CZECH TECHNICAL UNIVERSITY IN PRAGUE

DIPLOMA THESIS:

Development of SCADA Concept For Doosan Bobcat Engineering & Manufacturing for Assembly Process (Pilot Project)

By: RAUNAK CHHAJER,

FACULTY OF ELECTRICAL ENGINEERING,

Study Programme: CYBERNETICS AND ROBOTICS,

Specialization: ROBOTICS

Supervised By: Ing. PREMYSL NEMEC, Ph.D., EWE

Prague,
August 2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student:	Raunak Chhajer
Study programme:	Cybernetics and Robotics
Specialisation:	Robotics
Title of Diploma Thesis:	Development of SCADA Concept for Doosan Bobcat Engineering & Manufacturing for Assembly Process (pilot project)

Guidelines:

- Introduction to SCADA concept.
- Process Map for assembly process (pilot project).
- Flow Chart Development.
- Functional Description/Definitions.
- Technical Development:
  - hardware study (and comparison of makes where required),
  - logic/SCADA development.
- Final Integration.

Bibliography/Sources:

[1] Scada: Supervisory Control And Data Acquisition 4th Edition, Stuart A. Boyer
[2] PLC & SCADA SYSTEMS: Quick Reference Guide, Francis G.L.
[3] Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS 1st Edition, Tyson Macaulay, Bryan L. Singer

Diploma Thesis Supervisor: Ing. Přemysl Němec, Ph.D., EWE

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic **Head of Department**	prof. Ing. Pavel Ripka, CSc. **Dean**

Prague, February 20, 2017

I'd like to dedicate this work …

First and foremost to my *Dadisa, Late Ms. Matkoo Devi Parakh;* and my *Nana Papa, Late Mr. Ash Karan Parakh.* They have been nothing but an inspiration.

Secondly, I'd like to dedicate this whole journey to my *Parents, Nani Mummy, Namita Masi, Vandita Masi, Mukta Masi, Pramod Masaji, Birendra Masaji, Vipul Masaji, Ameesha, Disha, Dhwani, Ragini, Pranav and Vidushi.*

And last but not the least, to all my other family members and friends who made me smile through even failures and helped me stay strong through the worst; specially *Sanjanaa, Raghavendra, Sakhi, Devhuti, Saloni, Anand, Adarsh, Raghav, Eekshit, Vedant, Aneesh, Aditi.*

Again…

This is for you Dadisa and Nana Papa!

And thank you Mummy, Daddy and Nani Mummy again for always making me believe in myself.

ACKNOWLEDGEMENT

I'd like to further, express my gratitude to the following:

- Mr. Premysl Nemec and Mr. Fernando Ruano of Doosan Bobcat EMEA, Dobris for coming up with this project and selecting me for it.
- Mr. Daniel Kocourek and Mr. Jaroslav Melen for being amazing team members.
- Mr. Lukas and Mr. Karel of Rockwell Automation, and Mr. Zsolt of Crane Electronics for their constant support throught the project.
- Prof. Martin Hlinovsky for his constant guidance as a tutor.
- Prof. Kybic for being flexible with the deadline
- And last but not the least, Prof. Jiri Jakovenko for his constant Benevolence throughout my journey with Czech Technical University in Prague.

Declaration of Authenticity:

I declare that the presented work was developed independently (unless explicitly stated) and that I have listed all sources of information used within it in accordance with the methodological instructions for observing the ethical principles in the preparation of university thesis.

Prague, 15th August, 2017 ……………...................................

Signature

Abstract

This final diploma work of mine is a Pilot project as part of a the 'Leaks-Project' at Doosan-Bobcat EMEA s.r.o, Dobris. The pilot project deals with visualization of the torqueing process of hydraulic valves/hoses at the assembly line for guiding the operators through the correct flow of processes involved, while providing the management with means for analysis of the processes and performance of the workers to eliminate unnecessary wastes and improve traceability. This project intends to be the basis for further SCADA development and hence of the entire assembly line at Doosan Bobcat in the near future.

Keywords: Pilot, Leaks-Project, Doosan Bobcat, visualization, wastes, SCADA

Contents

List of Figures

List of Tables

Abbreviations

SCADA	Spervisory Control and Data Acquisition
HVAC	Heating, Ventilation and Air Conditioning
HMI	Human Machine Interface
RTU	Remote Termina Unit
PLC	Programmable Logical Controller
CSC	Critical and Significant Characteristics
PFMA	Process Failure Mode and Effects Analysis
RFID	Radio Frequency Identification
FTView	FactoryTalk View
ME	Machine Edition
SE	Site Edition
OPC	Open Platform Communications

1. Introduction to SCADA

1.1 SCADA as a Concept

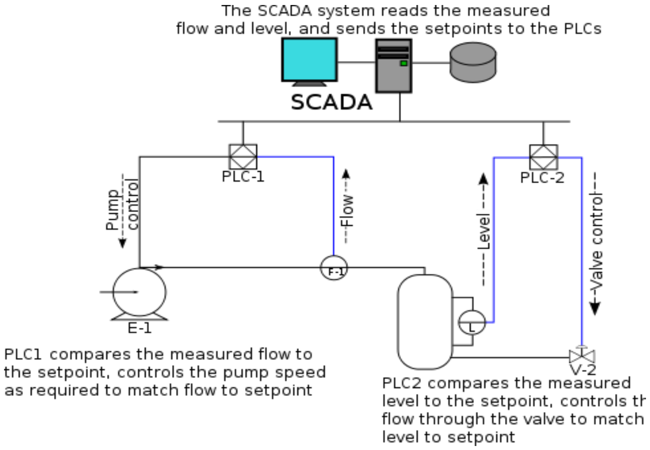
SCADA stands for *supervisory control and data acquisition*. It generally refers to an industrial control system: a computer system monitoring and controlling a process by collecting live field data from the sensors, from one place. The process can be industrial, infrastructure or facility-based as described below:

- *Industrial processes* include those of manufacturing, production, power generation, fabrication, and refining, and may run in continuous, batch, repetitive, or discrete modes.
- *Infrastructure processes* may be public or private, and include water treatment and distribution, wastewater collection and treatment, oil and gas pipelines, electrical power transmission and distribution, Wind Farms, civil defence siren systems, and large communication systems.
- *Facility processes* occur both in public facilities and private ones, including buildings, airports, ships, and space stations. They monitor and control HVAC, access, and energy consumption.

A SCADA System usually consists of the following subsystems:

- A Human-Machine Interface or HMI is the apparatus which presents process data to a human operator, and through this, the human operator monitors and controls the process.
- A supervisory (computer) system, gathering (acquiring) data on the process and sending commands (control) to the process.
- Remote Terminal Units (RTUs) connecting to sensors in the process, converting sensor signals to digital data and sending digital data to the supervisory system.
- Programmable Logic Controller (PLCs) used as field devices because they are more economical, versatile, flexible, and configurable than special-purpose RTUs.
- Communication infrastructure connecting the supervisory system to the Remote Terminal Units.

The term SCADA usually refers to centralized systems which monitor and control entire sites, or complexes of systems spread out over large areas (anything between an industrial plant and a country). Most control actions are performed automatically by Remote Terminal Units ("RTUs") or by programmable logic controllers ("PLCs"). Host control functions are usually restricted to basic overriding or *supervisory* level intervention. For example, a PLC may control the flow of cooling water through part of an industrial process, but the SCADA system may allow operators to change the set points for the flow, and enable alarm conditions, such as loss of flow and high temperature, to be displayed and recorded. The feedback control loop passes through the RTU or PLC, while the SCADA system monitors the overall performance of the loop.



The SCADA system reads the measured flow and level, and sends the setpoints to the PLCs

PLC1 compares the measured flow to the setpoint, controls the pump speed as required to match flow to setpoint

PLC2 compares the measured level to the setpoint, controls th flow through the valve to match level to setpoint

Figure 1

Data acquisition begins at the RTU or PLC level and includes meter readings and equipment status reports that are communicated to SCADA as required. Data is then compiled and formatted in such a way that a control room operator using the HMI can make supervisory decisions to adjust or override normal RTU (PLC) controls.

SCADA systems typically implement a distributed database, commonly referred to as a *tag database*, which contains data elements called *tags* or *points*. A point represents a single input or output value monitored or controlled by the system. Points can be either "hard" or "soft". A hard point represents an actual input or output within the system, while a soft point results from logic and math operations applied to other points. (Most implementations conceptually remove the distinction by making every property a "soft" point expression, which may, in the simplest case, equal a single hard point.) Points are normally stored as value-timestamp pairs: a value, and the timestamp when it was recorded or calculated.

A series of value-timestamp pairs gives the history of that point. It's also common to store additional metadata with tags, such as the path to a field device or PLC register, design time comments, and alarm information.

1.2 Components of SCADA

Human Machine Interface (HMI)

A Human-Machine Interface or HMI is the apparatus which presents process data to a human operator, and through which the human operator controls the process. An HMI is usually linked to the SCADA system's databases and software programs, to provide trending, diagnostic data, and management information such as scheduled maintenance procedures, logistic information, detailed schematics for a particular sensor or machine, and expert-system troubleshooting guides.

The HMI system usually presents the information to the operating personnel graphically, in the form of a mimic diagram. This means that the operator can see a schematic representation of the plant being controlled. For example, a picture of a pump connected to a pipe can show the operator that the pump is running and how much fluid it is pumping through the pipe at the moment. The operator can then switch the pump off. The HMI software will show the flow rate of the fluid in the pipe decrease in real time. Mimic diagrams may consist of line graphics and schematic symbols to represent process elements, or may consist of digital photographs of the process equipment overlain with animated symbols.

The HMI package for the SCADA system typically includes a drawing program that the operators or system maintenance personnel use to change the way these points are represented in the interface. These representations can be as simple as an on-screen traffic light, which represents the state of an actual traffic light in the field, or as complex as a multi-projector display representing the position of all of the elevators in a skyscraper or all of the trains on a railway.

An important part of most SCADA implementations is alarm handling. The system monitors whether certain alarm conditions are satisfied, to determine when an alarm event has occurred. Once an alarm event has been detected, one or more actions are taken (such as the activation of one or more alarm indicators, and perhaps the generation of email or text messages so that management or remote SCADA operators are informed). In many cases, a SCADA operator may have to acknowledge the alarm event; this may deactivate some alarm indicators, whereas other indicators remain active until the alarm conditions are cleared. Alarm conditions can be explicit - for example, an alarm point is a digital status point that has either the value NORMAL or ALARM that is calculated by a formula based on the values in other analogue and digital points - or implicit: the SCADA system might automatically monitor whether the value in an analogue point lies outside high and low limit values associated with that point. Examples of alarm indicators include a siren, a pop-up box on a screen, or a coloured or flashing area on a screen (that might act in a similar way to the "fuel tank empty" light in a car); in each case, the role of the alarm indicator is to draw the operator's attention to the part of the system 'in alarm' so that appropriate action can be taken. In designing SCADA systems, care is needed in coping with a cascade of alarm events occurring in a short time, otherwise the underlying cause (which might not be the earliest event detected) may get lost in the noise. Unfortunately, when used as a noun, the word 'alarm' is used rather loosely in the industry; thus, depending on context it might mean an alarm point, an alarm indicator, or an alarm event.

