TAMPEREEN
AMMATTIKORKEAKOULU

DEVELOPMENT OF MAINTENANCE ASSISTING SOFTWARE APPLICATION FOR MOBILE WORK MACHINE

Ville Saari

Master's thesis
March 2016
Automation Technology

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Automation Technology

SAARI, VILLE:
Development of Maintenance Assisting Software Application for Mobile Work Machine

Master's thesis 48 pages, appendices 6 pages
March 2016

Maintenance assisting software application has been helping in the maintenance of a mobile work machine, made for container handling, over a decade. Software application has been used to read alarm data and statistical information from the machine control system. These machine parameters have helped with machine service and designing of the preventative maintenance.

As a part of this thesis a user survey and a market comparison was executed to determine user needs for different machine parameters and other maintenance assisting functionalities in the software application.

This document serves as a functional description and specifies technical information needed for the implementation of a new version of the software application. With this specification all the essential characters and new developed characters can be transferred into the new platform.

Key points in the software application development are:

- Updating machine parameters and software application functionalities to serve modern-day requirements and customer demands to all machine variations. Customer demands are based on the user survey.
- Improved alarm data and statistical information analyzing to help the user in fault finding and therefore decrease the machine down-time.
- Changing the communication from serial bus to Ethernet using Omron FINS-commands. For the future Ethernet based communication gives better possibilities to implement wireless communication between machine and the software application.
- Harmonize user interface visualization to other machine related user interfaces.

Key words: maintenance, software application

CONTENTS

GLOSSARY

CAN	Controller Area Network
DEC	Decimal value
DA	Destination Address
DNA	Destination Network Address
ECU	Engine Control Unit
FINS	Factory Interface Network Service
GCT	Gateway Count
GUI	Graphical User Interface
HEX	Hexadecimal value
HS	Health and Safety
ICF	Information Control Field
KPI	Key Performance Indicator
PLC	Programmable Logic Controller
RSV	Reserved
SA	Source Address
SNA	Source Network Address
SID	Service ID
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

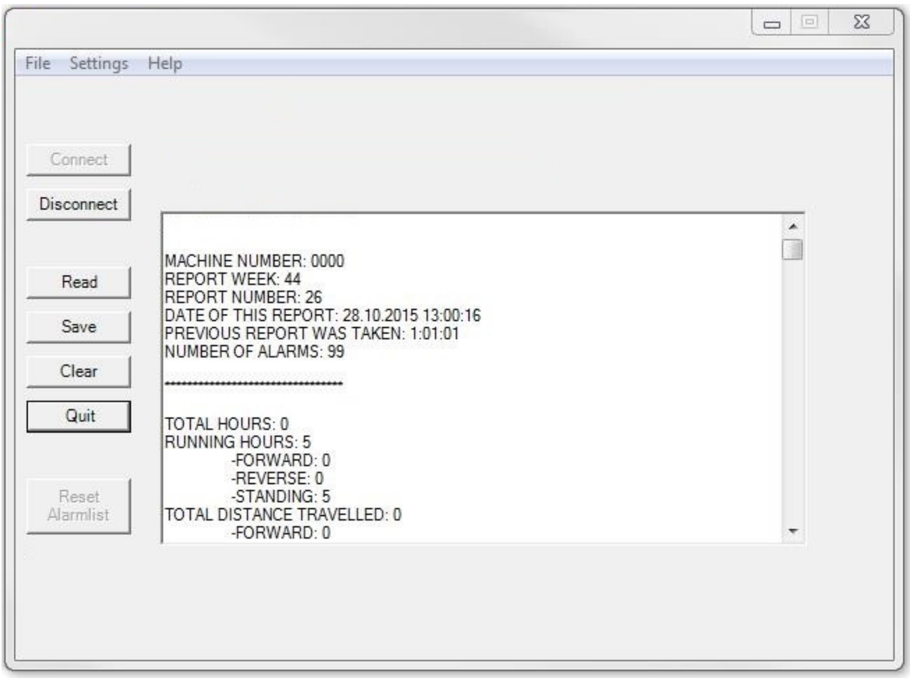
1 INTRODUCTION

1.1 Background and motivation

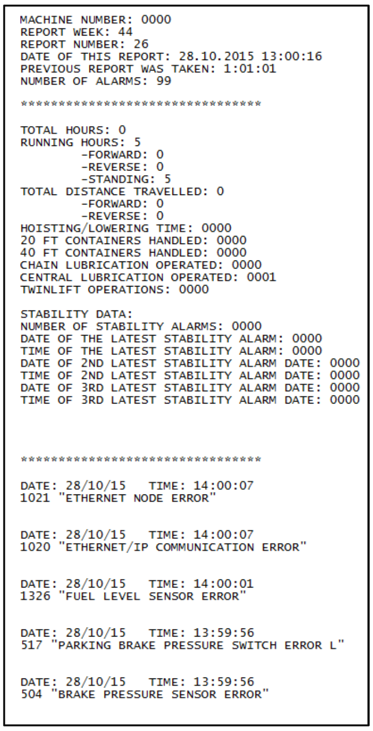
Maintenance assisting software application has been helping in the maintenance of a mobile work machine, made for container handling, over a decade. Software application has been used to read alarm data and statistical information from the machine control system. These machine parameters have helped with machine service and designing of the preventative maintenance.

This software application requires updating in order to answer modern-day requirements. The application is starting to have compatibility problems with the newest computer operating systems. Serial bus connection used is beginning to be more off an optional bus in the control systems and therefore needs to be changed to Ethernet based communication. Also machine level development has brought new requirements for the machine parameters.

Existent layout of the software application user interface and report can be seen in the pictures 1 and 2.



PICTURE 1: User interface



```
MACHINE NUMBER: 0000
REPORT WEEK: 44
REPORT NUMBER: 26
DATE OF THIS REPORT: 28.10.2015 13:00:16
PREVIOUS REPORT WAS TAKEN: 1:01:01
NUMBER OF ALARMS: 99

********************************

TOTAL HOURS: 0
RUNNING HOURS: 5
        -FORWARD: 0
        -REVERSE: 0
        -STANDING: 5
TOTAL DISTANCE TRAVELLED: 0
        -FORWARD: 0
        -REVERSE: 0
HOISTING/LOWERING TIME: 0000
20 FT CONTAINERS HANDLED: 0000
40 FT CONTAINERS HANDLED: 0000
CHAIN LUBRICATION OPERATED: 0000
CENTRAL LUBRICATION OPERATED: 0001
TWINLIFT OPERATIONS: 0000

STABILITY DATA:
NUMBER OF STABILITY ALARMS: 0000
DATE OF THE LATEST STABILITY ALARM: 0000
TIME OF THE LATEST STABILITY ALARM: 0000
DATE OF 2ND LATEST STABILITY ALARM DATE: 0000
TIME OF 2ND LATEST STABILITY ALARM DATE: 0000
DATE OF 3RD LATEST STABILITY ALARM DATE: 0000
TIME OF 3RD LATEST STABILITY ALARM DATE: 0000

********************************

DATE: 28/10/15   TIME: 14:00:07
1021 "ETHERNET NODE ERROR"

DATE: 28/10/15   TIME: 14:00:07
1020 "ETHERNET/IP COMMUNICATION ERROR"

DATE: 28/10/15   TIME: 14:00:01
1326 "FUEL LEVEL SENSOR ERROR"

DATE: 28/10/15   TIME: 13:59:56
517 "PARKING BRAKE PRESSURE SWITCH ERROR L"

DATE: 28/10/15   TIME: 13:59:56
504 "BRAKE PRESSURE SENSOR ERROR"
```

PICTURE 2: Report layout

1.2 Development targets

The aim of this document is to create a functional description and specify technical information needed for the implementation of a new version of the software application. With this specification all the essential characters and new developed characters can be transferred into the new platform.

Key points in the software application development are:

- Updating machine parameters and software application functionalities to serve modern-day requirements and customer demands to all machine variations. Customer demands are based on the user survey.
- Improved alarm data and statistical information analyzing to help the user in fault finding and therefore decrease the machine down-time.
- Changing the communication from serial bus to Ethernet using Omron FINS-commands. For the future Ethernet based communication gives better possibilities to implement wireless communication between machine and the software application.
- Harmonize user interface visualization to other machine related user interfaces.

2 USER SURVEY

2.1 Target

The aim of the survey was to gather information about the needs of different target groups regarding the available machine parameters. With the results from this survey the application will be developed to serve all target groups and also avoid unnecessary information and functionalities to make the application as easy as possible to use and maintain.

2.2 Implementation

User survey was implemented by using Google Forms -platform. Form included multiple-choice questions and free comment fields concerning necessity of the machine parameters and application functionalities. Survey was distributed via email to approximately 75 persons including inner and outer customers. Participants were selected from all departments (management, engineering, product support, commissioning, project delivery and service) to get as wide view as possible. Survey was divided into four main points, statistics, alarm, GUI remote use and others. Survey form is presented in appendix 1.

2.3 Results

The user survey was answered by 31 respondents, which makes the response rate approximately 40 per cent. With this response rate it is possible to define all the required machine parameters and application functionalities. Unfortunately most of the responses were received from inner customers who use this application monthly or less. Respondent profile can be seen from figure 1.

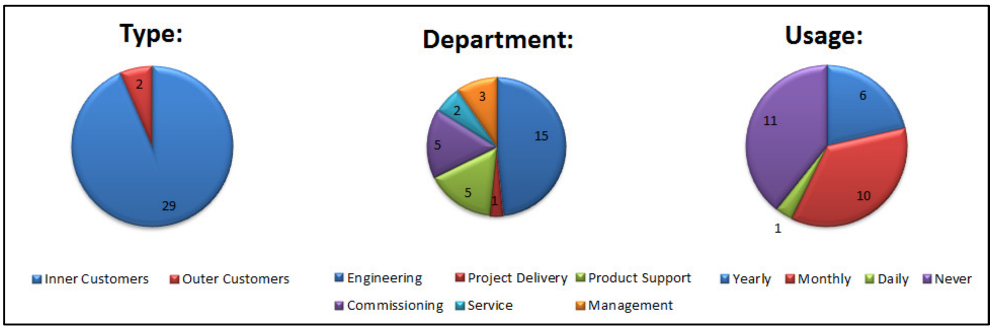


FIGURE 1. Respondent profile

In the survey the necessity of different parameters were made using numerical value from 1 to 5. One means that parameter is not relevant and five means that parameter is definitely needed in the software application.

