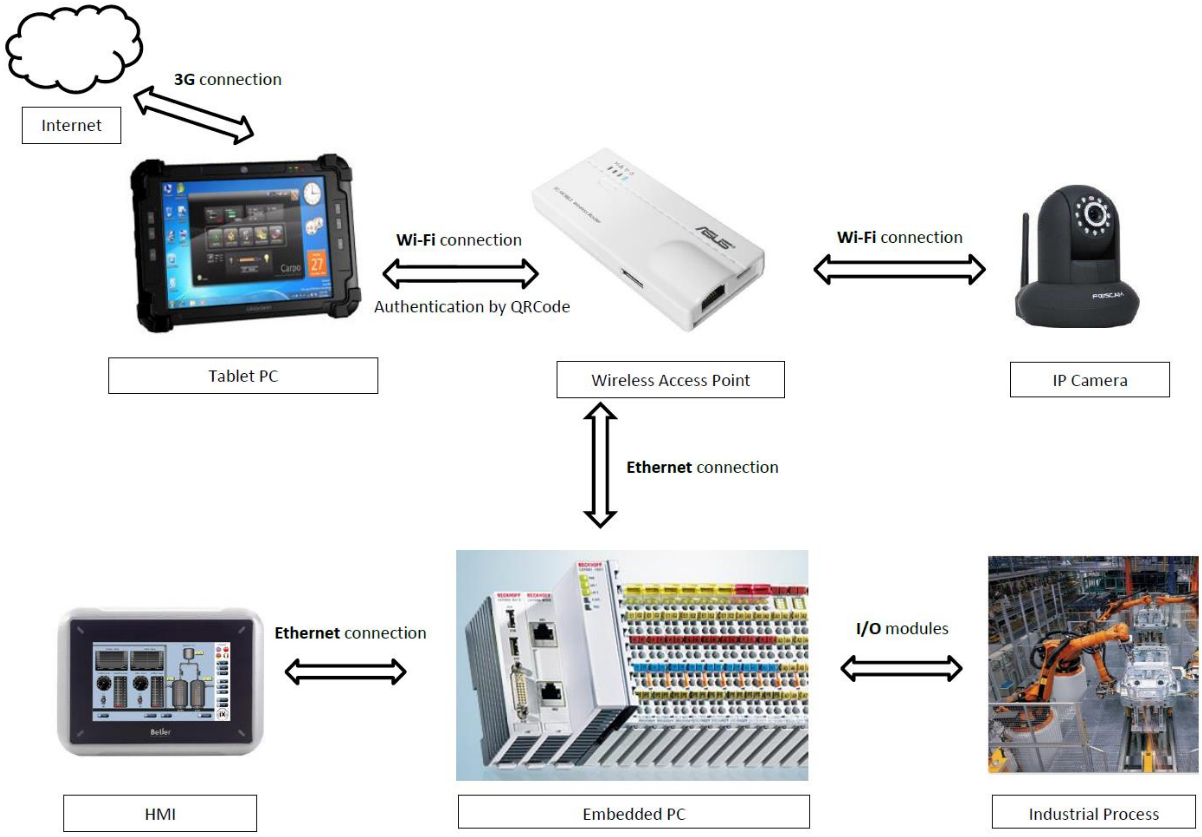

Figure 38 - Generic Hardware Architecture (Single System) [52], [53], [54], [55], [56], [57]

The embedded PC controls the industrial process, by reading inputs and changing outputs in the I/O modules, accordingly to the control algorithm loaded.

The HMI is connected to the embedded PC via an Ethernet connection, and displays the variables associated for controlling and monitoring of the industrial process.

The Tablet PC communicates with the embedded PC, via a connection to the wireless LAN created by the access point. The Tablet PC connects also to the internet with his 3G module.

The access point creates a wireless LAN in the industrial plant, and is connected to the embedded PC via an Ethernet connection.

The IP Camera is connected to the WLAN created by the access point, and can be accessed by devices in this network.

With the Figure 38 is possible to understand how each element communicates and is interconnected. However in an industrial plant, usually, there is not only one of these industrial systems, instead there are several. The real gain on inserting the mobile device is to have an architecture like the one presented in the Figure 39.

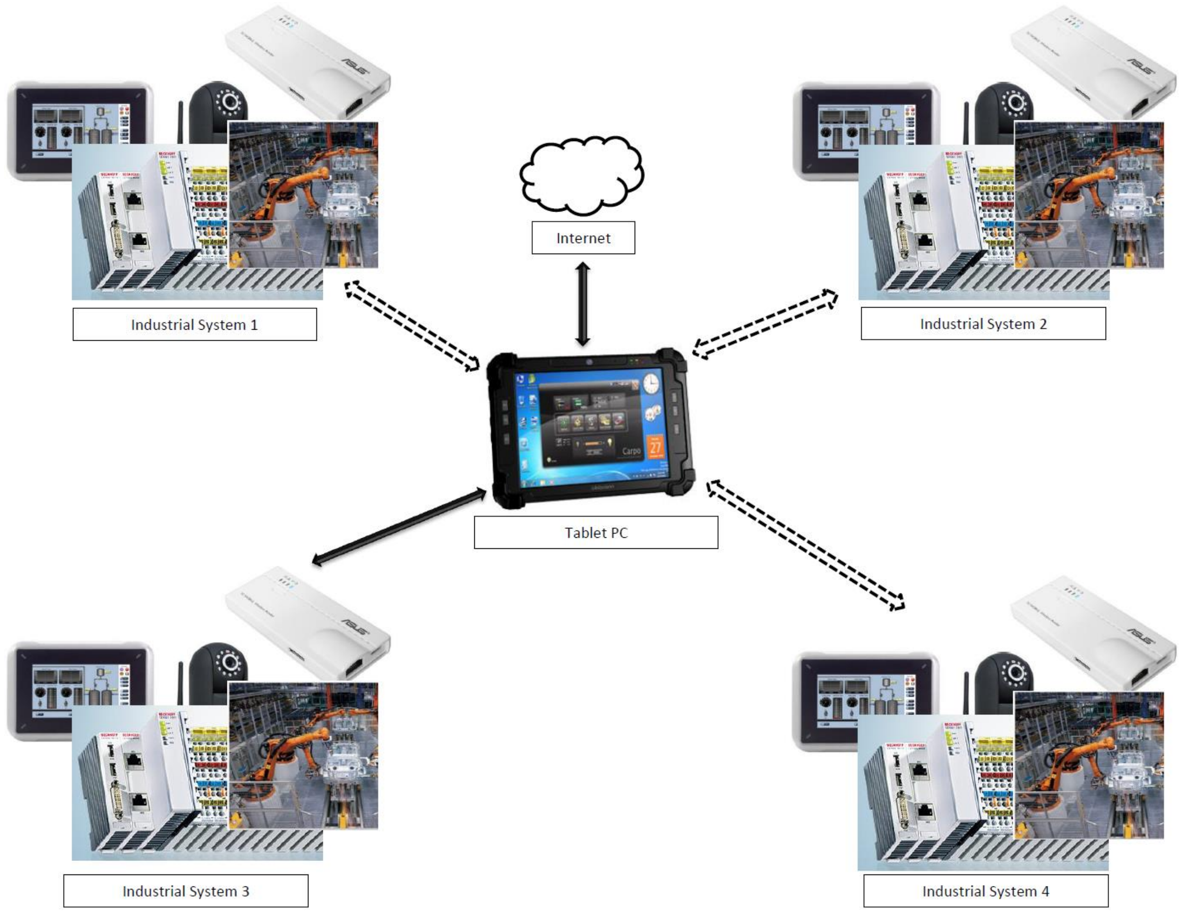

Figure 39 - Generic Hardware Architecture (Multiple Systems) [52], [53], [54], [55], [56], [57]

From the Figure 39 it is observed that the Tablet PC connects to several industrial systems, in order to have access to different industrial processes.

The advantage is the gain in mobility and the versatility of functions that can be associated to each industrial system.

These different industrial systems, mandatorily incorporate an industrial process, an embedded PC and an access point. The existence of these three components is the minimum requirement to integrate the Tablet PC. The industrial system can also contain a HMI panel and other devices, for example, an IP camera.

With the explanation of the hardware architecture is possible to see the fundamental integrating parts of an industrial system, and how the project adds value to the current setup. By adding a mobile device is possible to lower the cost of maintenance, since it is possible to provide access to several industrial systems with one Tablet PC instead of several dedicated HMI.

3.4.2 Software Architecture

The objective of this section is to define the software architecture, enumerating the different software components, defining how they are interconnected and organized to create the final system.

One of the main goals of this project is to guarantee a long life. Therefore, the architecture will be developed taking this aim in account.

It is also important, when developing the software architecture, to take in account the point of view of the final user and the software developer.

From the point of view of the final user because it will be the one that mostly use the product, where it expects a level of quality in its use from different criteria, such as, ease of use, reliability, security.

From the point of view of the software developer, because to add new functions or change the ones implemented, the developer needs to understand how it affects the system.

Taking in mind these points of view, it will be explained the software architecture, starting with an explanation of the structure chosen.

Architecture Diagram

For the software architecture a modular structure is chosen, so it is possible to add and remove modules, adding value to the product. If the client does not uses one functionality or wishes another, for the software developer should be simple to satisfy this needs.

In the Figure 40 is possible to see the major components of the software architecture.

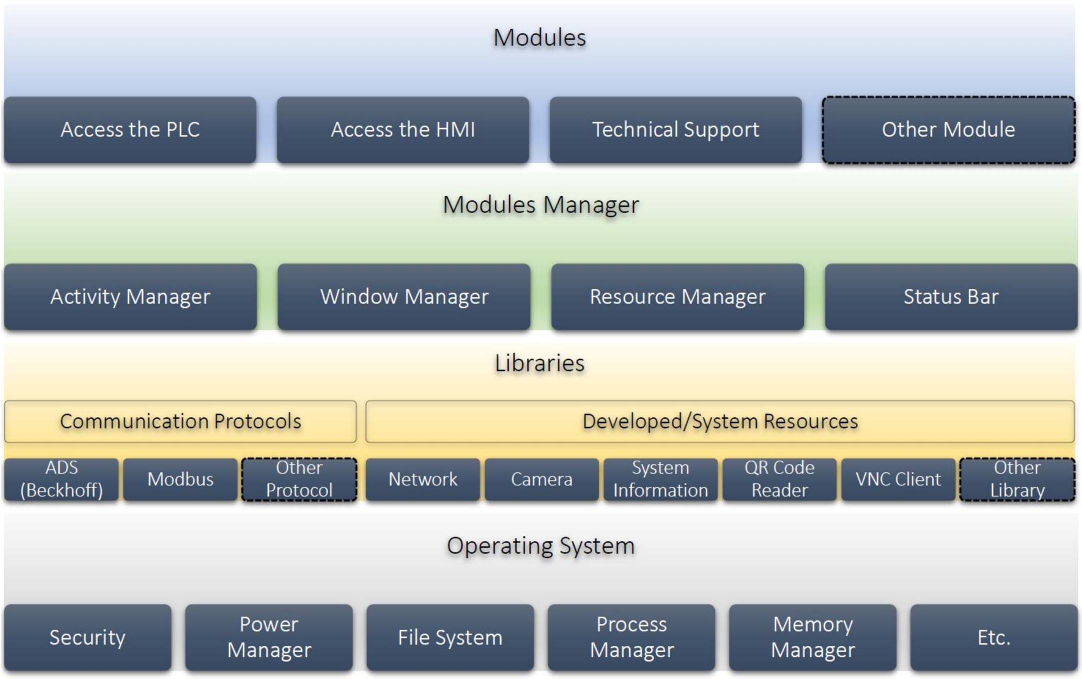

Figure 40 - Architecture Diagram

On top of the figure are the modules, which implement a set of functions that interact directly with the user. The important to notice is that another module can be easily added or an existing one can be removed.

Below is possible to see the **Modules Manager**, this is the most important part of the software, since it gives support for all applications. It creates the graphical interface for interaction with the software, and manages the activity of the different modules.

Under the libraries, which contain a set of different functions used by the different modules and the **Modules Manager**. A new library, like the modules, can be easily added or an existing one can be removed.

All the software runs on top of an operating system, which manages the hardware resources and provides common known services, as the ones show on the Figure 40.

Despite the software being only a system application, it is a project objective to hide the operating system from the user, so only the software created is accessible. This is done in order to provide a great ease of use, minimize run-time errors and prevent users to misuse the tools given.

The following subsection will present a class diagram description about the software in order to give a better insight on the software architecture.

Class Diagram

The class diagram presented on Figure 41 describes the object and information structures used by the software, here is possible to identify the different parts enumerated before in the Figure 40. A description of each different part of the figure will be made.

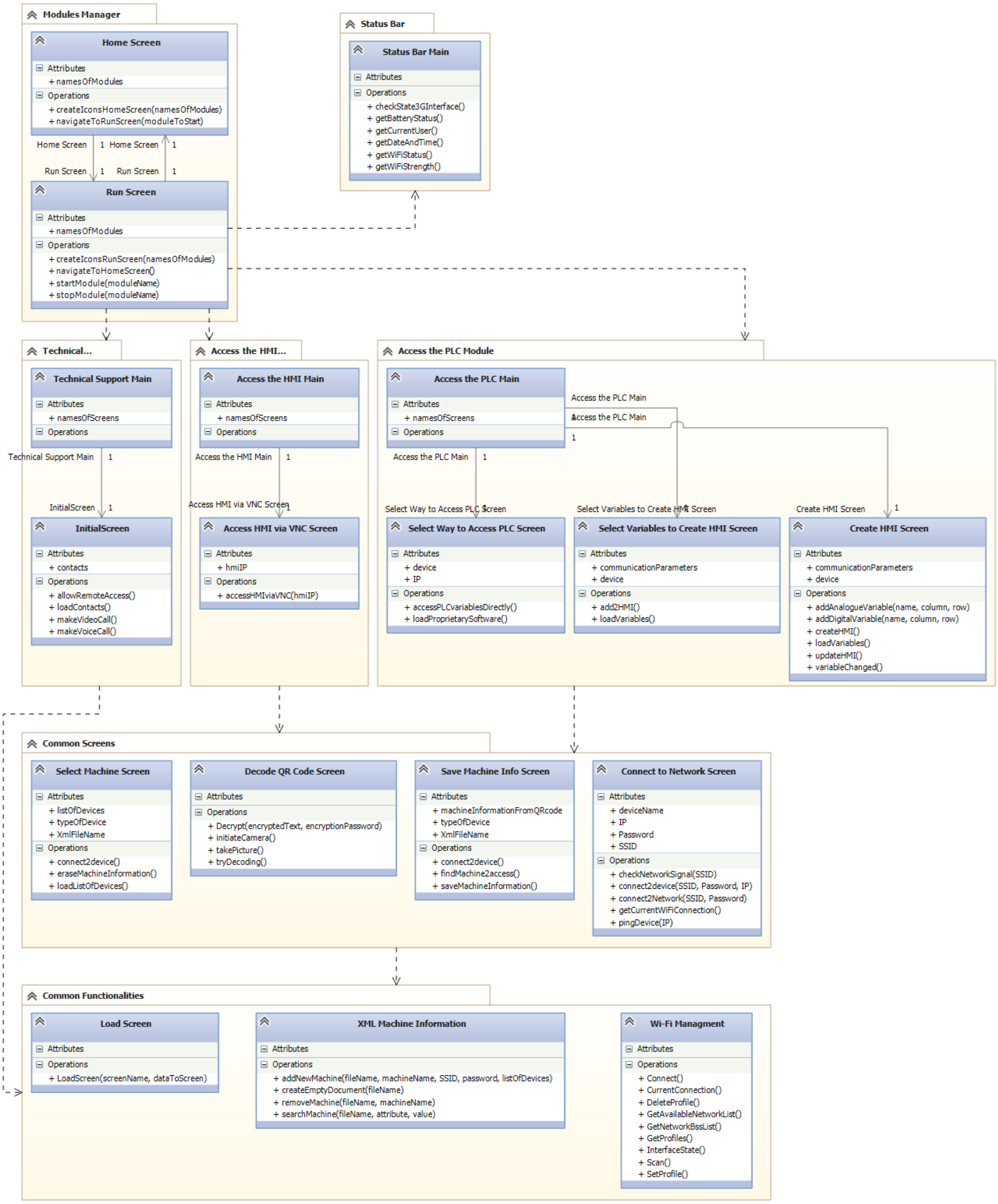

Figure 41 - Class Diagram

Modules Manager Class Diagram

On the top left corner of Figure 41 there is the **Modules Manager**, which contains the Home Screen and the Run Screen. The Home Screen creates the different icons of the modules loaded on the project, it also does the navigation to the Run Screen. The Run Screen loads and unloads the different modules loaded on the project. The Run Screen also creates the different icons of the modules loaded on the project on a panel on the left side, which also includes the icon to navigate back to the Home Screen.

Status bar Class Diagram

On the right top corner of Figure 41 there is the Status Bar, which displays useful information to the user. This status bar is presented on top of the screen where the user can read the presented information. The status bar obtains the information about the 3G interface, battery, current user, Wi-Fi, date and time.

Modules Class Diagram

On Figure 41, below the **Modules Manager** and the Status Bar, are the modules included in the project. It is possible to see that each module contains a main class, which is responsible to load the different screens of the application. The screens create the graphical user interface and interact directly with the user. These screens can be internal or external to the module.

The important thing to notice from the diagram is how each module integrates on the software. If another module is inserted the same pattern should be followed.

Technical Support Module Class Diagram

The **Technical Support Module** contains only a screen, the Initial Screen, and allows to grant remote access, load contacts and make a video or voice call to each of these contacts. The screen uses the common functions specified below the module as the Load Screen functionality.

Access the HMI Module Class Diagram

The **Access the HMI Module** contains one internal screen, the Access the HMI via VNC Screen, and four external screens, contained in the common screens.

The Select Machine Screen allows to display a list of devices, loaded from a file and filtered by type of device. It allows to erase this information of a machine and use the information to connect to a device.

The Decode QR Code Screen allows to use the camera to decode a QR code and to decrypt the information on it.

The Save Machine Info Screen allows to save the information decoded in the QR code, and to connect to a device using the read information.

The Connect to Network Screen allows to connect to a device, by connecting to the network where the device is located, and by checking the availability of the selected device.

The Access HMI via VNC Screen allows to access remotely to a HMI by using its IP address.

Access the PLC Module Class Diagram

The **Access the PLC Module** contains three internal screen and four external screens, contained in the common screens.

The screens contained in the common screens were already explained in the Access the HMI Module.

The Select Way to Access PLC Screen allows the user to select if he wants to access to the PLC variables directly or by launching the proprietary software.

The Select Variables to Create HMI Screen allows to load the variables of the PLC and to add which variables will be used to create a HMI.

The Create HMI Screen allows to create the HMI based on the variables previously selected. It loads the variables from the PLC and adds analogue and digital variables to create the HMI graphical interface. It also associates events to notice the software when the variables have changed in order to update the HMI graphical part.

Common Functions Class Diagram

Below the modules included in the software are a set of common functions, accessible through all the parts. This functions are used by more than one module and need to be independent of the rest of the software.

With the class diagram presented in Figure 41 is possible to understand better how the constituting parts of the software are organized, and how they interact between them.

3.5 Conclusion

In this chapter was defined clearly the problem tackled in this project and the solution proposed. With the requirements elicitation was possible to define a set of functions and uses case that the software should have.

The functions, use cases and requirements indicated were discussed in meetings with the team of engineers in Bresimar, in order to ensure a better coordination with the final vision of the product, and guarantee a viable project result.

The different components of the project were presented, as well how they are interconnected for the creation of the final system.

The hardware and software architecture were designed with a modularity concept, so it is possible to make changes demanded by different customers, creating a more competitive and complete product.

Chapter 4

SYSTEM DEVELOPMENT

This chapter will present the software framework and Tablet PC chosen for the project, as well the requirements that led to the choices. Next, the developed software in the scope of the project will be presented, including the application framework and architecture.

4.1 SOFTWARE FRAMEWORK

In the following subsections, the requirements analysis and reasons that lead to the choice of the software framework will be presented. Moreover, an overview of the selected framework is made.

4.1.1 Software Framework Requirements

Different software frameworks were evaluated while planning the project. Desirable properties were previously selected [58], [59], [60], leading to one understandable decision. The properties considered were the following:

- **Support**: A long term supported software framework is needed to ensure the longevity of the project. It should also be taken in account that the developed software will run on top of an operating system. So, both the framework and the OS need to offer: trusted warranty, long term compatibility and provide technical support;
- **Developer requirements:** The framework should be able to support the requirements previously stated for the project;
- **Usability**: The framework must improve the developer performance, enabling a reduction of work, which leads to a faster development;
- **Documentation and community:** The framework must have a well written documentation and a large community development associated. The project focus is to provide existing software features into the industrial automation field. It is not an objective of the project to develop features from scratch. Having access to a set of examples enables a faster project development.

4.1.2 Software Framework Chosen

With these criteria in mind the decision was to develop an application targeting the operating system Windows. The aim of the project is then to develop a windows-based application to run locally on the Tablet PC.

Due to the well-established market share of Microsoft operating system Windows, seen in the Figure 42 and in [61], as well the policy that the company assumed along the past years, is expected that warranty, compatibility and technical support are guaranteed on the software framework targeting this operating system.

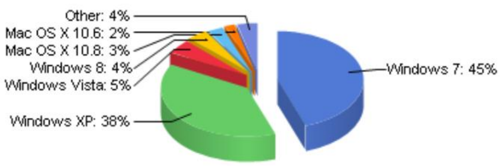

Figure 42 - Desktop Operating System Market Share (April 2013) [62]

The developer requirements can be fulfilled using .NET Framework [63], since is possible to design Windows client applications with great user experience, and have access to common operating system's functions such as network, documents and media management. The requirement to run proprietary software from the manufacturer in industrial automation in the Tablet PC is also satisfied, since they are primarily design to run in Microsoft Windows [64], [65], [66].

Usability is needed as the project has the requirement to be easily reconfigured, or updated, by persons not initially involved in the project. The requirement is met since the IDE used to develop .NET applications, Visual Studio, has a learning curve with a steep start. Another benefit is that the people who probably will be in future involved in the project, belong to the team of engineers in the company Bresimar, who develops Windows-based applications.

A well written documentation is provided by Microsoft [67], which attracts a large community of developers from long years of development in Desktop applications for Windows.

To conclude, the developed software will be an application running on top of the operating system, so the importance of the software framework chosen has a bigger weight to the software developer.

4.1.3 Overview of the .NET Framework

The .NET Framework is a development platform for building Windows applications. It consists of the common language runtime and the .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies. The .NET Framework provides a managed execution environment, simplified development and deployment, as well as integration with a variety of programming languages.

4.2 Tablet PC

In the following subsections, the requirements analysis and reasons that lead to the Tablet PC chosen for the project will be presented.

4.2.1 Tablet PC Requirements

For the choice of the Tablet PC, the requirements selected were the following:

- Operating System: the Tablet PC needs to run the operating system Windows, required due to the software framework chosen;
- Screen Size and Resolution: the Tablet PC needs to possess a touch screen with size of approximately 10 inch, and minimum resolution of 1024x768, in order to have a practical ease of use;
- Networks: the Tablet PC needs to connect to a wireless LAN, to communicate with the industrial devices, and to a mobile network, to have an internet access to provide the technical support functions on the industrial site;
- USB Interface: the Tablet PC needs to have, at least, one USB interface, in order to synchronize files of big size, for example, videos or manuals;
- Camera: the Tablet PC needs to have, at least, one camera, so it is possible to record machines information via the decoding of a QR Code, and provide technical support with a video call;
- Physical Robustness: the Tablet PC needs to have, at least, one integrity standard, in order to prove the adequate resistance to an industrial plant;
- Battery Life: the Tablet PC needs to have, at least, 5 hours of battery life, supposing the user will not use it more than 4 hours in a row;

- **Price:** the project has a commercial application, so the price should be evaluated accordingly to the various offers in the market since it will influence the final decision of the client;

4.2.2 Tablet PC Chosen

After a research of devices with these characteristics, in the Tablet PC market, the choice was the uTablet T10C, from the company Ubiqconn. A resume of the market research, with the main candidates, can be found in Appendix B -

Tablet PC Comparison.

In the Figure 43, it is presented a picture of the Tablet PC, and in the Table 35 its specifications [68].

Figure 43 - uTablet T10C [69]

Table 35 - uTablet T10C Specifications

Hardware	Intel Atom 1.6GHz Dual Core 2GB DDR3 32GB SDD	
Operating System	Windows 7 Professional	✓
Screen	10.4 inch 1024x768 Resistive Touch	✓ ✓
Networks	Wi-Fi IEEE 802.11 a/b/g/n 3.5G Bluetooth 4.0	✓ ✓
I/O interfaces	USB Ethernet Audio Jack	✓
Camera	Rear – 5MP w/ LED flash and autofocus Front - 2MP w/ audio input	✓
Physical Robustness	IP65 MIL STD 461F MIL STD 810G	✓
Battery	8 Hours 3800mAH, 7.4V	✓

It is possible to see from the list in Table 35 that the specified requirements (OS, Screen, Networks, I/O interfaces, Camera, Physical Robustness and Battery) are satisfied. The other characteristics of the Tablet PC can be found in the Appendix C - uTablet T10C Complete Specifications.

In the subsequent section, the developed software will be presented, targeting the software framework and Tablet PC chosen.

4.3 Software Development

In the following subsections, the application framework and architecture for the developed software will be presented.

Next, the implementation details for each part of the software will be shown. Starting with the Modules Manager, followed by the three core modules, the Technical Support Module, the Access the HMI Module and the Access the PLC Module.

4.3.1 Application Framework

The developed software will be a Windows Presentation Foundation (WPF) [70] application targeting the .NET Framework 4.0 [71].

WPF is a presentation system for building Windows client applications, which is included in the Microsoft .NET Framework, so it is possible to build applications that incorporate other elements of the .NET Framework class library.

WPF offers additional programming enhancements for Windows client application development with features that include Extensible Application Mark-up Language (XAML) [72], controls, data binding [73], layout, 2-D and 3-D graphics, animation, styles, templates, documents and media.

The application will be written in C# language, more specifically Visual C# [74]. Visual C# is a type-safe, and object-oriented programming language, which is an implementation of C# by Microsoft.

The integrated development environment (IDE) used will be Visual Studio 2012 [75]. This IDE from Microsoft is used to develop applications for all software platforms supported by Microsoft Windows.

4.3.2 Application Architecture

From the System Architecture, described in Chapter 3, is understood that is wanted an application with navigation ability [76]. The navigation application will host these standalone pages, which implement the different predicted functions to interact with the user.

Figure 44 presents the application architecture, and it is possible to see the internal structure of developed software. The figure is intended to explain how to design the software described in Chapter 3 - System Requirements and System Architecture using WPF technology.

Figure 44 represents, with white and black colour, the different software parts of the Modules Manager project: App, MainWindow, HomeScreenPage and RunScreenPage. Inside, with blue colour, it is possible to see the other projects hosted by the Modules Manager.

It will be explained what each part of the software in the figure represents, and what class they implement.

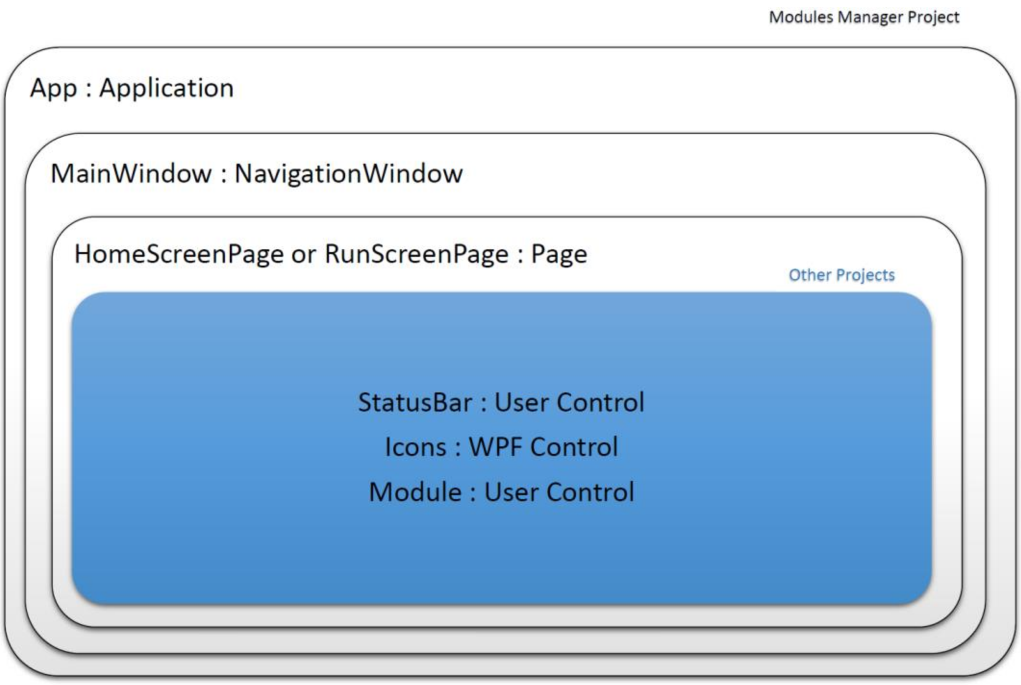

Figure 44 - Application Architecture Diagram

The App implements the Application class [77]. This class encapsulates WPF application-specific functions. These functions include the application lifetime, application scope, navigation, command-line parameter and exit code processing. The App starts the application and loads the MainWindow. The App also contains different resources used through all the application.

The MainWindow implements the NavigationWindow Class [78]. This class derives from Window Class [79] and extends it with the ability to navigate to and display content. In this project, the content is a .NET Framework object, more specifically a Page object. The pages hosted will be the HomeScreenPage or RunScreenPage.

The HomeScreenPage and RunScreenPage implement the Page Class [80]. This class encapsulates a page of content, which can be navigated to and hosted by a NavigationWindow. It manages a set of screens to be presented to the user. The first screen presents the icons of the modules loaded in the project. The second screen presents to the user the module running and a way to navigate back.

The HomeScreenPage and RunScreenPage contain WPF controls [81], which are common UI components used in Windows applications, such as Button, Label, etc., as well a set of User Control (UC) [82]. UC is a customizable control that can be reusable through the application. This component can enclose other controls, resources, and animation timelines, just like a WPF application.

The User Controls loaded can be the status bar, presented on top of the application, as well the different modules loaded by the project. This architecture fosters a project independence from the **Modules Manager** and these User Controls [83]. The solution will then contain other projects, created as UC, which implement the modules previously described.

The application developed in Visual Studio is then a unique solution composed by several projects [84]. Each project implements a different functionality of the software.

Figure 45 shows the organization of the solution in Visual Studio and is possible to see the main project, **Modules Manager**, in Bold, with the previously described components in Figure 44, and the other projects included, which implement the different software modules.

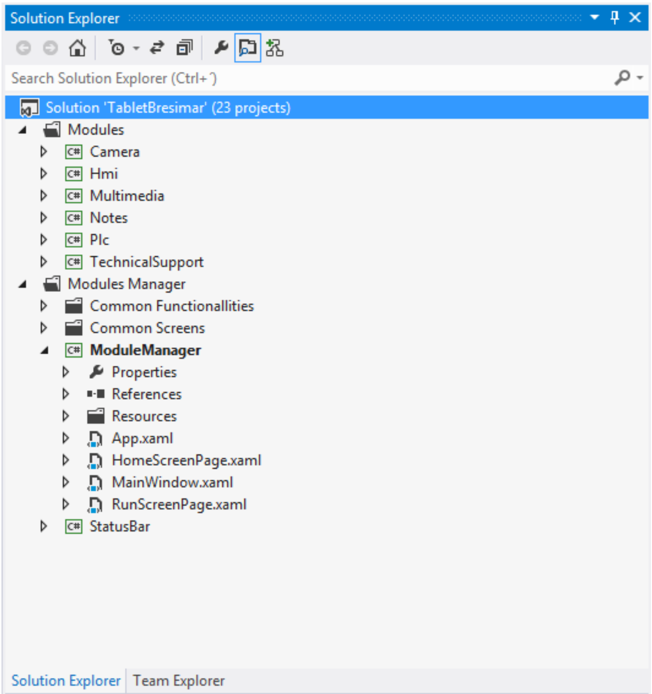

Figure 45 - Visual Studio Solution Explorer

It was presented that the RunScreenPage loads the different modules, and that these modules are defined in different projects of the solution as a User Control. It will be now shown how the different modules do the activity management inside the User Control.