Remote Terminal Unit (RTU)

The RTU connects to physical equipment. Typically, an RTU converts the electrical signals from the equipment to digital values such as the open/closed status from a switch or a valve, or measurements such as pressure, flow, voltage or current. By converting and sending these electrical signals out to equipment the RTU can control equipment, such as opening or closing a switch or a valve, or setting the speed of a pump.

Supervisory Station

The term "Supervisory Station" refers to the servers and software responsible for communicating with the field equipment (RTUs, PLCs, etc), and then to the HMI software running on workstations in the control room, or elsewhere. In smaller SCADA systems, the master station may be composed of a single PC. In larger SCADA systems, the master station may include multiple servers, distributed software applications, and disaster recovery sites. To increase the integrity of the system the multiple servers will often be configured in a dual-redundant or hot-standby formation providing continuous control and monitoring in the event of a server failure.

Programmable Logic Controller (PLC)

Control engineering has evolved over time. In the past humans were the main method for controlling a system. More recently electricity has been used for control and early electrical control was based on relays. These relays allow power to be switched on and off without a mechanical switch. The development of low cost computer has brought the most recent revolution, the Programmable Logic Controller (PLC). The advent of the PLC began in the 1970's and has become the most common choice for manufacturing controls.

When a process is controlled by a PLC it uses inputs from sensors to make decisions and update outputs to drive actuators. Actuators will drive the system to new states (or modes of operation). This means that the controller is limited by the sensors available, if an input is not available, the controller will have no way to detect a condition.

Many PLC configurations are available, even from a single vendor. But, in each of these there are common components and concepts. The most essential components are –

- Power supply – This can be built into the PLC or an external unit. Common voltage levels required by the PLC (with or without the power supply) are 24 Vdc, 120 Vac, 220 Vac.
- CPU (Central Processing Unit) – This is a computer where ladder logic is stored and processed.
- I/O (Input / Output) – Several input/output terminals must be provided so that the PLC can monitor the process and initiate actions.
- Indicator Lights – These indicate the status of the PLC including power on, program running, and a fault. These are essential when diagnosing problems.

The configuration of the PLC refers to the packaging of components. Typical configurations are listed below:

- Rack – A Rack is often large (up to 18" by 30" by 10") and can hold multiple cards. When necessary, multiple racks can be connected. These tend to be the highest cost, but also the most expensive, but also the most flexible and easy to maintain.
- Mini – These are similar in function to PLC Racks, but about half the size.
- Shoebox – A compact, all-in-one unit (about the size of a shoebox) that has limited expansion capabilities. Lower cost and compactness make these ideal for small applications.
- Micro – These units can be as small as a deck of cards. They tend to have fixed quantities of I/O and limited abilities, but costs will be lowest.
- Software – A software based PLC requires a computer with an interface card, but allows PLC to be connected to sensors and other PLCs across a network.

2. The Leaks Project

2.1 Objective

The current-status in a summary of the assembly lines in Doosan Bobcat EMEA s.r.o, Dobris, per it's Director of Production- Mr. Fernando Ruano is as following:

- The workers can move from one station to the other without any constrain. Hence, a very rigid flow, leading to a very high variability
- Diffuse working areas
- CSCs (Critical & Significant Characteristics) are not guaranteed. Hence there is a demand of PFMA.
- Traceability for the CSCs are not in place
- Work instructions are not always strictly maintained/followed
- High number of torque tools per assembly station

Hence to eliminate the above issues, it was suggested by him to substitute Current Torque tools by Electronic Torque tools for

- Cloud monitoring & Recording
- Introducing sequence of assembly procedures and respective parameters wirelessly
- Better Traceability and precision for each torque operation by the operator(s)

Thus, the pilot phase was objectified to

- Validate application of such a system into Doosan Bobcat Production
- Quantify system costs
- Quantify assembly improvements in terms of time, quality and control
- Define and standardize the operation structure(s)

2.2 Process Map of Pilot Project

After a closed discussion with the team, including:

- Mr. Fernando Ruano,
- Mr. Premysl Nemec,
- Mr. Daniel Kocourek,
- Mr. Jaroslav Melen, and
- Myself

the following general strategy was laid down:

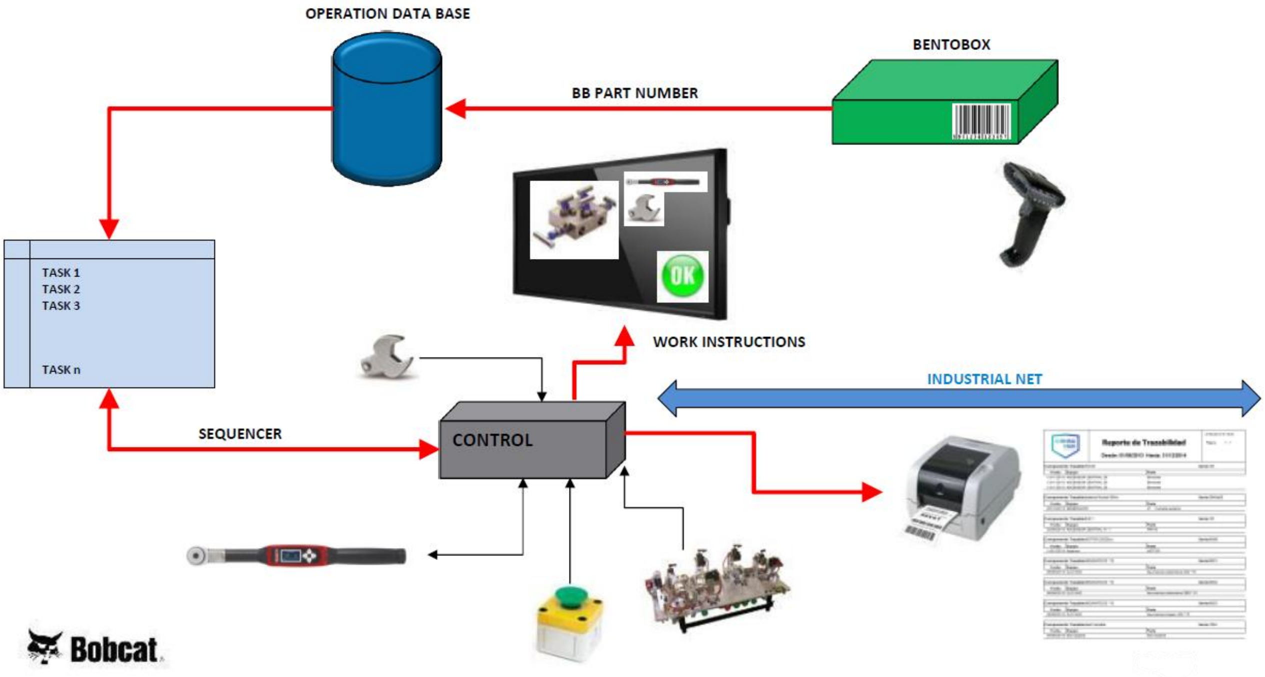


Figure 2

2.3 Flow Diagram

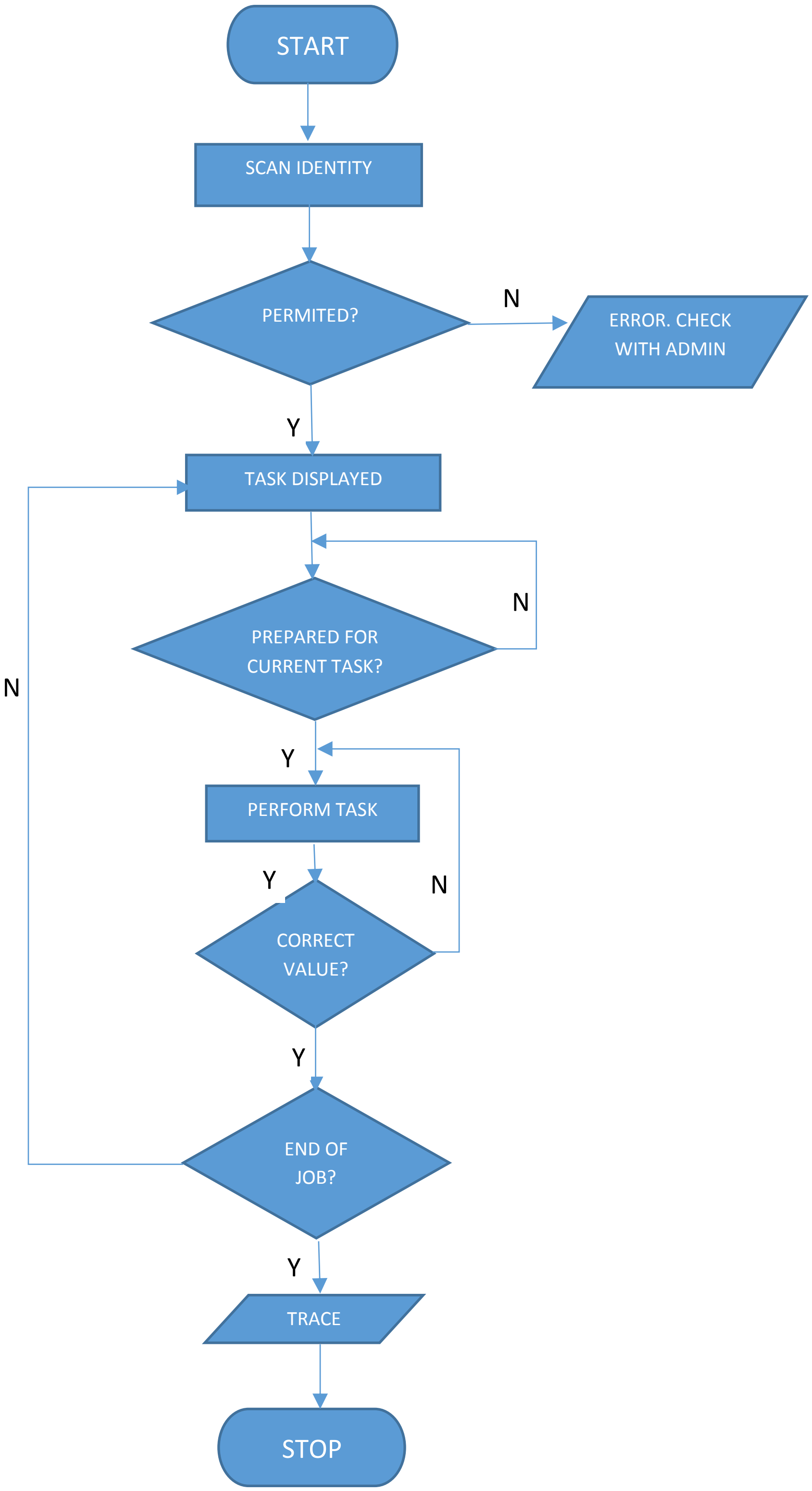


Figure 3

2.4 Functional Description/Definition

1. **<u>Identity</u>**
    - 1.1 **Employee Identification** by RFID scanner
    - 1.2 **Part Identification** by Barcode Scanner
    - 1.3 **Operation Identification**
        - **i.** as displayed on HMI
        - **ii.** manually entered by permitted person
2. **<u>Tasks (& Jobs)</u>** Loaded via PLC
3. **<u>Prepared</u>**
    - 3.1 **Feedback(Bidirectional)**
        - **i.** Digitally
        - **ii.** Manually by push button
    - 3.2 **Finishing of Task(Unidirectional)**
        - **i.** Digitally
        - **ii.** Manually by push button
4. **Time Measurement** by PLC
5. **Acceptance**
    - 5.1 **Triggering** by PLC
    - 5.2 **Calculation** by sensor's software and HMI
6. **Notification of End of Job** by PLC
7. **Traceability**
    - **7.1 Via Industrial NET**
    - **7.2 Printer**
    - **7.3 Storage of Data by PLC**
    - **7.4 RFID results**