2.3.1 Statistics

In a mobile work machine made for container handling, hour, distance, container handling calculations give the user a view from the usage of the machine. According the survey all these parameters can be seen necessary. Only driving direction specific hour and distance calculations are less relevant. The necessity of the hour, distance and container handling calculations is presented in figures 2, 3 and 4.

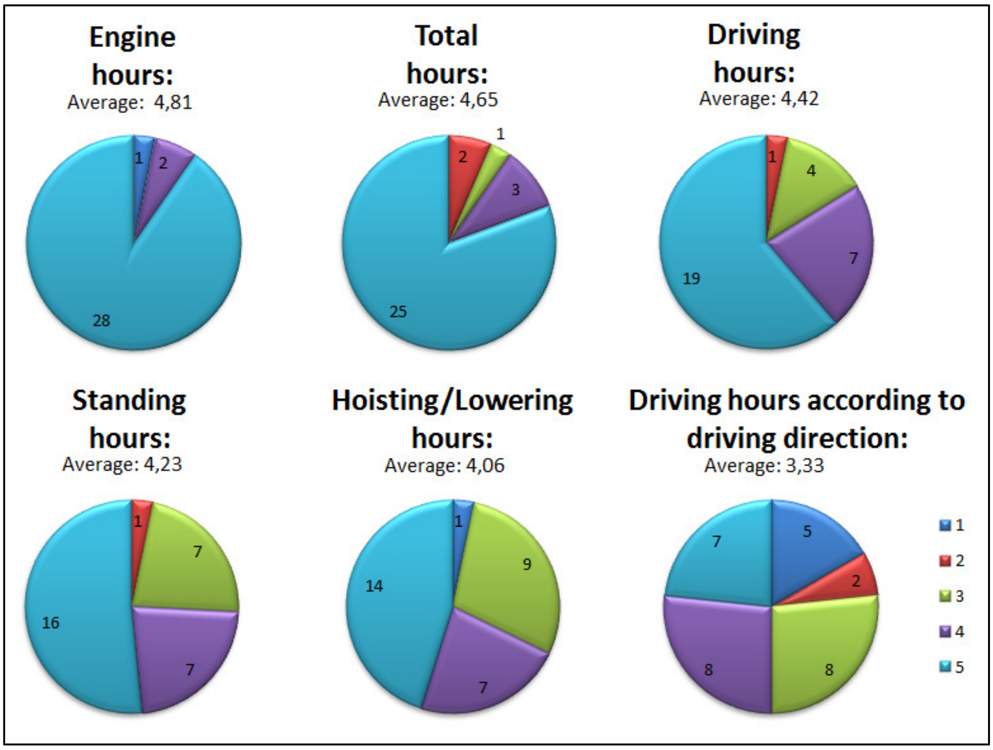


FIGURE 2. Necessity of hour calculations

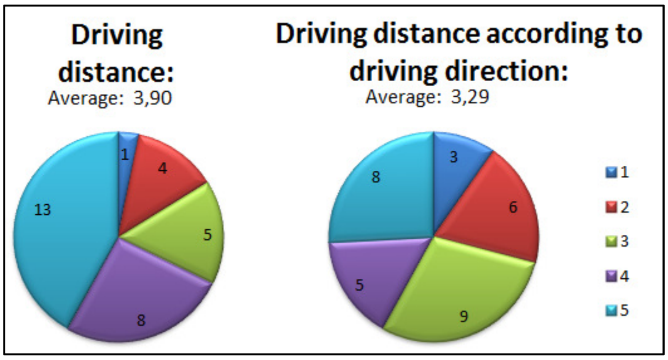


FIGURE 3. Necessity of distance calculations

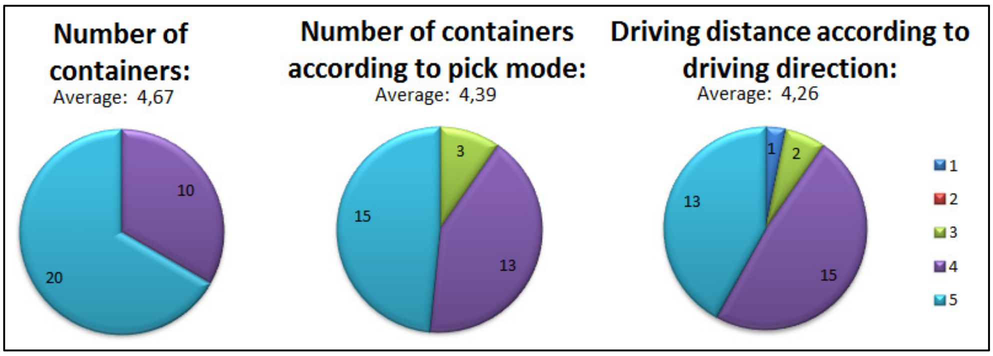


FIGURE 4. Necessity of container calculations

Environmental values are modern-day interests and this can also be seen from the necessity values provided by the survey. Energy consumption helps the user to compare the fuel consumption between different machines. Operation between different machines can vary but used energy versus fuel gives an equal comparison. The necessity of fuel and energy consumption is presented in figure 5.

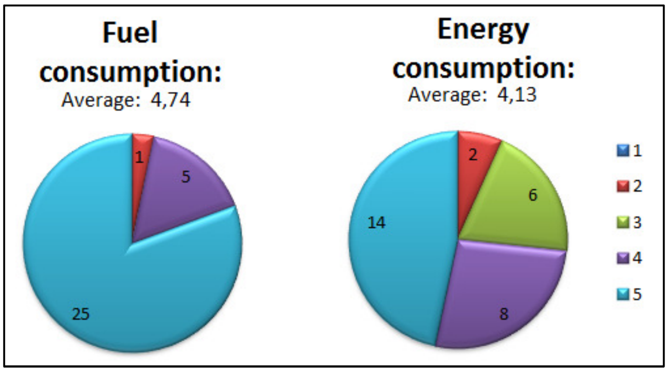


FIGURE 5. Necessity of energy calculations

Safety parameters are important to follow in order to improve safety at work. Follow-up of safety parameters helps to identify the possible training needs of the operator in order to improve safety and therefore prevent the worst case from happening. Necessity of safety parameters is presented in figure 6.

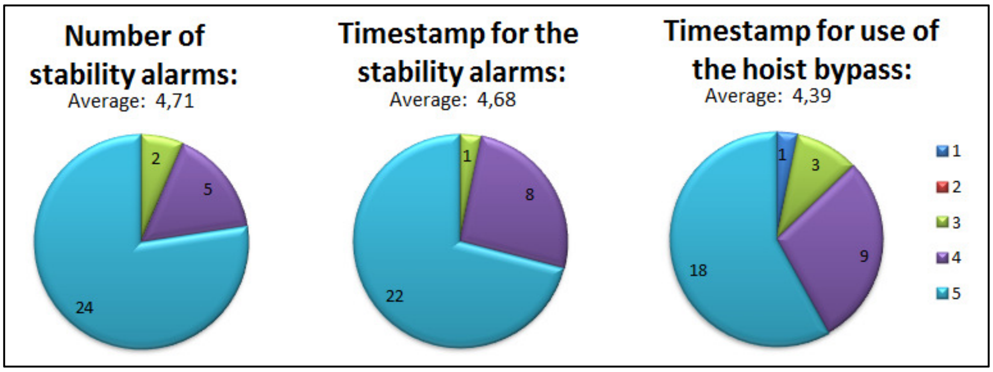


FIGURE 6. Necessity of safety calculations

In the open comments the following answers were received concerning statistics:

- Long term and short term statistics separately
- Description of machine speed, hoist speed etc. during occurrence of stability alarm
- Duty cycle profile
- Require the data to be retained for longer than 30 days.
- Improved ability to download data for driver ability management reports, no of alerts by driver, day, week, etc.
- Reset for short time memory. Total values shown.
- Hoist weights with hoist length.
- Data should not reset after 9999.
- Fuel consumption based on efficiency - km/hoist/lower/containers handled.
- Hoist and lowering meters to determine the wire rope lifecycle.
- Time from last maintenance (maintenance hours).
- Algorithm for rope life - km/hoist/lower/container weight and record this along with reset facility.
- Time of operation per day or per week for central lubrication. Helps to adjust the optimal time for lubrication.
- Timestamp from service box connection and maintenance key usage.
- Timestamp when Overrides (hoist bypass/ t-lock bypass) are used.

Overall statistic parameters are important for the users. As a conclusion from statistic part of the survey the software application needs to have a total and trip counters for the statistic information. In the survey the reset functionality to the statistical information was required by 55 per cent of the respondents. Also the range of different parameters needs to be verified to prevent loss of data.

2.3.1 Alarms

Correct and specific alarm information is important to the maintenance personnel to speed-up the fault diagnostic and to keep the machine down-time as short as possible. Necessity of alarm information features is presented in figure 7.