The module architecture, presented in the Figure 46, relates to the User Control loaded and hosted by the RunScreenPage. It was developed to allow different interfaces with the user. These interfaces have the possibility to be internal or external to the module project.

It will be explained the representation of each part of the software in the figure, and which class implements it.

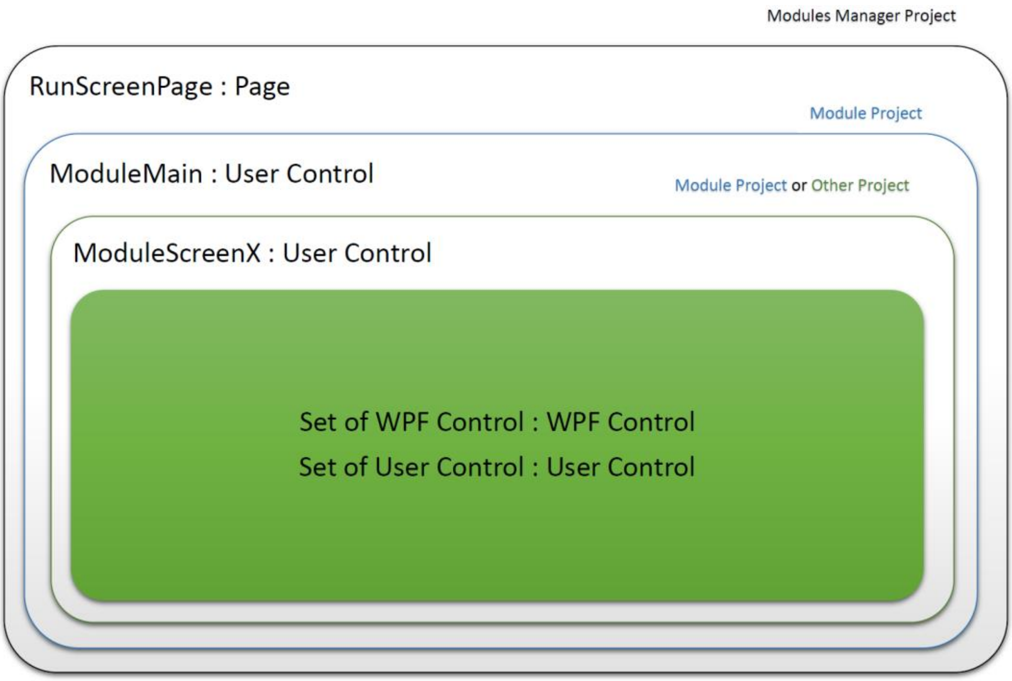

Figure 46 - Module Architecture Diagram

The RunScreenPage loads the ModuleMain which is implemented by the User Control Class. The User Control ModuleMain acts as a controller of User Controls. The objective is the following, the ModuleMain loads different User Control, like for example, the ModuleScreenX, which presents a user interface.

The ModuleScreenX is a User Control, which can be on the same project of the module, or on an external project. It contains a set of WPF controls, as well a set of user controls, to interact with the user. This UC represents a single screen to interact with the user, independently of the project. It can be re-used across different modules.

When the ModuleScreenX needs to load another User Control to replace it, triggers an event that is handled by ModuleMain. The ModuleMain proceeds to load a different module.

The ModuleScreenX has then to implement a delegate, and trigger an event, so it notifies the ModuleMain to replace it by another UC. This mechanism also enables to transmit data to the loaded user interface [85].

This architecture allows the existence of different screens to interact with the user, and these screens are independent of the module project.

In the next subsections the implementation of the **Modules Manager** will be presented, followed by the implementation of the modules with the highest priority in the scope of this project. These modules, as stated before, are: the **Technical Support Module**, the **Access the HMI Module** and the **Access the PLC Module**.

4.3.3 Modules Manager Implementation

The **Modules Manager** is responsible to launch the software and host the different modules developed. The implementation was divided into six phases, according to the functional requirements set in the software architecture. First, it was established a way to launch the application in full screen mode, when the operating system is started. Next, it was implemented a status bar to display on top of the application. Then the focus went to the implementation of the Home Screen and the Run Screen. Finally, the multilingual support was developed, as well a way to manage resources through the application.

FR-1: Full Screen on the OS launch

The functional requirement,

Table 15 - FR-1: Full Screen on the OS launch, established that the application should initiate in full screen mode, on the startup of the operating system.

The solution found, in Windows OS, was to use the "Local Group Policy Editor", to activate the option "Run these programs at user logon", and add a path to the executable of the developed software [86]. The option can be found in: "Computer Configuration/Administrative Templates/System/Logon" by running the command **gpedit.msc**.

For the application to initiate in full screen, it should be defined in Visual Studio that the Window of the application has the following properties:

- **WindowState="Maximized"**, so the application starts maximized;
- **WindowStyle="None"**, so the window does not have any borders, or buttons, which allow to resize or close the application.

The resolution of the screen of the Tablet PC should be taken in account, in order to use the available space to run the software in an efficient way.

FR-2: Status Bar

The functional requirement, Table 16 - FR-2: Status Bar, established that the software presents a status bar on top of the application, which should present the following information:

- Date and Time;
- Wireless Network Signal;
- Cellular Network Signal;
- Battery life;
- Supervisor Name;

The project Status Bar is defined as a WPF User Control, as stated before, this allows to reuse the component and keep it independent of the **Modules Manager** development.

To create the status bar it was used the StatusBar class [87], where each information item is a StatusBarItem [88]. It will be explained in the following subsections the implementation of each information item.

Date and Time

For Date and Time item it is defined on the graphical interface a StatusBarItem with a label as content.

The label is initialized with the current value of the date and time, which is updated every minute. This option introduces an error not bigger than one minute, regarding the date and time given by the Operating System.

The time value updating was done using the DispatcherTimer Class [89]. This class provides a timer integrated into the Dispatcher queue. The task is processed within a specified interval of time and with a specified priority.

Wireless Network Signal

To obtain the wireless network signal, it was used the Managed Wifi API[1] [90]. This library uses the Native Wifi API from Microsoft [91].

Graphically, it displays five bars to indicate the Wi-Fi signal strength [92], and a label to indicate the wireless network name to which the Tablet PC is currently connected.

The strength of the Wi-Fi signal is a value that goes from 0 to 100, where 0 is the minimum, and 100 is the maximum. This value colors the bars, in white or gray, each bar corresponding to a weight of 20.

The current Wi-Fi connection is a string containing the SSID of the current Wi-Fi connection [93].

The strength of the Wi-Fi signal and the current Wi-Fi connection are updated each 30 seconds, using the same method previously described for the Date and Time.

Cellular Network Signal

To obtain the cellular network signal the Network Interface Class from Microsoft was used [94].

It was developed a function to check the state of a network interface, by returning the value **True** if the operational state is “Up”.

---

[1] I would like to show my gratitude to ikonst, Kaalen and NN, as well to other people involved on the development and sharing of Managed Wi-Fi API

The state of 3G module interface is checked every 5 minutes and it is represented through a label with the content “3G”. The color of the label represents the state of the connection, being white if the state of the interface is “Up” and grey if the state of the interface is “Down”.

Battery life

For the graphical part of the battery life information, were defined two rectangles, which with the right size and position create the icon of a battery. Inside this icon, it is presented a label with the state of the battery life, more specifically the total time for the battery to discharge.

To get the battery state was used the information on the Microsoft web site from the Win32Class [95] and the example found in [96].

A function was developed where is possible to get the total time of discharge of the battery, or in case of that not being possible an empty string. Was not created more battery states information, in order to keep the status bar simple and language independent.

Summarizing, if the battery is in discharge the remaining time is asked, converted to hours and minutes, and the result sent on a string. The information is updated every 5 minutes.

Supervisor Name

The supervisor name is the name of the user that has a session initiated in Windows. To obtain this information the library mscorlib was used [97]. Then is defined, inside a StatusBarItem, a label where is set the content as the variable username.

Final Result

As the final result the status bar obtained is shown in Figure 47, where it is possible to see the items and how they are presented to the user.

Figure 47 - Status Bar: Implementation

To use it in the application it is just needed to add, after reference it, the User Control like a common UI component of WPF.

The component has the size of 1024x50, to fit the top of the screen of the Tablet PC with resolution 1024x768.

FR-3: Home Screen

As functional requirement, Table 17 - FR-3: Home Screen, was established that the application creates a set of icons, allowing the user to access the different modules loaded on the project.

The structure of the project was already explained, the project has a Main Window that implements the NavigationWindow Class and has as source the HomeScreenPage. The HomeScreenPage is the initial page presented to the user and it implements the Page class. It creates a screen with a set icons of the different modules loaded on the project. It is responsible to navigate to the RunScreenPage when an icon is clicked and to send the information of which module is to load.

Regarding the graphical part of the HomeScreenPage, the Status Bar created is defined on top, and a Grid 4x3 with the modules icons, filling the remaining space automatically. In the Figure 48 is shown the dimensions of the HomeScreenPage graphical part.

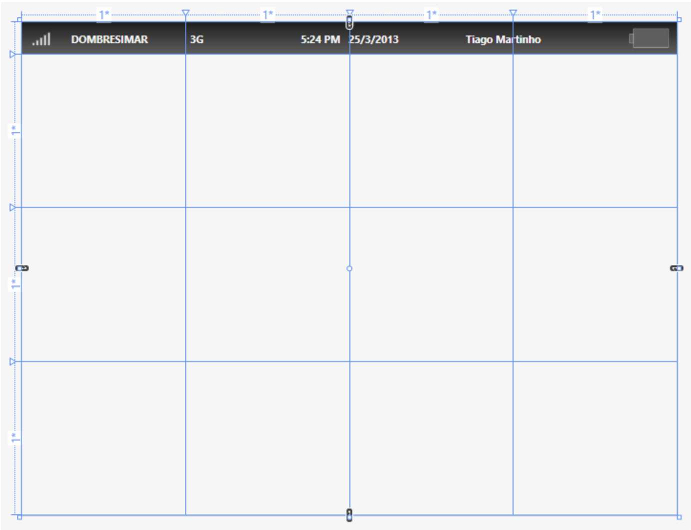

Figure 48 - Home Screen Page: Dimensions

The icons of the modules are placed in run-time using the function: createIconsHomeScreen. It receives the name of the modules in the project. The function fills the grid, from left to the right, till the grid is completed, or the modules are finished.

On the creation of the icons it is defined what happens when the icon is clicked. In this case it initiates the navigation to the RunScreenPage and passes the information of which module is to load.

If the user presses the Lock icon the **Modules Manager** does not run a module. Instead it secures the Tablet PC and put it in a low power mode consumption. To do so it was used the method SetSuspendState from Microsoft [98], associated with the icon of the module in the HomeScreenPage. To gain access again to the Tablet PC the user has to press the power button of the Tablet PC and enter the Windows OS credentials.

The image of the icon is an image with the same name of the module in the folder Resources/Images/Icons/ with the extension.png. The name that appears below the icon is a string that depends on the language selected. The icons used were taken from [99].

The final result obtained is shown in the Figure 49, where it is possible to see the Status Bar on top, as well the set of icons that initiate the navigation to the RunScreenPage and launch the according module.

Figure 49 - Home Screen Page: Implementation

FR-4: Run Screen

As functional requirement, Table 18 - FR-4: Run Screen, was established that when a module is started, a panel with the icons on the left side is created, and a blank space for the module to run is reserved on the right side.

It was also established that when an icon in the left panel created is clicked, the current module is terminated and a new one initiated. Clicking in the Back Home button in the left panel brings the user to the initial page.

The **Modules Manager** is then responsible for starting and stopping the module, more specifically, to load and unload the different user control representing the modules.

In the Figure 50 it is shown the dimensions of the RunScreenPage graphical part. It has the resolution of 1024x768, with the status bar filling the in the top 1024x50, and a Grid 3x2 filling the remaining space, with the dimensions presented in the figure.

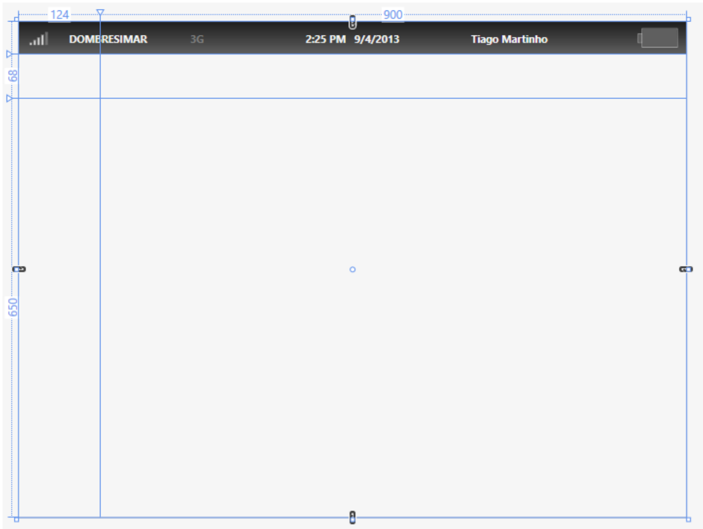

Figure 50 - Run Screen Page: Dimensions

On the first column and second row of the grid it is defined a button to come back to the HomeScreenPage. On the first column and third row it is defined a stack panel with the icons of the loaded modules. On the second column and second row it is defined a label with the title of the loaded module, below that a space for the module to run. The available space for the module to run is of 900x650, so the modules have to be developed according to this size.

To create the icons on the stack panel, it was used the same reasoning used in the HomeScreenPage. The actions associated with the buttons are that when clicking on the back button the user navigates to the HomeScreenPage. When clicking the icons on the stack panel on the left, the module that is running is unloaded, and the one which icon was clicked is loaded.

The final result obtained is presented in Figure 51. In here it is possible to see the status bar on top, the back button on the left upper corner, and the set of icons of the modules loaded on the project on the left. It is also possible to see the title of the module loaded in the project, as well the space reserved where the module runs.

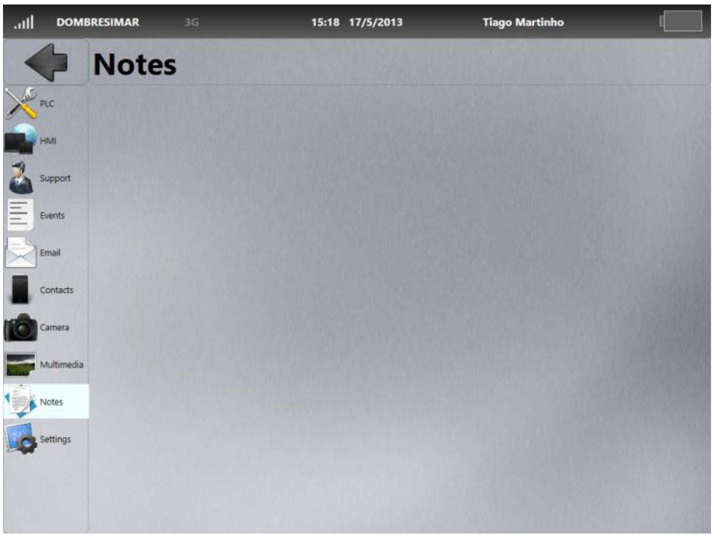

Figure 51 - Run Screen Page: Implementation

FR-5: Multilingual Support

As functional a requirement, Table 19 - FR-5: Multilingual Support, established that the software should have two changeable languages and support for the insertion of a new language.

It was decided that the strings used in the project are all be grouped in a file. This enables to change the language by changing the file from where the strings are loaded. It also enables to insert a new language, by inserting a new file with the according translations.

Following the references [100] and [101], it was acknowledged that this is an effort that should start on the beginning of the project. The effort of doing so is small, comparing to adding this functionality on the end. It is also acknowledged that the graphical interface should take into account this aspect, because in different languages the same message will have different lengths.

In the chosen solution it was created a folder with the name Resources, and inside a folder with the name Languages. Here in it were added two files of type XAML Resource Dictionary [102]. The files have the names English and Portuguese, one for the English language, and another for the Portuguese language.

The Resource Dictionary class provides a hash table/dictionary implementation that can be used to store data like, for example, languages and themes of the application.

The files have the content based on the example from Microsoft in [103]. The file contains the strings to use in the application with a key that identifies them.

To access this resource through all the application it is still needed to add the resource reference in App.xaml of the **Modules Manager**. After that, it is possible to access the strings in the graphical interface or programmatically. It is also possible to change the file where the strings are being loaded.

With these steps done the requirement is fulfilled: having two easily mutable languages, Portuguese and English, and a simple way to add a new language, by creating a new XAML and translating the value of the strings.

FR-6: Resource Management

The functional requirement, Table 20 - FR-6: Resource Management, established that there is information transversal to the different modules. This information is stored and provided by the **Modules Manager**.

This is done in two ways in the application, one is to have a folder with images used by different modules. This folder contains the icons of the modules and different images used along the application.

Another way is by defining a Resource Dictionary file in the folder resources, and to refer the resource on the App.xaml. This was previously explained for the language files, but is also used to have different themes for the application. The application accesses a file that contains a set of style definitions to override the look and feel of some WPF controls.

4.3.4 Technical Support Module Implementation

The **Technical Support Module** allows the operator to receive technical support from the Tablet PC. With this support the operator is able to make a video call or voice call in the industrial plant. This call enables to troubleshoot problems with an expert. This support can be also provided by having remote access to the Tablet PC. This access allows to reprogram or reconfigure remotely the machines in the industrial plant.

Resuming, this module will allow the user to do three main functions:

- Make a voice call;
- Make a video call;
- Grant remote access to reprogram the machines.

The video call and voice call are established using the proprietary software Skype. It is wanted that the operations are mostly automated. Operations as the login in Skype, as well the loading of the user contacts. It is also needed that the Skype and the developed software, switch focus automatically. The idea is to use Skype as an integrated component of the software, and not as a standalone application.

To allow remote access to the Tablet PC it was used two approaches. One uses the proprietary software TeamViewer, the other uses the Windows Desktop Sharing application.

The implementation was divided into two phases, according to the functions of making a call and granting remote access. Each implementation phase will be now explained.

Make a video call or voice call

The current method to start calls in the Skype client, publicized on Skype Developers [104], does not allow to have great control over the software. In Windows OS, the user has to do a manual login and it does not allow to regain focus on the developed software.

The decision was then to use the Skype application programming interface (API) for Windows [105]. This API allows to establish a communication channel with the Skype client, so it is possible to communicate with it through a different set of commands.

To facilitate the communication it was used the wrapper library Skype4COM [106], instead of using the text based command protocol. Skype4COM is an ActiveX component that represents the Skype API as objects. The component encapsulates the communication layer and protocol layer commands.

The implementation follows the next reasoning: when the module is loaded, it is verified if the software Skype is already running and if not it is started. Then the client application is attached with Skype. In the process it is verified if the user is logged in, and if not it asks the credentials. The module loads the contacts associated with the user and minimizes the Skype application.

After these initial steps the user can make a video or voice call to the list of contacts loaded. When the call is made the window of Skype is maximized and when the call finishes the window is minimized.

The implemented user interface is presented in Figure 52. It is possible to see a list of contacts to use in Skype. The two buttons on the left, allow to navigate up and down through the list. The two buttons on the right, allow to make a video call or a voice call.

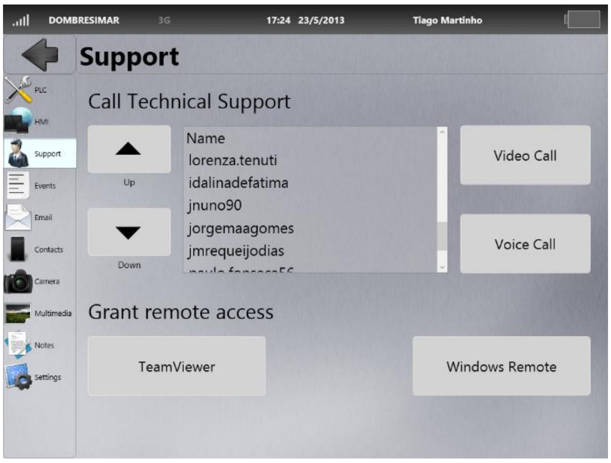

Figure 52 - Technical Support: Initial Screen Implementation

In order to improve the user experience some considerations were done. First the look and feel of the application was implemented in order to be touch friendly. It was also implemented a mechanism that appears in some mobile operating systems: the scroll bar of the applications list, when is touched it appears and disappears after a while. Two buttons to scroll up and down through the list of machines were provided in order to give better accuracy with simple interaction.

The user interface when a video call is established is the Skype application and it is presented in Figure 53.

Figure 53 - Technical Support: Video Call Implementation

When the call uses only voice, the application shows only the buttons from Skype application. It is possible to see in Figure 54 the user interface, where the difference from the initial screen is the Skype interface in the right upper corner.

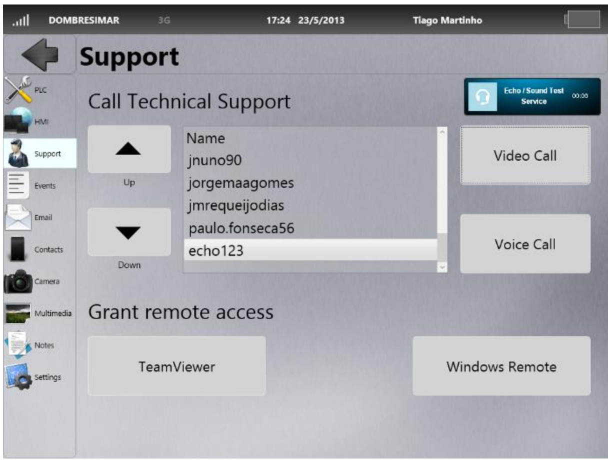

Figure 54 - Technical Support: Voice Call Implementation

Remote Access

To grant remote access to the Tablet PC was used the proprietary software TeamViewer. This choice was implemented, once Bresimar uses and has a license for the software. This software allows to grant remote access, and at the same time, among other functions, provide a way to make a video call.

To launch the external software it was used the ProcessStartInfo [107] together with the Process component [108]. These two classes enable to start and stop local system processes, allowing also to specify a set of values.

The software launched is a personalized version of TeamViewer QuickSupport [109]. The version to launch depends upon the current selected language of the developed software. The user interface is the initial screen with the software TeamViewer running on top of it.

This functionality should be used during a voice call, so it is possible to give the person to access remotely the accessing parameters. It is possible to see in Figure 55 the user interface. The language of the software depends upon the current language of the developed software.

Figure 55 - Technical Support: Remote Access Implementation

The second way to grant remote access is by using the desktop sharing application from Windows OS [110]. This tool is provided in the System Services of the Windows OS [111]. The choice to implement this functionality arose, because it creates a native way to grant remote access in the application developed, and it is a feature that comes with Windows OS.

To implement this functionality, it was used the information from Microsoft found in [112] and in [113]. It was also found an example of an implementation described by Microsoft the WinPresenterFinal[2] [114].

When the user presses the button with the "Windows Remote" description a remote access is configured. This is done by starting a Remote Desktop Session. From here the user has to give the Windows credentials to the person doing the remote access.

The Remote Desktop Connection comes by default with Windows OS and a demo can be found in [115]. The application can only connect to computers running the Professional, Ultimate, or Enterprise editions from the OS, a requirement satisfied by the Tablet PC chosen.

4.3.5 Access the HMI Module Implementation

The **Access the HMI Module** allows the operator to access a HMI panel remotely. With this access, the operator can monitor and control the industrial process by interacting with the HMI.

The remote access is made with Virtual Network Computing (VNC) [116] since most HMIs currently available in the market have VNC capability [117], [118], [119], [120].

To access the HMI via VNC two conditions must be met: the Tablet PC must be connected on the same network of the HMI, and the HMI IP address must be known. Since the objective is to create something simple to use, the access information is encoded on a QR code. This feature makes the interaction more automated, since it is not needed to ask that information to the user.

Using the rear camera a QR code is decoded, with this information the connection to networks and VNC access are done automatically. The read information of a machine is saved for future access once a decoding is made.

Resuming this module will allow the user to do two main functions:

- Access a HMI panel via VNC;
- Record machines information by decoding of a QR code;

The development of this module implements a set of common functions. For example, the access to the camera of the Tablet PC for decoding a QR code. These common functions were created independently of this module and re-used by other modules.

To simplify the implementation, and provide greater flexibility, the interaction with the user was divided into five screens. It was created five different user controls, which are loaded according to the user interaction. The implementation of each user control will be now explained.

Select Machine Screen

The "Select Machine Screen" is the screen presented to the user when the module is initially loaded. In this screen it is presented a list of machines previously recorded, and the possibility to read a QR code to record and access a machine.

The screen implemented is presented in Figure 56. It shows a list containing the information of the recorded machines. The two buttons, on the left, allow the navigation up and down through the list. The two buttons, on the right, allow to access the HMI and erase the information of a machine. Below, it is placed a button with a QR code image. This button is used to record and access the machines, via the decoding of a QR code.

[2] I would like to show my gratitude to termserv for the development and sharing of WinPresenterFinal

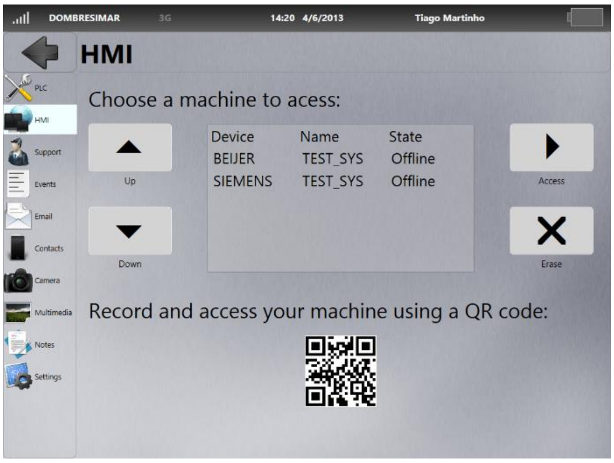

Figure 56 - Access the HMI: Select Machine Screen Implementation

The information of the recorded machines is loaded from an XML file. It will be explained in the implementation of the Save Machine Information Screen how this information is saved and structured.

To present the information to the user it was used ObservableCollection<T> Class [121]. This class embodies a dynamic data collection. It provides notifications that allow to know when items get added, removed, or when the whole list is refreshed. Using this class it is possible to bind the data to the graphical part, in this case the data was bind to a list view [122], following the example provided in [123] and in [124].

This allows to add new elements to the list which are automatically updated in the graphical interface. The advantage is, for example, to update the state of the machine programmatically, and it updates automatically the graphical part. It is also possible to receive information directly from the graphical interface, what allows to know which item is selected from the list and access that information directly.

The state of the machine is obtained by checking their connectivity. It was used the Ping Class [128] from the System.Net.NetworkInformation Namespace [129]. Each device is tested in background using the BackgroundWorker Class [125]. This class allows to execute an operation on a separate thread, which makes it possible to test each machine connectivity, without interfering with the graphical interface.

Decode QR Code Screen

In the "Decode QR Code Screen", Figure 57, it is presented to the user the image provided by the rear camera of the Tablet PC, as well the information to scan the QR code.

It decodes the QR code automatically, as soon as a QR Code is detected on the camera image. This mechanism was provided in order to simplify the decoding action.

Figure 57 - Access the HMI: Decode QR Code Screen Implementation

To read a QR code it was needed to use the camera to obtain an image from it. For getting access to the camera it was used the library directshow.net [126] because of its simplicity and proof of robustness.

To decode the QR code it was used a port for .NET Framework of Zxing, ZXing.Net[3] [127]. Zxing[4] [128] is an open source multi-format 1D/2D barcode image processing library implemented in Java.

The information encoded in the QR code, regarding the machine, has the following structure:

NetworkOfMachine31#PasswordOfTheNetwork#PLC_BECKHOFF#192.168.1.3#PLC_SIEMENS#192.168.1.7#HMI_BEIJER#192.168.1.4#HMI_SIEMENS#192.168.1.8.

The fields stored are separated by hashes in order to process the information easily in the software. The first field contains the SSID of the network created by the access point connected to the machine, NetworkOfMachine31. The next field is the password to this network, PasswordOfTheNetwork.

Then there are the list of devices connected in the network. Starts with the type and name of the device, PLC_BECKHOFF, followed by the IP of the device, 192.168.1.3. This pattern will be repeated for all the devices in that network. It is possible to see that the example system contains: two PLC's, one of Beckhoff and another of Siemens, and two HMI's one of Beijer and another of Siemens.

The information in the QR code was encrypted using the System.Security.Cryptography Namespace [129] and following the example in [130]. This mechanism was implemented in order to prevent the users to misuse the system.

To encrypt the QR code information it was developed the software in Figure 58. On the left side is possible to see the information to encrypt, and on the right side the information encrypted that will be present in the QR code.

[3] I would like to show my gratitude to micjahn and fabianhenzler, as well to other people involved on the development and sharing of the port ZXing.Net

[4] I would like to show my gratitude to Sean Owen and Daniel Switkin, as well to other people involved on the development and sharing of the Zxing library

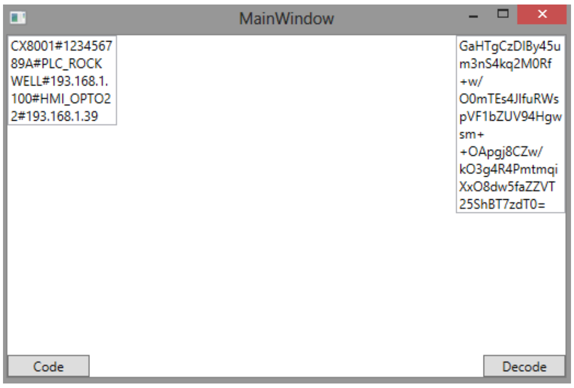

Figure 58 - Encrypt Information Software

Save Machine Information Screen

The "Save Machine Information Screen" presents the option to save the decoded information, for future access. Here the user has the option to give a name to the information of a machine to save. It is also possible to skip this step by hitting the cancel button, in this case, the information of a machine is not stored.

The screen implemented is presented in Figure 59. It is possible to see a textbox, where the name of the machine can be written, using the virtual keyboard, as well two buttons, one for saving the information and another for cancel the action.

Figure 59 - Access the HMI: Save Machine Information Screen Implementation

The machines information is composed by the following items:

- **Name of the machine:** asked to the user;
- **SSID of the wireless network:** created by the access point connected to the machine;
- **Password of the wireless network:** created by the access point connected to the machine;
- **List of Machines:** with the information of all the devices connected to the network, including the name of the device and IP address.

This information is stored in an XML file. This method suits the situation because XML is a standard easily transposed to other solutions.

To access and manipulate this data some functions are needed to interact with an XML file:

- To create an XML file, in case of the first run of the software;
- To add a new entry, whenever a new machine is recorded;
- To remove an entry, if a user wants to remove a machine recorded;

- To search within an XML file, to verify if a machine already exists in the system;

These functions were developed based on the several examples by Microsoft in [131], [132], as well the examples in [133], [134].

The information stored is structured in this way, each indentation is a lower level in the XML file:

List of Machines

|---------Machine

    |---------*Name*

    |---------*SSID*

    |---------*Password*

    |---------*List of Devices*

        |---------Device

            |---------*Name*

            |---------*IP address*

        |---------Device

            |---------*Name*

            |---------*IP address*

        |---------...

|---------Machine

    |---------*Name*

    |---------*SSID*

    |---------*Password*

    |---------*List of Devices*

        |---------Device

            |---------*...*

|---------Machine

    |---------*...*

|---------...

It is possible to see that, with this structure, the number of devices associated with a machine can grow indefinitely.

When using the **Access the HMI Module**, the information that matters from this list, is the information filtered to a single HMI device. That is, the information separated by Machine Name, SSID, Password, Device Name and Device IP. The reasoning is to display the information easily to the user within a list, as well to access directly each device with it.

For this purpose it was created an infoMachine class to store and access easily these parameters. The information is loaded and filtered from the XML file in order to display a list corresponding to devices recorded. The advantage is that information can be associated with a list to display graphically to the user. It was also added one more field, not present in the XML file: the state of the device. This state can be online or offline as the connectivity of the device is tested.