3. Hardware and Software Used

3.1 Products from Crane Electronics Ltd.

Torque Wrench (WrenchStar Multi)

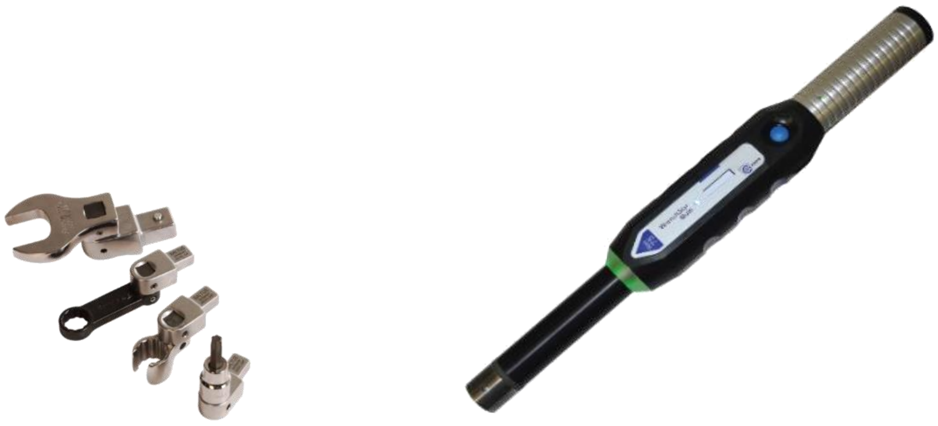


Figure 4

- Production Wrench with possibility of digital recording of torque and angle data
- 360° Colored light ring and vibration alert to realise status of readings and jobs regardless of how it is held
- Wireless connectivity using Radio Frequency to avoid danger of trailing cables
- Quick change battery pack; chargeable internally as well as externally
- Inter-changeable heads with automatic identification, allowing auto length compensation
- Display to tell how many jobs (from loaded) left
- Can record up to 200 readings offline
- Compatible with IQVu data collector and TCI line side controller (both from Crane Electronics)

TCI Multi Controller

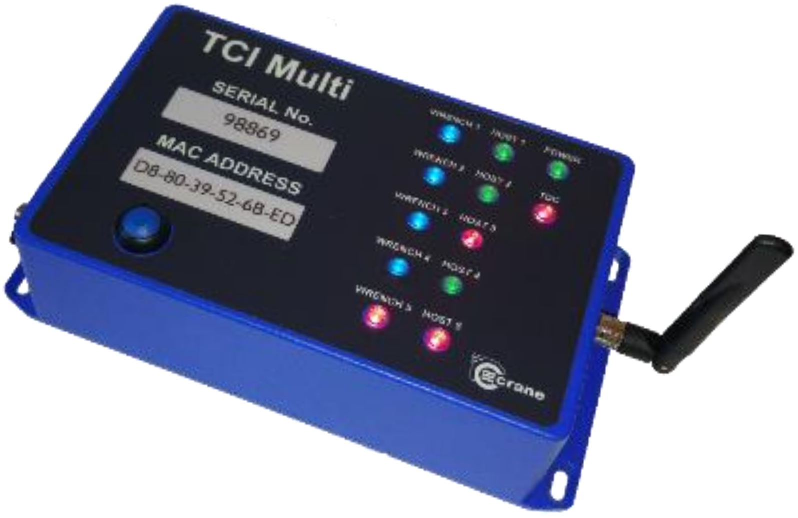
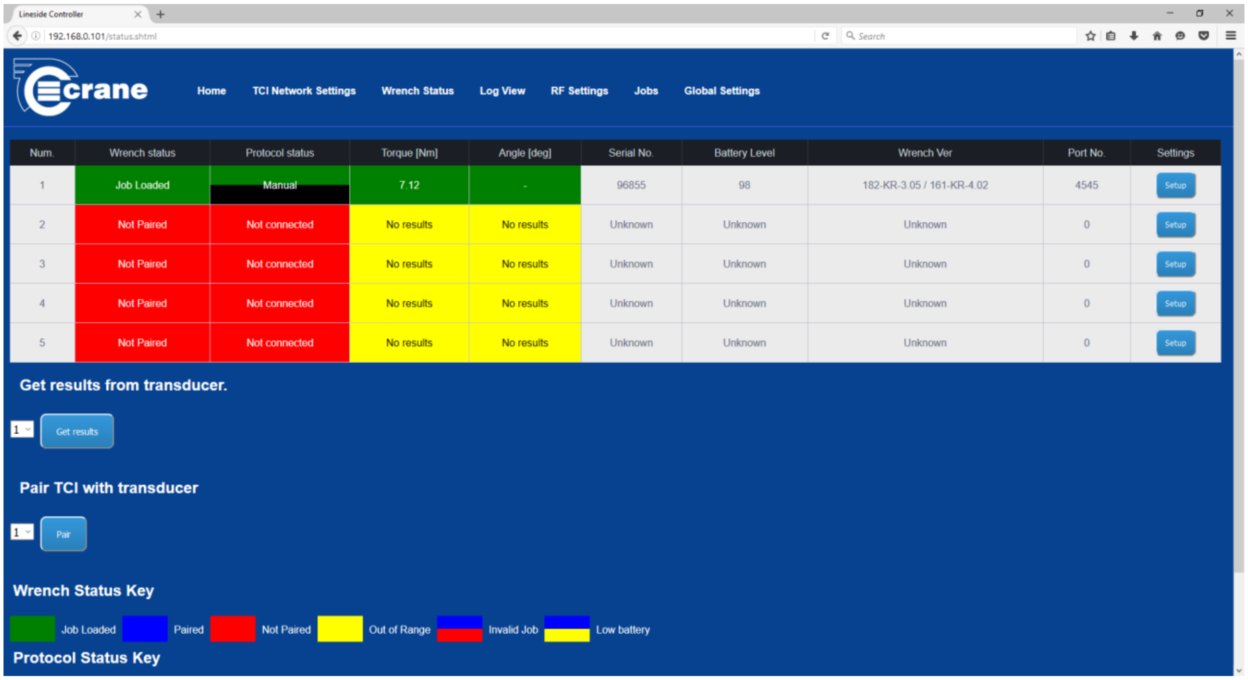


Figure 5

- RJ45 (Ethernet) connection with local network
- 2.4 GHz Radio Frequency to allow secure communication with the wrenches
- Uses stand alone and Open Protocol Communication
- Can communicate with 5 wrenches simultaneously
- Has a guaranteed wireless range of 10m
- Can store upto 20 jobs offline
- Results can be exported into CSV or SQL Database
- Can be setup easily via access to its web browser



Num.	Wrench status	Protocol status	Torque [Nm]	Angle [deg]	Serial No.	Battery Level	Wrench Ver	Port No.	Settings
1	Job Loaded	Manual	7.12	-	96855	98	182-KR-3.05 / 161-KR-4.02	4545	Setup
2	Not Paired	Not connected	No results	No results	Unknown	Unknown	Unknown	0	Setup
3	Not Paired	Not connected	No results	No results	Unknown	Unknown	Unknown	0	Setup
4	Not Paired	Not connected	No results	No results	Unknown	Unknown	Unknown	0	Setup
5	Not Paired	Not connected	No results	No results	Unknown	Unknown	Unknown	0	Setup

Get results from transducer.

1 Get results

Pair TCI with transducer

1 Pair

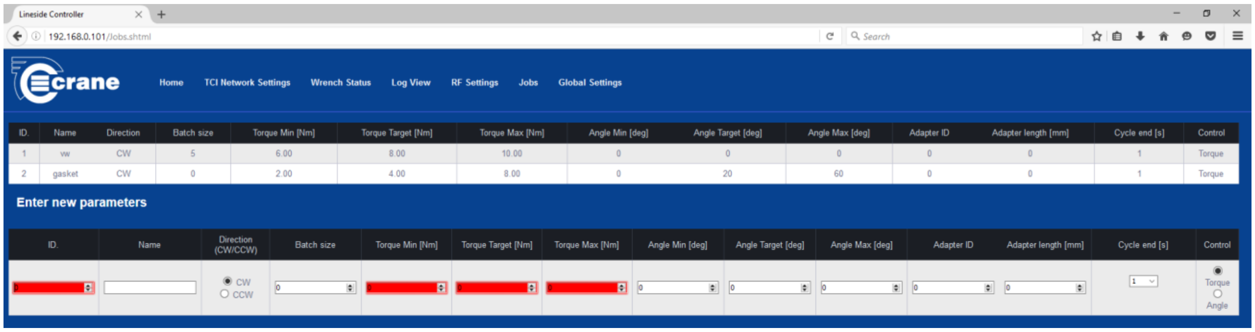
Wrench Status Key

🟩 Job Loaded 🟦 Paired 🟥 Not Paired 🟨 Out of Range Invalid Job Low battery

Protocol Status Key

Figure 6

- Data setup and monitoring is possible through above displayed browser, of the TCI
- All processes are setup on TCI before calling them using the PLC which communicates with it via open protocol
- Job ID is equivalent with the P-Set number (which can be called using the open protocol command)



ID.	Name	Direction	Batch size	Torque Min [Nm]	Torque Target [Nm]	Torque Max [Nm]	Angle Min [deg]	Angle Target [deg]	Angle Max [deg]	Adapter ID	Adapter length [mm]	Cycle end [s]	Control
1	vw	CW	5	6.00	8.00	10.00	0	0	0	0	0	1	Torque
2	gasket	CW	0	2.00	4.00	8.00	0	20	60	0	0	1	Torque