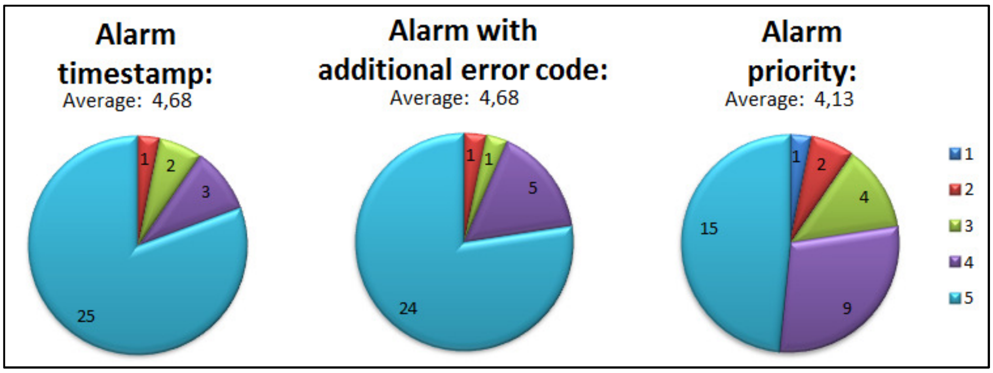


FIGURE 7. Necessity of alarm information

In the open comments the following answers were received concerning alarms:

- Long term top 10 alarms.
- Description of corresponding machine parameters during alarm occurring, example machine speed and spreader height.
- The alarms (particularly stability) need some review as today we have many irrelevant alarms (informal) which fill the list too fast and make the user numb for critical issues.
- Reset only once investigation is completed.
- Alarming system is too sensitive and generates too many alarms in the list that it is difficult to use it as an analytical tool.
- Worth considering having all alarms in the list, but counting occurrence and recording the latest alarm (time & date).
- Need to record all the automation alarm codes with time stamps

- The onboard system to send alerts to management, when example pre-determined level of stability alarms is activated. Currently it is possible to drive through alarms, i.e. I believe a high level stability alarm should stop the machine and only reset to continue following some maintenance intervention and possibly a driver investigation.
- I wish to trigger an HS incident requiring management investigation prior to the driver in question continuing, particularly if the event is a repeat for that asset/driver.
- Alarms, classified by the severity of operation
- Commonly customer required statistics and manufacturer interested statistic
- To save memory space, customers could reset faults, but some "Internal" serious faults should be kept for company use. Stability, driving speed history, spreader height history, priority 1 alarms etc.
- Would be nice to have sort of "triggers" (emergency stop, overheat, stability, overload...) and when active certain parameters are recorded in those situations (re. clarifying the situation when failure, accident etc. happened)
- Operation hour recording when operating with central lubrication alarm active.
- Record brake cooling temperatures in same data packet as the alarm and timestamp.
- Alerts to maintenance staff and operation management, including type, time, driver, and a record of when the alarm was activated, driver at the time, response by (mechanic name) and what was the solution/fix.
- Too many not-critical alarms. Possibility to reduce or just have display pop-up without recording these alarms
- Critical alarms should trigger recording of surrounding events.

Survey presents that it is important to know when, in which situation and what sort of alarm has occurred. Chronological order for the alarm list is more pleasing for 78 per cent of the respondents. In the software application alarm list should be made as a combination of chronological and frequency based listing. Reset functionality to the alarm information should also be added as it was required by 81 per cent of the respondents.

2.3.2 GUI viewer

Monitoring of the driver's GUI is more required functionality than remote operation and from the safety perspective this is also a better option. Remote connection should also cover maintenance view and not just driver's view of the GUI. Necessity of GUI remote connection is presented in figure 8.

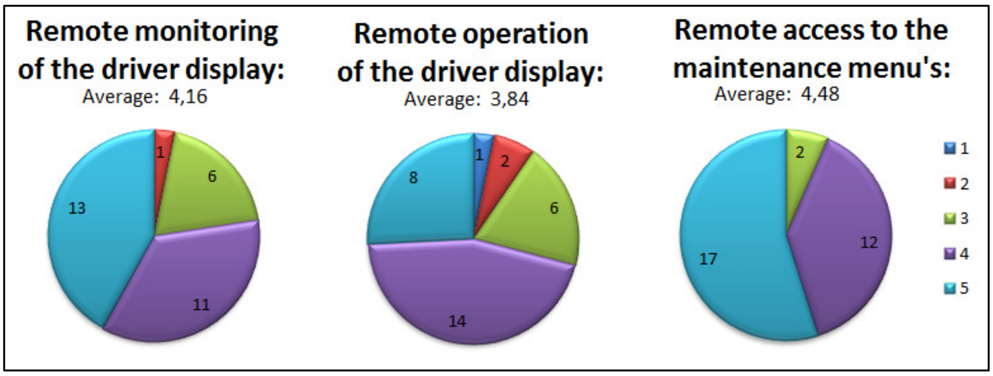


FIGURE 8. Necessity of GUI remote connection

In the open comments the following answers were received concerning remote connection to GUI:

- Possibility to save a test log or screens.
- Good to have but not mandatory.
- This shall cover most of it.
- Consider to highlight the critical stability in reporting and on operators screen.
- Available nodes and IP addresses on the display including all automation/smart units.
- Pin hole camera recording driver /maintenance person and jpg taken and sent to record the image of who presses the acknowledgement

GUI viewer is a new feature to the software application and according the survey it is a wanted feature. Where the software application contains the machine level statistics and alarms the GUI viewer gives the user an access to monitor the machine more accurately at sensor level.

2.3.1 Others

Even though world is moving towards digitalization, a printable report from the machine is still required. Also a lot of the archiving of maintenance operations is done by using manual paper work. Necessity of the printable report is presented in figure 9.

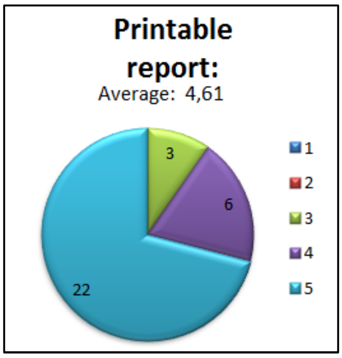


FIGURE 9. Necessity of printable report.

In the open comments the following answers were received concerning other functionalities:

- Maintenance counters.
- Printable report in excel for statistics.
- Driver ID is a key component of this process and that system requires many additional features. Event/alert reporting by driver, asset, day, month to a downloaded file for management use to review driver ability, identify retraining needs or removal from driving, etc.
- Possibility to save history in text format before reset.
- Report should be in format that could be inputted in excel and filtered with time, alarm, etc.
- USB to USB cable connector on machine. No need for USB-Serial adapter (Omron display data port, for example).

2.4 Survey conclusions

According the user survey the following parameters will be stored in to the memory of the control system and will be shown in the software application.

Statistic parameters:

- Machine type
- Machine number
- Engine hours (separate value for total and trip)
- Hybrid hours (separate value for total and trip)
- Traveling hours forward (separate value for total and trip)
- Traveling hours backward (separate value for total and trip)
- Hoisting and lowering hours (separate value for total and trip)
- Standing hours (separate value for total and trip)
- Traveling distance forward (separate value for total and trip)
- Traveling distance backward (separate value for total and trip)
- Number of 20ft containers (separate value for total and trip)
- Number of 40ft containers (separate value for total and trip)
- Number of twinlift container picks (separate value for total and trip)
- Number of critical stability events
- Timestamp from the 10 latest critical stability events
- Fuel consumption
- Energy consumption

Alarm information:

- Total number of alarm events
- Table for 100 alarms including:
  - Alarm identification number
  - Two additional error codes for intelligent units e.g. engine
  - Occurrence count
  - Latest occurrence
  - First occurrence

Other functionalities to be implemented to the software application are GUI remote monitoring, specific troubleshooting information for each alarm code, user manual implementation and printable report functionality. Also the format of printable report should easily support data analyzing.

For future improvements a possibility to add driver identification system and maintenance counters to the control system and therefore to the software application should be considered.

3 MARKET COMPARISON

3.1 Konecranes Truconnect®

Truconnect® is a remote monitoring application for Konecranes CXT® and SMA RTON® cranes. Remote connection is included in new cranes but can also be retrofitted to existing cranes (picture 3).

Application provides operating and productivity data from the crane usage. From the operating data the application estimates the remaining theoretical design working period and service life for crane components. Data is collected using a separate diagnostic unit that collects the information from the crane. The operating data is sent periodically to the Konecranes remote data center. The data center analyzes and compiles the information into graphical reports that the user can view online.

For prompt reaction to certain issues the user has possibility to receive notifications via email and/or text message thus giving the user the ability to address problems and take corrective actions.

User can also optimize crane usage by identifying operator training needs to improve crane safety and productivity with certain crane usage patterns provided by the application. (Konecranes)

PICTURE 3. Truconnect®. (Konecranes)

3.2 Bromma Green zone™

Green zone™ is a remote monitoring application for Bromma crane spreaders (picture 4).

Application provides operating and productivity data from the spreader usage. Application helps the user to prioritize maintenance operations and to resolve problems before they have effect on productivity.

Green zone consists of three different programs:

- Fleet doctor™
  - Monitors operating data of the spreader to detect decelerating performance and faults.
  - Provides troubleshooting information to the maintenance personnel in advance to minimize the spreader downtime when fault occurs.
- Roadmap™
  - Monitors production data of the spreader by following KPI's
  - Provides 10 most frequent fault areas and helps the user to design predictive maintenance to improve spreader performance.
- Workorder™
  - Help user in maintenance planning and scheduling. (Bromma)

PICTURE 4. Green zone™ (Bromma)

3.3 Agco Power WinEEM4

WinEEM4 is an online monitoring application for AGCO diesel engines (picture 5).

Application provides operating data from the engine, including error logs and engine usage profiles. Application can also be used for online monitoring of engine variables, perform test runs and uploading software to the ECU.

Connection is done via engine CAN bus. Remote monitoring is not included into this application.

PICTURE 5. WinEEM4

4 CONTROL SYSTEM FEATURES

4.1 Control system description

Simplified structure of the used control system consists of Omron CJ2 -programmable logic controller (PLC), Omron NS8 -graphical user interface (GUI) and Ethernet/IP bus for data connection. Software application will be connected to the control system via RJ-45 connector on the Ethernet/IP bus (picture 6).

PICTURE 6. Simplified structure of used control system.

4.2 Recording of the maintenance information to the control system

This chapter defines specification for the statistical and alarm information stored into the memory of the control system. Specified memory areas for statistic and alarm information can be found from appendix 2.

4.2.1 Statistics

Statistic information is stored in to the memory of the control system including following parameters:

Machine type:

- This value is a hard coded number in the control system. This can be used to show or hide certain data in different machine. Example in non-hybrid machine, the hours of the hybrid system are not shown. Also some visualization for the user interface can be done using this parameter. For example to add a picture of the type of the machine where connection is established.

Machine number:

- This parameter is to identify the machine. Operator can add the machine number from operator's display.

Engine hours (separate value for total and trip):

- This parameter shows total engine hours. Calculation is done when engine is running, engine rpm's over 400, with 1s resolution.

Hybrid hours (separate value for total and trip):

- This parameter shows total hybrid hours. Calculation is done when hybrid is activated with 1s resolution. This value is calculated only when selected machine type is hybrid.