The information is structured as it is presented in Table 36.

Table 36 - infoMachine Class

Device
+ Machine Name: string
+ SSID: string
+ Password: string
+ Device Name: string
+ Device IP address: string
+ State of Device: string

This information is associated with the list of machines in the Select Machine Screen.

Connect to Network Screen

In the "Connect to Network Screen", Figure 60, a connection to the machine is made and a progress bar indicating the status is shown. In this screen the Tablet PC connects to the network where the machine is, and is verified if the machine is reachable within that network.

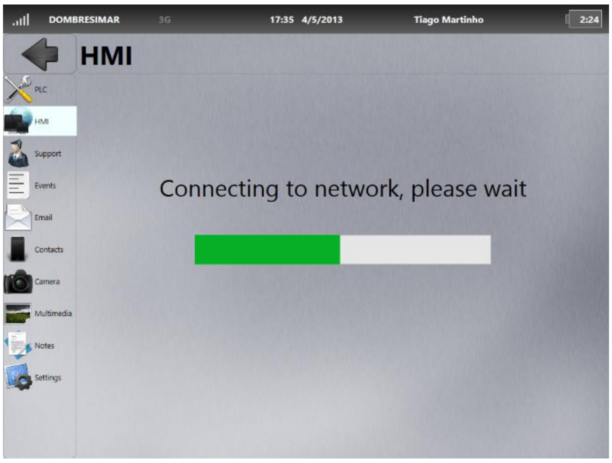

Figure 60 - Access the HMI: Connect to Network Screen Implementation

To control the Wi-Fi network adapter the Managed Wifi API was used again. This management has to be made within the application, since the user does not have access to the operating system, and so he cannot control to which network to connect.

In this step it is needed to: check if the Tablet PC is connected, do a broadcast to detect the available networks and to connect to a network.

To connect to a chosen wireless network, which was not previously saved by the operating system, it is needed to create a Wireless Profile for the wireless configuration of the network [135]. The profile defined was a WPA2-Personal based on the example from Microsoft in [136].

To check the availability of the HMI device it was again used the Ping Class [137]. Even if the Tablet PC is connected to the right network, it is also needed to guarantee that the HMI is connected and available for access.

If the Tablet PC is already connected to the network, and the machine available, this screen is jumped.

Access the HMI via VNC Screen

The “Access the HMI via VNC Screen”, allows automatic access the HMI via VNC protocol, using the parameters decoded from the QR code.

The screen implemented is presented in Figure 61 and it shows a reserved area of 640x480 pixels to interact with the HMI by remote access, and a button on top to go back to the “Select Machine Screen”.

Figure 61 - Access the HMI: Access the HMI via VNC Screen Implementation

The library VncSharp[5] [138] was used to connect directly using a VNC client. A Windows Form Host was used to host the control VncSharp [139], since the control was not made for WPF technology but for Windows Form technology [140].

User interactions

In this section is possible to understand how the different screens interrelate, according to the user interaction.

The first time the module is initiated the software does not have any information regarding the machines to access. The only option to the user is to record a machine, by decoding a QR code. The software will then have the parameters to establish a connection.

In Figure 62 it is presented the interaction with the user. If the user press the Access or Erase button, a message appears to say he still needs to record the machines. If the user press the button with a QR Code image, he has the possibility to read a QR code. With the information decoded he can access the HMI via VNC.

[5] I would like to show my gratitude to David Humphrey, as well to other people involved on the development and sharing of the VncSharp library

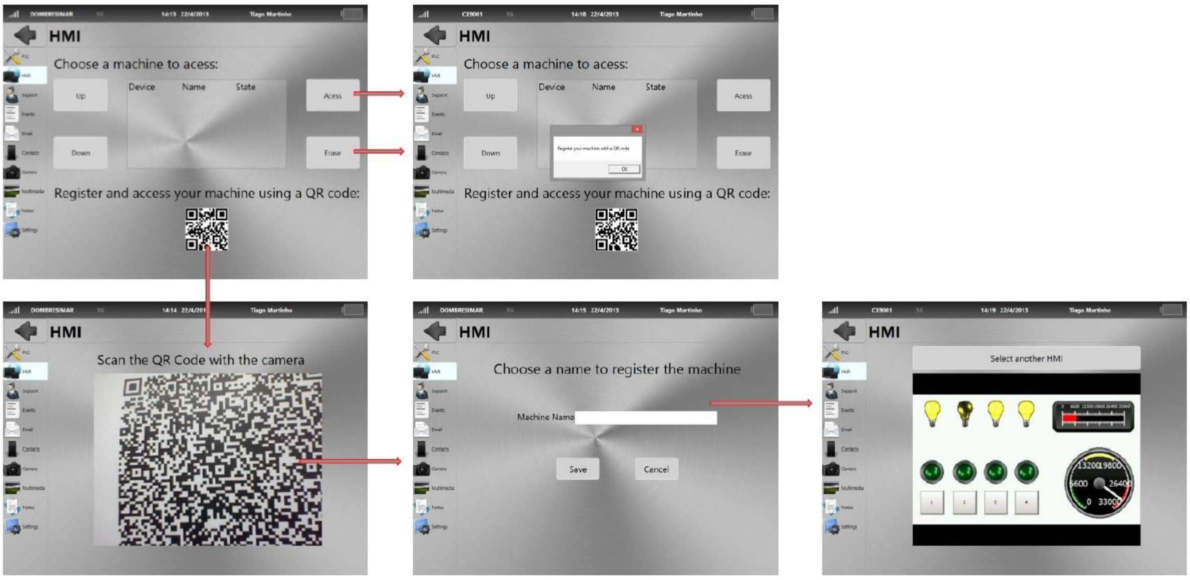

Figure 62 - Access the HMI: User Interaction 1

After the user records the machines he can also access the HMI by choosing from a list. Previously it was not referred of the possibility of the communication to fail. This can happen in two ways: first if the network is not active or not in range, and second if the machine is not active. Figure 63 presents the interaction with the user, and also a case where the communication fails. If all the conditions are met the HMI is accessed like in the previous case.

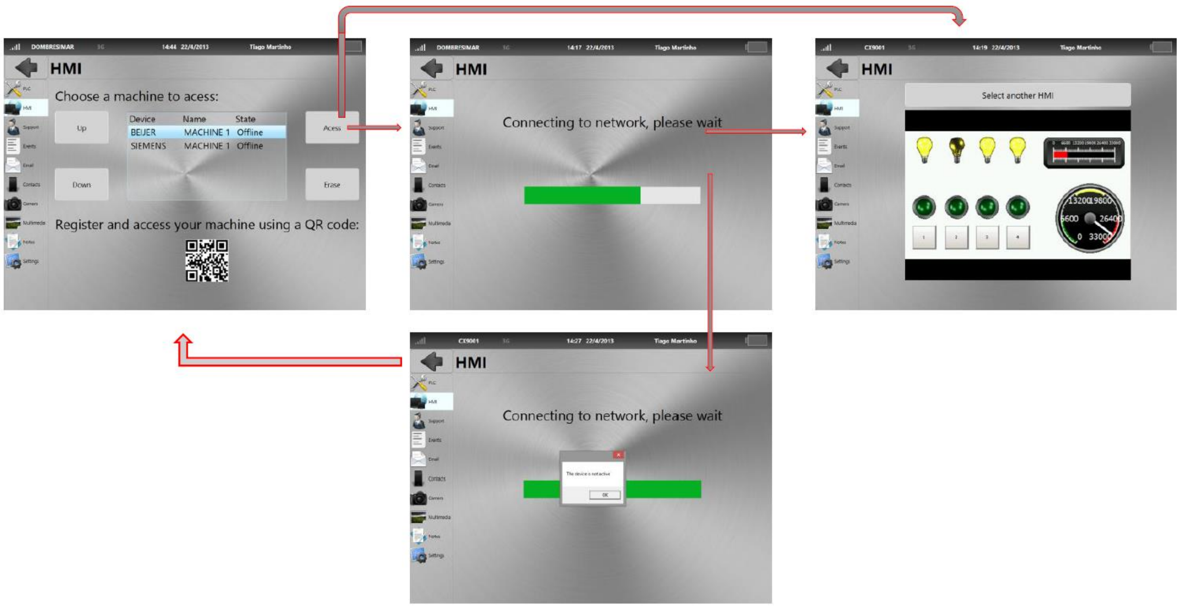

Figure 63 - Access the HMI: User Interaction 2

Another possibility to access the HMI, the user already saved the information of a machine, but he still wants to access it using a QR code. Figure 64 presents the user interaction for this choice. The user selects the button with the QR code image. After the decoding of the code, the software tries to connect the HMI the same way as the previous cases.

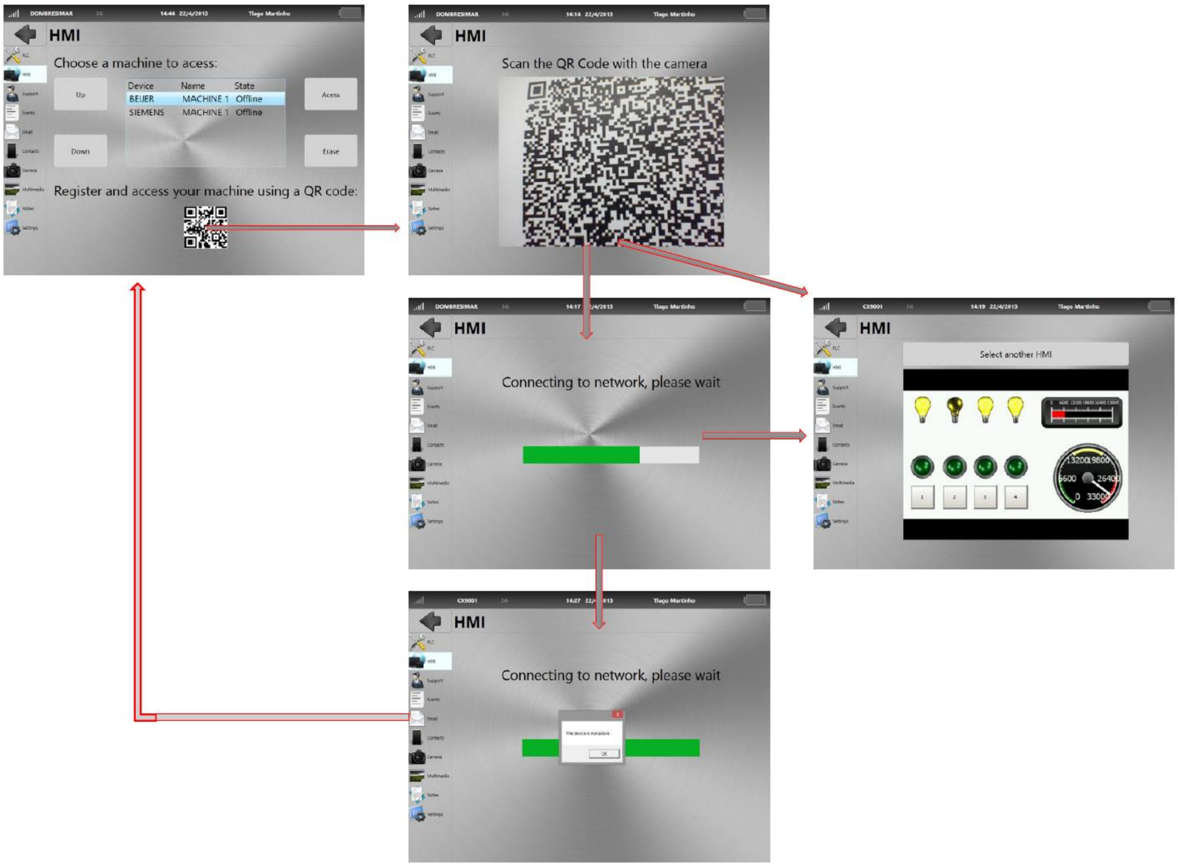

Figure 64 - Access the HMI: User Interaction 3

In this case, the device accessed is the first HMI that appears in the QR code information. If there is not a HMI in the QR code, the user will be notified and taken back to the “Select Machine Screen”.

The user also has the option to erase a previously registered information of a machine. This is done in the “Select Machine Screen”, by pressing one time the erase button to erase one machine, or holding the erase button to erase all the machines information. Figure 65 presents the interaction with the user.

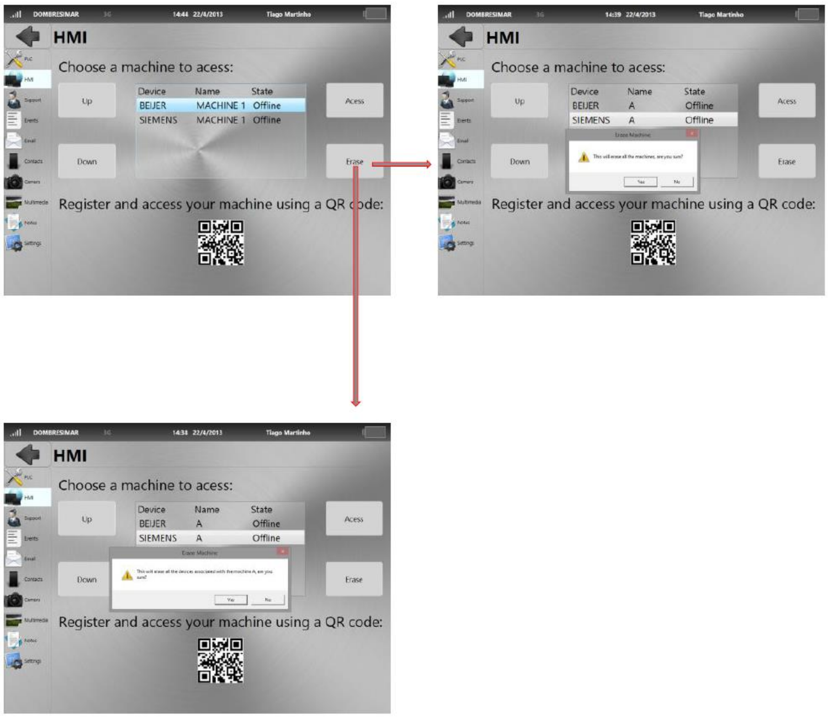

Figure 65 - Access the HMI: User Interaction 4

This finishes the **Access the HMI Module** implementation. The screens implemented were developed in side projects and independent of the rest of the software. To control the screens it was developed a user control, which loads the different screens and passes information between them.

With this architecture it is possible to have a flexible structure where each screen that interacts with the user can be developed independently. This enables to re-use the screens in different modules.

With this modular structure, it is also possible to add another different screen to interact with the user. For example, it is possible to create a new way to access the HMI without using a VNC client. It is also possible to add a new way to register the machines that not by decoding a QR Code but, for example, using near field communication (NFC) technology [141].

4.3.6 Access the PLC Module Implementation

The **Access the PLC Module** allows the operator to access a PLC, via a non-proprietary or proprietary protocol. With this access the operator is able to monitor and control the industrial process by interacting with the PLC. It allows also to reprogram the PLC using the proprietary software of the PLC.

For the **Access the PLC Module**, a decision was made to create a basis to grow in supporting different communications protocols. A starting point can be to give support to a proprietary and a non-proprietary protocol. The proprietary protocol can be, for example, the Automation Device Specification (ADS) protocol from Beckhoff [141], since Bresimar works with this manufacturer. The non-proprietary protocol can be the Modbus Protocol, for being one of the most popular protocol used in industrial automation [142].

To access the PLC via proprietary, or non-proprietary, protocol it is needed: first that the Tablet PC is connected on the same network of the PLC, second that the IP address of the PLC is known, and third that specific parameters for the communication are known.

Since the objective is to create something simple to use, the access information is encoded on a QR code, this makes the interaction more automated. The specific parameters for the communication are assumed and if incorrect they are asked to the user.

The user can access the PLC via the decoding of a QR code using the rear camera. The connection to networks and access is done automatically, the user only have to read the QR code with the camera. Once a first access is done the software records the information of a machine for future access.

Resuming this module will allow the user to do four main functions:

- Access a PLC panel via the proprietary protocol ADS;
- Create a HMI to interact directly with the variables of the PLC;
- Reprogram the PLC by accessing the proprietary software of the PLC;
- Record machines information by decoding of a QR code;

The functions used by the **Access the PLC Module** are very similar to the ones used in the **Access the HMI Module**. Since the previous module development was implemented to be modular, now is possible to re-use these components.

To simplify the implementation, and provide a greater flexibility, the interaction with the user was divided into eight screens. It were used eight different user controls, which are loaded according to the user interaction. Four of these eight screens are the ones developed for the **Access the HMI Module**: the Select Machine Screen, Decode QR Code Screen, Save Machine Information Screen and Connect to Network Screen. The implementation of the other screens will be now explained.

Select Way to Access the PLC Screen

When the user connects to the PLC he still needs to select how he is going to access it. In the select way to access PLC screen, the user is asked if he wants to access directly to the PLC variables, or access using the proprietary software of the PLC.

The screen implemented is presented in Figure 66. It is possible to see the two options to access the PLC, with the according titles and buttons. The buttons on bottom refer to the proprietary software, which are created in Run-Time, since they depend upon the PLC the user is accessing.

In this example the user is accessing a PLC of the manufacturer Beckhoff and so two shortcuts are created to launch two proprietary software of the manufacturer.

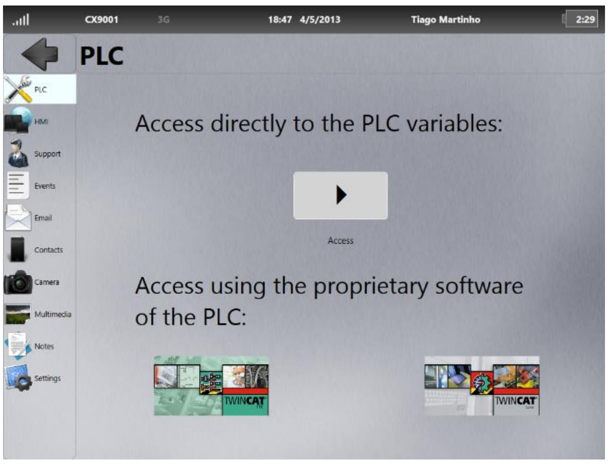

Figure 66 - Access the PLC: Select Way to Access PLC Screen Implementation

If the user selects to access the PLC using the proprietary software, the software is launched and covers all the screen. In Figure 67 it is possible to see an example where it was launched the software TwinCAT System Manager from Beckhoff [143].

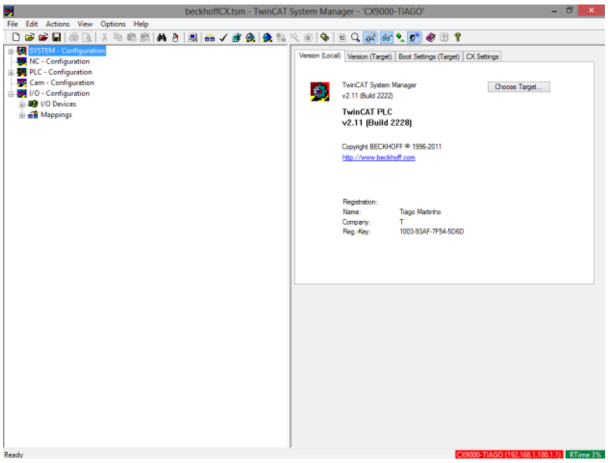

Figure 67 - Access the PLC: Proprietary Software Implementation

Access the PLC via X Protocol Screen

In the "Access the PLC via X Protocol Screen" it is verified if the communication parameters encoded on the QR Code allow to access the PLC. If the parameters are incorrect they are asked manually to the user otherwise the screen is skipped. This screen is loaded if the user selects to access the PLC variables directly.

The screen implemented is presented in Figure 68. In this particular case the screen is the Access the PLC via ADS Screen. This screen will vary from protocol to protocol, since different parameters are needed.

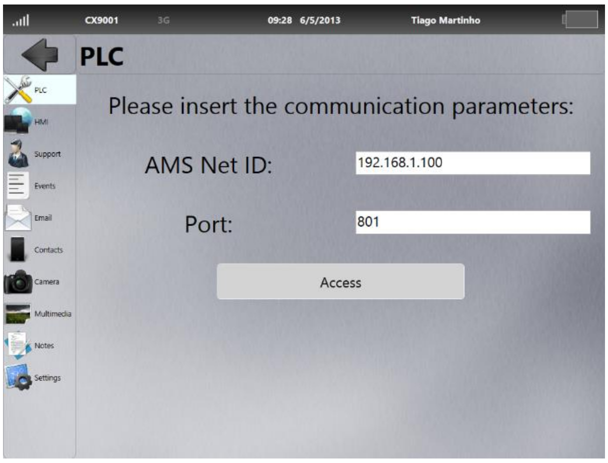

Figure 68 - Access the PLC: Access the PLC via X Protocol Screen Implementation

The parameters needed for the ADS communication are two, the AMS Net ID and the Port. If the parameters encoded on the QR code are not enough to assume these parameters, they are asked to the user as shown in Figure 68, two titles indicating the parameters and a button to access the PLC.

Select Variables from PLC Screen

In the "Select Variables from PLC Screen", the user selects the PLC variables he wants to control and monitor.

The screen implemented is presented in Figure 69. It is possible to see on the left a list containing all the variables of the PLC and a list on the right that contains the variables that the user wants to monitor and control.

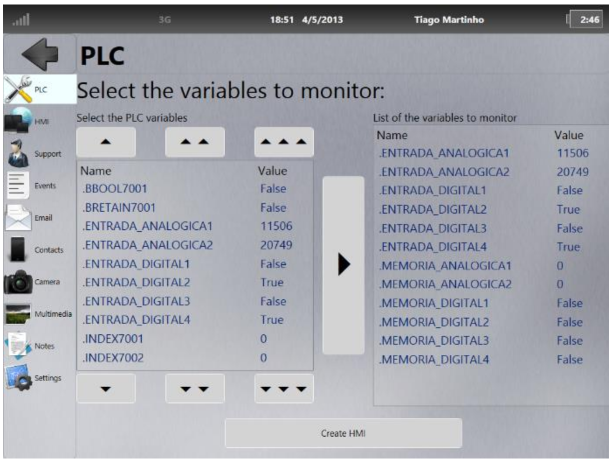

Figure 69 - Access the PLC: Select Variables from PLC Screen Implementation

It was implemented also three buttons on top and bottom of the list of the PLC variables to allow to navigate through the list more quickly. The user selects the variables to monitor and adds them using the button on the middle, with the arrow pointing right.

Once the user finishes the selection he presses the “Create HMI” button to navigate to the Create HMI Screen, which enables to monitor and control the process variables.

Create HMI Screen

In the “Create HMI Screen” is created a HMI for the user to monitor and control the PLC variables he previously selected.

The screen implemented is presented in Figure 70. It is possible to see the analogue variables represented by textboxes and the digital ones by light indicators.

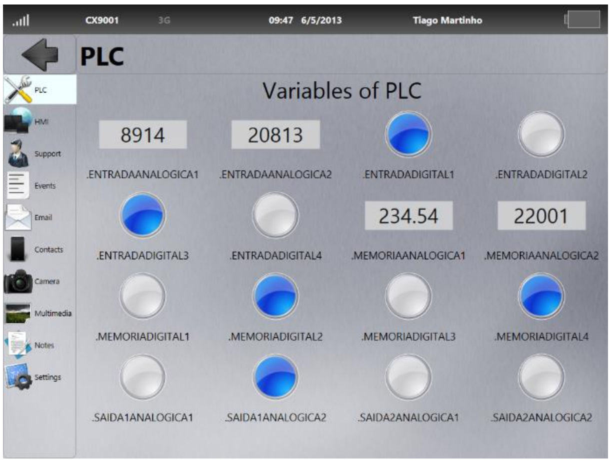

Figure 70 - Access the PLC: Create HMI Screen Implementation

The user can control the digital variables by pressing the light indicators, toggling their actual value.

The user can control the analogue variables, by writing a new value on the textbox, and pressing the return key of the virtual keyboard.

To implement support for different protocols the idea is to have one class that implements the protocol, with generic functions such as connect, disconnect, and load variables. So it was implemented a class to structure the different kinds of variable that different protocols demand.

These interfaces were integrated so it is possible to use different screens to interact with the user independently of the communication protocol.

User Interactions

In this section it is possible to understand how the different screens interrelate, according to the user interaction.

The first time the module is initiated the software does not have any information regarding the machines to access. The only option to the user is to record a machine, by decoding a QR code. The software will then have the parameters to establish a connection.

In Figure 71 it is presented the interaction with the user, similar to the **Access the HMI Module**. If the user press the Access or Erase button, a message appears to say he still needs to record the devices. If the user press the button with a QR Code image, he has the possibility to read a QR Code. With the information decoded he can connect to the PLC, so after he can access it via the proprietary software or directly to the variables.

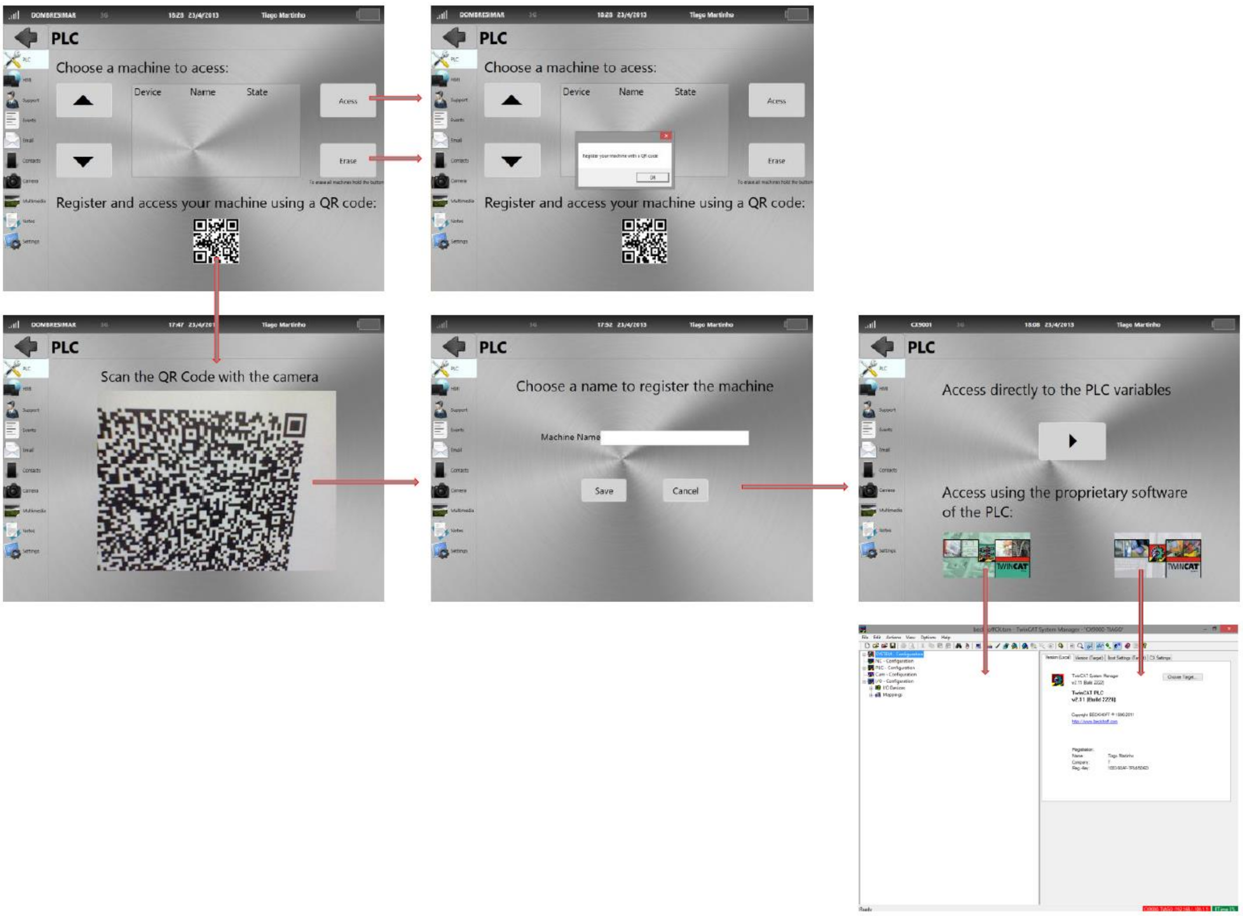

Figure 71 - Access the PLC: User Interaction 1

After the user records the machines he can also connect to the PLC by choosing from a list. In Figure 72 it is presented the user interaction.

Previously it was not referred the possibility of communication failing. This can happen in two ways, first if the network is not active or not in range, second if the device is not active. In Figure 72 it is presented the failing situations. If all the conditions are met the user can choose how to access the PLC.

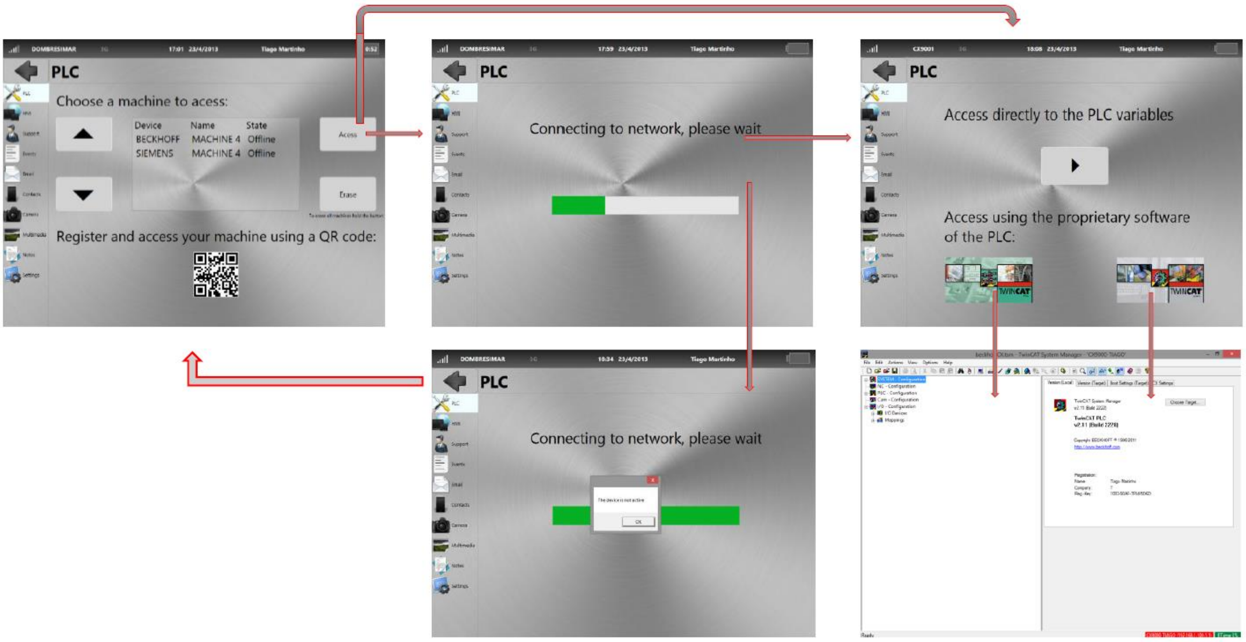

Figure 72 - Access the PLC: User Interaction 2

Another possibility to access the PLC, the user already saved the information of a machine, but he still wants to access it using a QR Code. In Figure 73 it is presented this user interaction. The user selects the button with the QR Code image. After the decoding of the code the software tries to connect the PLC the same way as the previous, where the user chosen from a list.

If the Tablet PC is already connected to the network where the device is connected and the device is active the access is made immediately. The device accessed is the first PLC that appears in the QR Code.