Enter new parameters

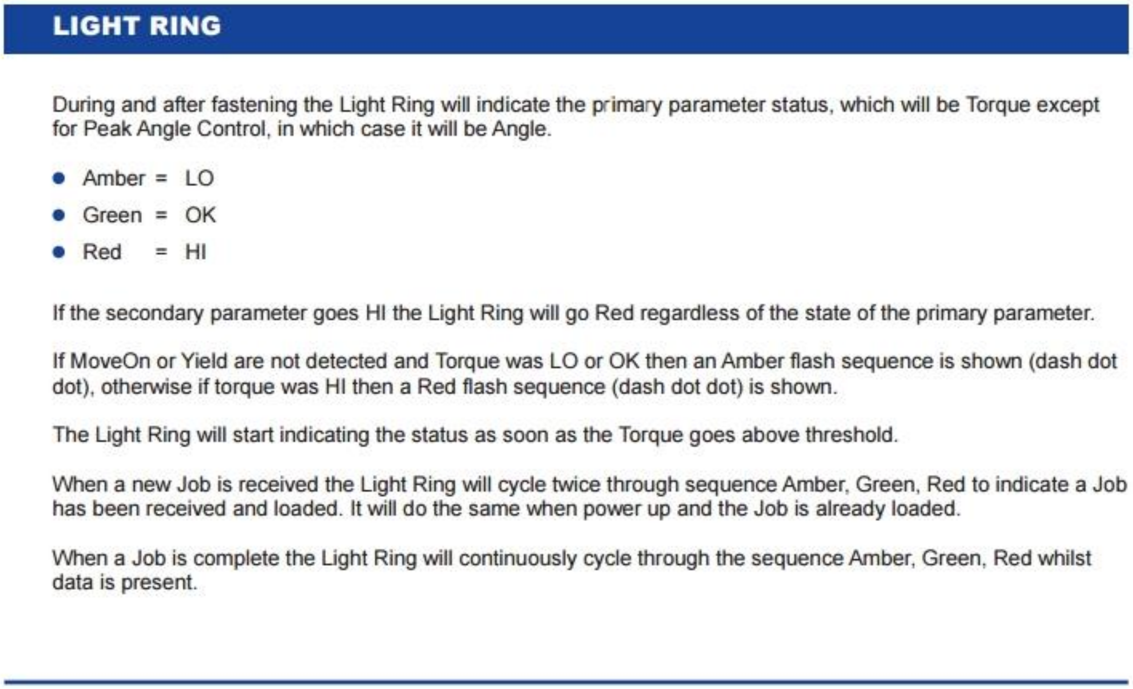
ID.	Name	Direction (CW/CCW)	Batch size	Torque Min [Nm]	Torque Target [Nm]	Torque Max [Nm]	Angle Min [deg]	Angle Target [deg]	Angle Max [deg]	Adapter ID	Adapter length [mm]	Cycle end [s]	Control
0		● CW ○ CCW	0	0	0	0	0	0	0	0	0	1	● Torque ○ Angle

Figure 7

The Open Protocol commands used are as following:

MID	Description
0001	Communication start
0002	Communication start acknowledge
0003	Communication stop
0004	Command error
0005	Command accepted
0010	PSet ID upload request
0011	PSet ID upload reply
0012	PSet data upload request
0013	PSet data upload reply
0018	Select PSet
0042	Disable tool
0043	Enable tool
0060	Last tightening result data subscribe
0061	Last tightening result data upload
0062	Last tightening result data acknowledge
0063	Last tightening result data unsubscribe
9999	Keep alive message

Figure 8



LIGHT RING

During and after fastening the Light Ring will indicate the primary parameter status, which will be Torque except for Peak Angle Control, in which case it will be Angle.

● Amber	=	LO
● Green	=	OK
● Red	=	HI

If the secondary parameter goes HI the Light Ring will go Red regardless of the state of the primary parameter.

If MoveOn or Yield are not detected and Torque was LO or OK then an Amber flash sequence is shown (dash dot dot), otherwise if torque was HI then a Red flash sequence (dash dot dot) is shown.

The Light Ring will start indicating the status as soon as the Torque goes above threshold.

When a new Job is received the Light Ring will cycle twice through sequence Amber, Green, Red to indicate a Job has been received and loaded. It will do the same when power up and the Job is already loaded.

When a Job is complete the Light Ring will continuously cycle through the sequence Amber, Green, Red whilst data is present.

Figure 9

3.2 Products from Rockwell Automation

Stratix 2000 Unmanaged Switch (5 Port)

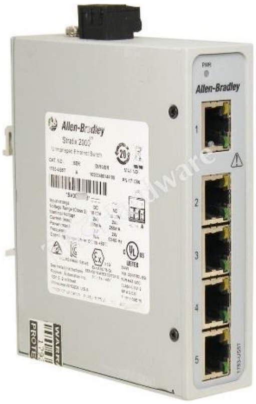


Figure 10

PanelView Plus 7 Performance Terminal

Figure 11

- Touch Screen
- Performance Model
- 15 XGA
- TFT Color
- Ethernet DLR
- 100-240VAC
- Windows CE OS License Pro

PLC- CompactLogix 5370

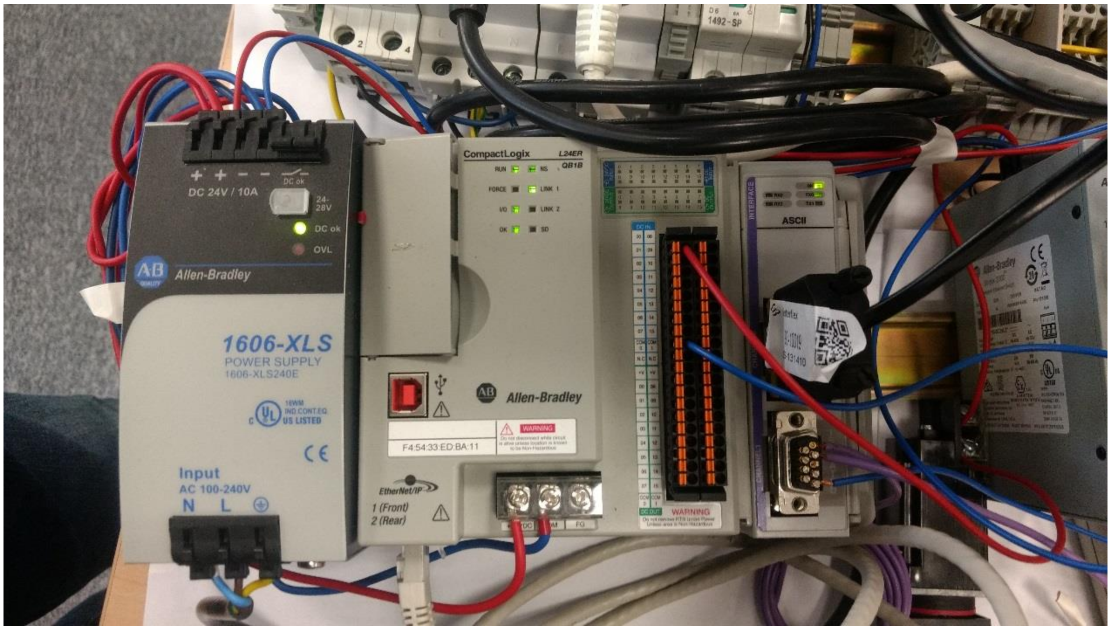


Figure 12

- 1768 / 1769 CompactLogix System
- CompactLogix 5370 L2 Controller
- 2 EtherNet/IP ports with Device Level Ring Capability
- 750KB Memory
- 750KB Memory
- 16 DC Inputs
- 16 DC Outputs
- Up To 4 1769 I/O Expansion Modules
- 8 EtherNet/IP and 120 TCP Connections
- 24V DC Input Embedded Power Supply,With 1GB SD Card
- Can Support Up To 2GB SD Card
- 1 USB Port

Studio 5000 Lite Edition (For PLC Programming)

- Single programming software for all disciplines
- Allows to easily configure devices with graphical wizards and automatically create tags
- Allows simplified programming with multiple languages (IEC61131) and modularity features
- Enables us to view the system and find what we need easily, with the Logical Organizer and Controller Organizer views
- Allows creation of code simultaneously with others and then compare and merge the changes
- Protects the design and execution of our Logix content with license-based content protection capabilities to ensure that only authorized users can view, modify or execute protection code.

<u>FactoryTalk View Studio- Machine Edition (For HMI Programming)</u>

It allows the following features:

- Defining of tags and graphic displays once and referring them throughout the application
- Streamlined development with a common editor for FTView ME and SE.
- Maximizing productivity by directly accessing tag information in the controller, eliminating the need to create HMI tags
- Full-featured graphics editor with powerful editing tools, drawing objects, pre-configured operator devices and animation capabilities
- Simulate individual displays using Test Run, or the entire application with Test Application
- Data Logging to log tag values that can be displayed in historical trend objects
- Alarming to quickly alert operators to conditions requiring immediate action
- Security to restrict operator access to specified displays
- RecipePlus for machine or process recipe management
- Runtime Language Switching
- Multiple-Version support
- Graphics Library with hundreds of graphic objects to drag and drop into our displays
- Reusable faceplates and global objects, speeding application development
- Object Explorer to quickly view the hierarchy of objects on a display
- Embed web-browser frame on a FTView ME display
- Manage users while the application is running
- Easily work on multiple projects, since each project has its own security configuration
- Support for both ActiveX® and OPC technologies

3.3 Other Components Used

Barcode Printer

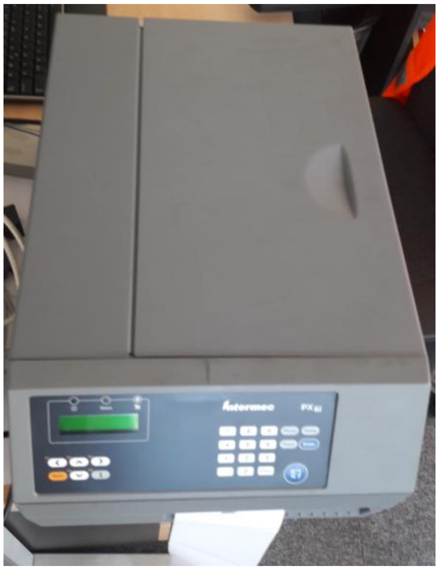


Figure 13

Push Button

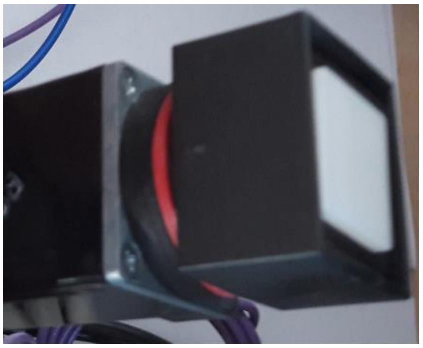


Figure 14

RFID Scanner

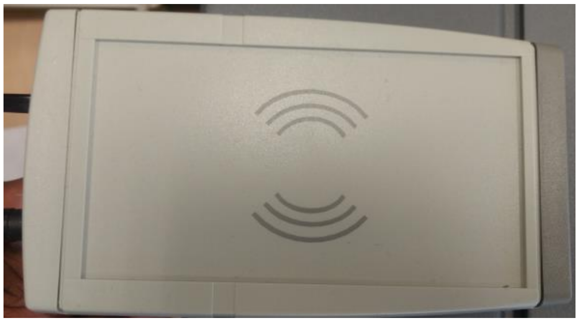


Figure 15

4. SCADA Development

4.1 Data Used: Jobs, Manifolds and Task Sequences

- **Barcode Letters:** Job to be performed
- **Name (Job#):** job number corresponding to size of wrench size for TCI
- **Moment Torque:** Ideal value of moment of torque needed (with tolerance of ±1Nm)
- **Wrench Size:** Size of wrench needed to torque the corresponding part
- **Part Number:** part numbers of the manifold and their corresponding hoses and tube lines
- **Sequence 0:** Manifold to be chosen
- **Sequence 1 onwards:** Sequence of tasks followed for each job