Traveling hours forward (separate value for total and trip)

- This parameter shows total traveling hours to driving direction forward. Calculation is done when actual machine speed is greater than 0,5km/h with 1s resolution.

Traveling hours backward (separate value for total and trip)

- This parameter shows total traveling hours to driving direction backward. Calculation is done when actual machine speed is greater than 0,5km/h with 1s resolution.

Hoisting and lowering hours (separate value for total and trip)

- This parameter shows total hoisting and lowering hours. Calculation is done when hoisting or lowering speed is greater than 0 m/min with 1s resolution.

Standing hours (separate value for total and trip)

- This parameter shows total standing hours. Calculation is done when machine is standing still without any movements for driving, hoisting or lowering and engine is running or the hybrid is activated, with 1s resolution.

Traveling distance forward (separate value for total and trip)

- This parameter shows total traveling kilometers to driving direction forward. Calculation is done when actual machine speed is greater than 0,5km/h with 0,1km resolution.

Traveling distance backward (separate value for total and trip)

- This parameter shows total traveling kilometers to driving direction backward. Calculation is done when actual machine speed is greater than 0,5km/h with 0,1km resolution.

Number of 20ft containers (separate value for total and trip)

- This parameter shows total number of picked 20ft containers.

Number of 40ft containers (separate value for total and trip)

- This parameter shows total number of picked 40ft containers

Number of twinlift container picks (separate value for total and trip)

- This parameter shows total number of done twinlift picks, 2 x 20ft container picks.

Number of critical stability events

- This parameter shows total number critical stability events. Stability event is determined to be critical when stability goes to less than 30 per cent.

Timestamp from the 10 latest critical stability events

- These parameters show timestamp for the 10 latest stability alarms.

Fuel consumption

- This parameter shows average engine fuel consumption in litres per hour.

Energy consumption

- This parameter shows average energy used for driving, hoisting and lowering in kilowatts.

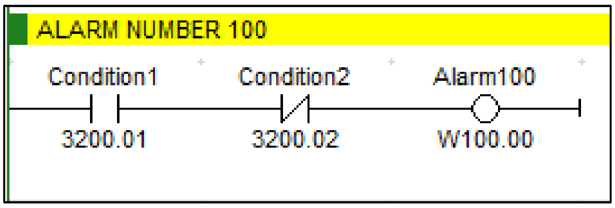
3.2.2 Alarms

Alarm information is stored in to the memory of the control system including following parameters:

- Total number of alarm events
- Table for 100 alarms including:
    - Alarm identification number
    - Two additional error codes for intelligent units e.g. engine
    - Occurrence count
    - Latest occurrence
    - First occurrence

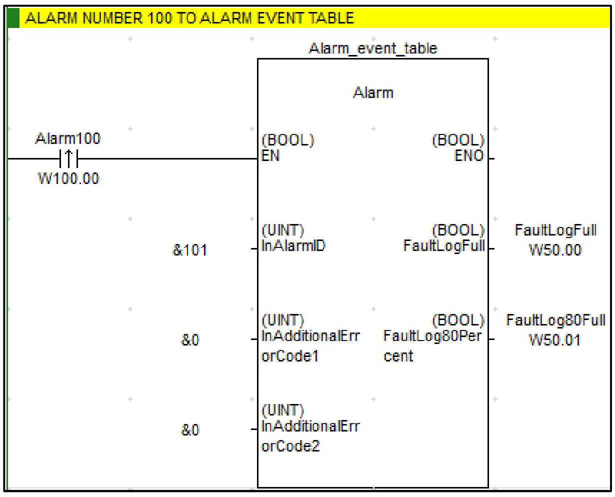
Alarm information handling in the control system is explained below. For an example, alarm identification 100 is used with description of ALARM 100 and without any additional error codes.

In the control system each alarm is specified to control output as a sign of an active alarm. ALARM 100 is activated if condition 1 is on and condition 2 is off (picture 7).



PICTURE 7. Alarm output control.

Saving the alarm data to the alarm event table is executed on the Alarm -function block. Function block execution is activated on the rising edge of the alarm bit. Inputs to the function block are alarm identification number and two additional error codes. As an outputs alarm log full and alarm log 80 per cent full information's are provided. These can be used to control service light on the machine (picture 8).



PICTURE 8 Alarm data function block

Alarm -function block handles alarm event table and contains variables shown in the table 1.

TABLE 1. Variables of Alarm -function block

Name	Data Type	Memory address	Initial Value	Comment
EN	BOOL	internal	false	Controls execution of the Function Block
InAlarmID	UINT	internal	0	Alarm number
InAdditionalErrorCode1	UINT	internal	0	First additional error code
InAdditionalErrorCode2	UINT	internal	0	Second additional error code
RowNumber	UINT	internal	0	Alarm list row number
AlarmAddedToList	BOOL	internal	false	Alarm was added to the list
FaultLogFull	BOOL	internal	false	Fault log is full
FaultLog80Percent	BOOL	Internal	false	Fault log is 80% full
YEARPLUSMONTH	WORD	A353	-	Year and month
DATEPLUSHOUR	WORD	A352	-	Date and hour
MINPLUSSEC	WORD	A351	-	Minutes and seconds
TotalNumberOfAlarms	UINT	E3_99	-	Total number of alarms
AlarmID	UINT[100]	E3_100	-	Alarm id from row 0 to 99
AdditionalCode1	UINT[100]	E3_200	-	Additional error code1 from row 0 to 99
AdditionalCode2	UINT[100]	E3_300	-	Additional error code2 from row 0 to 99
NumberOfAlarms	UINT[100]	E3_400	-	Number of alarms from row 0 to 99
LastYearMonth	WORD[100]	E3_500	-	Last year and month of alarms from row 0 to 99
LastDayHour	WORD[100]	E3_600	-	Last day and hour of alarms from row 0 to 99
LastMinuteSecond	WORD[100]	E3_700	-	Last minute and second of alarms from row 0 to 99
FirstYearMonth	WORD[100]	E3_800	-	First year and month of alarms from row 0 to 99
FirstDayHour	WORD[100]	E3_900	-	First day and hour of alarms from row 0 to 99
FirstMinuteSecond	WORD[100]	E3_1000	-	First minute and second of alarms from row 0 to 99

PLC code in the function block is done using structured text format. To improve the readability of the code all the comments are added using green color. Code structure is shown and explained below:

```
(* Initializing values *)
RowNumber := 0;
AlarmAddedToList := FALSE;
FaultLogFull := FALSE;
FaultLog80Percent := FALSE;

(* Total number of alarms *)
TotalNumberOfAlarms := TotalNumberOfAlarms + 1;
(* Adding Alarms to Fault List *)
(* WHILE loop is executed as long as AlarmAddedToList or FaultLogFull variables becomes true *)
WHILE AlarmAddedToList = FALSE AND FaultLogFull = FALSE DO

(* If AlarmID with correct additional error can be found from rows 0 to 99 then number of alarms is added by one
and last occurence time is updated *)
IF AlarmID[RowNumber] = InAlarmID AND AdditionalCode1[RowNumber] = InAdditionalErrorCode1 AND
AdditionalCode2[RowNumber] = InAdditionalErrorCode2 THEN
                NumberOfAlarms[RowNumber] := NumberOfAlarms[RowNumber] +1;
                LastYearMonth[RowNumber] := YEARPLUSMONTH;
                LastDayHour[RowNumber] := DATEPLUSHOUR;
                LastMinuteSecond[RowNumber] := MINPLUSSEC;
                AlarmAddedToList := TRUE; (* AlarmAddedToList is set to true and WHILE loop execution is
ended *)
END_IF;

(* If an alarm row with content of 0 can be found from rows 0 to 99  then number of alarms is added by one and first
occurence time is updated *)
IF AlarmID[RowNumber] = 0 THEN
                AlarmID[RowNumber] := InAlarmID ;
                AdditionalCode1[RowNumber] := InAdditionalErrorCode1;
                AdditionalCode2[RowNumber] := InAdditionalErrorCode2;
                NumberOfAlarms[RowNumber] := 1;
                FirstYearMonth[RowNumber] := YEARPLUSMONTH;
                FirstDayHour[RowNumber] := DATEPLUSHOUR;
                FirstMinuteSecond[RowNumber] := MINPLUSSEC;
                AlarmAddedToList := TRUE; (* AlarmAddedToList is set to true and WHILE loop execution is
ended *)
END_IF;

(* Row number is added by one because alarm id in this row wasn't 0 or the same as the currently active alarm Id *)
RowNumber := RowNumber + 1; IF RowNumber >= 100 THEN (* Alarm rows 0 to 99 already contained alarm ID
that is different than the current active alarm id -> fault log is full *)
                FaultLogFull := TRUE; (* FaultLogFull is set to true and WHILE loop execution is ended *)
END_IF;
END_WHILE;

IF AlarmID[80] <> 0 THEN (* alarm rows 0 to 80 already contain an error code. Alarm list will soon become full if
download/reset is not executed *)
                FaultLog80Percent := TRUE;
END_IF;
```

5 SOFTWARE APPLICATION FEATURES AND USER INTERFACE

5.1 Basic requirements

The application shows main statistics and alarm history for the maintenance personnel. Software application will be done to support Windows based operating system. The actual source code will be done using C^{++} -programming language and the visual features of the user interface will be done using software called QT Creator.

Using this application the service personnel can print a diagnostics report to be attached to the maintenance archives. Report will be in Microsoft Excel format, so the further analysis from the report can be easily done.