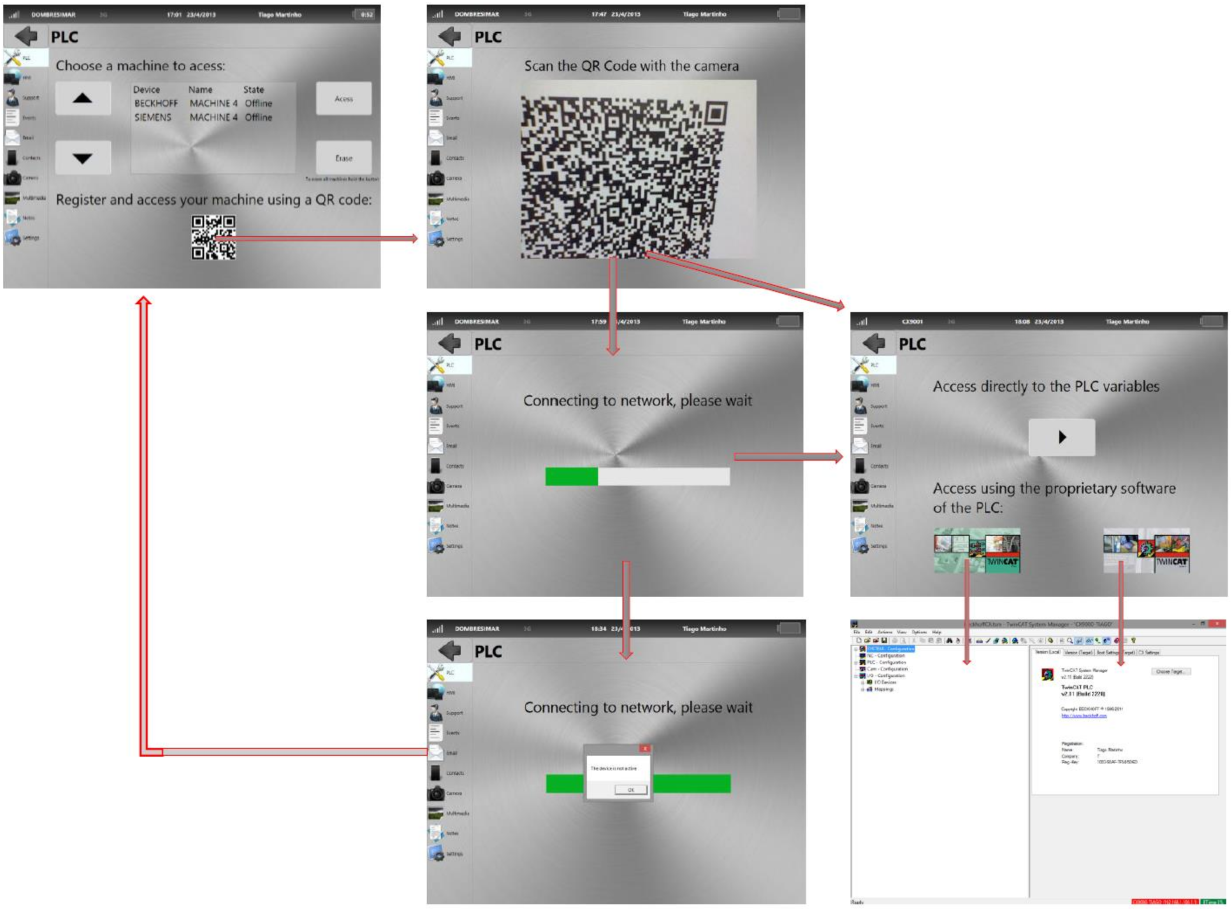

Figure 73 - Access the PLC: User Interaction 3

To access the PLC variables directly, the user selects the first option in the "Select Way to Access the PLC Screen". After the software tries to connect to the PLC if the parameters are correct the Access the PLC via X Protocol Screen is skipped, otherwise they are asked to the user. In Figure 74 it is presented the interaction with the user. After that, the user selects the variables he wants to monitor and control, a HMI screen is created to monitor and control the process.

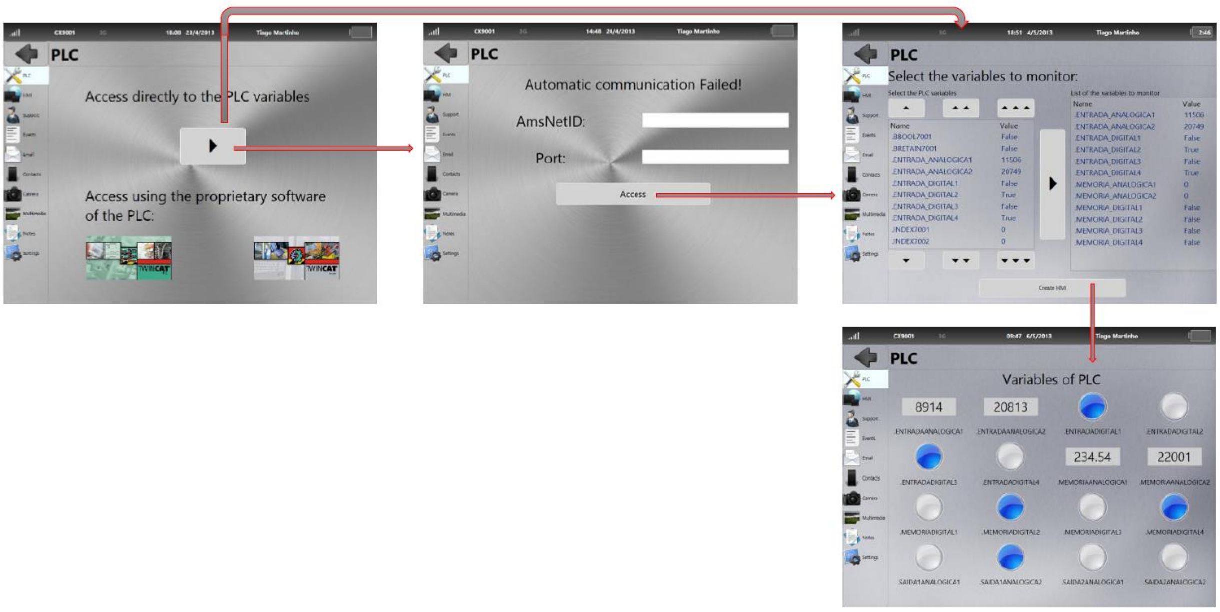

Figure 74 - Access the PLC: User Interaction 4

The user also has the option to erase the information of a machine. This is done by pressing one time the erase button, to erase one machine, or holding the erase button, to erase all the machines information. In Figure 75 it is presented the interaction with the user.

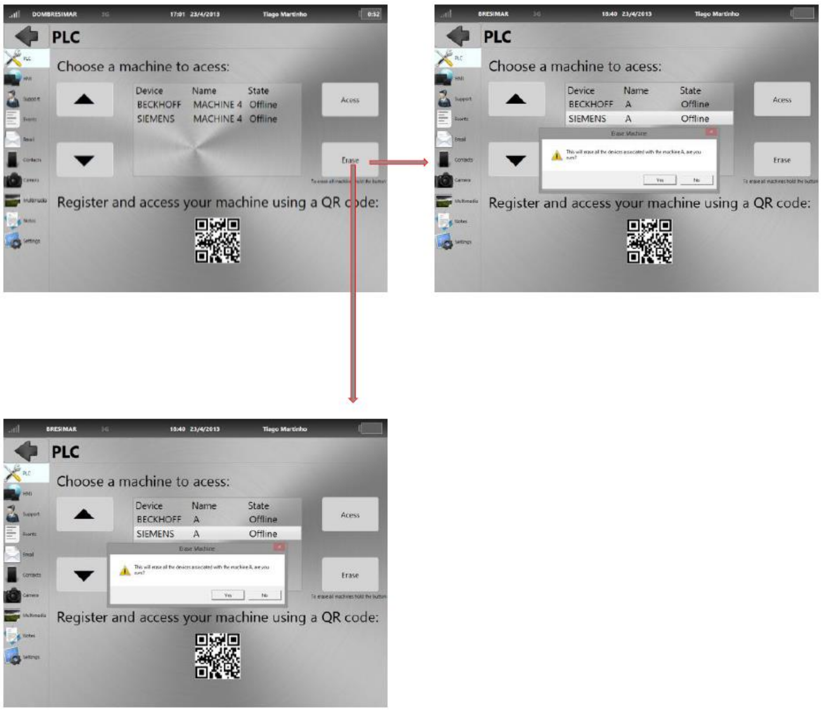

Figure 75 - Access the PLC: User Interaction 5

This finishes the **Access the PLC Module** implementation. The screens implemented were developed in side projects and independent of the rest of the software. To control the screens it was developed a user control, which loads the different screens and passes information between them.

With the architecture created, it is possible that each screen that interacts with the user can be developed independently. This brings more independence between each functionality. It was possible to observe that the screens implemented for the **Access the HMI Module** were re-used in this module.

4.4 Conclusion

In this chapter were presented the requirements that led to the choices of the framework and Tablet PC for the project.

Next, it was presented the developed software, targeting the framework and mobile device chosen. It was possible to see implemented the use cases and requirements previously discussed.

As aimed, with the developed software and the architecture used, it is now possible to develop features independently and fit them in the application. It is also possible to improve and test the software more easily by parts.

In this chapter were presented the requirements that led to the choices of the framework and Tablet PC for the project.

Chapter 5

VALIDATION

This chapter describes the conducted tests and validation to the developed software solution. The software interactions for a specific underlying hardware are presented. The integrated hardware simulates an actual field situation.

The chapter starts with the steps taken to integrate the hardware, and after it presents the conducted tests to the developed software.

5.1 TEST SYSTEM

In this section, the test system will be described and how it allows to test the developed software. After, it will be explained the configurations needed to integrate the different devices in the test system. Finally, the developed test system will be presented.

This system allows to test the software with a specific underlying hardware and communication protocol. The aim is to simulate a real case scenario of an industrial system so it will be focused in controlling and monitoring.

5.1.1 Description of the Test System

In this subsection, it will be described the hardware integrated to test the functions of the developed software. Starting with the devices used and the architecture of the system. Next, it will be explained how each device allows to test the developed software.

As it was referred in the hardware architecture, a system where the project can be inserted has the minimum requirement of having an IPC, a wireless access point and the mobile device, an industrial Tablet PC. It can additionally contain a HMI or an IP camera.

The system used for testing has one of each device to check the most number of software features.

The system will be composed by:

- **Industrial Tablet PC,** the uTablet T10C [69];
- **Embedded industrial PC,** the CX9000 from Beckhoff [144];
- **Wireless access point,** the WL-330N3G from Asus [145];
- **Graphic touch HMI,** the T4A from Beijer Electronics [146];
- **IP camera,** the FI8910W from Foscam [54];

The devices will interconnect as it is presented in Figure 76, where is possible to see how each device communicates.

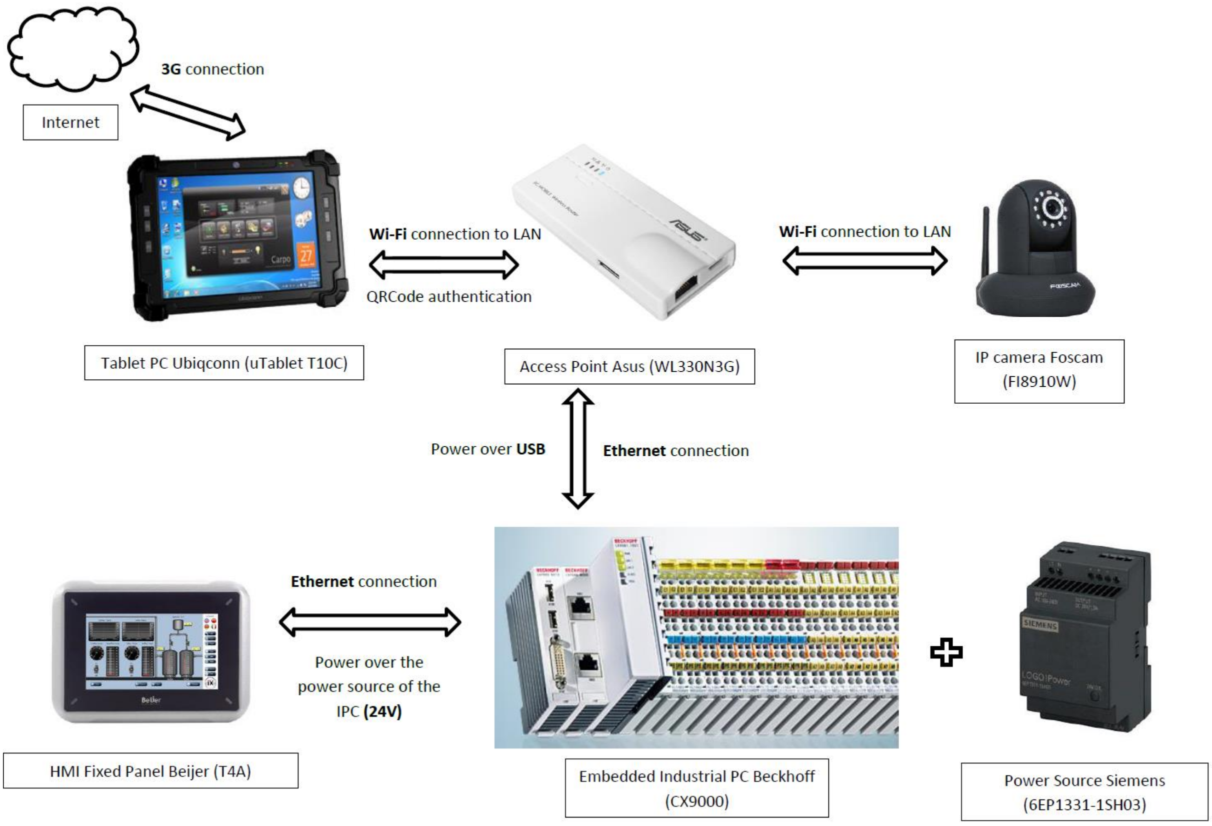

Figure 76 - Test System Architecture [52], [53], [54], [55], [56], [147]

The Tablet PC is the mobile device where the developed software will run. It will allow to examine the different developed software functions. The Tablet PC connects to the access point via Wi-Fi, in order to have access to the different devices. The Tablet PC needs also to guarantee an internet connection via the 3G module, in order to provide technical support functions.

The wireless router allows the Tablet PC to connect to the different devices wirelessly. These devices include the PLC where the router is connected, as well the HMI connected to the PLC.

The PLC allows to test the **Access the PLC Module**, by testing the connection to a PLC using the proprietary communication protocol ADS of Beckhoff. It will allow to test the access to the variables directly, as well the access via the proprietary software running on the Tablet PC. The PLC will have to be programmed in order to simulate a real case scenario. This PLC is available to the Tablet PC via the access point connected to it.

The HMI allows to evaluate the **Access the HMI Module**, by testing the remote access to a HMI via VNC. This HMI is connected to the PLC via an Ethernet, and it is accessible to the Tablet PC since it is on the same LAN shared by the router.

The IP Camera allows to check if it is possible to access a device in the network created by the wireless router. The IP Camera is connected wirelessly to the WLAN create by the access point.

5.1.2 Configurations of the Test System

In the following subsections, it will be explained how to configure the different devices, in order to simulate an industrial field situation.

Configuration of the PLC

The PLC CX9000, since it is already a PC, does not require an external PC to be programmed. It contains interfaces to interact with machines, the I/O modules, both analogue and digital.

Figure 77 presents a picture of the equipment, which contains the USB, DVI and Ethernet interfaces on the left, and the inputs and outputs modules on the right. In the middle is placed the IPC, which is a PC with an industrial robustness. The PLC used contains an ARM processor and all flash technology. Which, with the proper casing, provides the robustness to industrial environment.

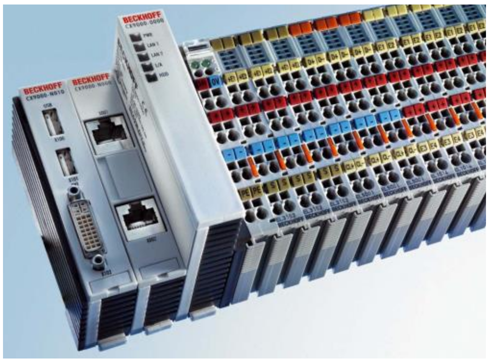

Figure 77 - Beckhoff CX9000 [56]

To set the interface with the IPC it is required to connect a power source and connect the I/O modules used. The PC communicates with the IPC via the Ethernet interface.

Once these connections are done, it is needed to check that the IP address of the Tablet PC is on the same range of the IPC. It may be needed to make a hard reset if the IP address of the PLC is not known. Then, if needed, the IP address of the PC is changed in the network settings of Windows and verified using the command prompt with the command IPCONFIG. To verify the connection is done a ping to the IPC IP address, with the command PING.

With the above steps successfully done, is used the proprietary software of Beckhoff, TwinCAT System Manager, to do a Broadcast Search and add a route dialog from the IPC to the PC.

Figure 78 presents these steps, it is possible to see the command prompt windows with the successful ping statistics. On the right bottom corner, it is possible to ensure that the software is successfully connected to the PLC.

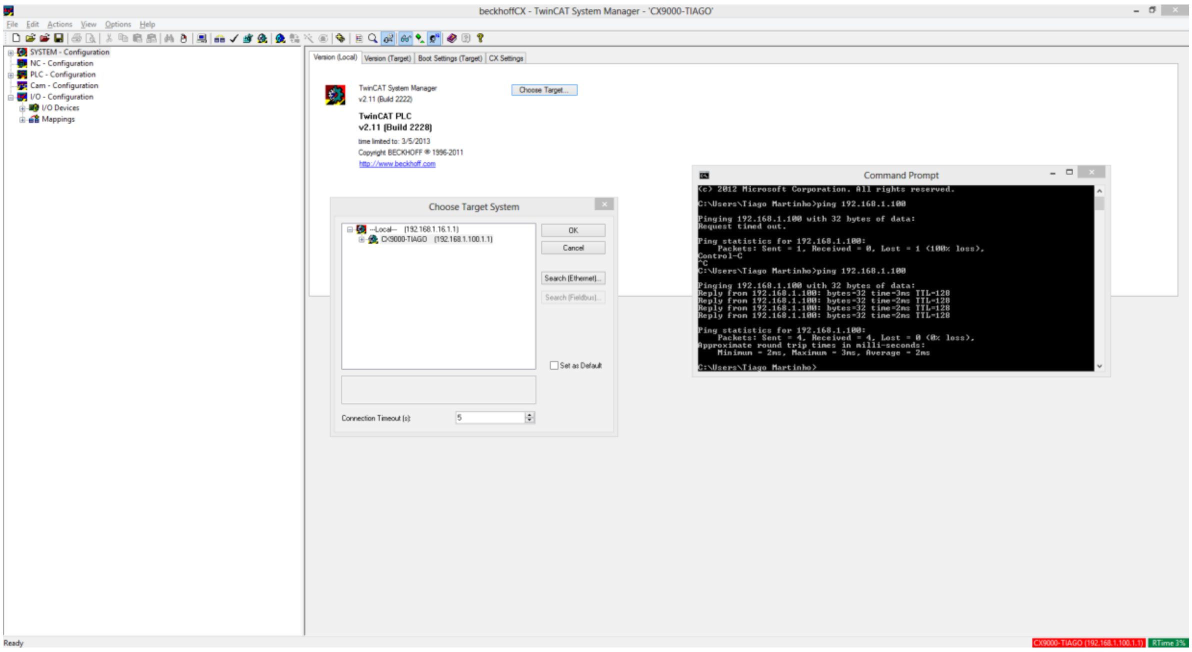

Figure 78 - PLC Configuration: Testing Connectivity

The software TwinCAT System Manager allows to manage all configurations related with the PLC. Some examples are: to set communications parameters, map variables of the PLC program with the physical I/O of the machine, listing the modules connected to the PLC, and specify the actions the PLC does when it starts.

The software, as the other software provided by Beckhoff, communicates using the proprietary protocol Automation Device Specification (ADS), which follows the structure in Figure 79. The upper part regards the PC, which communicates using the ADS protocol to a Message Router. The Message Router is connected, via TCP/IP or COM ports, to another Message Router in the lower part, which regards the IPC.

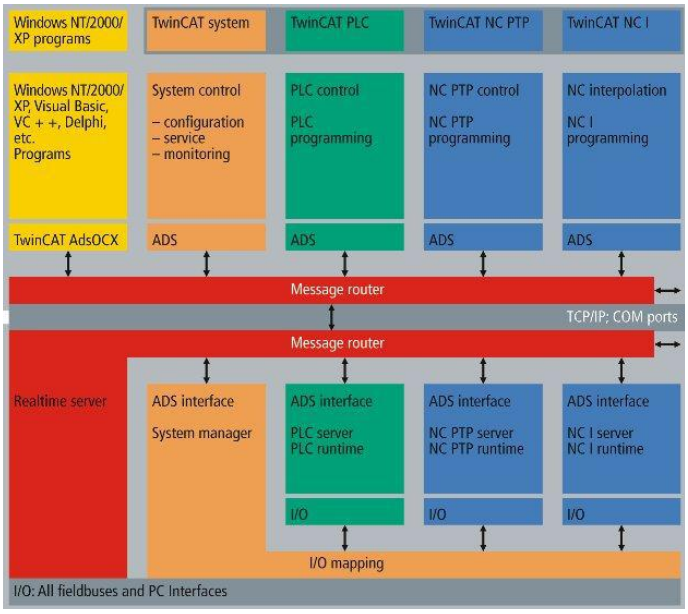

Figure 79 - ADS Architecture [148]

The IPC will then work as a "soft PLC", where the software in the IPC creates a virtual PLC, which communicates to an AMS message router running on the IPC. The Tablet PC communicates then to this

AMS message router via TCP/IP, which redirects the information to the virtual PLC. With this architecture is then possible to have different levels of abstraction, in the mode as the communication with the PLC is done.

After these connections and configurations had been successfully done, it was needed to reprogram the PLC in order to be able to control its outputs regarding the inputs and memory states.

To reprogram the PLC it was used the proprietary software of Beckhoff, the TwinCAT PLC Control [149]. This software is based in CODESYS [150], which respects the standard IEC 61131-3. It contains the more known PLC programming languages: IL, LD, FBD, SFC, ST and CFC [151].

In Figure 80, it is presented the user interface with the software, where is shown a developed program for the PLC. To test the developed program it is possible to download it to a real machine, or to emulate a virtual one in the PC.

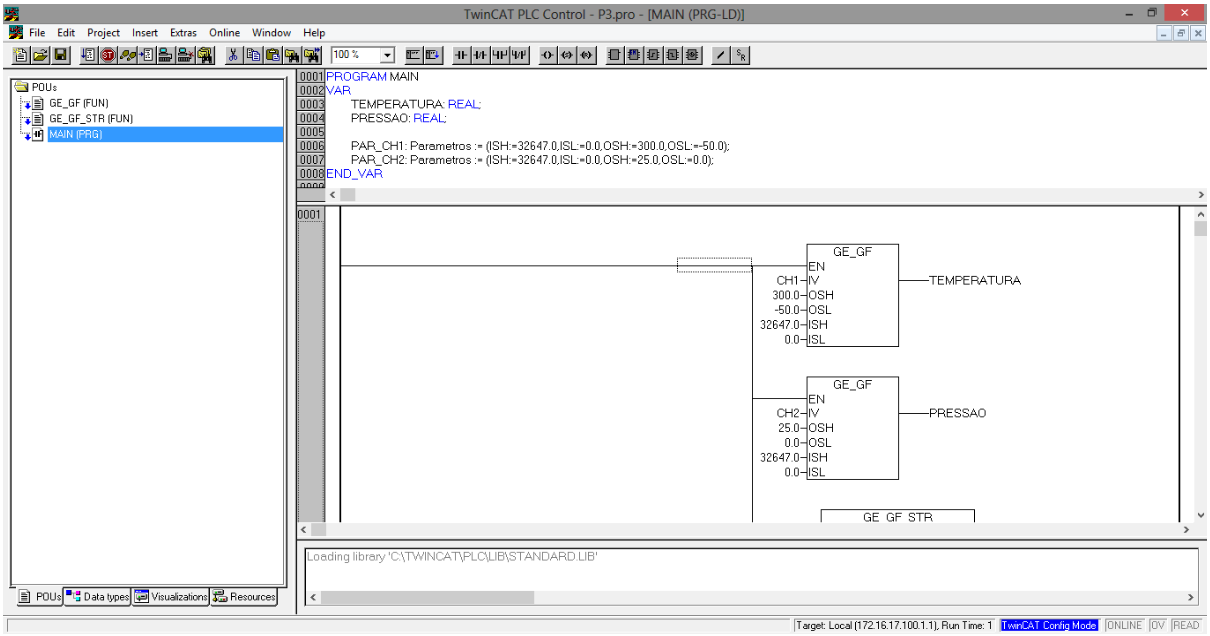

Figure 80 - PLC Configuration: Programming the PLC

After the program is created and downloaded to the IPC, it is still needed to use the software TwinCAT System Manager. This is done in order to map statically the declared variables in the program, to the I/O modules connected to the system.

At the creation of the PLC program, TwinCAT PLC Control generated a PLC-Configuration file, with the name of the project and *.tpy extension. The file contains all the variables declared as global in the project.

To achieve the mapping is needed to select the option "Linked To", in order to link the variable to the physical I/O in the PLC. After the mapping is done, the configurations are stored by clicking the button “Activate Configuration”.

The graphical interface is presented in Figure 81. On the left, it is presented a diagram, and under the section with name PLC Configuration, it is possible to see the variables declared in the developed PLC program. The Inputs represented with the yellow colour and the outputs represented with the red colour. Below, under the section with name I/O Configuration, it is possible to see the modules connected to the IPC. The analogue inputs represented with the green colour, the digital inputs represented with the yellow colour and the digital outputs represented with the red colour.

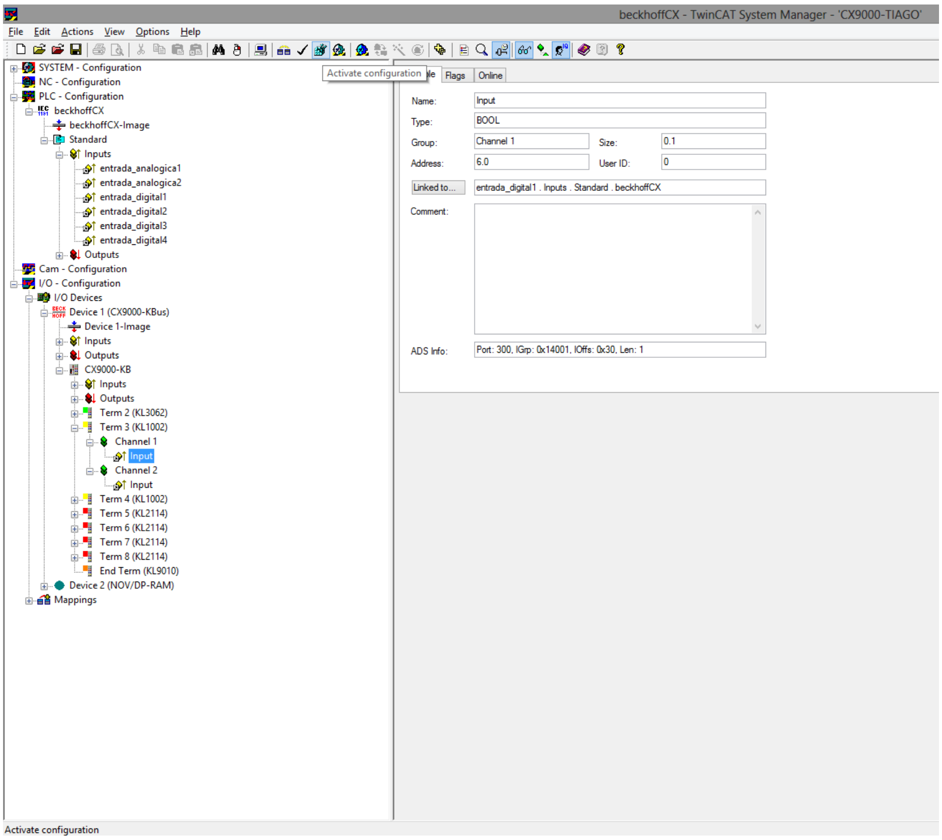

Figure 81 - PLC Configuration: Mapping Variables

With the steps fulfilled in this subsection, it was understood how to communicate with the IPC, how to develop programs for it, and how to map the physical interfaces.

Configuration of the Wireless Router

The wireless router will work in Access Point Mode, where it connects to a network with an Ethernet cable to share a wireless Ad-Hoc network. On this mode, the firewall, IP sharing and Network Address Translation (NAT) function is by default disabled. The goal is to create a wireless LAN for communication between the Tablet PC and the PLC, in order to access the PLC wirelessly.

The steps to make the configurations are the following:

1. Set the router to the default settings by pressing the reset button for 30 seconds;
2. Access the configuration menu of the router, using a browser, through the default IP address;
3. Set the mode to Access Point;
4. Choose the name for the network (SSID) and the password;

In Figure 82 is shown the graphical user interface of the configurations made.

Figure 82 - Wireless Router Configuration

If the PLC has DHCP [152] enabled, it is not needed any further configuration. If not, it is necessary to change the IP address of the wireless router to be on the same range as the IPC.

In order to test the configurations, the steps in Configuration of the PLC were repeated wirelessly.

Configuration of the HMI

The HMI Panel allows to visualize and edit the variables of the PLC, with this it enables to control and monitor the industrial process.

To develop HMI interfaces for the panel, and apply different parameters, was used the proprietary software of Beijer Electronics, iX Developer 2.0 [153].

First a controller is added, the PLC. The HMI will communicate using the proprietary protocol ADS from Beckhoff. The software of Beijer already provides support for this communication protocol. The settings of the controller will be Default Station (Zero) and HMI AMS Net ID (IP of the HMI with .1.1 in the end). Then on the Stations separator it is necessary to put the IP address, AMS net ID and the port of the PLC. If not known, these parameters are obtained from the software TwinCAT System Manager.

In the software TwinCAT System Manager, it is still needed to add the HMI panel, in the section route settings, with the according IP address and AMS Net ID; otherwise the PLC cannot detect the HMI.

After the device is added, the Tags are imported in the software iX Developer, that is, the variables of the program loaded in the PLC that should be monitor and control with the HMI. The file that contains this information is the one with the name of the project with *.tpy extension. The same file used before to do the mapping of the variables to the I/O modules.

With the variables imported, it is then possible to add elements to the HMI project and define their actions. These actions can be, for example, when the button is pressed toggle, reset or set a variable, or just to monitor variables by presenting them to the user, in the form of text boxes, indicators or gauges.

In order to test the configuration it was developed a program for the HMI, which contains a button and an indicator. When the button is pressed it toggles the variable state, which is displayed by the light indicator. It is possible to see the developed test program in Figure 83, with the light indicator on the left and the button to toggle the state on the right.

Figure 83 - HMI Configuration: Test Program

For the scope of this project, the remote access to the HMI is needed, so it is possible to interact with it using a VNC client. To do so the remote access server option needs to be enabled in the software iX Developer 2.0. In Figure 84 it is presented the configuration made.

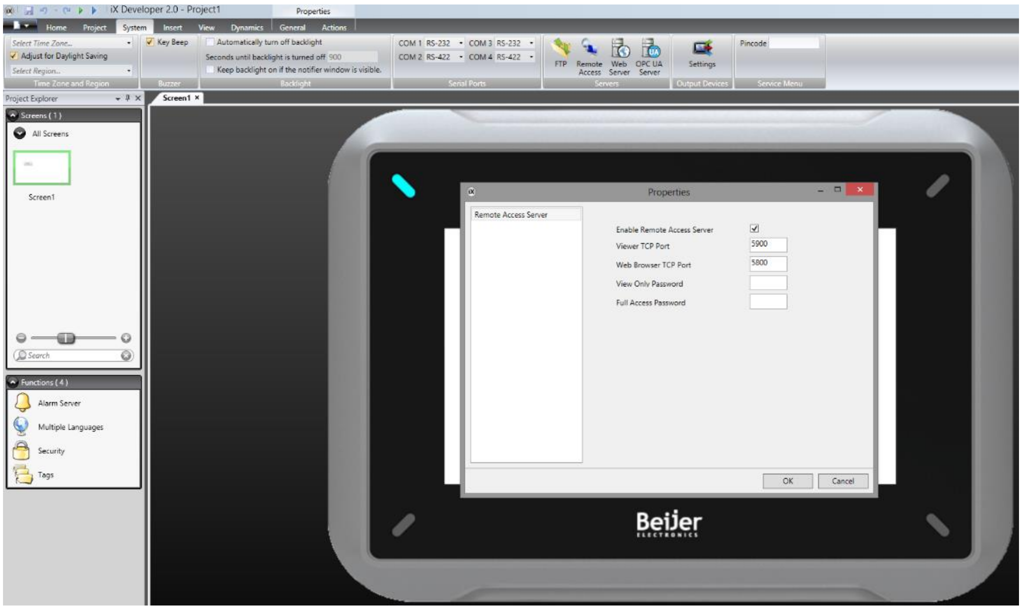

Figure 84 - HMI Configuration: Enabling Remote Access

Configuration of the IP Camera

The IP Camera allows to have live video of a physical place in the industrial plant.