		BARCODE LETTERS	**MM812AM**	E17/2SPEED
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7024226	0
JOB#2	30	19	7218929	1
JOB#2	30	19	7218929	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7235446	4
JOB#2	30	19	7223110	5
JOB#2	30	19	7223070	6
JOB#2	30	19	7225502	7
JOB#2	30	19	7222874	8
JOB#2	30	19	7226593	9
JOB#2	30	19	7235370	10
JOB#2	30	19	7222874	11

Table 1

		BARCODE LETTERS	**MM812BM**	E19/EH
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7309426	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#1	18	14	7238571	3
JOB#1	18	14	7238576	4
JOB#2	30	19	7235446	5
JOB#2	30	19	7223111	6
JOB#2	30	19	7223111	7
JOB#2	30	19	7235446	8
JOB#2	30	19	7223110	9
JOB#2	30	19	7223070	10
JOB#2	30	19	7225502	11
JOB#2	30	19	7222874	12
JOB#2	30	19	7226593	13
JOB#2	30	19	7235370	14
JOB#2	30	19	7222874	15

Table 2

		BARCODE LETTERS	**MM812CM**	E20/EH
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7309426	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#1	18	14	7238571	3
JOB#1	18	14	7238576	4
JOB#2	30	19	7235446	5
JOB#2	30	19	7223111	6
JOB#2	30	19	7223111	7
JOB#2	30	19	7235446	8
JOB#2	30	19	7223110	9
JOB#2	30	19	7223070	10
JOB#2	30	19	7225502	11
JOB#2	30	19	7222874	12
JOB#2	30	19	7222876	13
JOB#2	30	19	7222874	14
JOB#2	30	19	7235370	15

Table 3

		BARCODE LETTERS	**MM812DM**	E19/MAN
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7024225	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7223111	4
JOB#2	30	19	7223111	5
JOB#2	30	19	7235446	6
JOB#2	30	19	7223110	7
JOB#2	30	19	7223070	8
JOB#2	30	19	7225502	9
JOB#2	30	19	7222874	10
JOB#2	30	19	7226593	11
JOB#2	30	19	7235370	12
JOB#2	30	19	7222874	13

Table 4

		BARCODE LETTERS	**MM812EM**	E17Z,DX17zEU,E20/MAN
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7024225	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7223111	4
JOB#2	30	19	7223111	5
JOB#2	30	19	7235446	6
JOB#2	30	19	7223110	7
JOB#2	30	19	7223070	8
JOB#2	30	19	7225502	9
JOB#2	30	19	7222874	10
JOB#2	30	19	7222876	11
JOB#2	30	19	7235370	12
JOB#2	30	19	7222874	13

Table 5

		BARCODE LETTERS	**MM812FM**	E17/1SPEED/ALARM
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7230751	0
JOB#2	30	19	7218929	1
JOB#2	30	19	7218929	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7235446	4
JOB#2	30	19	7223110	5
JOB#2	30	19	7223070	6
JOB#2	30	19	7225502	7
JOB#2	30	19	7222874	8
JOB#2	30	19	7226593	9
JOB#2	30	19	7235370	10
JOB#2	30	19	7222874	11

Table 6

		BARCODE LETTERS	**MM812GM**	E17/1SPEED
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7024226	0
JOB#2	30	19	7218929	1
JOB#2	30	19	7218929	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7235446	4
JOB#2	30	19	7223110	5
JOB#2	30	19	7223070	6
JOB#2	30	19	7225502	7
JOB#2	30	19	7222874	8
JOB#2	30	19	7226593	9
JOB#2	30	19	7235370	10
JOB#2	30	19	7222874	11

Table 7

		BARCODE LETTERS	**MM812HM**	E17/2SPEED/ALARM
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7230751	0
JOB#2	30	19	7218929	1
JOB#2	30	19	7218929	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7235446	4
JOB#2	30	19	7223110	5
JOB#2	30	19	7223070	6
JOB#2	30	19	7225502	7
JOB#2	30	19	7222874	8
JOB#2	30	19	7226593	9
JOB#2	30	19	7235370	10
JOB#2	30	19	7222874	11

Table 8

		BARCODE LETTERS	**MM812IM**	E19/MAN/2AUX
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7024225	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7223111	4
JOB#2	30	19	7223111	5
JOB#2	30	19	7235446	6
JOB#2	30	19	7223110	7
JOB#2	30	19	7223070	8
JOB#2	30	19	7225502	9
JOB#2	30	19	7222874	10
JOB#2	30	19	7226593	11
JOB#2	30	19	7235370	12
JOB#2	30	19	7222874	13

Table 9

		BARCODE LETTERS	**MM812JM**	E20/EH/2AUX
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7309426	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#1	18	14	7238571	3
JOB#1	18	14	7238576	4
JOB#2	30	19	7235446	5
JOB#2	30	19	7223111	6
JOB#2	30	19	7223111	7
JOB#2	30	19	7235446	8
JOB#2	30	19	7223110	9
JOB#2	30	19	7223070	10
JOB#2	30	19	7225502	11
JOB#2	30	19	7222874	12
JOB#2	30	19	7222876	13
JOB#2	30	19	7235370	14
JOB#2	30	19	7222874	15

Table 10

		BARCODE LETTERS	**MM812KM**	E19/EH/2AUX
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7309426	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#1	18	14	7238571	3
JOB#1	18	14	7238576	4
JOB#2	30	19	7235446	5
JOB#2	30	19	7223111	6
JOB#2	30	19	7223111	7
JOB#2	30	19	7235446	8
JOB#2	30	19	7223110	9
JOB#2	30	19	7223070	10
JOB#2	30	19	7225502	11
JOB#2	30	19	7222874	12
JOB#2	30	19	7226593	13
JOB#2	30	19	7235370	14
JOB#2	30	19	7222874	15

Table 11

		BARCODE LETTERS	**MM812LM**	E17z,E20/MAN/2AUX
NAME	Moment Torque [Nm]	Wrench size [mm]	PART NUMBER	**SEQUENCE**
			7024225	0
JOB#2	30	19	7223313	1
JOB#2	30	19	7223313	2
JOB#2	30	19	7235446	3
JOB#2	30	19	7223111	4
JOB#2	30	19	7223111	5
JOB#2	30	19	7235446	6
JOB#2	30	19	7223110	7
JOB#2	30	19	7223070	8
JOB#2	30	19	7225502	9
JOB#2	30	19	7222874	10
JOB#2	30	19	7222876	11
JOB#2	30	19	7235370	12
JOB#2	30	19	7222874	13

Table 12

4.2 Tags

Folder: BarcodeDB\...

TAG NAME	**TAG TYPE**	**DATA SOURCE TYPE**	**INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)**
...\BarCodeDBNumber	Analog	Memory	0
...\DeleteEnable	Digital	Memory	0
...\FolderName	String	Memory	recipe (address on panel where .csv files are stored)
...\FileName	String	Device	[PLC]T0_Station.TCI_WRENCH.cmd_Recepture
...\LoadEnable	Digital	Memory	0
...\Save Enable	Digital	Memory	0

Table 13

Folder: BarcodeDB\Controller\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[0].sts_ID
...\String\02	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[1].sts_ID
...\String\03	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[2].sts_ID
...\String\04	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[3].sts_ID
...\String\05	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[4].sts_ID
...\String\06	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[5].sts_ID
...\String\07	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[6].sts_ID
...\String\08	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[7].sts_ID
...\String\09	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[8].sts_ID
...\String\10	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[9].sts_ID
...\String\51	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[0].sts_ReceptureName
...\String\52	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[1].sts_ReceptureName
...\String\53	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[2].sts_ReceptureName
...\String\54	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[3].sts_ReceptureName
...\String\55	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[4].sts_ReceptureName
...\String\56	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[5].sts_ReceptureName
...\String\57	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[6].sts_ReceptureName
...\String\58	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[7].sts_ReceptureName
...\String\59	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[8].sts_ReceptureName
...\String\60	String	Device	::[PLC]Program:BarCode_Scanner.BarCodeDB[9].sts_ReceptureName

Table 14

Folder: BarcodeDB\HMI\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Memory	
...\String\02	String	Memory	
...\String\03	String	Memory	
...\String\04	String	Memory	
...\String\05	String	Memory	
...\String\06	String	Memory	
...\String\07	String	Memory	
...\String\08	String	Memory	
...\String\09	String	Memory	
...\String\10	String	Memory	
...\String\51	String	Memory	
...\String\52	String	Memory	
...\String\53	String	Memory	
...\String\54	String	Memory	
...\String\55	String	Memory	
...\String\56	String	Memory	
...\String\57	String	Memory	
...\String\58	String	Memory	
...\String\59	String	Memory	
...\String\60	String	Memory	

Table 15

Folder: BarcodeDB\Name\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\01	String	Memory	
...\02	String	Memory	
...\03	String	Memory	
...\04	String	Memory	
...\05	String	Memory	
...\06	String	Memory	
...\07	String	Memory	
...\08	String	Memory	
...\09	String	Memory	
...\10	String	Memory	
...\11	String	Memory	
...\12	String	Memory	

Table 16

Folder: BarCodeDB\comp\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String21	String	Memory	_812AM
...\String22	String	Memory	_812BM
...\String23	String	Memory	_812CM
...\String24	String	Memory	_812DM
...\String25	String	Memory	_812EM
...\String26	String	Memory	_812FM
...\String27	String	Memory	_812GM
...\String28	String	Memory	_812HM
...\String29	String	Memory	_812IM
...\String30	String	Memory	_812JM
...\String31	String	Memory	_812KM
...\32	String	Memory	_812LM