The user interface of the software application is divided into five separate main views:

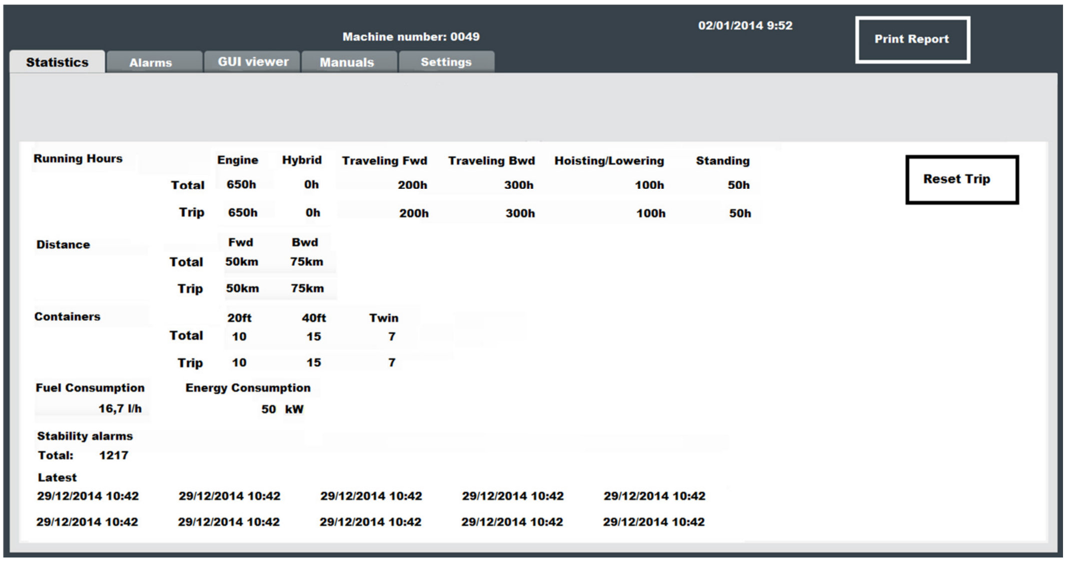
- Statistics
- Alarms
- GUI remote viewer
- Manuals
- Setting

5.2 Statistics

In the statistics view user can see all the main statistics from the machine use. All machine parameters consist of 2 values. Total value which shows machine lifetime statistics and trip statistics which can be reset with intervals the user defines, for example during scheduled maintenance (picture 9).

Statistic information is read to the user interface using FINS-command called memory area read. This FINS-command is executed every 1 second by the software application.

Trip values can be reset from the control system using FINS-command called memory area fill. Software application sends value 0000_{hex} to all memory areas which contains a trip value. This FINS-command is executed when user enables reset trip -button on the user interface.



PICTURE 9. Statistics view

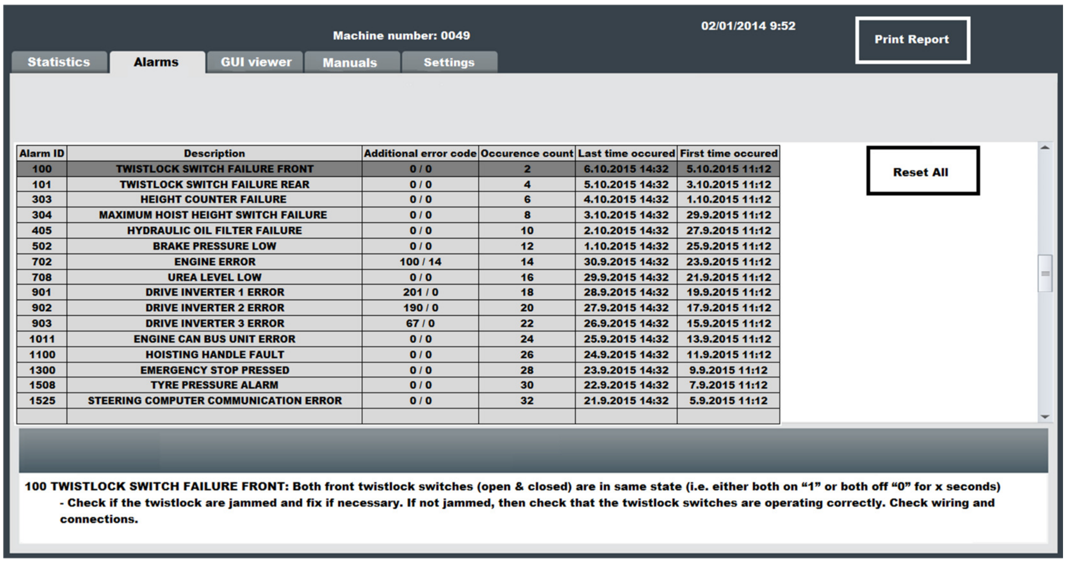
5.3 Alarms

In the alarm view user can see machine alarm history with timestamps and occurrence frequencies. User can sort the alarm list using all columns on the list to help the user to make the list as easy as possible to interpret. When user activates the alarm row a troubleshooting window will open on the bottom of the screen. Troubleshooting window contains specific information from the alarm. It shows the terms which caused the selected alarm and also troubleshooting tips to be checked first (picture 10).

Software application reads alarm id number, additional error codes, occurrence count, first and last occurrence time from the control system using FINS-command called memory area read.

Software application contains separately parameterized file which contains cross-reference list to match the alarm identification number to the correct alarm description and troubleshooting information. To support multilingual user interface this parameter file will be added for every language separately. This separately parameterized file makes it possible to add and maintain the alarm list without the need of changing the actual source code of the software application.

From the reset all -button the user can reset alarm history from the control system using FINS-command called memory area fill. When alarm list is resetted using this software application the control system handles resetting of the alarm list also in the GUI interface on the machine, so that all the alarm lists on the machine are synchronized.



PICTURE 10. Alarms view

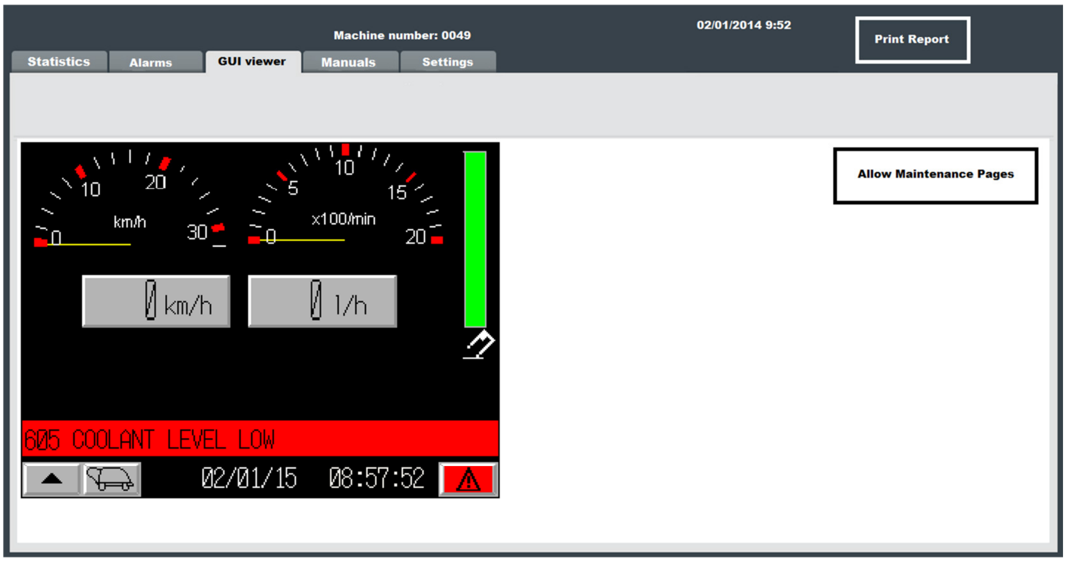
5.4 GUI viewer

GUI viewer allows maintenance personnel to monitor machines GUI remotely. This helps maintenance personnel to resolve problems immediately by connecting and seeing exactly what the local operator sees (picture 11).

From the GUI viewer the maintenance personnel can allow maintenance page access to the local operator but for safety reasons cannot operate any controls on the GUI remotely.

From Allow Maintenance Pages –button, the software application writes $FFFF_{hex}$ to the specified memory address on the control system using FINS-command called memory area write. After this the control system allows maintenance view access for the local operator. If the local operator has not touched the display in 2 minutes then control system will close the maintenance view automatically and operator display will change back to operator’s view.

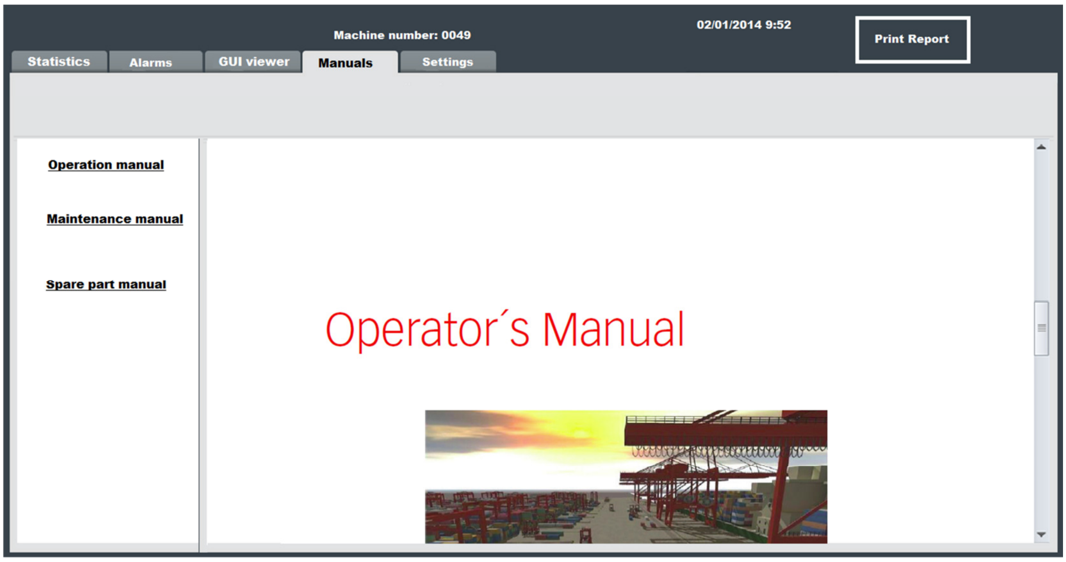
GUI viewer is done by accessing http://192.168.1.50/monitor.htm via web browser (Omron 2008).