To configure the IP camera, first it is needed to connect it on the same network of the PC where the configurations are made. After that, it was used the proprietary software of Foscam, the IP Camera Tool PC [154], to discover the camera and change the IP settings.

Next, the camera was accessed with the default IP address via a browser. On the menu "Device Management", under wireless LAN settings, it was added the wireless network that the camera should connect to, in this case, the network created by the access point.

Figure 85 presents the configurations done.

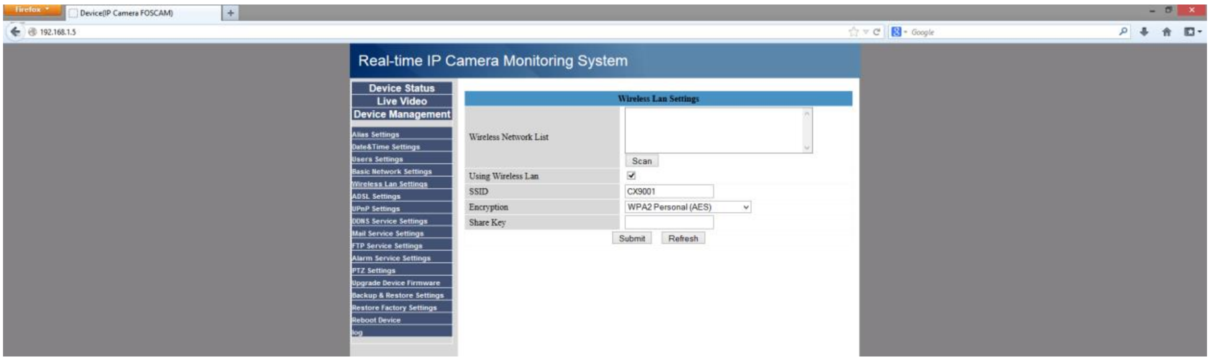

Figure 85 - IP Camera Configuration

To validate these steps the Tablet PC is connected to the wireless network created by the access point, and the IP camera accessed through the browser. By inserting the already known IP, and chosen the option Live Video, it is obtained live video from the camera as it is possible to see in Figure 86.

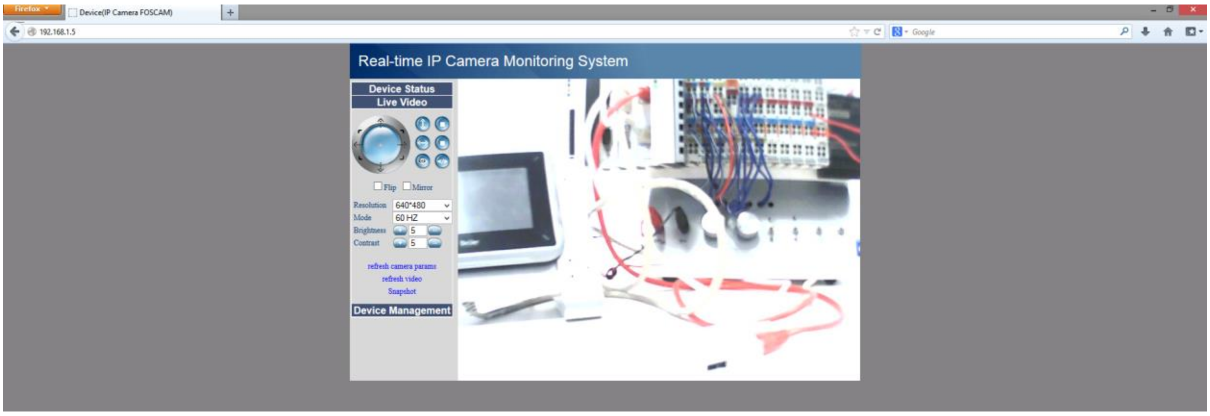

Figure 86 - IP Camera Live Video

Configuration of the 3G Module

The 3G module needs to be configured to have Internet access in the industrial plant, so it is possible to obtain technical support functions like the video call or remote access.

The steps are: insert a SIM card on the module of the Tablet PC, set the APN configurations and check the Internet connection.

In order to access the industrial devices trough Wi-Fi and still have internet access in the industrial site, it is still needed to do some configurations in the Tablet PC. First to change the order in which the interfaces are accessed by the networks services, putting in first place the mobile broadband connection. The configuration is presented in Figure 87 and can be found in the Control Panel->Network and Internet->Network connections, by selecting the top toolbar the option Advanced->Advanced Settings.

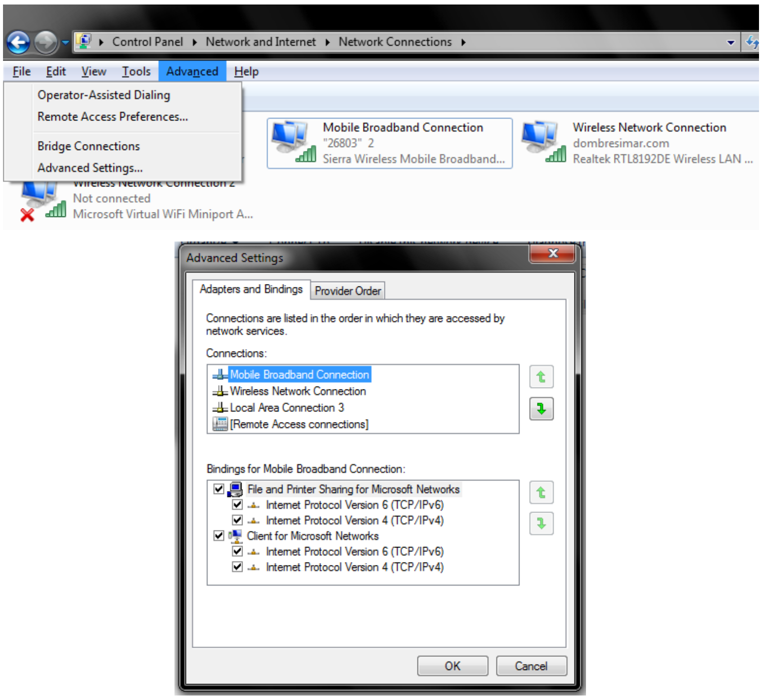

Figure 87 - 3G Module Configuration: Adapter Priority

Second, to use a static IP address of the Wi-Fi connection when two connections are used. The IP address must to be in the range of the access point connected to the machines, in order for the Tablet PC does not presents two routes when accessing the networks services.

5.1.3 Development of the Test System

Once these devices are all integrated, it is still needed to develop software that takes advantage of the hardware. For this reason was developed a new program for the PLC, and a new program for the HMI. It is then possible to do several tests to the developed software.

Since the aim of the project is to control and monitor industrial processes, the program loaded in the PLC should make use of inputs, outputs and memory registers of the PLC.

The IPC used was linked to a development Kit containing two analogue inputs and four digital inputs. The I/O modules attached to the IPC contained two analogue inputs, four digital inputs and sixteen outputs.

The mapping done for the digital variables of the program was the following: the four digital inputs of the PLC are connected to four digital outputs; the four digital memory register of the PLC are connected to four digital outputs.

In Table 37, it is presented this mapping of the digital variables, with the name of the module on top, and the variable that turns on the different digital outputs of the PLC.

Table 37 - Digital Variables Mapping

Output Module 1		Output Module 2	
1 - Digital Input 1	2 - Digital Input 2	1 - Digital Input 3	2 - Digital Input 4
3 - Digital Memory 1	4 - Digital Memory 2	3 - Digital Memory 3	4 - Digital Memory 4

The mapping done for the analogue variables of the program was the following: the value read from the two analogue inputs has 15 bits. Two limits were established which turn on two outputs. For the analogue memory registries was used the same reasoning.

In Table 38, it is presented this mapping, with the name of the module on top, and the value of the variable that turns on the different digital outputs of the PLC.

Table 38 - Analogue Variables Mapping

Output Module 3		Output Module 4	
1 - Analogue Input 2 > 11 000	2 - Analogue Input 2 > 22 000	1 - Analogue Memory 2 > 11 000	2 - Analogue Memory 2 > 22 000
3 - Analogue Input 1 > 11 000	4 - Analogue Input 1 > 22 000	3 - Analogue Memory 1 > 11 000	4 - Analogue Memory 1 > 22 000

The PLC program, written in Ladder language, has the function to toggle the outputs according the different inputs and memory registers, with the reason previously explained.

With this program it is possible to simulate an industrial process since it is possible to change and visualize the state of inputs and outputs, as well to modify memory registers.

In Figure 88, it is presented the PLC, linked to the kit, running the program created. The two potentiometers, in the figure, are connected to the two analogue inputs terminals, with the green colour and the number two. The four switches, are connected to the four digital inputs terminals, with the yellow colour and the numbers four and five. It is also possible to see in the figure the sixteen outputs, with the red colour and the numbers six, seven, eight and nine.

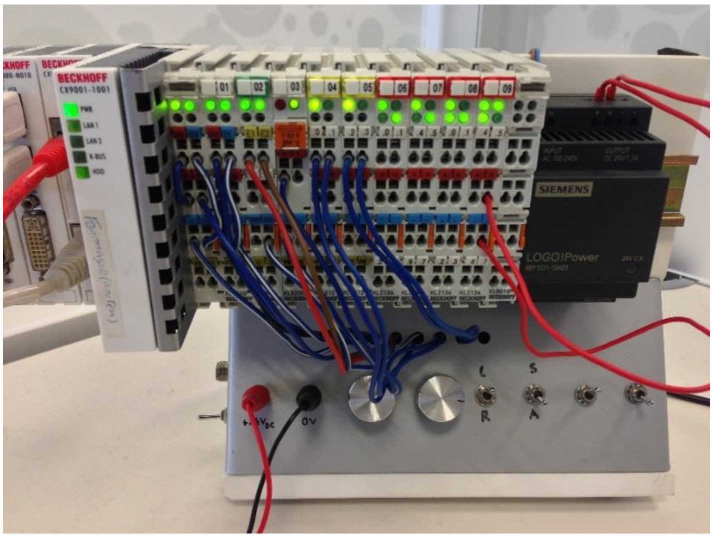

Figure 88 - Test System Development: PLC program running

In Figure 89, is shown the value of the variables using the software TwinCAT PLC Control. It is now described the value of the variables according to the reason previously stated.

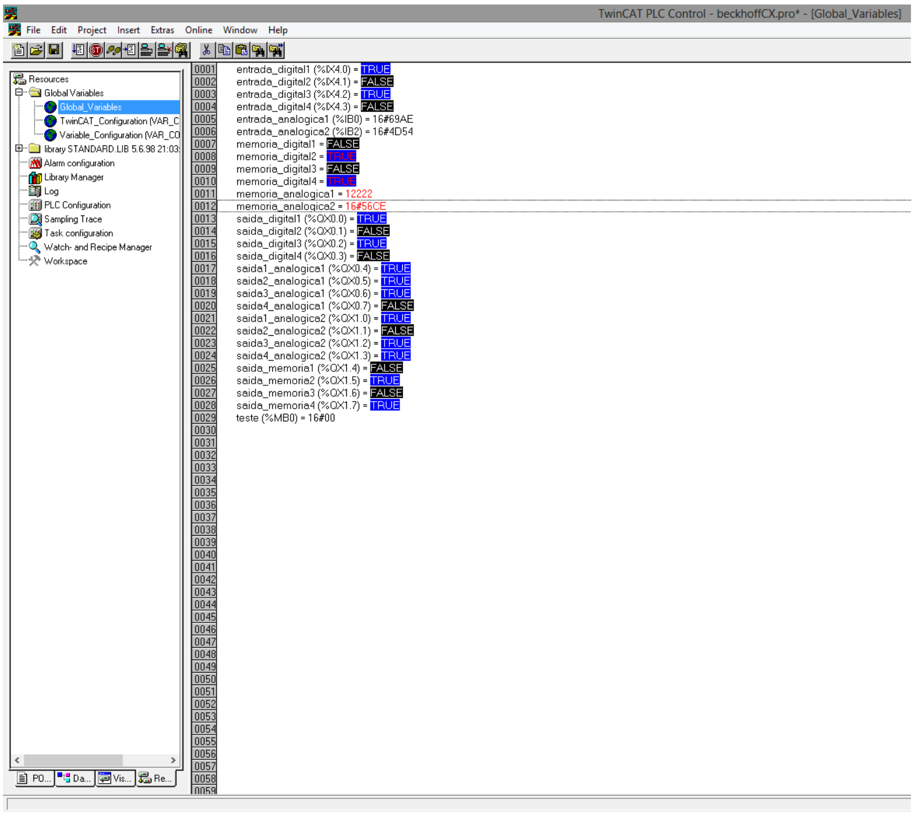

Figure 89 - Test System Development: value of PLC variables

The digital inputs **entrada_digital1** and **entrada_digital3** are **True**. The digital inputs **entrada_digital2** and **entrada_digital4** are **False**. Therefore, the digital outputs **saida_digital1** and **saida_digital3** are **True**, and the digital outputs **saida_digital2** and **saida_digital4** are **False**. The values are linked as saw in the Table 37 - Digital Variables Mapping.

The analogue input **entrada_analogica1** is bigger than 22.000. The analogue input **entrada_analogica2** is bigger than 11.000 but smaller than 22.000. Therefore, the digital outputs **saida1_analogica1**, **saida2_analogica1** and **saida3_analogica1** are **True**, and the digital output **saida4_analogica1** are **False**. The values are linked as saw in the Table 38 - Analogue Variables Mapping.

The digital memories **memoria_digital2** and **memoria_digital4** are **True.** The digital memories **memoria_digital1** and **memoria_digital3** are **False**. Therefore, the digital outputs **saida_memoria2** and **saida_memoria4** are **True**, and the digital outputs **saida_memoria1** and **saida_memoria3** are **False**. The values are linked as saw in the Table 37 - Digital Variables Mapping.

The analogue memory **memoria_analogica1** is bigger than 11.000 but smaller than 22.000. The analogue memory **memoria_analogica2** is bigger than 22.000. Therefore, the digital outputs **saida1_analogica2**, **saida3_analogica2** and **saida4_analogica2** are **True**, and the digital output **saida2_analogica2** are **False**. The values are linked as saw in the Table 38 - Analogue Variables Mapping.

With the software TwinCAT PLC control is possible to monitor and control the variables of the PLC, but to do that in an efficient way it is necessary to use the HMI.

It was then developed an interface for the HMI in order to control and monitor the variables of the program loaded in the PLC, the result is presented in Figure 90.

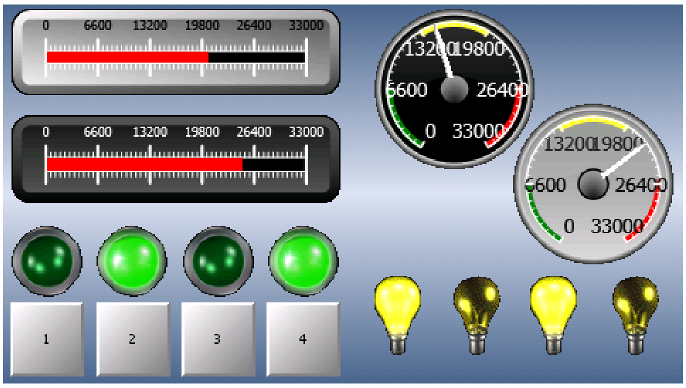

Figure 90 - Test System Development: HMI Program

The reasoning is the following, the two linear meters on the left upper corner are linked to the analogue inputs, and present their current value.

The four icons of green lights on the left lower, corner represent the digital memory register, the buttons allow to toggle the value of the variable.

The two circular meters on the right upper corner are linked to the analogue memories and present their current value.

The four icons of lamps on the right lower corner are associated with the four digital inputs. They represent the current state of the inputs, True or False.

With this test system is possible to conduct different tests to the developed software. These test system acts as a simulator of a real case scenario in industrial automation.

5.2 Software Testing

In this section, it will be explained how the developed software works with the integrated test system.

It was made a functional testing of the software, since unit and integration tests were made in the development phase. The tests made verify if the functions developed match the expected outputs.

In each sub-section, it will be described the conducted test to the fully integrated system, in order to prove that it meets the requirements stated.

5.2.1 Recording Information of a machine

With this test, it was verified that it is possible to record information of the machines by reading a QR code.

It were used four QR codes, in order to check if the software stores the information correctly.

The first QR code contained the information of the test system developed:

CX9001#XPTO1234ABCD#PLC_BECKHOFF#192.168.1.3#HMI_BEIJER#192.168.1.4#PLC_SIEMENS#192.168.1.7#HMI_SIEMENS#192.168.1.8

As explained in Chapter 4, Decode QR Code Screen, the first two parameters regard the network where the machine is connected. The SSID of the network is CX9001, and its password is XPTO1234ABCD.

Next, it is presented a list of information regarding the devices connected to the machine. It is possible to see that there is the PLC of Beckhoff and the HMI of Beijer. It was added two more devices of Siemens, a PLC and a HMI, to test the robustness of the software. These last two are virtual devices to test a software failure case. More specifically, the fact that the network is available but the devices are not.

The following QR code contained an existing network and four virtual devices, to test a software failure case. More specifically, the fact that the Tablet PC is connected to a different network and has to change. The information of the QR code was the following:

DOMBRESIMAR#passwordDaRede#PLC_ROCKWELLn1#193.168.1.100#HMI_OPTO22n1#193.168.1.39#PLC_ROCKWELLn2#193.168.1.4#HMI_EXTER#192.168.0.233

The third QR code contained a virtual network and two virtual devices, to test a software failure case. More specifically, the fact that the Tablet PC tries to connect to a network that does not exist. The information of the QR code was the following:

CX8001#123456789A#PLC_ROCKWELLn3#193.168.1.100# HMI_OPTO22#193.168.1.39

Finally, the fourth QR code was a dummy QR code to test a software failure case, where the information contained in the code is not valid.

The test was done with the **Access the HMI Module**, but it could also be made with the **Access the PLC Module** since they share the same modules of implementation and lead to the same results.

The steps to make the test were: to load the **Access the HMI Module**, by clicking the icon with the name HMI. Then to press the button with the QR code image.

The interaction with the software and the obtained results are shown in Figure 91. This interaction is similar to the first three valid QR code.

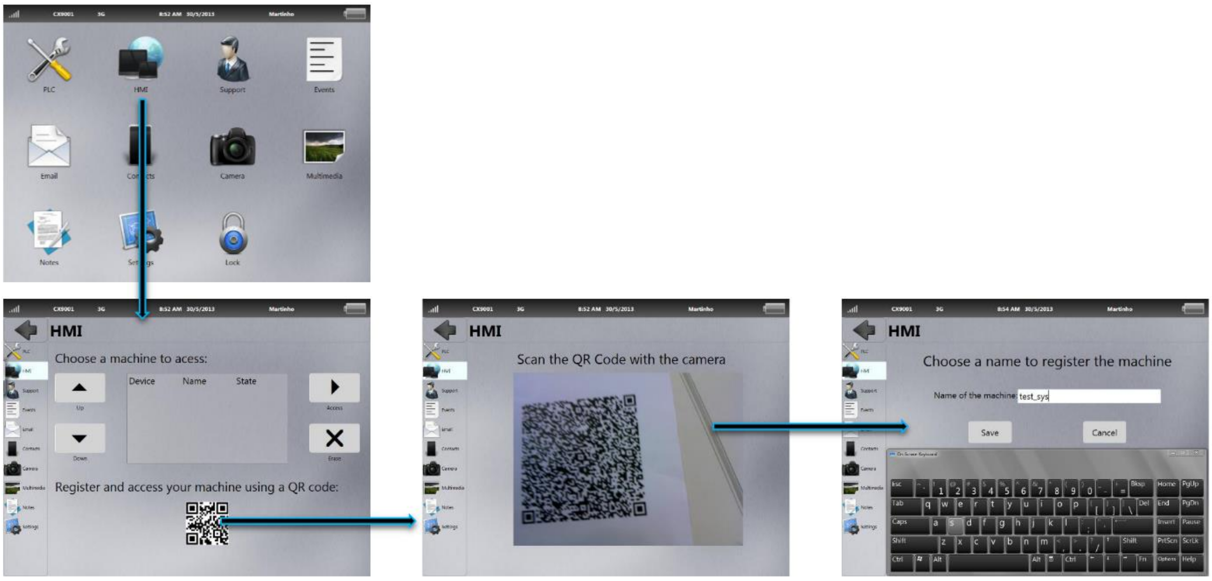

Figure 91 - SW Testing: Record Machines

It is possible to verify that the data was saved as it is presented in Figure 92. On the left, it is possible to see the information regarding the integrated test system, saved with the name TEST_SYS. On the middle, it is presented all the information saved. Finally, on the right, it is presented the information of the machines saved in Access the PLC module, since the module records all the devices associated with the machine, PLC and HMI.

Figure 92 - SW Testing: Recorded Machines

It is possible to verify from the Figure 92 that the information was properly stored by the software. Nevertheless, the resulting XML file can be checked in Appendix D - Software Testing - D.1 XML File.

In case of the fourth QR code, with the invalid information, a message is presented to the user. It is possible to see the interaction in Figure 93.

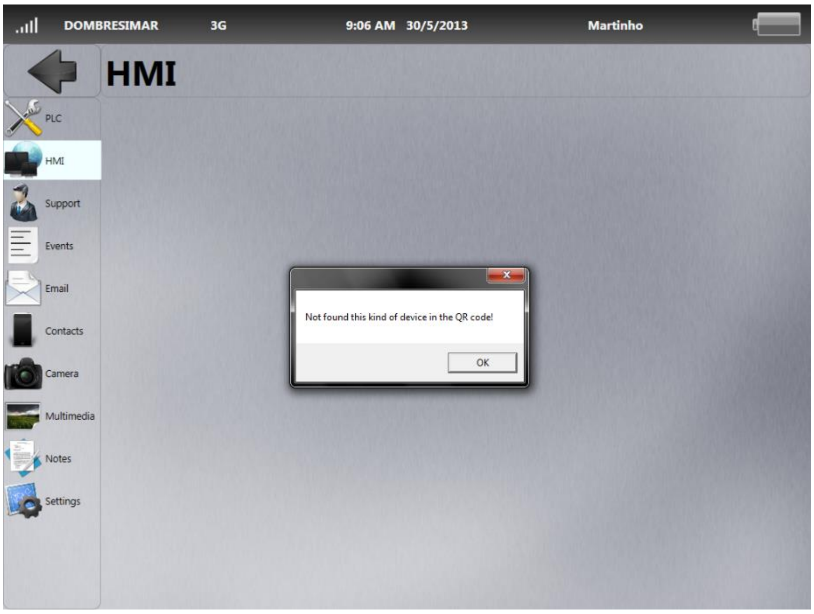

Figure 93 - SW Testing: Invalid QR Code

When trying to save information of a machine without a name, the user is notified with a tooltip that appears near the save button as shown in Figure 94.

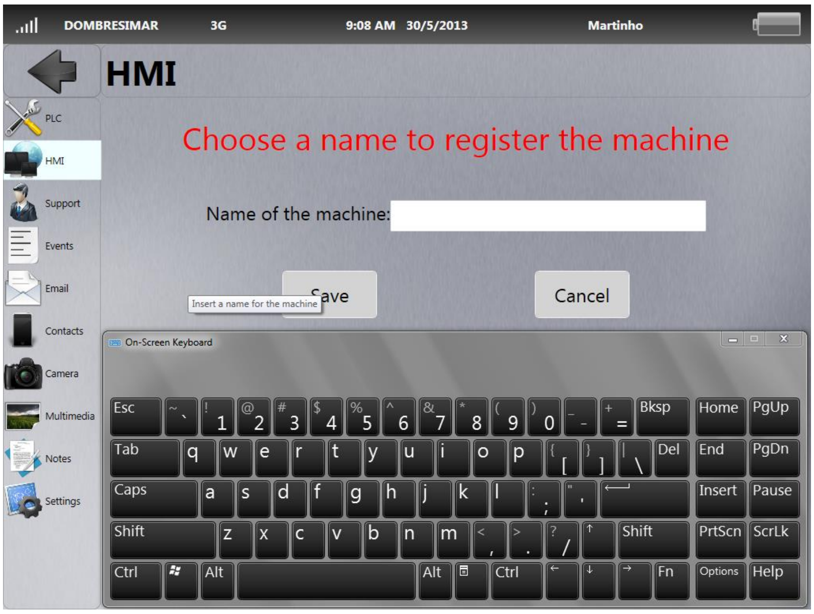

Figure 94 - SW Testing: Record Machine without Name

When trying to save information of a machine with the same name, the user is notified with a tooltip that appears near the save button as shown in Figure 95.

Figure 95 - SW Testing: Record Machine with the Same Name

The erase button worked as expected, allowing to remove information of a machine already recorded by pressing the erase button, also to erase all the information already recorded by holding the erase button. The interaction with the user is presented in Figure 96.

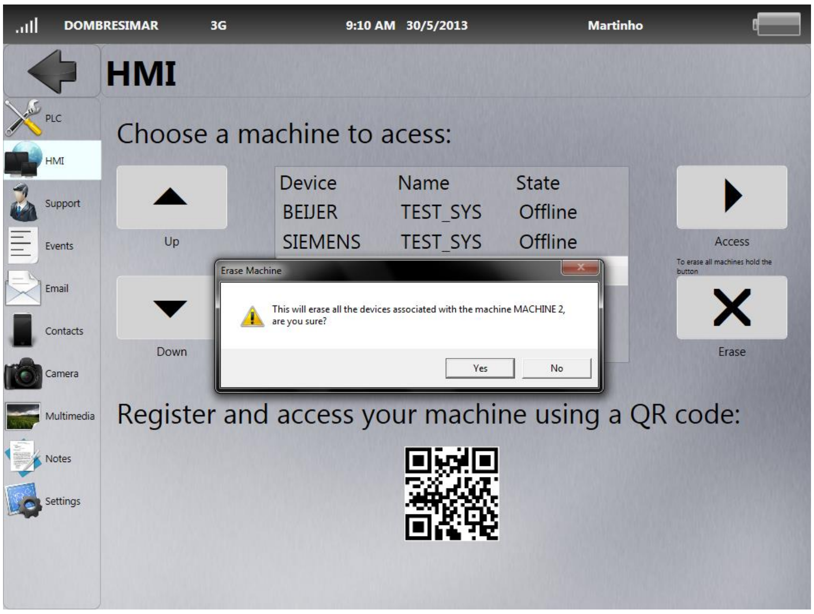

Figure 96 - SW Testing: Erasing Machines

5.2.2 Accessing the HMI via VNC

With this test, it was verified that it is possible to access the HMI of Beijer via VNC.

The steps to make the test were: to load the **Access the HMI Module**, by clicking the Icon with the name HMI. Then to choose a machine, previously recorded, from the list and press the button with the name Access.

The interaction with the software and the obtained results are shown in Figure 97.

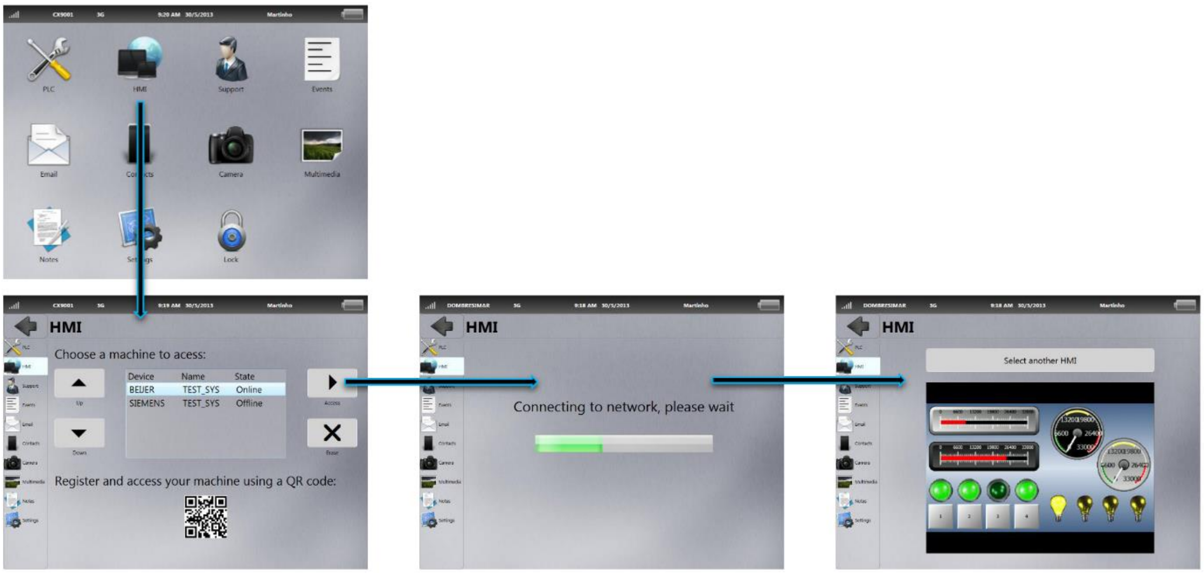

Figure 97 - SW Testing: Access HMI via VNC

It is possible to verify that the HMI was accessed via VNC, and it was possible to control and monitor the process by interacting with it.

In case the network is not in range or the device is not active, the user is notified with a message saying the device is not active as presented in Figure 98.

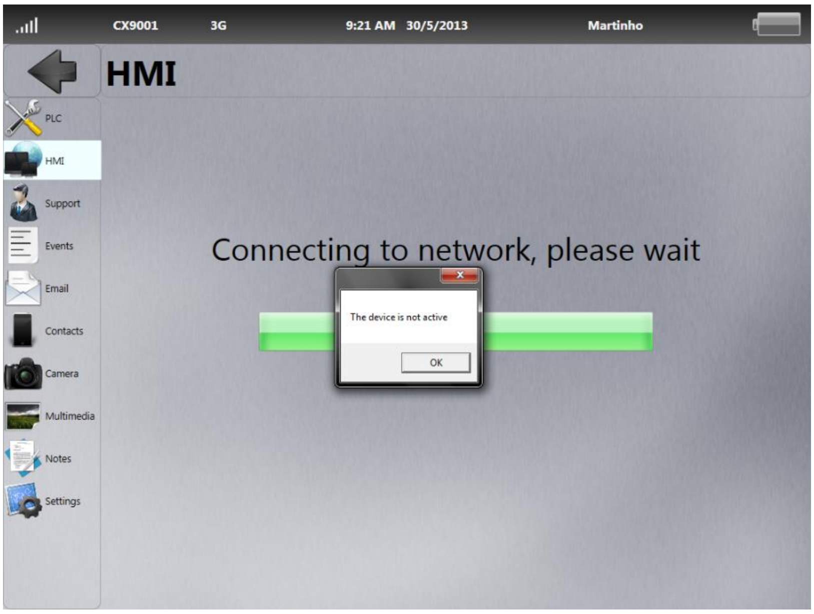

Figure 98 - SW Testing: HMI device is not active

5.2.3 Accessing to the PLC variables directly

With this test, it was verified that it is possible to access directly to the variables of the PLC from Beckhoff, using the ADS communication protocol.

The steps to make the test were: to load the **Access the PLC Module**, by clicking the Icon with the name PLC. Then to choose a machine, previously recorded, from the list and press the button with the name Access. After, the user selects the option to access directly to the PLC variables. A list of all the PLC variables is presented and the user selects which ones to monitor. Finally a HMI was created, so the variables were monitored and controlled.

The interaction with the software and the obtained results are shown in Figure 99.