Table 17

Folder: JobRecepture\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\JobReceptureNumber	Analog	Memory	0
…\DeleteEnable	Digital	Memory	0
…\FolderName	String	Memory	recipe (address on panel where .csv files are stored)
…\FileName	String	Device	[PLC]T0_Station.TCI_WRENCH.cmd_Recepture
…\LoadEnable	Digital	Device	::[PLC]Program:MainProgram.LoadNEWRecipe
…\SaveEnable	Digital	Memory	0

Table 18

Folder: JobRecepture\Controller\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Memory	
...\String\02	String	Memory	
...\String\03	String	Memory	
...\String\04	String	Memory	
...\String\05	String	Memory	
...\String\06	String	Memory	
...\String\07	String	Memory	
...\String\08	String	Memory	
...\String\09	String	Memory	
...\String\10	String	Memory	
...\String\11	String	Memory	
...\String\12	String	Memory	

Table 19

Folder: JobRecepture\Controller\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\Float\01	Analog	Memory	0
...\Float\02	Analog	Memory	0
...\Float\03	Analog	Memory	0
...\Float\04	Analog	Memory	0
...\Float\05	Analog	Memory	0
...\Float\06	Analog	Memory	0
...\Float\07	Analog	Memory	0
...\Float\08	Analog	Memory	0
...\Float\09	Analog	Memory	0
...\Float\10	Analog	Memory	0
...\Float\11	Analog	Memory	0
...\Float\12	Analog	Memory	0

Table 20

Folder: JobRecepture\HMI\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Memory	
...\String\02	String	Memory	
...\String\03	String	Memory	
...\String\04	String	Memory	
...\String\05	String	Memory	
...\String\06	String	Memory	
...\String\07	String	Memory	
...\String\08	String	Memory	
...\String\09	String	Memory	
...\String\10	String	Memory	
...\String\11	String	Memory	
...\String\12	String	Memory	

Table 21

Folder: JobRecepture\HMI\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\Float\01	Analog	Memory	0
...\Float\02	Analog	Memory	0
...\Float\03	Analog	Memory	0
...\Float\04	Analog	Memory	0
...\Float\05	Analog	Memory	0
...\Float\06	Analog	Memory	0
...\Float\07	Analog	Memory	0
...\Float\08	Analog	Memory	0
...\Float\09	Analog	Memory	0
...\Float\10	Analog	Memory	0
...\Float\11	Analog	Memory	0
...\Float\12	Analog	Memory	0

Table 22

Folder: JobRecepture\Name\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\01	String	Memory	
...\02	String	Memory	
...\03	String	Memory	
...\04	String	Memory	
...\05	String	Memory	
...\06	String	Memory	
...\07	String	Memory	
...\08	String	Memory	
...\09	String	Memory	
...\10	String	Memory	
...\11	String	Memory	
...\12	String	Memory	

Table 23

Folder: UserDB\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\UserDBNumber	Analog	Memory	0
…\DeleteEnable	Digital	Memory	0
…\FolderName	String	Memory	recipe (address on panel where .csv files are stored)
…\FileName	String	Device	[PLC]T0_Station.TCI_WRENCH.cmd_Recepture
…\LoadEnable	Memory	Device	0
…\SaveEnable	Digital	Memory	0

Table 24

Folder: UserDB\Controller\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\Float\01	Analog	Device	::[PLC]Program:RFid.UserDB[0].sts_Role
...\Float\02	Analog	Device	::[PLC]Program:RFid.UserDB[1].sts_Role
...\Float\03	Analog	Device	::[PLC]Program:RFid.UserDB[2].sts_Role
...\Float\04	Analog	Device	::[PLC]Program:RFid.UserDB[3].sts_Role
...\Float\05	Analog	Device	::[PLC]Program:RFid.UserDB[4].sts_Role
...\Float\06	Analog	Device	::[PLC]Program:RFid.UserDB[5].sts_Role
...\Float\07	Analog	Device	::[PLC]Program:RFid.UserDB[6].sts_Role
...\Float\08	Analog	Device	::[PLC]Program:RFid.UserDB[7].sts_Role
...\Float\09	Analog	Device	::[PLC]Program:RFid.UserDB[8].sts_Role
...\Float\10	Analog	Device	::[PLC]Program:RFid.UserDB[9].sts_Role

Table 25

Folder: UserDB\Controller...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Device	::[PLC]Program:RFid.UserDB[0].sts_ID
...\String\02	String	Device	::[PLC]Program:RFid.UserDB[1].sts_ID
...\String\03	String	Device	::[PLC]Program:RFid.UserDB[2].sts_ID
...\String\04	String	Device	::[PLC]Program:RFid.UserDB[3].sts_ID
...\String\05	String	Device	::[PLC]Program:RFid.UserDB[4].sts_ID
...\String\06	String	Device	::[PLC]Program:RFid.UserDB[5].sts_ID
...\String\07	String	Device	::[PLC]Program:RFid.UserDB[6].sts_ID
...\String\08	String	Device	::[PLC]Program:RFid.UserDB[7].sts_ID
...\String\09	String	Device	::[PLC]Program:RFid.UserDB[8].sts_ID
...\String\10	String	Device	::[PLC]Program:RFid.UserDB[9].sts_ID
...\String\51	String	Device	::[PLC]Program:RFid.UserDB[0].sts_Name
...\String\52	String	Device	::[PLC]Program:RFid.UserDB[1].sts_Name
...\String\53	String	Device	::[PLC]Program:RFid.UserDB[2].sts_Name
...\String\54	String	Device	::[PLC]Program:RFid.UserDB[3].sts_Name
...\String\55	String	Device	::[PLC]Program:RFid.UserDB[4].sts_Name
...\String\56	String	Device	::[PLC]Program:RFid.UserDB[5].sts_Name
...\String\57	String	Device	::[PLC]Program:RFid.UserDB[6].sts_Name
...\String\58	String	Device	::[PLC]Program:RFid.UserDB[7].sts_Name
...\String\59	String	Device	::[PLC]Program:RFid.UserDB[8].sts_Name
...\String\60	String	Device	::[PLC]Program:RFid.UserDB[9].sts_Name

Table 26

Folder: UserDB\HMI\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Memory	
...\String\02	String	Memory	
...\String\03	String	Memory	
...\String\04	String	Memory	
...\String\05	String	Memory	
...\String\06	String	Memory	
...\String\07	String	Memory	
...\String\08	String	Memory	
...\String\09	String	Memory	
...\String\10	String	Memory	

Table 27

Folder: UserDB\HMI\...

TAG NAME	TAG TYPE	DATA SOURCE TYPE	INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)
...\String\01	String	Memory	
...\String\02	String	Memory	
...\String\03	String	Memory	
...\String\04	String	Memory	
...\String\05	String	Memory	
...\String\06	String	Memory	
...\String\07	String	Memory	
...\String\08	String	Memory	
...\String\09	String	Memory	
...\String\10	String	Memory	
...\String\51	String	Memory	
...\String\52	String	Memory	
...\String\53	String	Memory	
...\String\54	String	Memory	
...\String\55	String	Memory	
...\String\56	String	Memory	
...\String\57	String	Memory	
...\String\58	String	Memory	
...\String\59	String	Memory	
...\String\60	String	Memory	

Table 28

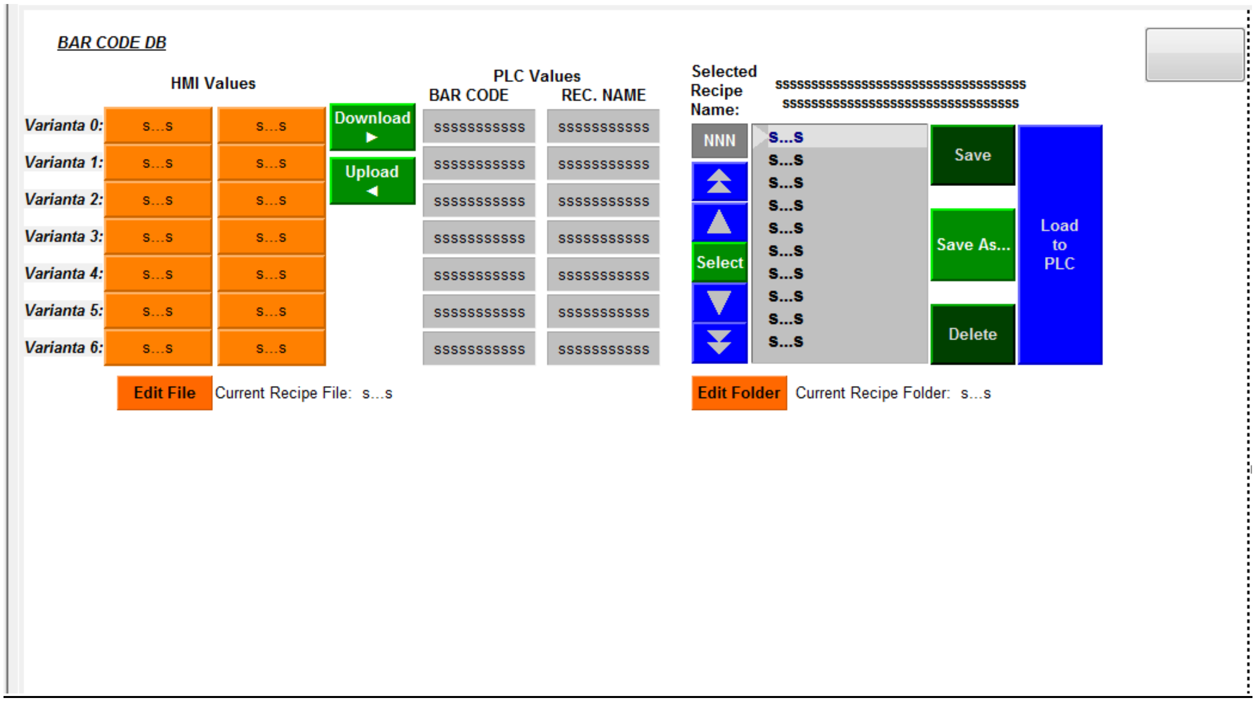
Folder: UserDB\Name\...