PICTURE 11. GUI viewer view

5.5 Manuals

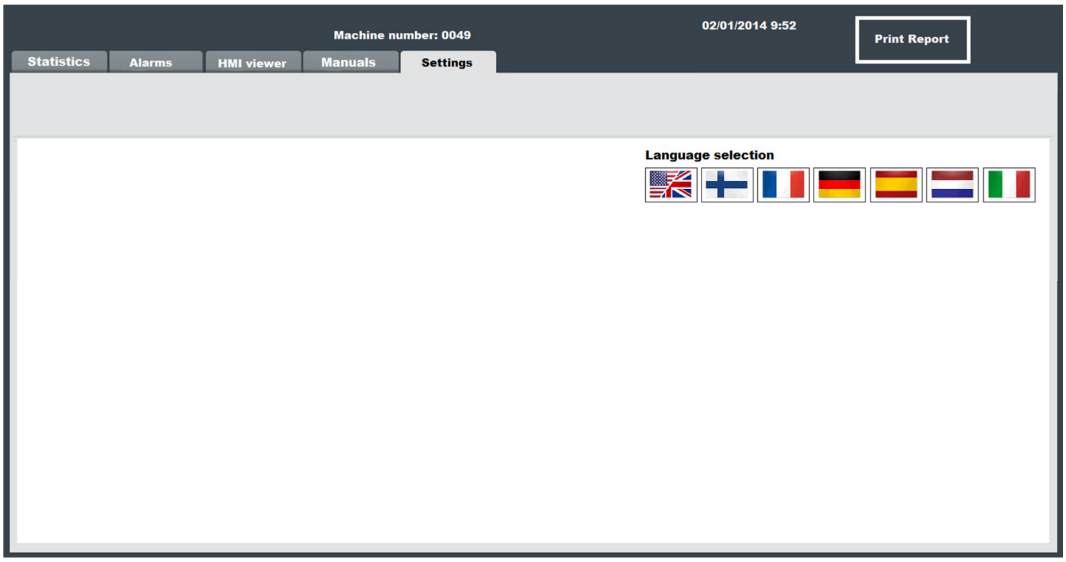
Operation, maintenance and spare part manuals are implemented to the Manuals view. Manuals are implemented as pdf –files and updating the manual does not require any changes to the source code of the software application. Just updating of new manual files is needed (picture 12).



PICTURE 12. Manuals view

5.6 Settings

From the setting view maintenance personnel can select preferred language to be used in the application. Selected language determines language of all text tags in the user interface and the used parameter file in the alarm view (picture 13).

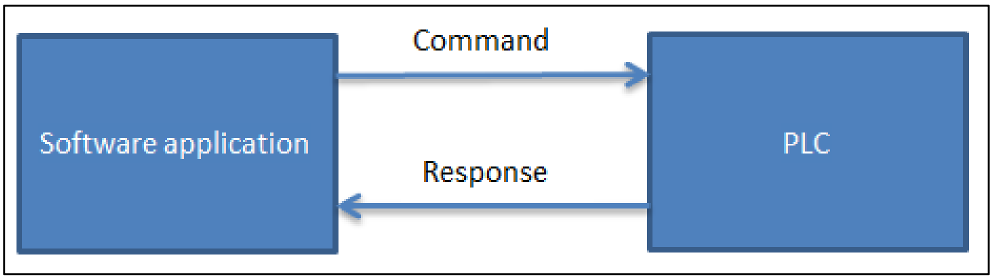


PICTURE 13. Settings view

6 DATA TRANSFER BETWEEN SOFTWARE APPLICATION AND CONTROL SYSTEM

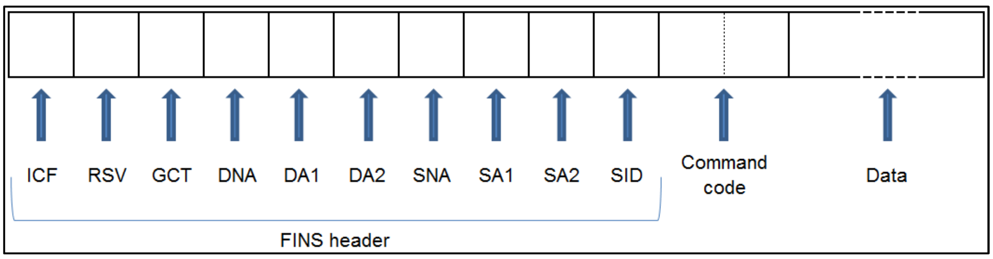
6.1 Overview of Omron FINS -protocol

Data transfer between software application and control system is done using Omron FINS –messages using UDP -protocol via Ethernet/IP bus. Software application sends FINS –command to the PLC and receives the response from the PLC (picture 14). This chapter defines needed message structures to fulfill all the functionalities designed for the software application.

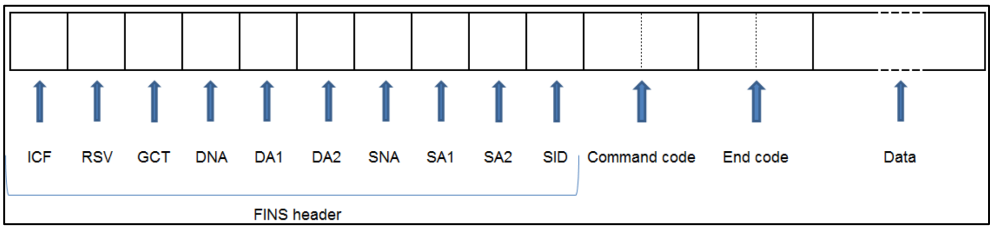


PICTURE 14. Data transfer sequence

FINS commands are a command system for message services across different OMRON networks. They can be used for various control operations, such as sending and receiving data, changing controller operating modes, executing forced set and reset operations and performing file operations. FINS commands can be used in serial and Ethernet networks. FINS -command and -response frame variables are presented in pictures 15 and 16.

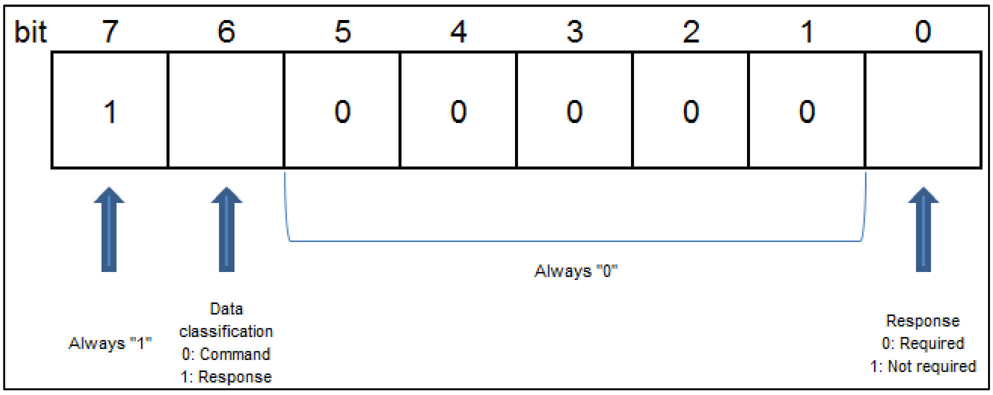


PICTURE 15. FINS command frame configuration.



PICTURE 16. FINS response frame configuration.

The ICF determines the form of the FINS frame. If the frame is configured to be a command then also the need of the response message can be configured. Configuration of the ICF is shown in the picture 17.



PICTURE 17. ICF configuration

RSV is reserved for the system use and should always contain a value of 00_{hex}.

GCT is used when communicating across different network layers. If communication takes place trough different network layers this value is set to 07_{hex}, otherwise 02_{hex} when sending. In the response message this value is decreased by one at each network level and result is received. This value is for the system use.

DNA specifies the location of the FINS command destination and SNA specifies the location of the FINS command source. DNA and SNA are configured within the ranges of 00_{hex} when communicating in the local network and 01_{hex} to $7F_{hex}$ when communicating to remote network addresses.

DA1 is destination node address and SA1 is the source node address. These are configured within the ranges of 00_{hex} when communicating internally in the PLC, 01_{hex} to 20_{hex} when communicating in the Controller Link Network, 01_{hex} to FE_{hex} when communicating in the Ethernet network and FF_{hex} for broadcasting. In the Controller Link Network DA1 determines destination node address and SA1 is similar to the source. In the Ethernet network DA1 is the last address part of the destination IP address and SA1 is similar for the source.

DA2 and SA2 are configured according to the unit type used in the connection. For CPU unit 00_{hex} is used, FE_{hex} for Controller Link unit or Ethernet unit, 10_{hex} to $1F_{hex}$ for CPU Bus unit and $E1_{hex}$ for Inner Board.

SID is used to identify the data transmission. This can be set between 00_{hex} and FF_{hex}. FINS response frame contains same SID as the FINS command it is responding to.

Command code defines the data content of the command and response frame. All the command codes can be found from the appendix 3.

End code defines the status of the FINS response. When sent command is executed normally the 0000_{hex} is received. Other end codes refer to an error in the communication (Omron 2010).

6.2 Used FINS –commands in the software application

When using Ethernet UDP -protocol the FINS header consists of following parameters:

- ICF = 80_{hex}
- RSV = 00_{hex}
- GCT = 02_{hex}
- DNA = 00_{hex}
- DA1 = FD_{hex} = Last digit of the PLC IP address = 192.168.1.253

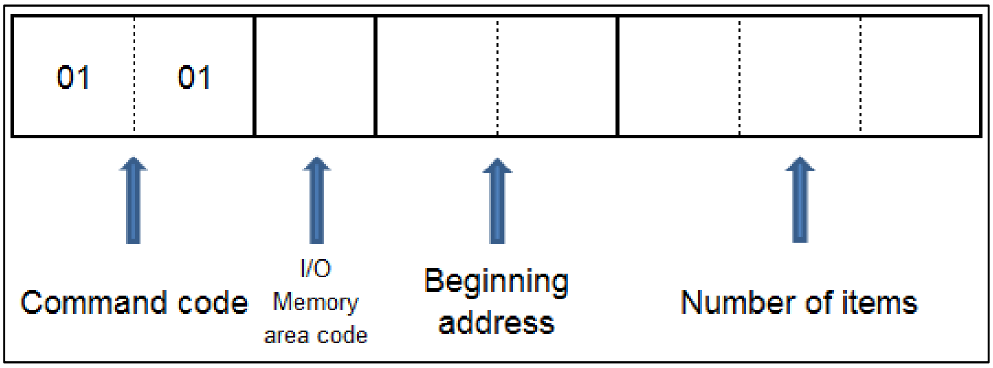
- DA2 = 00_{hex}
- SNA = 00_{hex}
- SA1 = 63_{hex} = Last digit of the software application IP address = 192.168.1.99
- SA2 = 00_{hex}
- SID = varies between 01_{hex} to 07_{hex} to identify sent command

6.2.1 Memory area read

Memory area read –command reads the contents of the specified memory addresses. This is used to read statistics and alarm information from the PLC.