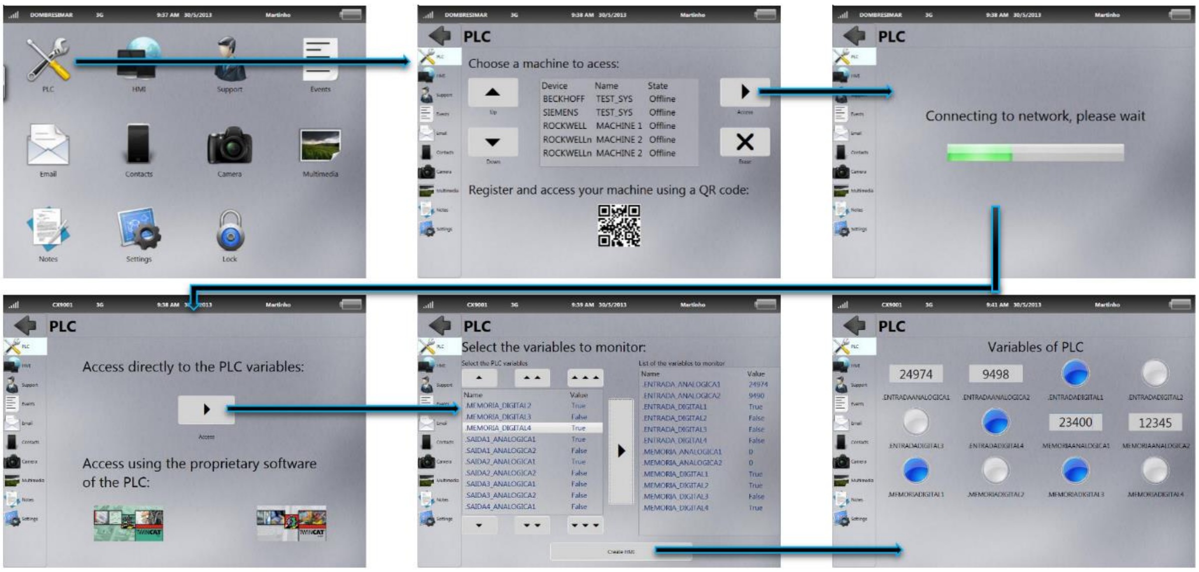

Figure 99 - SW Testing: Accessing PLC variables directly

It is possible to verify that the variables of the PLC were accessed and with them a HMI created.

It is then possible to change the value of the variables, as well to monitor their state.

In Figure 100 is presented the state of the HMI before and after the changes were made. The analogues were changed to the maximum and minimum, the digital inputs and digital memories were toggled, and the analogue memory registries were changed.

Figure 100 - SW Testing: Interacting with HMI created

In Figure 101, it is possible to verify that, if a variable is added again the user is notified, with a tooltip.

Figure 101 - SW Testing: Add again the variable to monitor

It is possible to verify with this test that the variables of the PLC were accessed, and a HMI was created in order to monitor and control them.

5.2.4 Accessing the PLC via proprietary Software

With this test, it was verified that it is possible to access to a PLC of Beckhoff through the proprietary software of the PLC.

The steps to make the test were: to load the **Access the PLC Module**, by clicking the Icon with the name PLC. Then choose a machine from the list and press the button with the name Access. After selected one of the options under the title with the content “Access using the proprietary software of the PLC”.

The interaction with the software and the obtained results are shown in Figure 102.

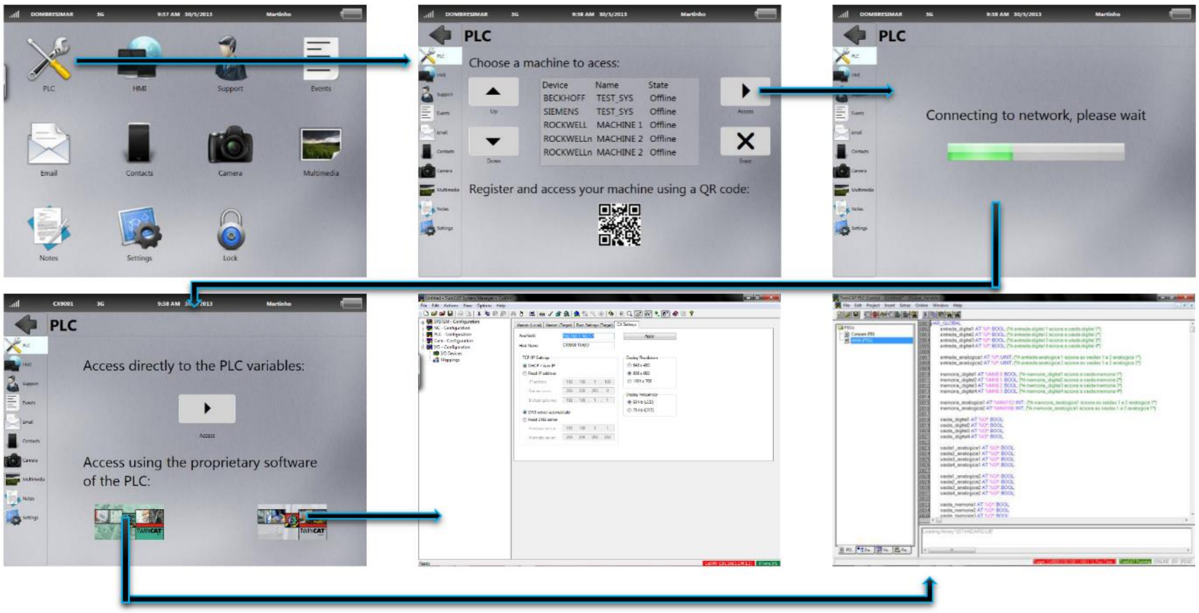

Figure 102 - SW Testing: Accessing PLC via Software

It is possible to verify that it was made an access to the PLC with the proprietary software TwinCAT System Manager. With this access it is possible to change PLC configurations directly from the Tablet PC.

It is also possible to verify that it was made an access to the PLC with the proprietary software TwinCAT PLC Control. With this access it is possible to reprogram the PLC or check the state of the PLC variables directly from the Tablet PC.

5.2.5 Making a video call and a voice call

With this test, it was verified that it is possible to make a video call, and a voice call, from the Tablet PC, using the proprietary software Skype.

The steps to make the test were: to load the **Technical Support Module**, by clicking the Icon with the name Support. Then choose a contact from the list and press the button with the name Video Call or Voice Call. The interaction with the software and the obtained results are shown in Figure 103.

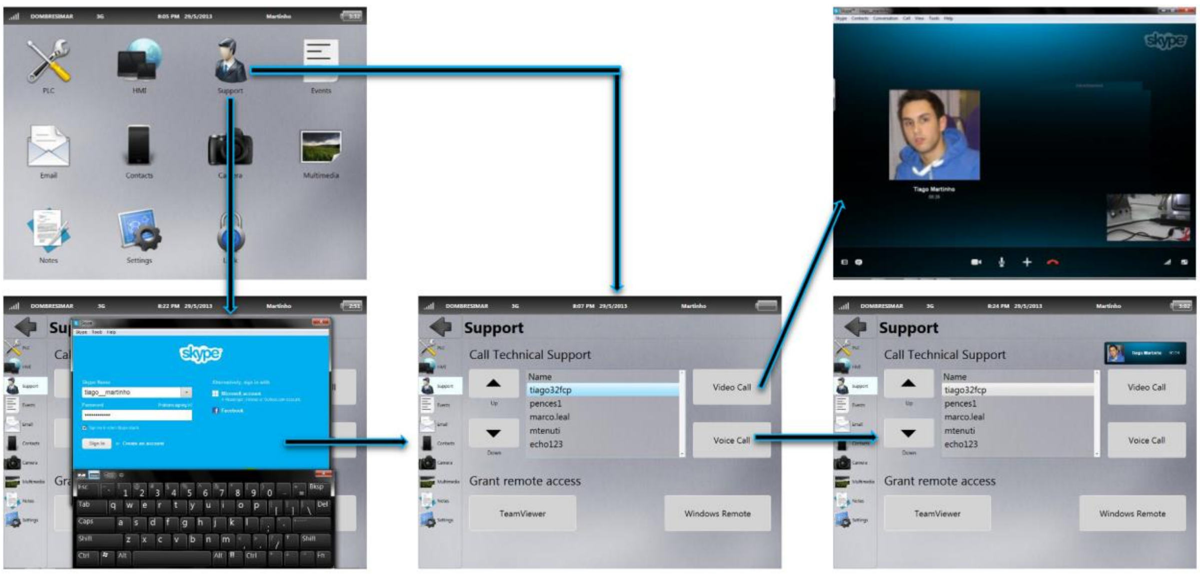

Figure 103 - SW Testing: Video Call and Voice Call

It is possible to verify from the test that a video call and voice call were established. Additionally it is possible to see that, if a login is not made in Skype, the credentials are asked and inserted via the virtual keyboard.

5.2.6 Providing Remote Access

With this test, it was verified that it is possible to provide remote access to the Tablet PC, using the proprietary software TeamViewer, in order to access the devices and reprogram the PLC.

The steps to make the test were: to load the **Technical Support Module**, by clicking the Icon with the name Support. Then, press the button with the TeamViewer name and allow remote access in the software TeamViewer.

The interaction with the software and the obtained results are shown in Figure 104.

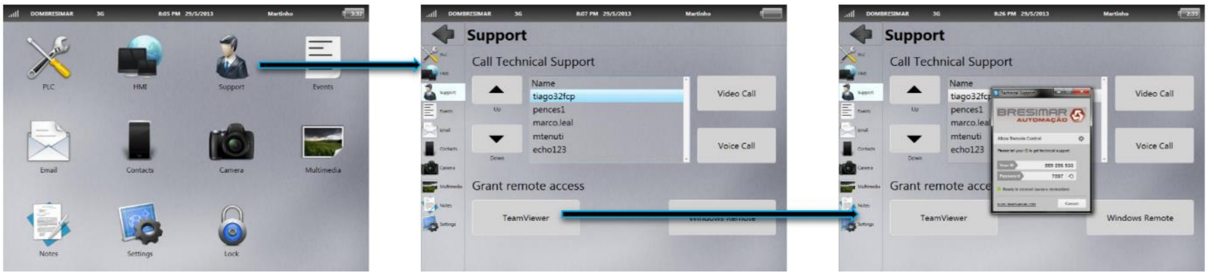

Figure 104 - SW Testing: Allowing Remote Access with TeamViewer

In the PC, where the remote access is done, it is possible to interact with the Tablet PC and access the software functions available, as seen in Figure 105. These functions include the sharing of files, video call, VoIP, etc.

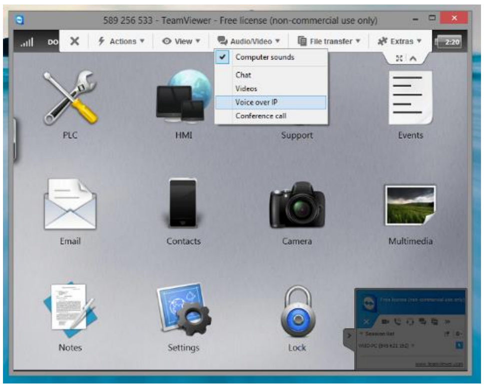

Figure 105 - SW Testing: TeamViewer Functions

It was also possible to access the HMI remotely by using the **Access the HMI Module**. In Figure 106, it is shown the remote access to the HMI.

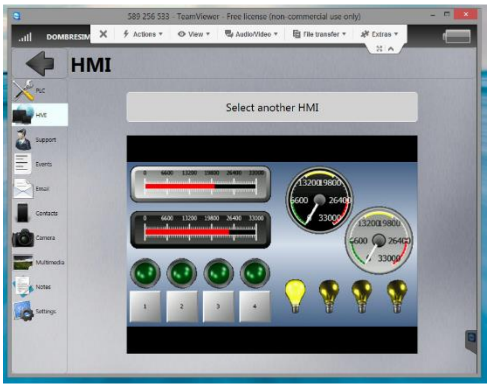

Figure 106 - SW Testing: Accessing the HMI remotely with TeamViewer

The software allowed to access the variables of the PLC remotely using the **Access the PLC Module** as presented in Figure 107.

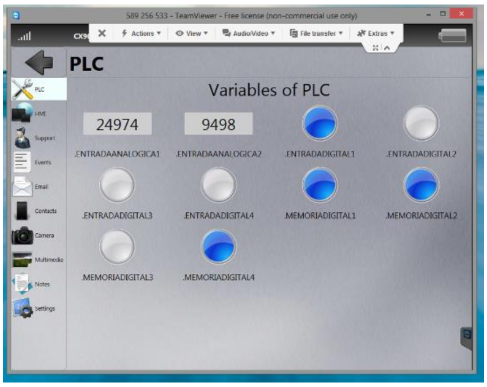

Figure 107 - SW Testing: Accessing the PLC variables remotely with TeamViewer

It furthermore allowed to reprogram the PLC remotely, by accessing the PLC with the proprietary software running on the Tablet PC. In Figure 108, it is shown the remote access where it is possible to download the code running in the PLC and edit it.

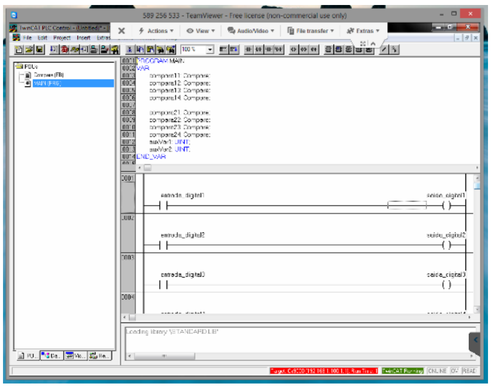

Figure 108 - SW Testing: Reprogramming the PLC remotely

The second mechanism to remotely access the Tablet PC was also tested. Using the Remote Desktop Connection is possible to have remote access to the Tablet PC, as it is presented in Figure 109. On the Tablet PC it is needed to press the button with the description "Windows Remote" and to acknowledge the message that appears.

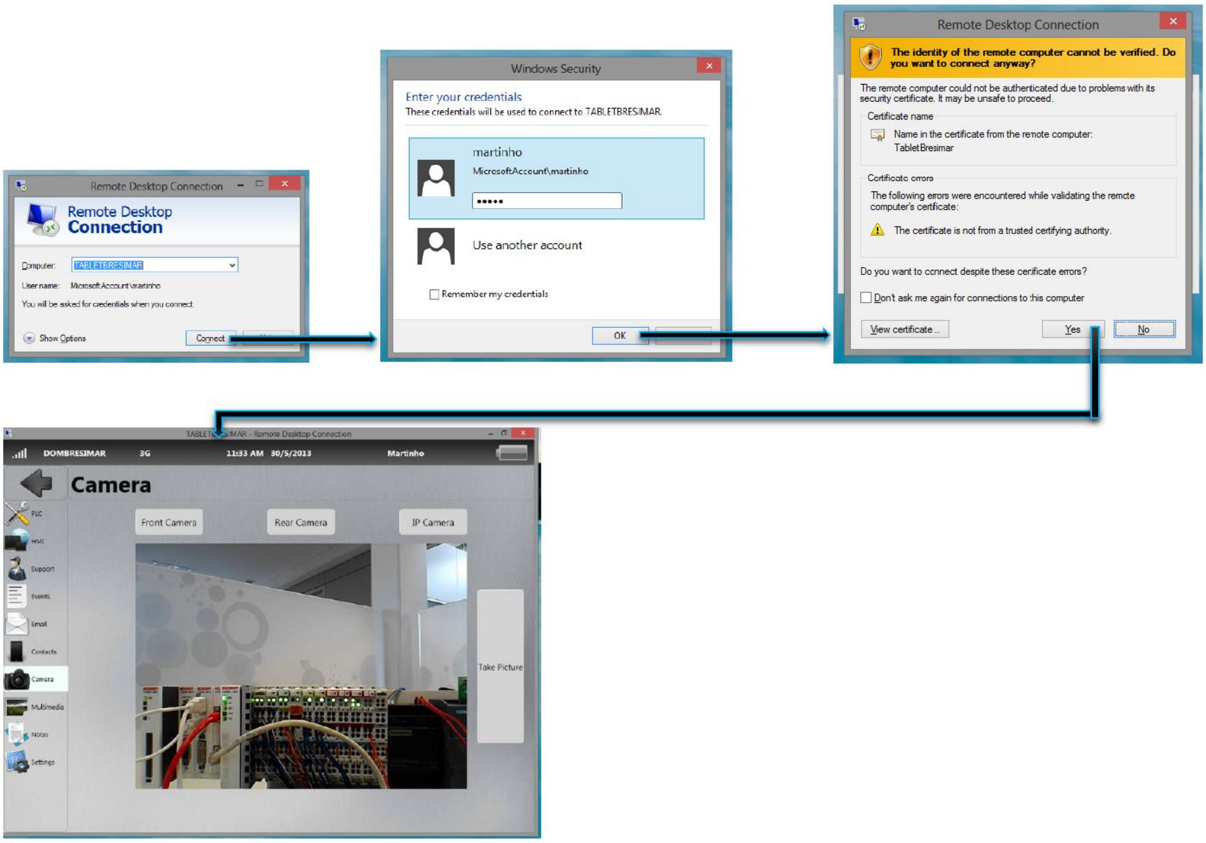

Figure 109 - SW Testing: Remote Desktop Connection

It is possible to verify that a remote access to the Tablet PC was possible in order to access the devices connected in the network created. Additionally, it was possible to reprogram remotely the PLC by accessing the Tablet PC.

5.2.7 Testing Camera and Multimedia

With this test, it was verified that it is possible to access to both cameras of the Tablet PC, as well to access the IP Camera. It was also verified that the pictures taken were accessible through the **Multimedia Module**.

The steps to make the test were: to load the **Camera Module**, by clicking the Icon with the name Camera. Then, it was tested the front camera by pressing the button with the name Take Picture. After that, the Multimedia Module was accessed, and it was confirmed that the photo had been stored. The obtained results are shown in Figure 110.

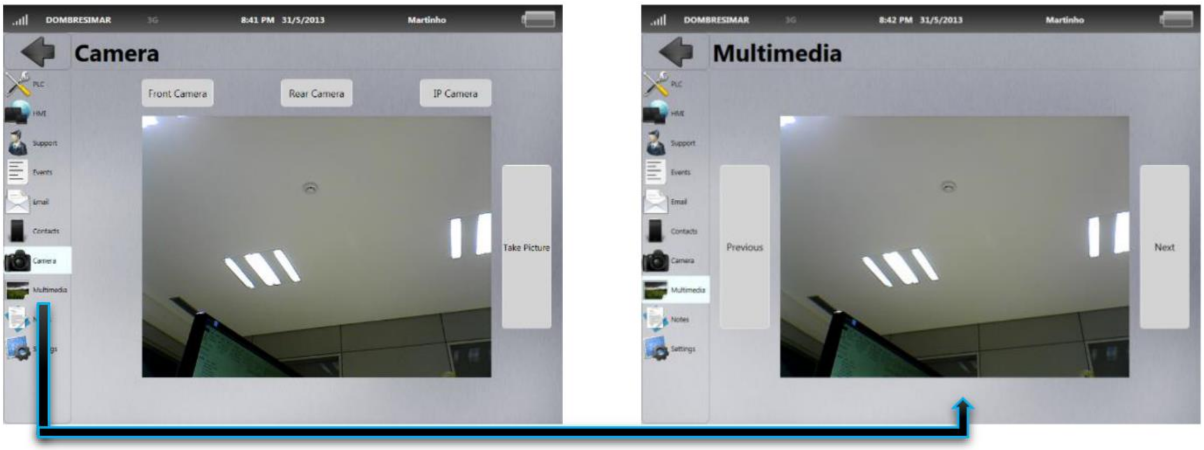

Figure 110 - SW Testing: Front Camera

Then, it was tested the rear camera by pressing the button with the name Rear camera and then the button with the name Take Picture. After that, the Multimedia Module was accessed and confirmed that the photo was stored and is accessible. The obtained results are shown in Figure 111.

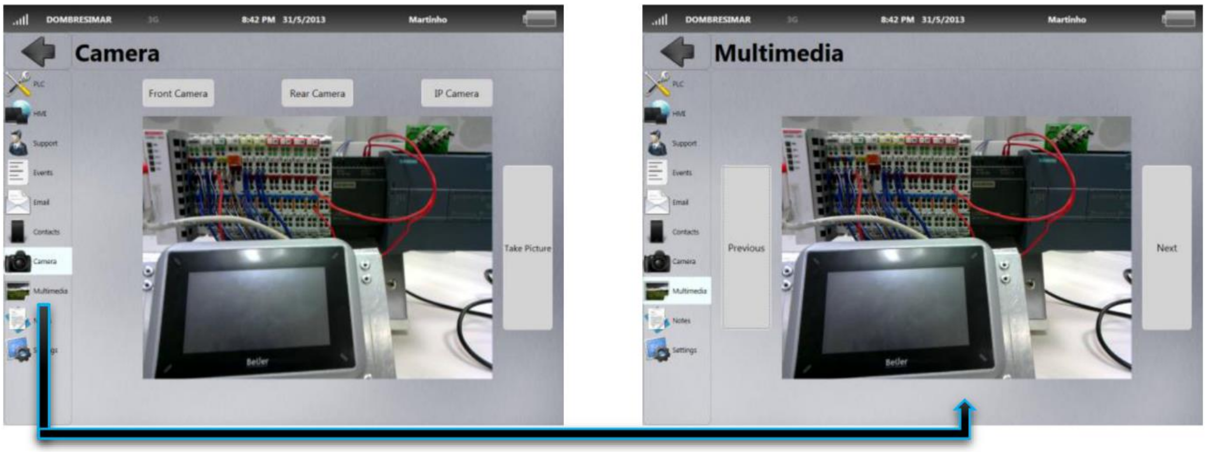

Figure 111 - SW Testing: Rear Camera

Regarding the IP Camera the communications settings were hard coded, and the Tablet PC should be in the network where the camera is present. For these two reasons, it was not included in the implementation part.

Following these conditions, an access was made and the buttons controlling the movement of the camera were successfully tested.

In Figure 112, it is possible to see an image with the user interface while accessing the IP camera.

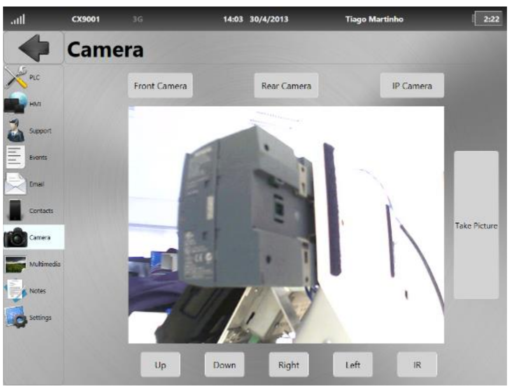

Figure 112 - SW Testing: IP Camera Access

A picture was saved and to verify it the **Multimedia Module** was accessed and navigated trough the gallery. It is possible to see the picture taken in Figure 113.

Figure 113 - SW Testing: IP Camera Photo

It is possible to verify with the tests made that it was possible to access and use the two cameras of the Tablet PC, as well to access, control and take a picture of the IP Camera connected in the WLAN.

5.3 Conclusion

In this chapter it was presented the test system developed in order to test the developed software. This system enabled to develop a set of tests to the software for a specific underlying hardware.

The different testing made to the developed software focused in the mains functions of the developed software. This test allowed to verify the developed software and in the next chapter it is possible to see an integration in a real field scenario.

Chapter 6

CASE STUDY

This chapter will present a real case scenario[6] where this thesis project can be inserted, and can be seen as the crucial test case, to validate the work in a real environment.

The chapter starts with a description of the industrial environment visited, followed by a description of how the integration with the existing system was made.

Next, it is presented the tests carried out to the developed software on the industrial field.

At the end of the chapter a reflection is made, in order to understand how the insertion of the device adds value to the current setup. In the conclusion, it is also stated the feedback received, by the field Engineers of the industrial plant.

6.1 INDUSTRIAL ENVIRONMENT DESCRIPTION

The visited industrial plant belongs to the company Bosch and it is located in Aveiro, Portugal [155]. Inside the plant there is a sector responsible for doing stress tests to boilers and water-heaters. The system developed in the scope of this thesis was tested in this sector. The main objective was to integrate the project with the existing system, so it was possible to validate the developed software.

The system responsible for the stress tests is composed by a main device, which acts as a control and data acquisition system, and a total of 24 functional testing stations, test rigs, connected to the main device.

The main device is an industrial PC (IPC), and it has associated a HMI, which can be remotely accessed. This device enables to: visualize and control settings, access all the variables of the process, set alarms and see trend graphs for each test rig.

Each test rig has associated a PLC, I/O modules and an Ethernet adapter. The functions of these devices are: to acquire the values of different testing variables (for example: temperature or pressure) with the I/O modules, process them with a program loaded in the PLC, and send the data to a database present in the main equipment using the Ethernet adapter.

In Figure 114, it is possible to see the Data Communication System, with the parts previously described. The test rigs, separated in two blocks of 8 and two blocks of 4, all connected to the Switch Layer 2. On the right the main device, the IPC, connected to the Switch Layer 2 and the Bosh Network Server. Finally the HMI, that is associated with the IPC, connected to the Switch Layer 2.

[6] I would like to show my gratitude to the Engineers in Bosch Aveiro: Sandra Fernandes and Reginaldo Ferreira for the availability and time spent, so it was possible to make the study presented in this chapter.

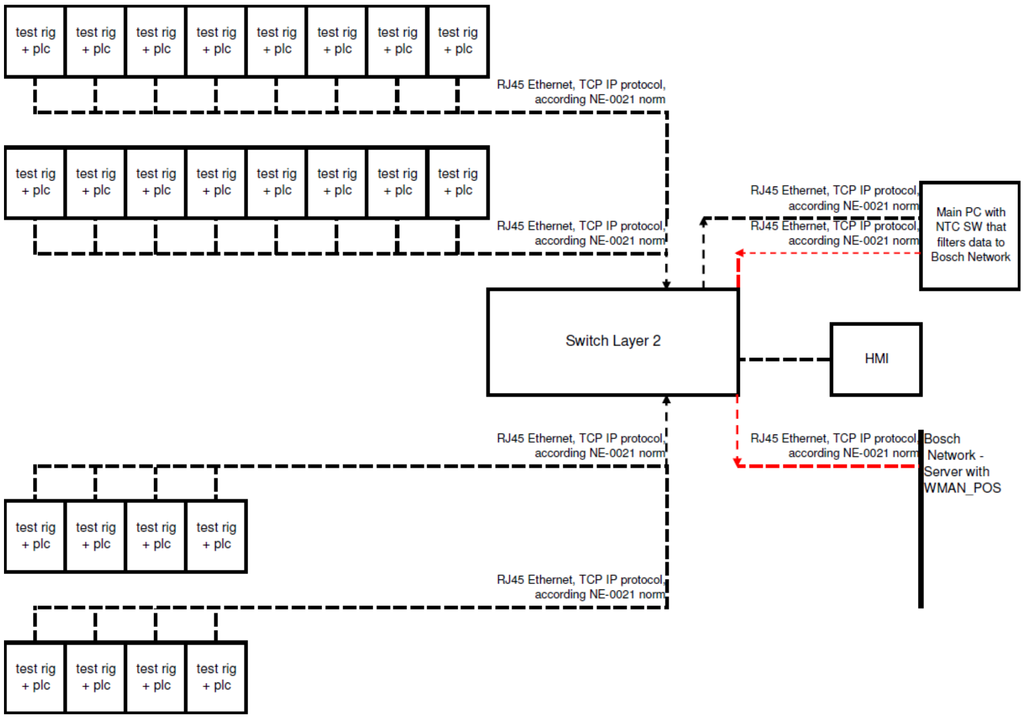

Figure 114 - Case Study: Data Communication System

Besides this system, there is a HMI that can connect directly to each test rig, which communicates via the Switch Layer2. This HMI is mobile, since it was encapsulated in a rugged casing and was added an adapter to connect to the power grid. To have a better understanding was taken a picture of the device, and it is shown in Figure 115.

Figure 115 - Case Study: Mobile HMI

6.2 Integration of the Project with Industrial System

As stated before, the objective is not to modify the existing system, but instead to integrate seamlessly this thesis project. Therefore, the configurations were only made on the wireless access point and the Tablet PC.

To integrate the developed application in this industrial system, the access point was connected to the previously shown industrial area network and configured with an IP address in the same range of the IPC, the main device.

The communication from the Tablet PC to the network, through the access point, was tested by pinging the different devices in the network. Then, the IPC was accessed, and registered, using the proprietary software from Beckhoff, TwinCAT System Manager [143].

After the communication was check, a QR code was encoded, containing the network information and the communications parameters of the devices, according with the previously explained section, the network information contains the SSID of the network created by the access point, and the according password. The communications parameters of the devices, contain the type of device and its name, along with the according IP address.

The devices on the specific network are the IPC, which contains the variables associated with all the test rigs, and the two HMI: the fixed one, connected to the main device, and the mobile one, that can connect to a test rig.

6.3 Field Tests

After the integration was successfully made, and the communications tested, different tests to the developed software were carried out.

The first functionality tested was the access to the variables of the PLC directly. It is possible to see from the Figure 116 that a list, containing all the variables of the PLC, was obtained successful. This step was achieved after recording the machine with the QR code created, and accessing the corresponding machine.

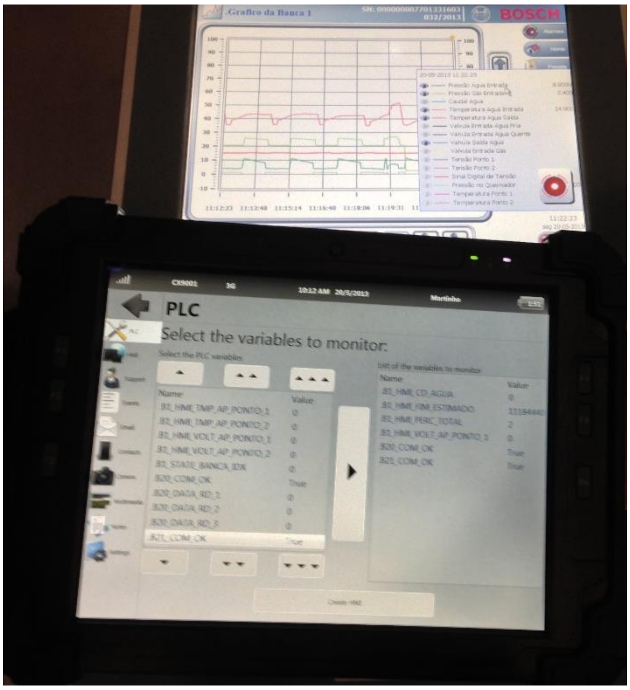

Figure 116 - Field Test to Access the PLC variables directly: Select Variables

In Figure 117, it is possible to see the HMI dynamically created with the selected variables. With this HMI it was possible to read and write the variables of the PLC.

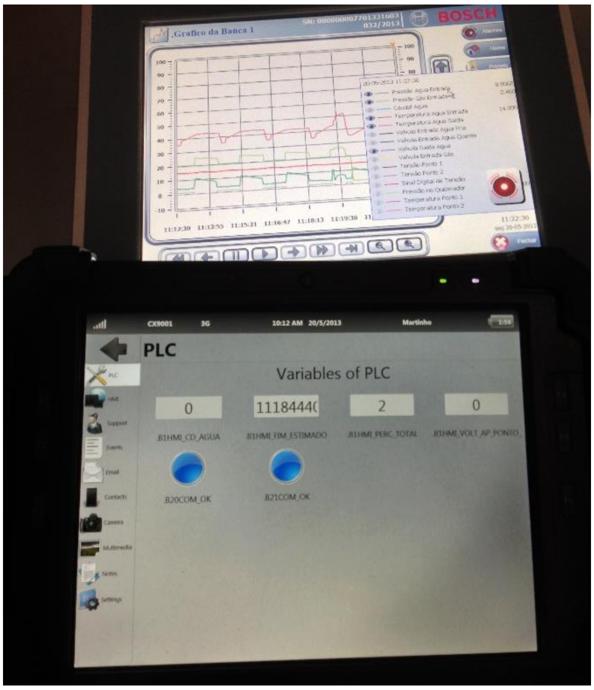

Figure 117 - Field Test to Access the PLC variables directly: Create HMI

The next functionality tested was the access to the PLC using the proprietary software. In this test the access was made using the proprietary software from Beckhoff, TwinCAT PLC Control. With this software was possible to download the source code from the PLC.