TAG NAME	**TAG TYPE**	**DATA SOURCE TYPE**	**INTIAL VALUE(in case of source = memory)/ ADDRESS(in case of source = device)**
...\01	String	Memory	
...\02	String	Memory	
...\03	String	Memory	
...\04	String	Memory	
...\05	String	Memory	
...\06	String	Memory	
...\07	String	Memory	
...\08	String	Memory	
...\09	String	Memory	
...\10	String	Memory	
...\11	String	Memory	
...\12	String	Memory	

Table 29

4.3 Displays and Definitions

BarCodeDB



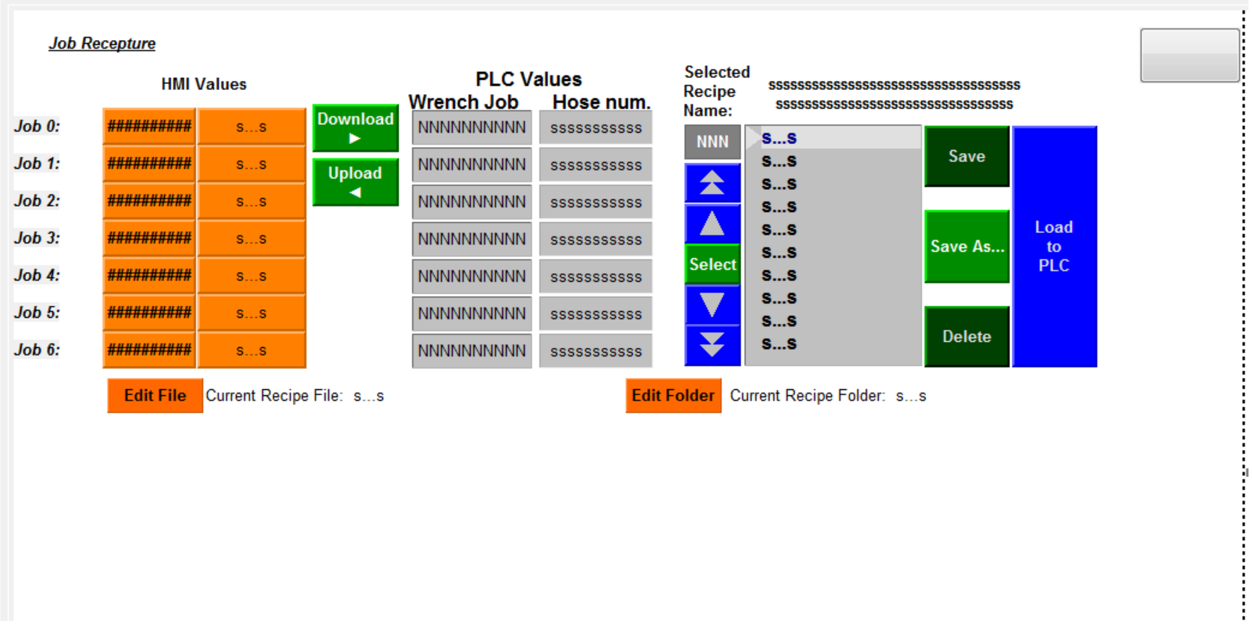
Current Recipe File: s...s

Current Recipe Folder: s...s

Figure 16

- Under the section **'PLC Values'**:
  - The Objects under the heading **'BAR CODE'** are associated with the tag values of **'BarCodeDB\Controller\String\01'** … **'BarCodeDB\Controller\String\12'** in ascending order from top to bottom (**for example, MM812AM)**
  - The Objects under the heading **'REC. NAME'** are associated with the tag values of **'BarCodeDB\Controller\String\51'** … **'BarCodeDB\Controller\String\62'** in ascending order from top to bottom (**for example, _812AM for MM812AM)**
- These PLC values can be uploaded to corresponding cells under the heading **'HMI VALUES'** by pressing the Upload Button defined by the macro:**'MacroBarCodeDBUpload'**
- **'Current Recipe File'** displays the current recipe loaded to the PLC (i.e, in the format **_812AM**) given by the tag **'…FileName'**
- 'Current Recipe Folder' displays the value given by tag **'…FolderName'**
- The right most displayed object enlists all the files in the displayed folder. If the user has the permission based on the role assigned to him/her, he/she can select a recipe file manually and load it to the PLC using the corresponding buttons having their usual meanings.

JobRecepture



	HMI Values		PLC Values Wrench Job	Hose num.
Job 0:	##########	s...s	NNNNNNNNNN	sssssssssss
Job 1:	##########	s...s	NNNNNNNNNN	sssssssssss
Job 2:	##########	s...s	NNNNNNNNNN	sssssssssss
Job 3:	##########	s...s	NNNNNNNNNN	sssssssssss
Job 4:	##########	s...s	NNNNNNNNNN	sssssssssss
Job 5:	##########	s...s	NNNNNNNNNN	sssssssssss
Job 6:	##########	s...s	NNNNNNNNNN	sssssssssss

Selected Recipe Name: ssssssssssssssssssssssssssssssssssssss
ssssssssssssssssssssssssssssssssssss

NNN

s...s
s...s
s...s
s...s
s...s
s...s
s...s
s...s
s...s
s...s

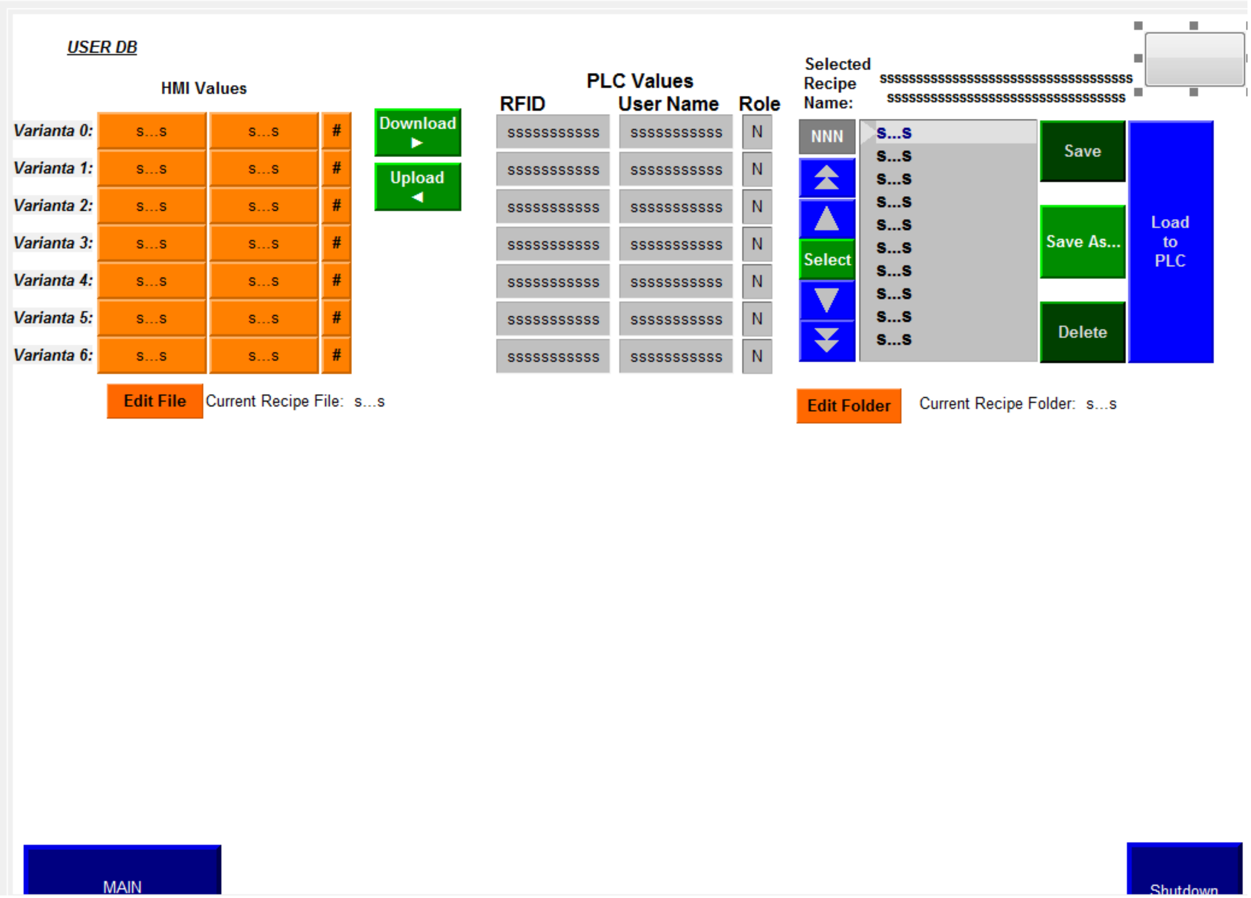
Current Recipe File: s...s

Current Recipe Folder: s...s

Figure 17

- Under the section **'PLC Values'**:
  - The Objects under the heading **'Hose num'** are associated with the tag values of **'JobRecepture\Controller\String\01'** … **'BarCodeDB\Controller\String\10'** in ascending order of sequence of tasks from top to bottom (**for example, 7223313):** [associated with part numbers of each job]
  - The Objects under the heading **'Wrench Job'** are associated with the tag values of **'JobRecepture\Controller\Float\01'** … **'JobRecepture\Controller\Float\10'**in ascending order from top to bottom (**for example, 2 for 7223313):** [associated with the Job#]
- These PLC values can be uploaded to corresponding cells under the heading **'HMI VALUES'** by pressing the Upload Button defined by the macro:**'MacroJobReceptureUpload'**
- **'Current Recipe File'** displays the current recipe loaded to the PLC (i.e, in the format **_812AM**) given by the tag **'…FileName'**
- 'Current Recipe Folder' displays the value given by tag **'…FolderName'**
- The right most displayed object enlists all the files in the displayed folder. If the user has the permission based on the role assigned to him/her, he/she can select a recipe file manually and load it to the PLC using the corresponding buttons having their usual meanings.

UserDB



	HMI Values			RFID	User Name	Role
Varianta 0:	s...s	s...s	#	sssssssssss	sssssssssss	N
Varianta 1:	s...s	s...s	#	sssssssssss	sssssssssss	N
Varianta 2:	s...s	s...s	#	sssssssssss	sssssssssss	N
Varianta 3:	s...s	s...s	#	sssssssssss	sssssssssss	N
Varianta 4:	s...s	s...s	#	sssssssssss	sssssssssss	N
Varianta 5:	s...s	s...s	#	sssssssssss	sssssssssss	N
Varianta 6:	s...s	s...s	#	sssssssssss	sssssssssss	N

Current Recipe File: s...s

Current Recipe Folder: s...s

Figure 18

- Under the section **‘PLC Values’**:
    - The Objects under the heading **‘RFID’** are associated with the tag values of **‘UserDB\Controller\String\01’** … **‘UserDB\Controller\String\10’**
    - The Objects under the heading **‘User’** are associated with the tag values of **‘UserDB\Controller\String\51’** … **‘UserDB\Controller\String\60’**
    - The Objects under the heading **‘User’** are associated with the tag values of **‘UserDB\Controller\Float\01’** … **‘UserDB\Controller\Float\10’**

    (Each of the Above Headings having their usual meanings)
- These PLC values can be uploaded to corresponding cells under the heading **‘HMI VALUES’** by pressing the Upload Button defined by the macro:**’MacroUserDBUpload’**
- **‘Current Recipe File’** displays the current recipe loaded to the PLC (i.e, in the format **_812AM**) given by the tag **‘…FileName’**
- ‘Current Recipe Folder’ displays the value given by tag **‘…FolderName’**
- The right most displayed object enlists all the files in the displayed folder. If the user has the permission based on the role assigned to him/her, he/she can select a recipe file manually and load it to the PLC using the corresponding buttons having their usual meanings.