In Figure 118, it is presented the proprietary software running directly in the Tablet PC. It is possible to see from the picture that the program in the PLC was successfully downloaded to the Tablet PC. As seen in the Figure 118, the user has now the possibility to edit the source code.

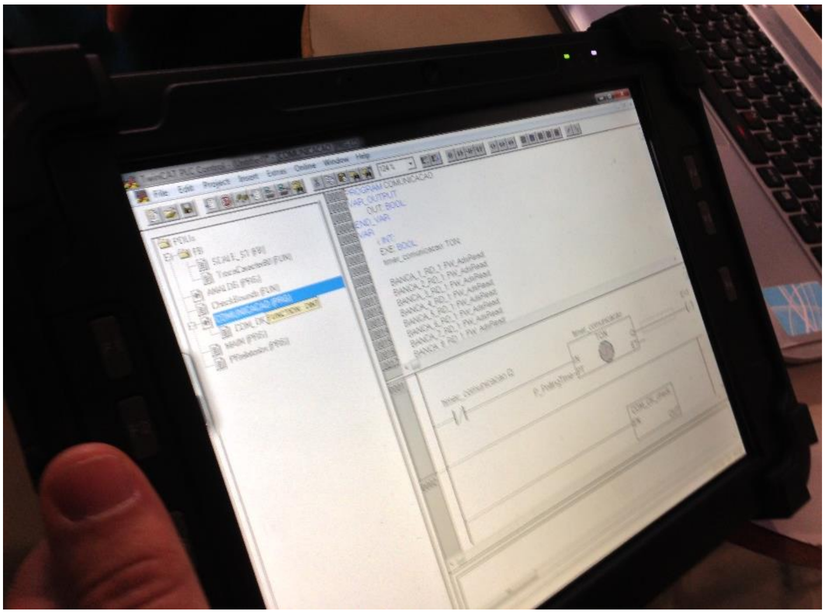

Figure 118 - Field Test to Access to the PLC using the proprietary software

The next tested functionality was the access to the HMI through VNC. Here a VNC connection was established, in order to have access to the HMI in the network remotely.

First it was made an access to the HMI connected to the main equipment. In the Figure 119 is possible to see the screen of the HMI on the top, and below the screen of the Tablet PC, where the remote access is being made.

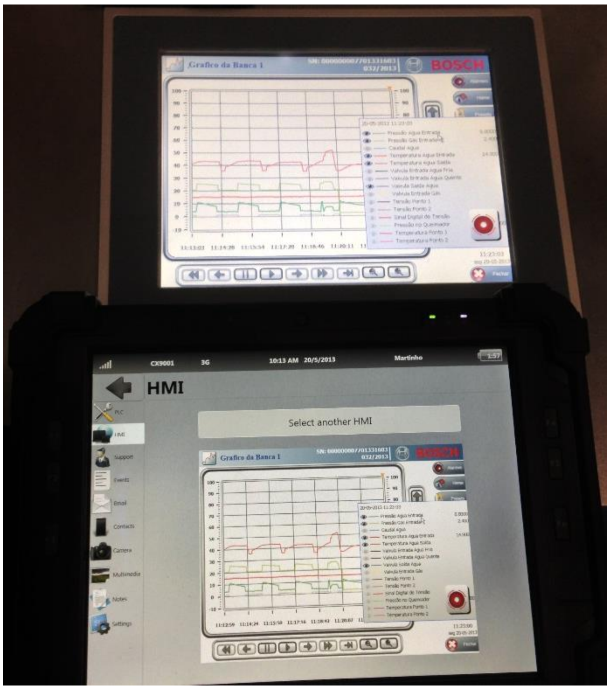

Figure 119 - Field Test to Access through VNC to the Main HMI

Second it was made an access to the mobile HMI connected to a test rig. In the Figure 120 it is possible to see the screen of the HMI on the right, and the screen of the Tablet PC on the left, where the remote access is being made.

Figure 120 - Field Test to Access through VNC to the Mobile HMI

It is possible to interact in the Tablet PC with the HMI remotely.

The next tested functionality was the technical support that can be provided with the Tablet PC. For the test of this functionality it was used the 3G connection of the Tablet PC.

First, it was tested the remote access to the Tablet PC. In the Figure 121 it is possible to see the access made with the PC, using the proprietary software TeamViewer. In the PC screen on the right, it is possible to see, and interact, with the screen of the Tablet PC, on the left.

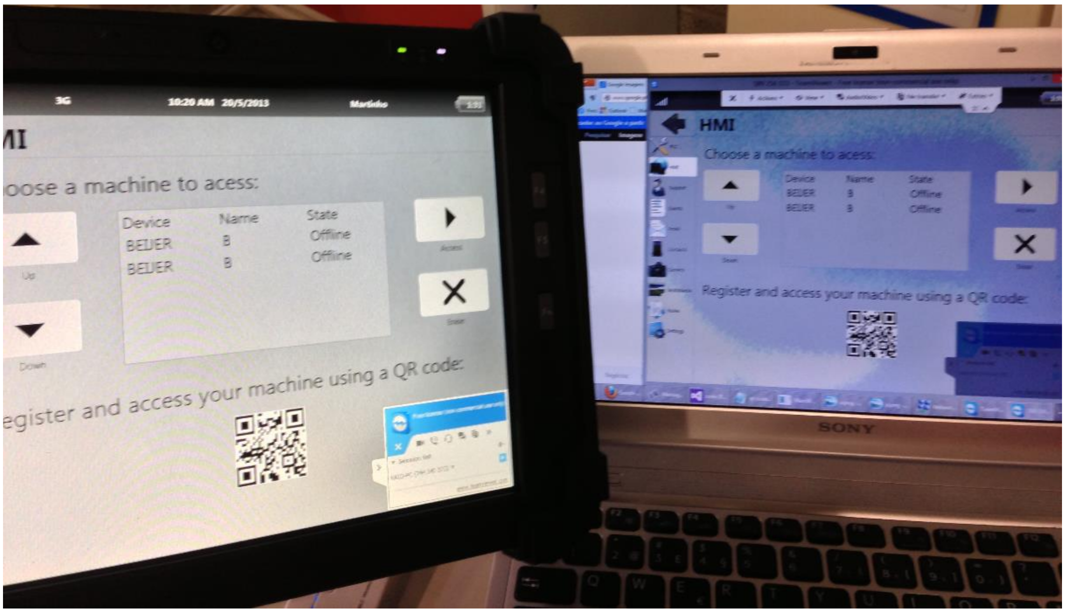

Figure 121 - Field Test to have Remote Access to the Tablet PC

After that it was tested the video call functionality from the Tablet PC. In the Figure 122 it is possible to see a video call established between the Tablet PC, on the left, and a PC, on the right.

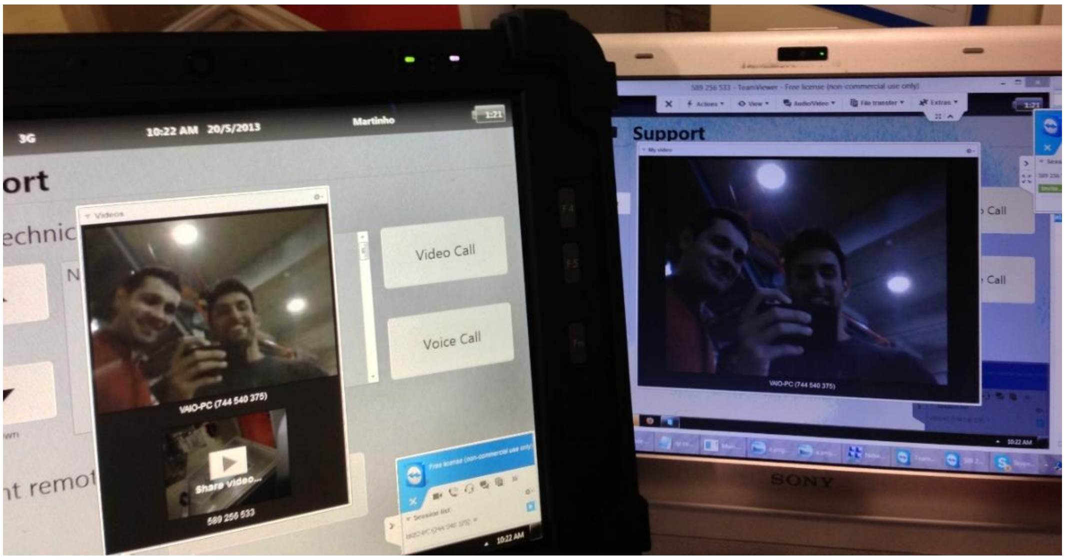

Figure 122 - Field Test to Make a Video Call from the Tablet PC

This ends the testing made on the industrial field. The other functions, implemented in the scope of this project, do not need to be tested on the field. Since the functions are, more or less, independent of the external conditions they are exposed. In this group fits the use of the camera and multimedia functions, as well, the possibility to take notes.

6.4 Conclusion

This case study allowed to make tests in a real field situation, and a validation, in a more concrete way.

This experiment allowed furthermore to receive feedback from a future possible user of the project developed. This feedback is very useful and so it is stated in the following topics:

- the access to the PLC variables directly seems interesting, but in practical terms does not have a great use unless it shows some trend graphs, and has the possibility to save information;
- the possibility to reprogram the PLC remotely is very interesting, and with a great practical use;
- the access to the HMI through VNC is good, and has a great practical use, it could be added the functionality to access it in Full Screen mode;
- the possibility to use the camera of the Tablet PC to take pictures, as well to have access to IP cameras, is very interesting, and with a good practical use;
- the access to the email would not have a great use, and could have problems of validation due to company policies;
- the possibility to take notes, and to record events, in the Tablet PC seems interesting, and with a practical use;
- the physical buttons of the Tablet PC could have default actions associated by the user;

In this particular case study, it was shown interest by the project, and stated that it provides a gain using an application like this one. The gain comes, not only by replacing the existing mobile HMI with a more practical solution, but also to add the new functions predicted in the scope of this thesis.

Chapter 7

CONCLUSION AND FUTURE WORK

This chapter concludes the work done in this thesis. A review of its content is going to be made. Next, it will be presented the main results achieved. Finally, it will be recommended future work for the project.

7.1 THESIS SUMMARY

This thesis started with the introduction of its theme and area of study. In here it was explained how the problems were discovered and stated the objectives for the project.

After the state-of-the-art study was presented, in Chapter 2, which included an analysis of theoretical and technological solutions, related with the theme of this project. The research presented registered patents, together with hardware and/or software solutions, provided by different companies in the industrial automation market.

In Chapter 3, the problem and proposed solution of the thesis were clearly formulated. In this chapter were also defined a set of use cases, requirements and UI mock-ups for the software project. Moreover, was defined the system architecture independently of the specific underlying hardware or software framework chosen.

Following, in Chapter 4, it was described the system development, including the decisions taken in the choice of the software framework and mobile device, as well the implementation of the different software functions.

Next, in Chapter 5, it was made the system validation. Here, it was described the integration of different equipment in order to develop a test system, which simulates a real field situation. The hardware integrated allowed to evaluate the different software functions developed.

This thesis included also a case study, in Chapter 6, where the project was applied to a real scenario in the industrial automation sector. The tests conducted on the industrial field allowed to validate the system developed, as well to get feedback of a possible future user of the project.

7.2 THESIS RESULTS

In the competitor analysis it was concluded that, at the moment of writing this thesis, no commercial solution offers a modular approach where the hardware and software are brought together, in order to create new technological tools in the industrial automation area. The thesis is innovative, not in terms of technology, but in terms of use case. By inserting already existing technology in industrial automation, it can lead to a new product category in the field, allowing functions like the reprogramming of machines remotely, or technical support via video call on industrial plant.

In the System Requirements and System Architecture, the use cases and requirements defined help to avoid future implementation problems and to explain the system in a precise way. The fact that the project was discussed in several meetings with the engineering team of Bresimar helped to establish the system basis, without being narrowed into implementation details.

The advancements in the System Development provides a modular basis so each software and functionality can now be tested, and developed, in an independent way by many developers. It allows the potential to develop different functions and attach them in the application. The advantage comes in bringing a faster and more flexible software development.

The prototype developed during the thesis was validated on a real situation of industrial automation area. The feedback received from a possible future user of the project was useful to evaluate the potential of the idea, as well to define a guideline for future improvements.

7.3 Future Work

Despite the developed work and the results achieved with the available time, there is still some key points that could improve the working prototype.

In this scope, the future work should include the following tasks:

- Describe different kinds of software testing for each software module defined. This includes automated tests to individual functions, so it improves the robustness of the software. It includes as well usability testing with industrial operators, so it improves the ease of use of the product;
- To develop a more uniform and engaging user interface, involving designers and customer feedback, in order to improve the overall user experience of the product;
- Add support for a non-proprietary protocol, for example, Modbus, so it increase the range of industrial applications where the project can be inserted;
- Improve the functionality to access the PLC variables directly, by adding trend graphs, the possibility to store information, and add more objects to create the HMI;

The development of the other modules predicted should also be included in the future work of the project. It is then needed to specify the functional requirements and use cases, as well to extend the class diagram for the modules to develop.

The future work can also follow a totally different approach. Having an architecture where the Tablet PC acts only as a terminal and the software is running in the cloud, accessible via a mobile broadband connection. This approach brings some trades-offs that could be examined in future development. To run the software via a mobile connection, it is needed an adequate broadband connection. The advantage is that the software is shareable through the devices, allowing a flexible and direct control over the functions provided in the industrial plant. By having this multi-layer architecture, it is possible to improve security, as well to run the software in a multitude of devices with different hardware architectures.

REFERENCES

[1] M. Meeker, "Internet Trends," in *D: All Things Digital*, California, 2013.

[2] Vision Mobile, "Developer Economics," BlueVia, London, 2012.

[3] G. Rieger and U. Sinn, "Mobile control and monitoring system". Patent US 7657492 B2, 2 February 2010.

[4] R. J. Kretschmann, "Mobile human/machine interface for use with industrial control systems for controlling the operation of process executed on spatially separate machines". Patent US 6167464 A, 26 December 2000.

[5] R. Mody and G. Rao, "Tablet computer system for industrial process design, supervisory control, and data management". Patent US 20040201602 A1, 14 October 2004.

[6] Sintesi, "Shd: Sintesi HMI Device," [Online]. Available: http://www.sintesi-am.eu/index.php?option=com_content&view=article&id=82&Itemid=116&lang=en. [Accessed 21 May 2013].

[7] Microsoft, "Welcome to Windows CE," [Online]. Available: http://msdn.microsoft.com/en-us/library/ms905511.aspx. [Accessed 21 May 2013].

[8] Automation.com, "Introduction to Profibus-DP," [Online]. Available: http://www.automation.com/library/articles-white-papers/fieldbus-serial-bus-io-networks/introduction-to-profibus-dp. [Accessed 21 May 2013].

[9] EtherCAT technology Group, "EtherCAT," [Online]. Available: http://www.ethercat.org/en/ethercat.html. [Accessed 21 May 2013].

[10] The Modbus Organization, "Modbus FAQ," [Online]. Available: http://www.modbus.org/faq.php. [Accessed 21 May 2013].

[11] CAN in Automation, "CANopen," [Online]. Available: http://www.can-cia.org/?canopen. [Accessed 21 May 2013].

[12] Rockwell Automation, "System Design According to IEC/EN 62061," [Online]. Available: http://www.ab.com/en/epub/catalogs/3377539/5866177/3378076/7555771/#]. [Accessed 21 May 2013].

[13] International Organization for Standardization, "ISO 13849-1:2006," [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=34931. [Accessed 21 May 2013].

[14] Sintesi , "shd_1.png," [Online]. Available: http://www.sintesi-am.eu/images/stories/prodotti/shd%201.png. [Accessed 21 May 2013].

[15] Siemens, "SIMATIC Mobile Panels," [Online]. Available: http://www.automation.siemens.com/mcms/human-machine-interface/en/operator-interfaces/mobile-panel/Pages/Default.aspx. [Accessed 21 May 2013].

[16] Siemens, “SCADA System SIMATIC WinCC,” [Online]. Available: http://www.automation.siemens.com/mcms/human-machine-interface/en/visualization-software/scada/Pages/Default.aspx. [Accessed 21 May 2013].

[17] Siemens, “SIMATIC Mobile Panel 277 IWLAN Version 2,” [Online]. Available: http://www.automation.siemens.com/mcms/human-machine-interface/en/operator-interfaces/mobile-panel/series-270/simatic-mobile-panel-277-iwlan/pages/default.aspx. [Accessed 21 May 2013].

[18] PROFIBUS and PROFINET International Organization, “ProfiNET Overview,” [Online]. Available: http://www.profibus.com/technology/profinet/overview/. [Accessed 21 May 2013].

[19] MPL , “IP Rating Table,” [Online]. Available: http://www.mpl.ch/info/IPratings.html. [Accessed 21 May 2013].

[20] Siemens, “mobile-panel-277-f-iwlan-172.jpg,” [Online]. Available: http://www.automation.siemens.com/mcms/human-machine-interface/en/operator-interfaces/mobile-panel/series-270/simatic-mobile-panel-277-iwlan/PublishingImages/mobile-panel-277-f-iwlan-172.jpg. [Accessed 21 May 2013].

[21] Invensys, “Datasheet_Wonderware_TabletComputerSeriesD_01_12.pdf,” [Online]. Available: http://global.wonderware.com/EN/PDF%20Library/Datasheet_Wonderware_TabletComputersSeriesD_01-12.pdf. [Accessed 21 May 2013].

[22] Beijer Electronics, “TREQ-VMx - Powerful and Versatile Mobile Data Terminals (MDT),” [Online]. Available: http://www.beijerelectronicsinc.com/product/vehicle/treq-vm-vmx.php. [Accessed 21 May 2013].

[23] Bosch, “CAN Specification,” [Online]. Available: http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can2spec.pdf. [Accessed 21 May 2013].

[24] Beijer Electronics, “treqvm_vmx_duo-2.jpg,” [Online]. Available: http://beijerinc.com/img/product/gallery/treqvm_vmx_duo-2.jpg. [Accessed 21 May 2013].

[25] Motorola Solutions, “Enterprise Tablet,” [Online]. Available: http://www.ab.com/db/encompass/bps_dir_update.show_page?x_directory_id=8932&x_region=AMERICAS. [Accessed 21 May 2013].

[26] Motorola Solutions, “49408.jpg,” [Online]. Available: http://www.ab.com/encompass/secure/extranet/images/company/49408.jpg. [Accessed 21 May 2013].

[27] J.A.S. Automação Industrial, “IHM OPEN,” [Online]. Available: http://jasautomacao.com.br/ihmopen.html. [Accessed 21 May 2013].

[28] J.A.S. Automação Industrial, “IHM.png,” [Online]. Available: http://jasautomacao.com.br/images/IHM.png. [Accessed 21 May 2013].

[29] Microsoft, “Virtual Private Networking: An Overview,” [Online]. Available: http://technet.microsoft.com/en-us/library/bb742566.aspx. [Accessed 21 May 2013].

[30] mySCADA technologies, “mySCADA Mobile,” [Online]. Available: http://www.myscada.org/wp/myscada-mobile/. [Accessed 21 May 2013].

[31] Rockwell Automation, “EtherNet/IP Network,” [Online]. Available: http://ab.rockwellautomation.com/Networks-and-Communications/Ethernet-IP-Network. [Accessed 21 May 2013].

[32] Siemens , “S7 PLC controllers,” [Online]. Available: http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/Pages/Default.aspx. [Accessed 21 May 2013].

[33] Apple, “mySCADA mzl.vlljnozj.320x480-75.jpg,” [Online]. Available: http://a1907.phobos.apple.com/us/r1000/095/Purple2/v4/24/bf/4d/24bf4d74-4e6d-a3be-230a-00a86ad5e6a9/mzl.vlljnozj.320x480-75.jpg. [Accessed 21 May 2013].

[34] Opto 22, “Opto iPAC and Opto aPAC,” [Online]. Available: http://www.opto22.com/lp/opto_ipac.aspx. [Accessed 21 May 2013].

[35] Apple, “Opto iPac mzl.oknwrfyd.320x480-75.jpg,” [Online]. Available: http://a337.phobos.apple.com/us/r1000/088/Purple/v4/fe/af/0e/feaf0e15-ed2b-3168-ba5f-4774141879d1/mzl.oknwrfyd.320x480-75.jpg. [Accessed 21 May 2013].

[36] Google, “Modbus Monitor,” [Online]. Available: https://play.google.com/store/apps/details?id=com.Bhavan.Hubble&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5CaGF2YW4uSHViYmxlIl0.]. [Accessed 21 May 2013].

[37] Google, “modbusMonitor.jpg,” [Online]. Available: https://lh4.ggpht.com/lN-FUIpPAfHPY86vbteWCaRM7St6H_4rnzUBH3uniWP0QJMLcgq8rwnF43r7jYM4djM. [Accessed 21 May 2013].

[38] Inductive Automation, “Ignition Overview,” [Online]. Available: http://www.inductiveautomation.com/scada-software;jsessionid=52DE1F793C8CC24931FC7988D2A4C960. [Accessed 21 May 2013].

[39] OPC Foundation, “ OPC Unified Architecture,” [Online]. Available: https://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC. [Accessed 21 May 2013].

[40] OPC Foundation, “What is OPC?,” [Online]. Available: http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC. [Accessed 21 May 2013].

[41] Inductive Automation, “expand.png,” [Online]. Available: http://www.inductiveautomation.com/resources/com.inductiveautomation.website.BasePage/images/products/ignitionoverview/expand.png. [Accessed 21 May 2013].

[42] MBLogic, “MB HMI - Intro,” [Online]. Available: http://mblogic.sourceforge.net/mbhmi/mbhmiintro.html. [Accessed 21 May 2013].

[43] Likindoy, “Cascadas - Protocol for Web Based HMI,” [Online]. Available: http://www.likindoy.org/wiki/index.php/Cascadas_-_Protocol_for_Web_Based_HMI. [Accessed 21 May 2013].

[44] International Engineering Consortium, “Voice over Internet Protocol,” [Online]. Available: http://www.teliqo.com/voip/. [Accessed 21 May 2013].

[45] Borland, “Sucessful Projects Begin with Well-Defined Requirements,” Ziff Davis Media, 2007.

[46] Microsoft, “Skype,” [Online]. Available: http://www.skype.com/en/. [Accessed 21 May 2013].

[47] Techcrunch, "Windows 8: You'll Absolutely Hate It At First (But Give It A Chance Anyway)," [Online]. Available: http://techcrunch.com/2012/06/17/windows-8-youll-absolutely-hate-it-at-first-but-give-it-a-chance-anyway/. [Accessed 21 May 2013].

[48] Apple, "iOS Human Interface Guidelines," [Online]. Available: http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html. [Accessed 21 May 2013].

[49] Android Developers, "Design Principles," [Online]. Available: http://developer.android.com/design/get-started/principles.html. [Accessed 21 May 2013].

[50] J. Wilk, "Unlocking the Proprietary Mess," *GE ESL Magazine,* p. 8, 2005.

[51] The Modbus Organization, "Modbus Home Page," [Online]. Available: http://www.modbus.org/. [Accessed 21 May 2013].

[52] Ubiqconn, "prod_43.jpg," [Online]. Available: http://www.ubiqconn.com/Modules/Product/prod_43_1_S-201303081055172740.jpg. [Accessed 21 May 2013].

[53] Asus, "P_500.jpg," [Online]. Available: http://pt.asus.com/websites/Global/products/O7MOua9hMhFNXeUh/P_500.jpg. [Accessed 21 May 2013].

[54] Foscam, "Foscam FI8910W Wireless IP Camera," [Online]. Available: http://foscam.us/foscam-fi8910w-wireless-ip-camera.html. [Accessed 21 May 2013].

[55] Beijer Electronics, "TxA-4_front.jpg," [Online]. Available: http://www.beijerelectronics.com/web/imagebank.nsf/0/D1E73E6AB5DDF85CC12578C4002F4C49/$file/be_480px_TxA-4_front.jpg. [Accessed 21 May 2013].

[56] Beckhoff, "CX9000__web.jpg," [Online]. Available: http://www.beckhoff.com/images/embedded_pc/CX9000__web.jpg. [Accessed 21 May 2013].

[57] RTC Magazine, "intel1_large.jpg," [Online]. Available: http://www.rtcmagazine.com/files/images/1322/rtc1008td_intel1_large.jpg. [Accessed 21 May 2013].

[58] Slashdot, "How Do You Decide Which Framework to Use?," [Online]. Available: http://slashdot.org/story/06/02/24/0148201/how-do-you-decide-which-framework-to-use. [Accessed 21 May 2013].

[59] G. Galvin, "Which Software Framework Should You Choose?," [Online]. Available: http://blog.galvintech.com/application-development/which-software-framework-should-you-choose-478.htm. [Accessed 21 May 2013].

[60] Smashing Magazine, "Frameworks Round-Up: When To Use, How To Choose?," [Online]. Available: http://coding.smashingmagazine.com/2008/01/04/frameworks-round-up-when-to-use-how-to-choose/. [Accessed 21 May 2013].

[61] W3Schools, "OS Platform Statistics and Trends," [Online]. Available: http://www.w3schools.com/browsers/browsers_os.asp. [Accessed 21 May 2013].

[62] Net Applications, "Desktop Operating System Market Share," [Online]. Available: http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0&qptimeframe=M. [Accessed 21 May 2013].

[63] Microsoft, “Overview of the .NET Framework,” [Online]. Available: http://msdn.microsoft.com/en-us/library/zw4w595w.aspx. [Accessed 21 May 2013].

[64] Beckhoff , “The Windows Control and Automation Technology,” [Online]. Available: http://www.beckhoff.com/english.asp?twincat/default.htm?id=347915987603. [Accessed 21 May 2013].

[65] Rockwell Automation, “IntelliCENTER Software,” [Online]. Available: http://literature.rockwellautomation.com/idc/groups/literature/documents/br/mcc-br003_-en-p.pdf. [Accessed 21 May 2013].

[66] Siemens, “Simatic Industrial Software,” [Online]. Available: http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-industrial-software_en.pdf. [Accessed 21 May 2013].

[67] Microsoft, “MSDN Library,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms123401.aspx. [Accessed 21 May 2013].

[68] Ubiqconn, “uTablet T10C Specifications,” [Online]. Available: http://www.ubiqconn.com/Default.aspx?Name=Product&Op=ProductD&pid=43&parentid=7. [Accessed 21 May 2013].

[69] Ubiqconn, “Product Document,” [Online]. Available: http://www.ubiqconn.com/DispProductPic.aspx?pid=43&index=1. [Accessed 21 May 2013].

[70] Microsoft, “Windows Presentation Foundation,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms754130.aspx. [Accessed 21 May 2013].

[71] Microsoft, “.NET Framework 4,” [Online]. Available: http://msdn.microsoft.com/en-us/library/w0x726c2%28v=vs.100%29.aspx. [Accessed 21 May 2013].

[72] Microsoft, “XAML Overview (WPF),” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms752059.aspx. [Accessed 21 May 2013].

[73] Microsoft, “Data Binding Overview,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms752347.aspx. [Accessed 21 May 2013].

[74] Microsoft, “Visual C#,” [Online]. Available: http://msdn.microsoft.com/en-us/library/vstudio/kx37x362.aspx. [Accessed 21 May 2013].

[75] Microsoft, “Visual Studio Resources,” [Online]. Available: http://msdn.microsoft.com/en-us/vstudio/cc136611.aspx. [Accessed 21 May 2013].

[76] P. Stovell, “WPF Navigation,” [Online]. Available: http://paulstovell.com/blog/wpf-navigation. [Accessed 21 May 2013].

[77] Microsoft, “Application Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.application.aspx. [Accessed 21 May 2013].

[78] Microsoft, “NavigationWindow Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.navigation.navigationwindow.aspx. [Accessed 21 May 2013].

[79] Microsoft, “Window Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.window.aspx. [Accessed 21 May 2013].

[80] Microsoft, “Page Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.controls.page.aspx. [Accessed 21 May 2013].

[81] Microsoft, “Controls,” [Online]. Available: http://msdn.microsoft.com/en-us/library/bb613551.aspx. [Accessed 21 May 2013].

[82] Microsoft, “UserControl Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.controls.usercontrol.aspx. [Accessed 21 May 2013].

[83] Microsoft, “Create a WPF user control,” [Online]. Available: http://msdn.microsoft.com/en-us/library/cc294992.aspx. [Accessed 21 May 2013].

[84] Microsoft, “How to: Create Multi-Project Solutions,” [Online]. Available: http://msdn.microsoft.com/en-us/library/vstudio/h970wzkb.aspx. [Accessed 21 May 2013].

[85] Akadia, “Delegates and Events in C# / .NET,” [Online]. Available: http://www.akadia.com/services/dotnet_delegates_and_events.html. [Accessed 21 May 2013].

[86] kovacsz1, “How to start a program automatically in Windows,” [Online]. Available: http://letitknow.wordpress.com/2012/09/06/how-to-start-a-program-automatically-in-windows-8/. [Accessed 21 May 2013].

[87] Microsoft, “StatusBar Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.controls.primitives.statusbar.aspx. [Accessed 21 May 2013].

[88] Microsoft, “StatusBarItem Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.controls.primitives.statusbaritem.aspx. [Accessed 21 May 2013].

[89] Microsoft, “DispatcherTimer Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.threading.dispatchertimer.aspx. [Accessed 22 May 2013].

[90] ikonst, “Managed Wifi API,” [Online]. Available: http://managedwifi.codeplex.com/. [Accessed 21 May 2013].

[91] Microsoft, “About the Native Wifi API,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms705969.aspx. [Accessed 21 May 2013].

[92] Actipro Software Support , “Cell phone signal strength (5 bars),” [Online]. Available: http://www.actiprosoftware.com/community/thread/20132/cell-phone-signal-strength-5-bars. [Accessed 21 May 2013].

[93] ikonst, “Wifi Examples cs,” [Online]. Available: http://managedwifi.codeplex.com/SourceControl/changeset/view/69709#213980. [Accessed 21 May 2013].

[94] Microsoft, “NetworkInterface Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.net.networkinformation.networkinterface.aspx. [Accessed 21 May 2013].

[95] Microsoft, “Win32_Battery class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394074%28v=vs.85%29.aspx. [Accessed 21 May 2013].

[96] K. Patel, “Get Laptop Battery Status using C# and WMI,” [Online]. Available: http://www.c-sharpcorner.com/uploadfile/kirtan007/get-laptop-battery-status-using-C-Sharp-and-wmi/.. [Accessed 21 May 2013].

[97] R. Kumar, “How to get System Environment information in WPF,” [Online]. Available: http://www.c-sharpcorner.com/uploadfile/raj1979/how-to-get-system-environment-information-in-wpf/. [Accessed 21 May 2013].

[98] Microsoft, “Application.SetSuspendState Method,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.forms.application.setsuspendstate.aspx. [Accessed 24 May 2013].

[99] Find Icons, “Icon Search Engine,” [Online]. Available: http://findicons.com/. [Accessed 21 May 2013].

[100] Microsoft, “WPF Globalization and Localization Overview,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms788718%28v=vs.100%29.aspx. [Accessed 21 May 2013].

[101] M. Pearl, “5 Lessons learnt in localization / multi language support in WPF,” [Online]. Available: http://geekswithblogs.net/MarkPearl/archive/2011/03/01/5-lessons-learnt-in-localization--multi-language-support-in.aspx. [Accessed 21 May 2013].

[102] Microsoft, “ResourceDictionary Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.resourcedictionary.aspx. [Accessed 21 May 2013].

[103] Microsoft, “How to: Use a ResourceDictionary to Manage Localizable String Resources,” [Online]. Available: http://msdn.microsoft.com/en-us/library/bb613547.aspx. [Accessed 21 May 2013].

[104] Skype , “Skype Developer - Skype URIs,” [Online]. Available: http://dev.skype.com/skype-uri. [Accessed 23 May 2013].

[105] Skype , “Skype Developrs - Skype API on Windows,” [Online]. Available: http://dev.skype.com/desktop-api-reference#Windows. [Accessed 23 May 2013].

[106] Skype, “Skype Developers - Skype4COM,” [Online]. Available: http://dev.skype.com/accessories/skype4com. [Accessed 23 May 2013].

[107] Microsoft, “ProcessStartInfo Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.diagnostics.processstartinfo.aspx. [Accessed 23 May 2013].

[108] Microsoft, “Process Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.diagnostics.process.aspx. [Accessed 23 May 2013].

[109] TeamViewer, “TeamViewer Custom Design,” [Online]. Available: https://www.teamviewer.com/en/download/customdesign.aspx. [Accessed 23 May 2013].

[110] Microsoft, “Windows Desktop Sharing,” [Online]. Available: http://msdn.microsoft.com/en-us/library/bb968809.aspx. [Accessed 23 May 2013].

[111] Microsoft, “System Services,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ee663297%28v=vs.85%29.aspx. [Accessed 23 May 2013].

[112] Microsoft, “About Windows Desktop Sharing,” [Online]. Available: http://msdn.microsoft.com/en-us/library/aa373260%28v=vs.85%29.aspx. [Accessed 23 May 2013].

[113] termserv, “Windows Desktop Sharing API,” [Online]. Available: http://blogs.msdn.com/b/rds/archive/2007/03/08/windows-desktop-sharing-api.aspx. [Accessed 23 May 2013].

[114] termserv, “Writing a Desktop Sharing Application,” [Online]. Available: http://blogs.msdn.com/b/rds/archive/2007/03/23/writing-a-desktop-sharing-application.aspx. [Accessed 23 May 2013].

[115] Microsoft, “Remote Desktop Connection,” [Online]. Available: http://windows.microsoft.com/en-us/windows7/products/features/remote-desktop-connection. [Accessed 23 May 2013].

[116] T. Richardson, Q. Stafford-Fraser, K. R. Wood and A. Hopper, “Virtual Network Computing,” *IEEE Internet Computing,* vol. II, no. 1, p. 7, 1998.

[117] Beijer Electronics, “Remote Access Viewer,” [Online]. Available: http://www.beijerelectronics.com/web/web_en_be_com.nsf/AllDocuments/F6587801834E3830C1257966004706C4. [Accessed 21 May 2013].

[118] Siemens , “Simatic Thin Clients,” [Online]. Available: http://www.automation.siemens.com/mcms/pc-based-automation/en/industrial-pc/industrial-monitors/thin-clients/simatic-thin-client/Pages/Default.aspx. [Accessed 21 May 2013].

[119] Automation World.com, “VNC Enables HMI Remote Access,” [Online]. Available: http://www.automationworld.com/information-management/vnc-enables-hmi-remote-access. [Accessed 21 May 2013].

[120] Rockwell Automation, “HMI Features and Extensions,” [Online]. Available: http://www.rockwellautomation.com/resources/downloads/rockwellautomation/pdf/events/raotm/sessions/L15_viz_ftview_pvp6_ppt.pdf. [Accessed 21 May 2013].

[121] Microsoft, “ObservableCollection<T> Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms668604.aspx. [Accessed 26 May 2013].

[122] Microsoft, “ListView Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.windows.forms.listview.aspx. [Accessed 26 May 2013].

[123] R. Stephens, “WPF Programmer's Reference,” in *Windows Presentation Foundation with C# 2010 and .NET 4*, Wiley Publishing, Inc., 2010, p. 434.

[124] Microsoft, “How to: Create and Bind to an ObservableCollection,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms748365.aspx. [Accessed 26 May 2013].

[125] Microsoft, “BackgroundWorker Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.componentmodel.backgroundworker.aspx. [Accessed 26 May 2013].

[126] Sourceforge, “Direct Show Net Library,” [Online]. Available: http://directshownet.sourceforge.net/. [Accessed 23 May 2013].

[127] micjahn, “ZXing.Net,” [Online]. Available: http://zxingnet.codeplex.com/. [Accessed 23 May 2013].

[128] srowen, “ZXing - multi-format 1D/2D barcode image processing library,” [Online]. Available: http://code.google.com/p/zxing/ . [Accessed 23 May 2013].

[129] Microsoft, “System.Security.Cryptography Namespace,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.security.cryptography.aspx. [Accessed 23 May 2013].

[130] H. Correa, “Encrypt and Decrypt a string in C#,” [Online]. Available: http://hectorcorrea.com/blog/encrypt-and-decrypt-a-string-in-c-sharp. [Accessed 23 May 2013].

[131] Microsoft, “How to read XML from a file by using Visual C#,” [Online]. Available: http://support.microsoft.com/kb/307548 . [Accessed 24 May 2013].

[132] Microsoft, “XmlTextWriter Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.xml.xmltextwriter.aspx. [Accessed 24 May 2013].

[133] A. Stellman, “Understanding C#: Simple LINQ to XML examples (tutorial),” [Online]. Available: http://broadcast.oreilly.com/2010/10/understanding-c-simple-linq-to.html . [Accessed 24 May 2013].

[134] S. Agarwal, “LINQ To XML Tutorials with Examples,” [Online]. Available: http://www.dotnetcurry.com/ShowArticle.aspx?ID=564 . [Accessed 24 May 2013].

[135] Microsoft, “Wireless Profile Samples,” [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa369853%28v=vs.85%29.aspx. [Accessed 31 May 2013].

[136] Microsoft, “WPA2-Personal Profile Sample,” [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa370032%28v=vs.85%29.aspx. [Accessed 31 May 2013].

[137] Microsoft, “Ping Class,” [Online]. Available: http://msdn.microsoft.com/en-us/library/system.net.networkinformation.ping.aspx. [Accessed 24 May 2013].

[138] D. Humphrey, “VncSharp: A .NET VNC Client Library,” [Online]. Available: http://cdot.senecac.on.ca/projects/vncsharp/index.html. [Accessed 24 May 2013].

[139] Microsoft, “Walkthrough: Hosting a Windows Forms Control in WPF by Using XAML,” [Online]. Available: http://msdn.microsoft.com/en-us/library/ms742875.aspx. [Accessed 24 May 2013].

[140] Microsoft, “Windows Forms,” [Online]. Available: http://msdn.microsoft.com/en-us/library/dd30h2yb.aspx. [Accessed 24 May 2013].

[141] Beckhoff, “ADS Protocol,” [Online]. Available: http://download.beckhoff.com/download/Software/TwinCAT/TwinCAT2/Infosystem/1033/chm/TcAdsAmsSpec.chm. [Accessed 21 May 2013].

[142] J. McConahay, “Using MODBUS for Process Control and Automation,” Moore Industries, 2011 .

[143] Beckhoff, “TwinCAT System Manager,” [Online]. Available: http://www.beckhoff.com/english.asp?twincat/SysMang.htm?id=159876032042705. [Accessed 21 May 2013].

[144] Beckhoff, “Embedded PC series CX9000, CX9010,” [Online]. Available: http://www.beckhoff.com/english.asp?embedded_pc/embedded_pc_series_cx9000_cx9010.htm?id=97337455631 . [Accessed 27 May 2013].

[145] Asus, “WL-330N3G,” [Online]. Available: http://pt.asus.com/Networks/Wireless_Routers/WL330N3G/. [Accessed 27 May 2013].

[146] Beijer Electronics, “Operator panel T4A,” [Online]. Available: http://www.beijerelectronics.com/web/web_en_be_com.nsf/alldocuments/7997E41B9D7849EAC125794C002C94CF. [Accessed 27 May 2013].

[147] siemens7, “6EP1331 -1SH03-250x250.jpg,” [Online]. Available: http://siemens7.com/media/siemens-3/6EP1331-1SH03/6EP1331-1SH03-250x250.jpg. [Accessed 27 May 2013].

[148] Beckhoff , “Beckhoff Information System - ADS introduction,” [Online]. Available: http://infosys.beckhoff.com/english.php?content=../content/1033/tcadscommon/html/tcadscommon_intro.htm&id=. [Accessed 27 May 2013].

[149] Beckhoff, "TwinCAT PLC Control," [Online]. Available: http://www.beckhoff.com/english/twincat/twincat_plc.htm?id=159876035060. [Accessed 21 May 2013].

[150] Codesys, "CODESYS," Industrial IEC 61131-3 PLC Programming, [Online]. Available: http://www.codesys.com/. [Accessed 27 May 2013].

[151] M. Tiegelkamp and J. Karl-Heinz, IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids, Springer, 2001.

[152] Internet Systems Consortium, "DHCP," [Online]. Available: http://www.isc.org/downloads/dhcp/. [Accessed 31 May 2013].

[153] Beijer Electronics, "iX Software," [Online]. Available: http://www.beijerelectronics.com/ix. [Accessed 27 May 2013].

[154] Foscam, "Support," [Online]. Available: http://foscam.us/tools-support.html. [Accessed 27 May 2013].

[155] Bosch, "About Bosch in Portugal," [Online]. Available: http://www.bosch.pt/en/pt/our_company_10/locations_11/locations-detail_15104.html. [Accessed 21 May 2013].

[156] Apple, "mySCADA Mobile jpg," [Online]. Available: http://a304.phobos.apple.com/us/r1000/075/Purple2/v4/13/ce/e7/13cee7dd-92d1-1007-3d77-08045e34ef50/mzl.ssjkdewb.320x480-75.jpg. [Accessed 31 May 2013].

[157] Sweet Williams, "Scada Mobile," [Online]. Available: http://www.sweetwilliamsl.com/iweb/smhome. [Accessed 21 May 2013].

[158] Sweet Williams, "Scada Mobile png," [Online]. Available: http://www.sweetwilliamsl.com/iweb/home_files/shapeimage_2.png. [Accessed 31 May 2013].

[159] Prosoft Technology, "Prosoft i-View," [Online]. Available: http://www.prosoft-technology.com/Products/ProSoft-Software/Mobile-Applications. [Accessed 21 May 2013].

[160] Apple, "ProSoft i-View jpg," [Online]. Available: http://a1231.phobos.apple.com/us/r1000/067/Purple/v4/f5/fe/05/f5fe05b9-404b-bb1a-eeff-578ed70f897a/mzl.gdcscvqn.320x480-75.jpg. [Accessed 31 May 2013].

[161] Novus Automation, "SuperView Mobile," [Online]. Available: http://www.superview.com.br/site/default.asp?TroncoID=829054&SecaoID=0&SubsecaoID=0. [Accessed 21 May 2013].

[162] Google, "SuperView Mobile jpg," [Online]. Available: https://lh4.ggpht.com/STAHZ2tGoigNHuGkCXDj1BE6ptgocxYJG8SKOFWhlmMubWAunNktBabLfWTwFFzHpHfG. [Accessed 31 May 2013].

[163] LLC Tesla, "Tesla SCADA," [Online]. Available: http://teslascada.com/news/scada_based_on_opc_ua_for_android_devices/2012-06-26-2. [Accessed 21 May 2013].

[164] LLC Tesla, "teslascada.jpg," [Online]. Available: http://teslascada.ucoz.ru/_nw/0/s69849353.jpg. [Accessed 31 May 2013].

[165] Automation Direct, “C-more Remote HMI,” [Online]. Available: http://c-more.automationdirect.com/remote_hmi/. [Accessed 21 May 2013].

[166] Automation Direct, “C-more png,” [Online]. Available: http://c-more.automationdirect.com/remote_hmi/images/iphone.png. [Accessed 31 May 2013].

[167] Innovative Werks, “SCADAView,” [Online]. Available: https://play.google.com/store/apps/details?id=com.innovativewerks.plc&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5pbm5vdmF0aXZld2Vya3MucGxjIl0.. [Accessed 21 May 2013].

[168] Google, “SCADAView jpg,” [Online]. Available: https://lh4.ggpht.com/w50q5Bdn2DZNxhN_5hfaE15WO6Md4rL4V_dLLQLXhiDKMWMS4N2bQIqL6rYA-w0ExA. [Accessed 31 May 2013].

[169] Opto22, “opto apac .png,” [Online]. Available: http://www.opto22.com/images/landingpages/opto_apac_160x309.png. [Accessed 31 May 2013].

[170] Omrom, “DxM Mobile HMI,” [Online]. Available: http://www.windowsphone.com/en-us/store/app/dxm-mobile-hmi-for-omron-plcs/cd9286fe-568b-4c6a-92b7-cd630909c1a0. [Accessed 21 May 2013].

[171] Microsoft, “DxM Mobile HMI png,” [Online]. Available: http://cdn.marketplaceimages.windowsphone.com/v8/images/deb4fca8-540f-4792-a0b0-df5b4b206fb2?imageType=ws_screenshot_large&rotation=0. [Accessed 31 May 2013].

[172] ISW, “ISWvis Mobile,” [Online]. Available: http://www.iswvis-mobile.at/en. [Accessed 21 May 2013].

[173] ISW Industriesoftware, “ISWvia Mobile,” [Online]. Available: http://www.iswvis-mobile.at/sites/default/files/styles/colorbox-full/public/screenshots/screenshot_2013-04-17-11-28-33.jpg. [Accessed 31 May 2013].

[174] Iconics, “Mobile HMI,” [Online]. Available: http://www.iconics.com/Home/Products/Web-Solutions/MobileHMI.aspx. [Accessed 21 May 2013].

[175] Iconics, “Mobile HMI png,” [Online]. Available: http://www.iconics.com/IconicsWebsite/media/General_Images/Products/Web%20Solutions/SilverlightDotCom1.jpg. [Accessed 2013 May 2013].

[176] Automation Software Engineering, “S7Droid,” [Online]. Available: http://www.automation-se.de/. [Accessed 21 May 2013].

[177] Automation Software Engineering, “S7Droid jpg,” [Online]. Available: http://www.automation-se.de/index.php/s7droid/s7droidpro. [Accessed 31 May 2013].

[178] Google, “Modbus Monitor png,” [Online]. Available: https://lh6.ggpht.com/rI-jbJ7h9YzVQSZ3wtAiYSEFh3dFHnhVWNub0RRX5U1dkPqayAbYjic8jiBaqlm8sQ. [Accessed 2013 May 2013].

[179] xplore technologies, “iX104C5 DMSR Tablet,” [Online]. Available: http://www.xploretech.com/products/ix104c5_DMSR. [Accessed 21 May 2013].

[180] ARMOR, “X10gx Rugged Tablet Computer,” [Online]. Available: http://www.drsarmor.com/x10gx.php. [Accessed 21 May 2013].

[181] Panasonic, “Toughpad FZ-A1,” [Online]. Available: http://www.panasonic.com/business/toughpad/us/secure-tablet-specs.asp. [Accessed 21 May 2013].

[182] Panasonic, “Toughbook H2,” [Online]. Available: http://www.panasonic.com/business/toughbook/ultra-mobile-rugged-mca-toughbook-h2.asp. [Accessed 21 May 2013].

[183] “Teamviewer,” [Online]. Available: http://www.teamviewer.com/en/index.aspx.

[184] Microsoft, “System.Net.NetworkInformation Namespace,” [Online]. Available: http://msdn.microsoft.com/en-us/library/3ew4thdx.aspx . [Accessed 24 May 2013].

[185] Autobase Inc, “Autobase HMI/SCADA Viewer,” [Online]. Available: http://www.windowsphone.com/en-us/store/app/autobaseviewer/8bff6844-071f-4271-8644-4d21342db3cd. [Accessed 21 May 2013].

[186] SilverEye Apps, “SilverEye,” [Online]. Available: http://www.windowsphone.com/en-us/store/app/silvereye/0aee9c55-0583-482f-9458-db3ff2678878. [Accessed 21 May 2013].

[187] NFC Forum, “About NFC,” [Online]. Available: http://www.nfc-forum.org/aboutnfc/. [Accessed 4 June 2013].

Appendix A

NATIVE APPLICATIONS

A.1 mySCADA TECHNOLOGIES

mySCADA Mobile (iOS) [30]

Main Features:

- Remotely control, monitor and display technological processes, via animations, alarms, trends and charts;
- Designed to operate over VPN connections or local wireless access;
- Support to Ethernet/IP, Modbus TCP, Siemens S7 industrial communication protocols;

Requirements:

- The PLC has to be connected to a Wi-Fi Router or to a VPN Server;

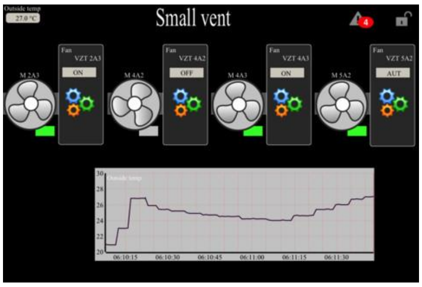

Figure 123 - Appendix A: mySCADA Mobile UI [156]

A.2 SWEETWILLIAM

ScadaMobile (iOS) [157]

Main Features:

- Monitor and Control PLC variables through local or remote wireless access;
- Simultaneous connections to a number of remote PLCs and RTUs;
- Tag Configuration is made by simply importing files created in Excel, or a Text Editor containing the specification of variables;
- Connects directly to PLCs without routing through servers or personal computers, using direct TCP/IP links between iOS devices and PLCs;
- Support to Modbus devices, and OMRON, Allen Bradley, OPTO22, SIEMENS controllers;

Requirements:

- The PLC has to be connected to a Wi-Fi Router or to the internet;

Figure 124 - Appendix A: ScadaMobile UI [158]

A.3 PROSOFT TECHNOLOGY

ProSoft i-View (iOS) [159]

Main Features:

- Remote monitoring and control of process values within an Ethernet/IP and/or Modbus TCP/IP network, using a wireless 802.11 (Wi-Fi) and/or cellular network connection;
- Barcode Reader (QR Code) with phone camera and send that barcode to a specific PAC/PLC tag;
- Import files from Excel, Word or Open Office containing the specification of variables;

Requirements:

- The PLC has to be connected to a Wi-Fi Router or to the Internet;

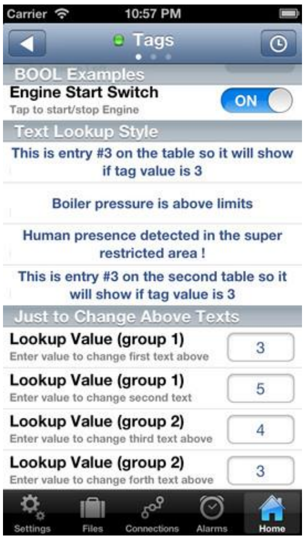

Figure 125 - Appendix A: ProSoft i-View UI [160]

A.4 Novus Automation

SuperView Mobile (Android OS) [161]

Main Features:

- Interface composed by list-type screens, graphics and map (GPS) that allows visualization of the data for both reading and writing;
- Alarms integrated into the Android OS notification system
- Support for Modbus TCP protocol (Wi-Fi, 3G, GPRS, etc.);

Requirements:

- The PLC has to be connected to a Wi-Fi Router or to the Internet;

Figure 126 - Appendix A: SuperView Mobile UI [162]

A.5 LLC Tesla

TeslaSCADA (Android OS) [163]

Main Features:

- SCADA based on OPC UA, which is a highly efficient and secure standard that allows encrypted transmission, authenticated and authorized access;
- Remotely control, monitor and display technological processes, via animations, alarms, trends and charts;

Requirements:

- Based on OPC UA;
- Smartphone with internet connection;

Figure 127 - Appendix A: Tesla SCADA UI [164]

A.6 Automation Direct

C-more Remote HMI (iOS) [165]

Main Features:

- Monitor and control screen operations of the C-more panel as if touching the panel itself;
- Users can save jpeg screen captures to review, email and print if needed;
- Allows each account to be configured in different levels of access;
- Option for each account to associate tags to activate alarms, events or notifications to alert local operators;

Requirements:

- Needs a C-more touch panel with Ethernet port;
- The HMI has to be connected to a VPN connection;

Figure 128 - Appendix A: C-more UI [166]

A.7 Innovative Werks

SCADAView (Android OS) [167]

Main Features:

- Alarm Notification;
- Support for Modbus TCP communication protocol;
- Simultaneous connect up to 15 PLCs an monitor up to 60 data points;

Requirements:

- GE 9030 and GE 9070 PLC, or Modbus devices;

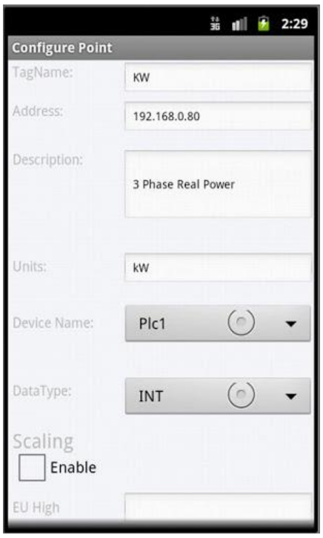

Figure 129 - Appendix A: SCADAView UI [168]

A.8 OPTO 22

Opto iPac and Opto aPac (iOS and Android OS) [34]

Main Features:

- See digital point status. Turn digital outputs on or off;
- View analog point values. Write values to analog outputs;
- Stop or run the current strategy. Start and stop charts;
- Automatically discover all Opto 22 devices on your network;

Requirements:

- Opto 22 SNAP PAC controllers, SNAP I/O units, and E1/E2 I/O units;
- Secured wireless LAN network with access to controller and I/O network;

Figure 130 - Appendix A: Opto aPAC UI [169]

A.9 OMRON

DxM Mobile HMI for Omron PLCs (Windows Phone OS) [170]

Main Features:

- Allows to read, write and control all the variables of work for industrial programmable controllers;
- Define lists of variables and also design interfaces, adding, dragging and dropping controls of type 'Textbox', 'Button' or 'slide bar';

Requirements:

- Use of Omron FINS Hostlink SYSMAC CJ / CS PLC;

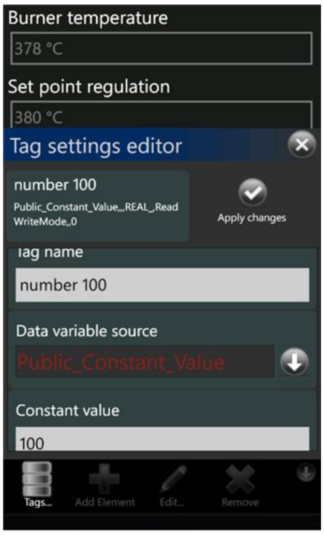

Figure 131 - Appendix A: DxM Mobile HMI UI [171]

A.10 ISW INDUSTRIESOFTWARE

ISWvis Mobile (Android OS) [172]

Main Features:

- SIMATIC S7 support (S7-1200, S7-300, S7-400) and Siemens Logo Controller;
- PC editor to design complex interfaces;

Requirements:

- Siemens PLC connected through VPN connection;

Figure 132 - Appendix A: ISWvis Mobile UI [173]

A.11 ICONICS

Mobile HMI (Windows Phone OS) [174]

Main Features:

- Graphic interface with alarms, trends, charts and grids;
- Overlay real-time data with Microsoft Bing Map;
- Receive Push Notifications for alarm subscriptions;

Requirements:

- Built on standards OPC Classic and OPC Unified Architecture;

Figure 133 - Appendix A: Mobile HMI UI [175]

A.12 AUTOMATION SOFTWARE ENGINEERING

S7Droid (Android OS) [176]

Main Features:

- Define several Siemens S7 PLC within several Data points;

Requirements:

- Siemens PLC: S7-1200, S7-300, S7-400, connected to a Wi-Fi router;

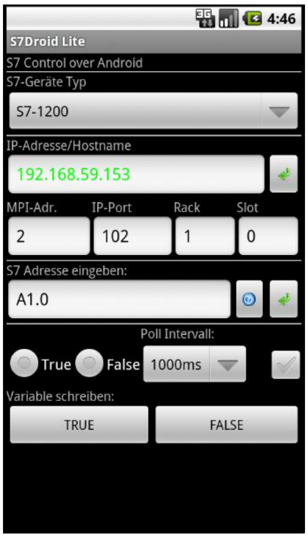

Figure 134 - Appendix A: S7Droid UI [177]

A.13 Hubble

Modbus Monitor (Android OS) [36]

Main Features:

- Multi-threaded Modbus client that polls any standard Modbus TCP and RTU devices over TCP/IP Transport;
- Full Protocol Address Range;
- Import and export list in common CSV format;

Requirements:

- Modbus TCP and RTU Device connected to a wireless router;

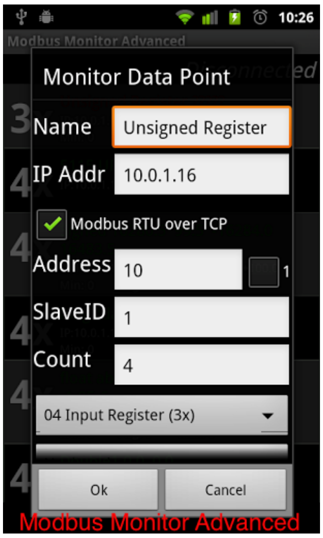

Figure 135 - Appendix A: Modbus Monitor UI [178]

Appendix B

TABLET PC COMPARISON




	uTablet T10C (Ubiqconn)	iX104C5 DMSR (Xplore technologies)	ARMOR X10gx (Armor)	ToughPad A1 (Panasonic)	Toughbook H2 (Panasonic)
System	Intel Atom 1.6GHz Dual Core 2GB DDR3, 32GB SDD	Intel Core i7 2.13GHz 4GB DDR3, SSD	Intel Core Duo 1.2GHz 2GB DDRIII, 64GB SDD	Marvel 1.2GHz Dual Core 1GB DDRIII, 16GB Flash	Intel Core i5 1.7GHz Dual Core 4GB DDRIII, 128GB SDD
OS	Windows 7 (Embedded)	Windows 7	Windows 7 Ultimate	Android 4.0	Windows 7 Professional
Screen	10.4 inch Resistive Touch 1024 x 768	10.4 inch Resistive Touch 1024 x 768	10.4 inch Resistive Touch 1024 x 768	10.1 inch Capacitive Multi-Touch 1024 x 768	10.1 inch Resistive Touch 1024 x 768
Network	Wi-Fi IEEE 802.11 a/b/g/n 3.5G or LTE Bluetooth 4.0	Wi-Fi IEEE 802.11 a/b/g/n HSDPA/HSUPA, EVDO Bluetooth 2.1	Wi-Fi IEEE 802.11 a/g/n - Bluetooth 2.0	Wi-Fi IEEE 802.11 a/b/g/n LTE Bluetooth 2.1	Wi-Fi IEEE 802.11 a/b/g/n 3G or LTE Bluetooth 2.1
I/O Interface	USB Ethernet Audio Jack	USB - Audio Jack	USB - Audio Jack	micro USB - Audio Jack	USB Ethernet Audio Jack
Camera	2 MP w/ audio input in Front 5 MP w/ LED flashlight in Rear	- 3 MP in Rear w/ audio input	- -	2 MP w/ audio input in Front 5 MP w/ LED flashlight in Rear	- 2MP w/ LED flashlight in Rear
Resistance	MIL-STD-810G,IP65 and MIL-STD-461F	MIL-STD-810G and IP67	MIL-STD-810G and IP65	MIL-STD-810G and IP65	MIL-STD-810G e IP65
Battery	8 Hours 3800 mAH, 7.4V	6.5 Hours 68.5Whr	- 2400 mAH, 11.1V	10 Hours 4590 mAH, 7.4V	6.5 Hours 3200 mAH, 7.2V
Price	**1100€**	**3 547 €**	**2 675 €**	**1070€ (+ 310€ LTE)**	**3 245 €**

Figure 136 - Appendix B: Tablet PC Comparison [69], [179], [180], [181], [182]

Appendix C

uTablet T10C Complete Specifications

System	CPU	Intel® Atom™ N2600 1.6GHz Dual Core Processor (Cedar Trail)
	Memory	2GB DDRIII 800 SO-DIMM
	Storage	32GB SATA half-slim size SSD
	Display	10.4-inch LED backlight screen with 5-wire Resistive Touch
	Display Resolution	1024(W) x 768(H) XGA
	Display Brightness	250nits / optional 500nits for sunlight readable
	Audio	1 x High quality speaker
	Communication	Wi-Fi IEEE 802.11 a/b/g/n and Bluetooth 4.0
	RFID	ISO/IEC 14443A, 14443B, 15693 Mifare 1K/4K, Ultralight NFC-IP1 Protocol } optional by snap-on module
	Webcam	2 Mega-pixel camera audio input (Front bezel) 5 Mega-pixel camera with LED Flashlight (Rear bezel)
	TPM	TPM 1.2 (option)
	WWAN	3.5G or LTE (option)
	GPS	Snap-on module (option)
	Light-Sensor	1
In Front Control		1 x Power button, 1 x Lock button, 1 x RF button, 5 x Program Function Button
I/O Ports	USB Port	2 x USB 2.0 type A
	Audio Jack	1 (for microphone and headphone)
	Docking Connector	1
	DC Jack	1
	Ethernet	1 x Micro USB
Mechanical & Environment	Dimensions/Weight	275.4(W) x 203.6(H) x 25(D) mm , 1.1kg (2.43lb)
	Certifications	CE, FCC, UL, VCCI
	Operating System	WES7–WS7P, Win7 Pro. (Embedded)
	Protection Class	IP65
	Drop	6-ft drop (MIL-STD-810G Method 516.6 Procedure IV)
	Mechanical Shock	Operating : 40g, 11ms, Terminal sawtooth (MIL-STD-810G Method 516.6 Procedure 1, Functional Shock) Non-operating : 75g, 8~13ms, Terminal sawtooth (MIL-STD-810G Method 516.6 Procedure V, Crash Hazard Shock)
	Operating Temperature	-20°C to +50°C (MIL-STD-810G Method 501.5 and 502.5)
	Storage Temperature	-30°C to +70°C (MIL-STD-810G Method 501.5 and 502.5)
	Humidity	5-95% without condensation (MIL-STD-810G Method 507.5)
Power Management	Battery	Internal Smart Lithium Polymer battery, 3800mAH, 7.4V
	Power Adapter	AC 100V ~ 240V, 50~60Hz input; 19VDC@3.42A, 65W

Figure 137 - Appendix C: uTablet T10C Complete Specifications [69]

Appendix D

SOFTWARE TESTING

D.1 XML FILE

ListOfMachines.XML


```
<?xml version="1.0" encoding="utf-8"?>
<LIST_OF_MACHINES>
<MACHINE>
		<NAME>TEST_SYS</NAME>
		<SSID>CX9001</SSID>
		<PASSWORD> XPTO1234ABCD </PASSWORD>
		<LIST_OF_DEVICES>
			<DEVICE>
				<NAME>PLC_BECKHOFF</NAME>
				<IP>192.168.1.3</IP>
			</DEVICE>
			<DEVICE>
				<NAME>HMI_BEIJER</NAME>
				<IP>192.168.1.4</IP>
			</DEVICE>
			<DEVICE>
				<NAME>PLC_SIEMENS</NAME>
				<IP>192.168.1.7</IP>
			</DEVICE>
			<DEVICE>
				<NAME>HMI_SIEMENS</NAME>
				<IP>192.168.1.8</IP>
			</DEVICE>
		</LIST_OF_DEVICES>
	</MACHINE>
	<MACHINE>
		<NAME>MACHINE2</NAME>
		<SSID>CX8001</SSID>
		<PASSWORD>123456789A</PASSWORD>
		<LIST_OF_DEVICES>
			<DEVICE>
				<NAME>PLC_ROCKWELL</NAME>
				<IP>193.168.1.100</IP>
			</DEVICE>
			<DEVICE>
```

```
				<NAME>HMI_OPTO22</NAME>
				<IP>193.168.1.39</IP>
			</DEVICE>
		</LIST_OF_DEVICES>
	</MACHINE>
	<MACHINE>
		<NAME>MACHINE1</NAME>
		<SSID>DOMBRESIMAR</SSID>
		<PASSWORD>PasswordDaRede</PASSWORD>
		<LIST_OF_DEVICES>
			<DEVICE>
				<NAME>PLC_ROCKWELLn1</NAME>
				<IP>193.168.1.100</IP>
			</DEVICE>
			<DEVICE>
				<NAME>HMI_OPTO22n1</NAME>
				<IP>193.168.1.39</IP>
			</DEVICE>
			<DEVICE>
				<NAME>PLC_ROCKWELLn2</NAME>
				<IP>193.168.1.4</IP>
			</DEVICE>
			<DEVICE>
				<NAME>HMI_EXTER</NAME>
				<IP>192.168.0.233</IP>
			</DEVICE>
		</LIST_OF_DEVICES>
	</MACHINE>
</LIST_OF_MACHINES>
```