1. The above three displays can be seen and manipulated by only the people with the role of an Administrator
2. The values change in the background depending on the tasks performed by operator without being visible to him/her due to the control associated to the ActiveX object created for each of the displays.
3. During the runtime of the application: the admin can directly get to the **Main** display by pressing on the corresponding button at the bottom left of each of the above displays, or similarly **Shutdown** the application by pressing button at the bottom right corner.
4. All other operators without the permission to navigate the above three displays can only see the following display:

Main

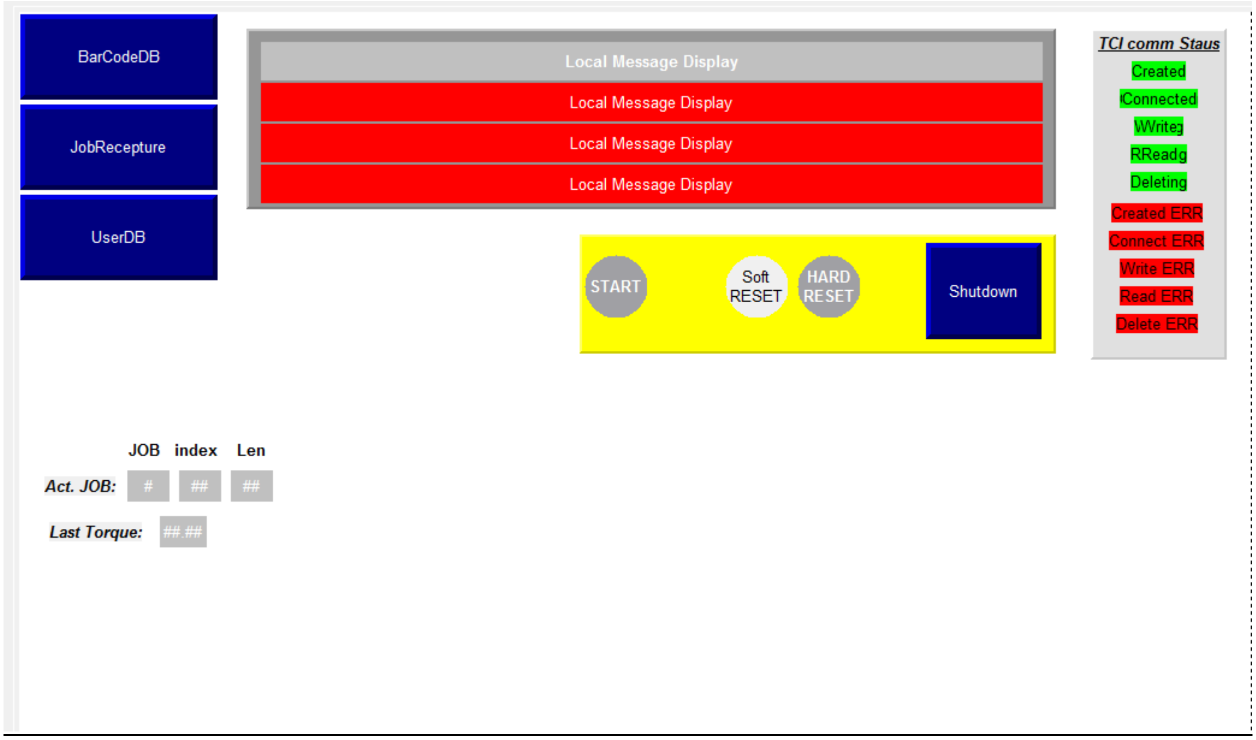
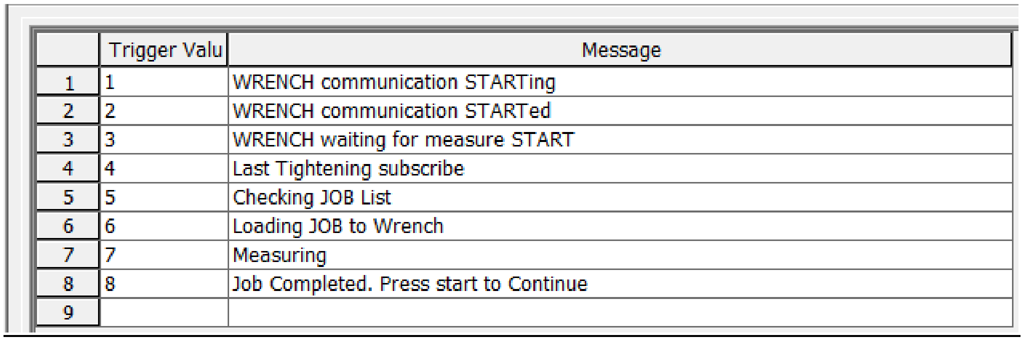


Figure 19

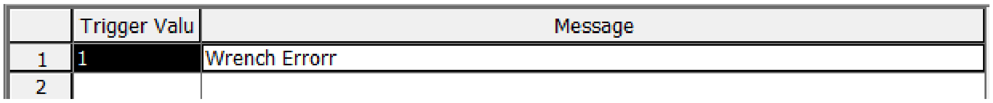
- When the Operator is working, he/she needs to either press the **push button** or the **start button on the display** to let the PLC know that he/she is starting a new task.
- The **Soft Reset Button** is to reset all software communications
- The **Hard Reset Button** is to reset all Hardware
- The **index** value corresponds to the sequence number of the current task of the corresponding job, and is compared with to display the image of the part to make it sure for the operator that he/she is working with the right part
- The **Last Torque** value communicates with the wrench, displaying the last value that it achieved successfully
- The **Local Message Displays** are as following in sequential order from top to bottom:

Local Status Message



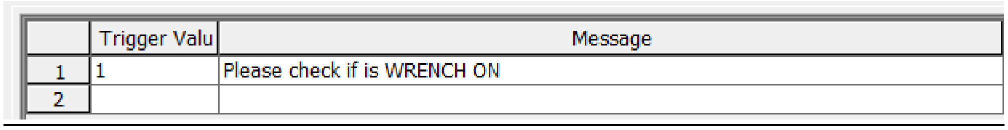
	Trigger Valu	Message
1	1	WRENCH communication STARTing
2	2	WRENCH communication STARTed
3	3	WRENCH waiting for measure START
4	4	Last Tightening subscribe
5	5	Checking JOB List
6	6	Loading JOB to Wrench
7	7	Measuring
8	8	Job Completed. Press start to Continue
9		

Local Error Message 0



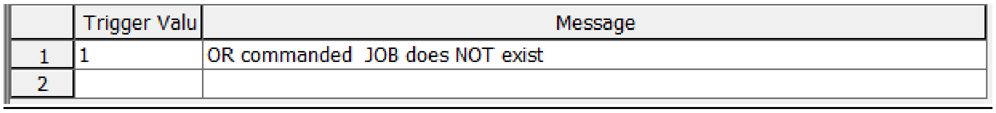
	Trigger Valu	Message
1	1	Wrench Errorr
2		

Local Error Message 1



	Trigger Valu	Message
1	1	Please check if is WRENCH ON
2		

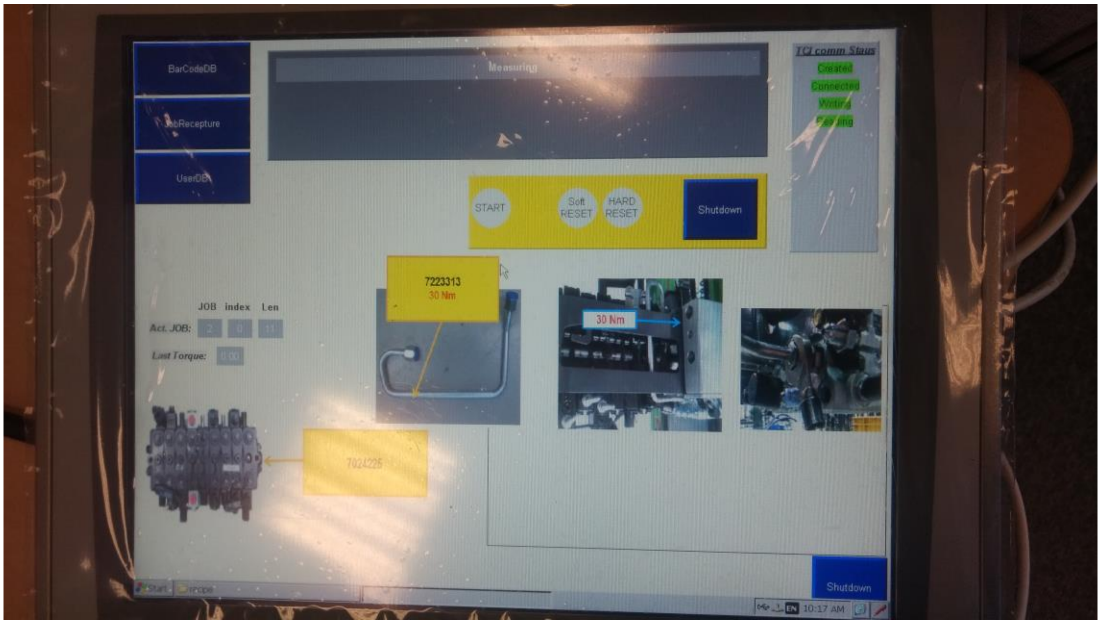
Local Error Message 2



	Trigger Valu	Message
1	1	OR commanded JOB does NOT exist
2		

Figure 20

5. Final Integration and Conclusion



JOB index Len
Act. JOB: 2 0 11
Last Torque: 0.00

Figure 21

As can be seen, the above image displays a sample look of the **Main** Display showing the corresponding **Manifold, Part, and other messages and values.**

And below, can be seen an image of the hoses tested upon

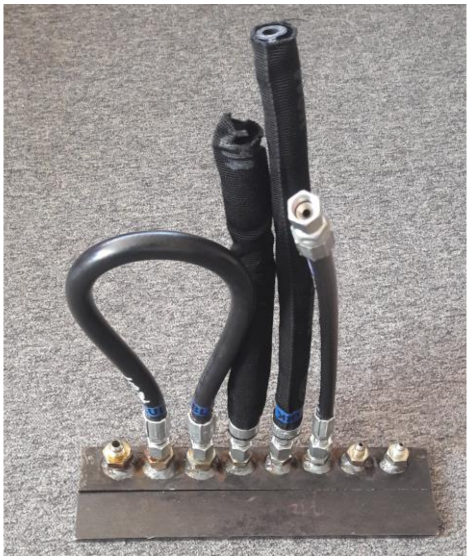


Figure 22

Hence I conclude that the test run of the Pilot Project was successful, besides some of the parts and cables were still missing.

6. References

- **Manuals of the products used**
- http://sds-group.com/content/plc-scada
- www.automation.com
- www.scada.com
- www.rockwellautomation.com