Reading statistic information, memory areas E3_1 to E3_82, is executed with following command sent from the software application (picture 18):

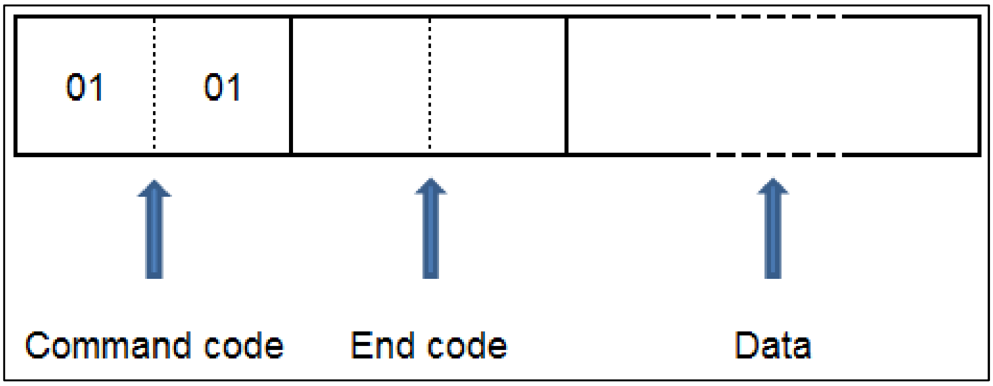
- SID = 01_{hex}
- Command code = 0101_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = 0001_{hex} = E3_1
- Number of items = 000052_{hex} = 82 words



PICTURE 18. Memory area read –command

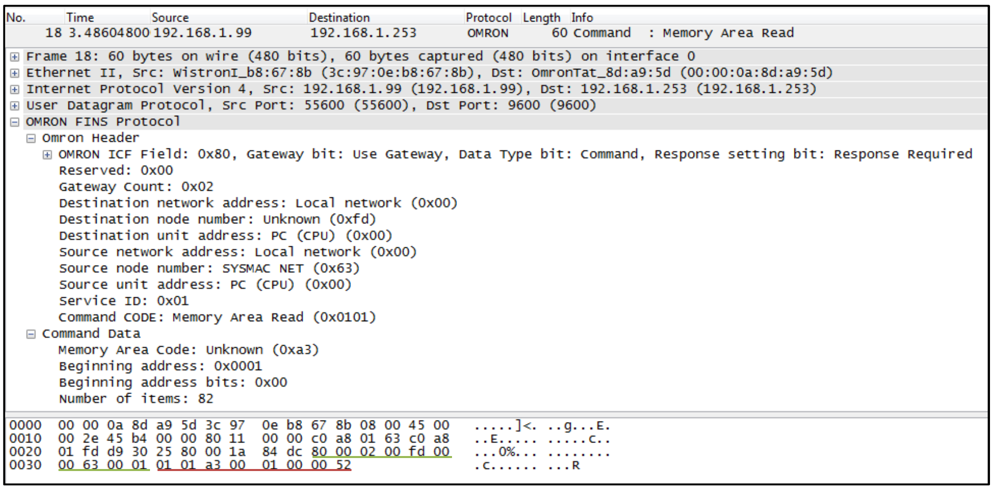
Response from the control system is received in following format (picture 19):

- Command code = 0101_{hex}
- End code = 0000hex = normal completion
- Data: Values of the selected memory areas

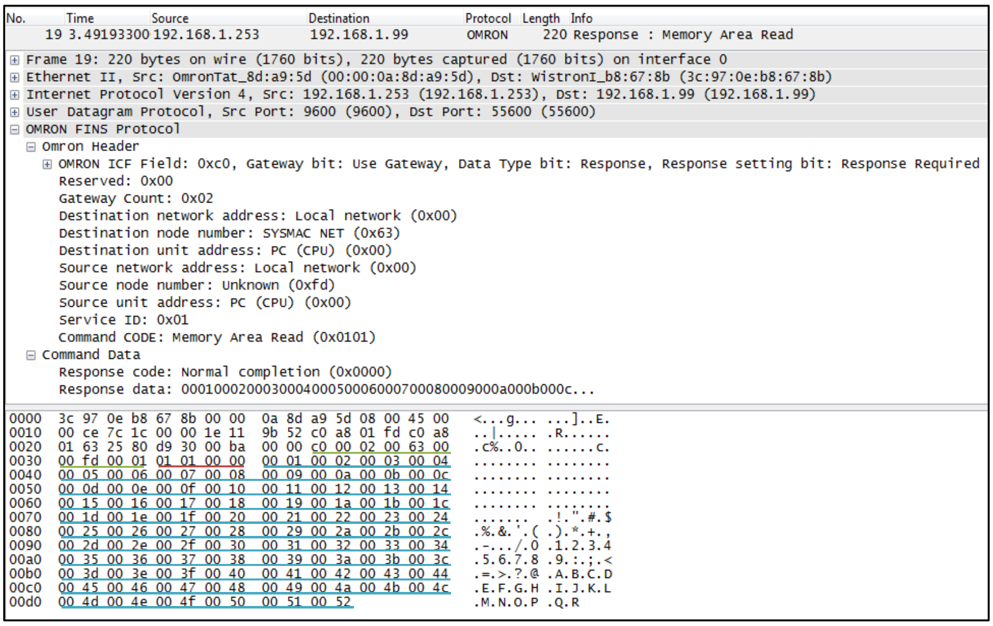


PICTURE 19. Memory area read –response

Below is an example from the memory area read command/response for statistics data. In the example memory address values match to the actual memory address, E3_1 = 1_{dec}, E3_2 = 2_{dec}, E3_3 = 3_{dec} and so on. Pictures 20 and 21 are taken using Wireshark Ethernet analyzer. In the pictures FINS frame is underlined with green color, command/response frame with red color and received data with blue color.



PICTURE 20. Memory area read -command on Wireshark for statistics data



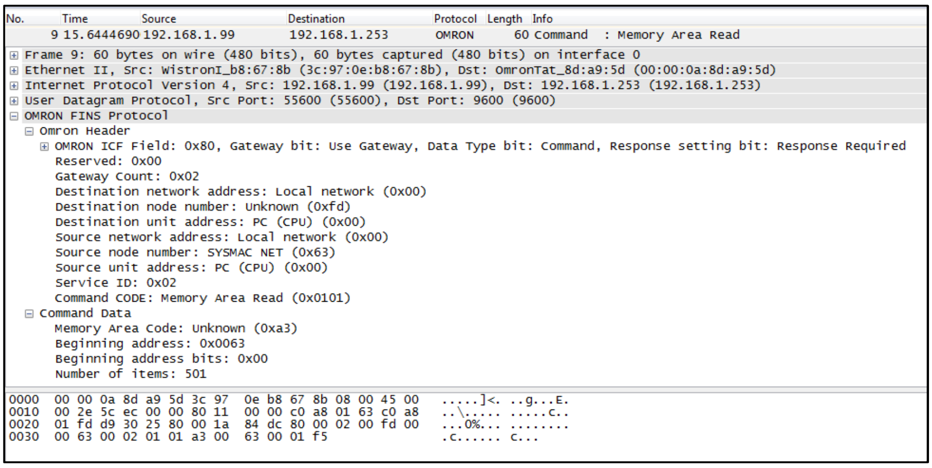
PICTURE 21. Memory area read -response on Wireshark for statistics data

Reading of the alarm information is divided into 2 different FINS –commands, because the maximum number of read memory addresses is 999 on the Ethernet network.

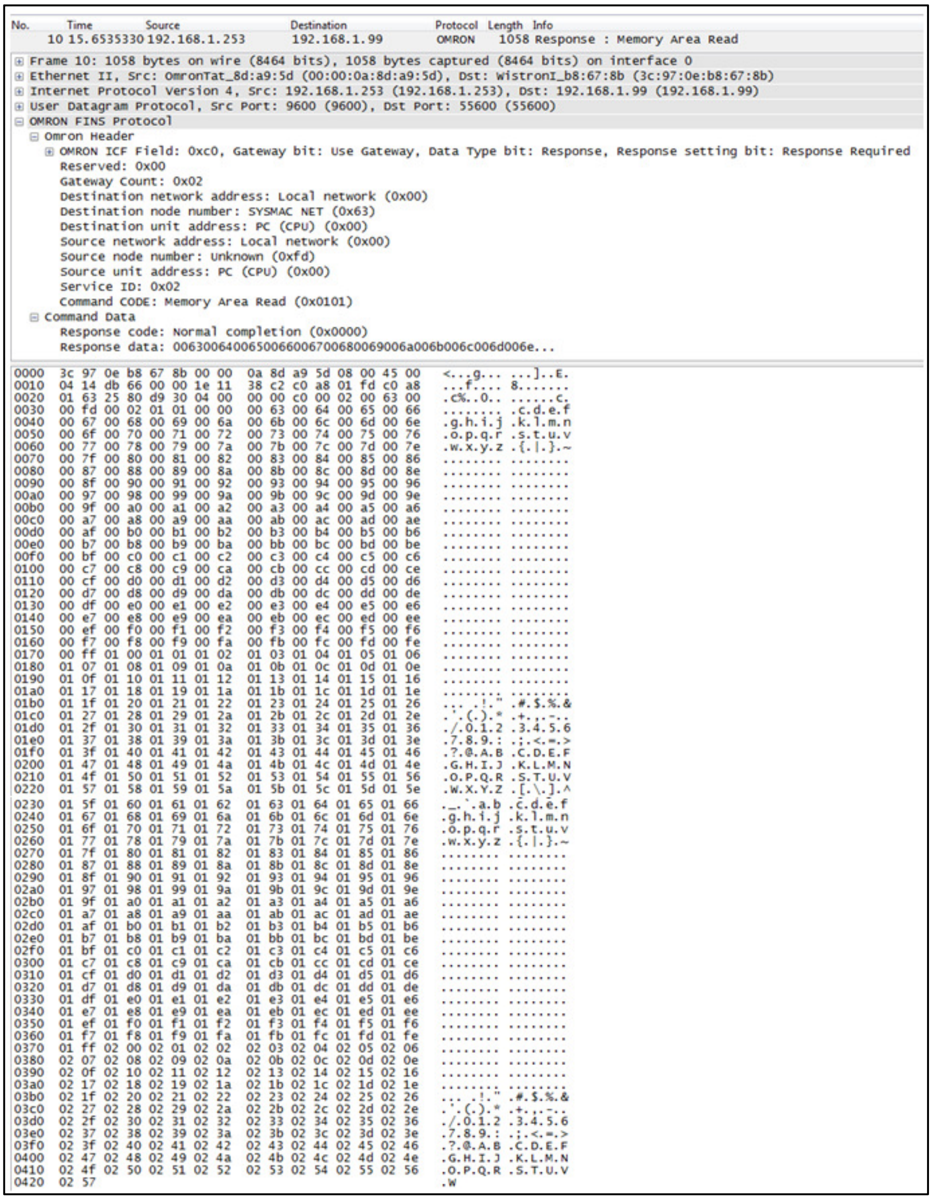
First command reads memory addresses E3_99 to E3_599, then command is in format:

- SID = 02_{hex}
- Command code = 0101_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = 0063_{hex} = E3_99
- Number of items = $0001F5_{hex}$ = 501 words

An example from the first command to read alarm information is presented in the pictures 22 and 23. In the example memory address values match to the actual memory address, E3_99 = 99_{dec}, E3_100 = 100_{dec}, E3_101 = 101_{dec} and so on. Pictures are taken from Ethernet analyzer software called Wireshark.



PICTURE 22. Memory area read -command on Wireshark for alarm data (E3_9 - E3_599)

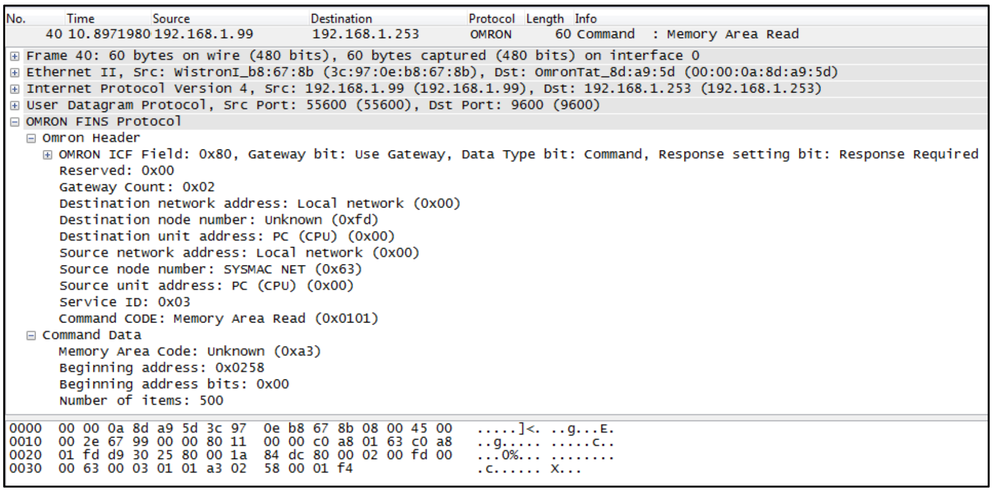


PICTURE 23. Memory area read -response on Wireshark for alarm data (E3_9 - E3_599)

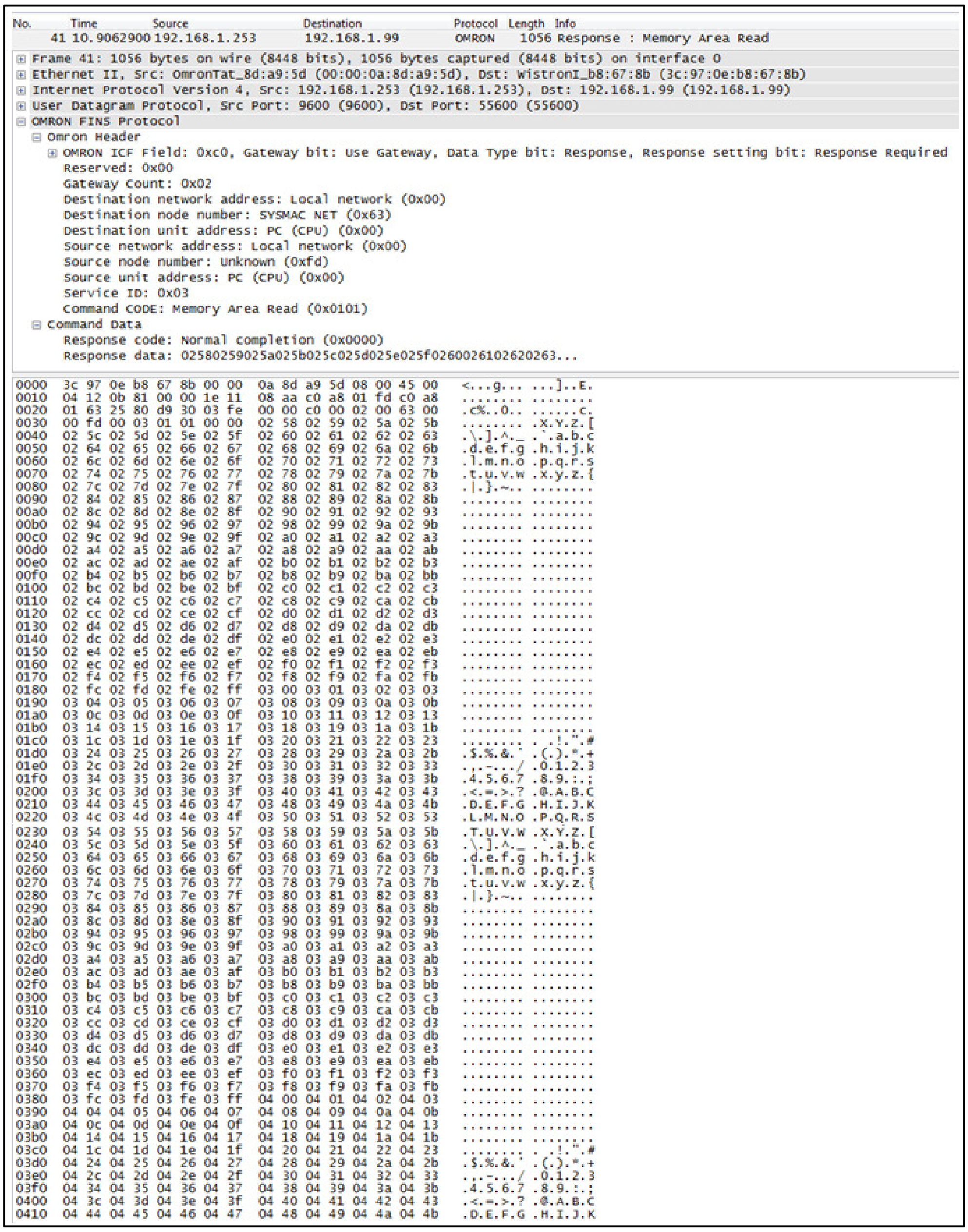
Second command reads memory addresses E3_600 to E3_1099, then command is in format:

- SID = 03_{hex}
- Command code = 0101_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = 0258_{hex} = E3_600
- Number of items = $0001F4_{hex}$ = 500 words

An example from the second command to read alarm information is presented in the pictures 24 and 25. In the example memory address values match to the actual memory address, E3_600 = 600_{dec}, E3_601 = 601_{dec}, E3_602 = 602_{dec} and so on. Pictures are taken from Ethernet analyzer software called Wireshark.



PICTURE 24. Memory area read -command on Wireshark for alarm data (E3_600 - E3_1099)



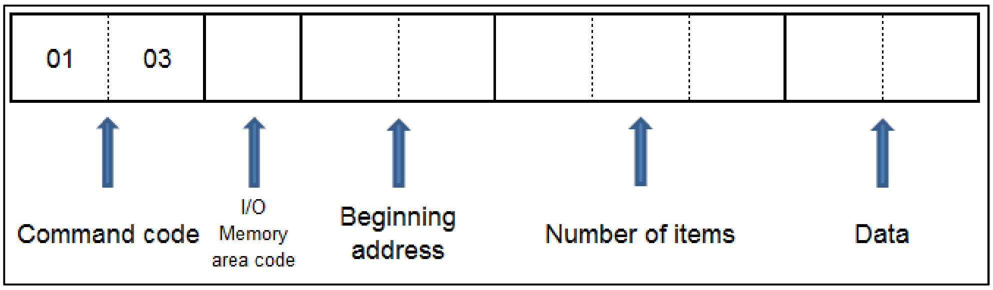
PICTURE 25. Memory area read -response on Wireshark for alarm data (E3_600 - E3_1099)

6.2.2 Memory area fill

Memory area fill –command write specified data to the specified memory addresses. This is used to reset statistics and alarm information from the PLC.

Resetting statistical trip indicators, memory areas E3_70 to E3_82, is executed with following command sent from the software application (picture 26):

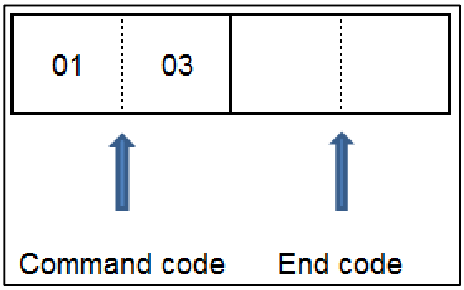
- SID = 04_{hex}
- Command code = 0103_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = 0046_{hex} = E3_70
- Number of items = $00000D_{hex}$ = 13 words
- Data = 0000_{hex} = 0_{dec}



PICTURE 26. Memory area fill -command

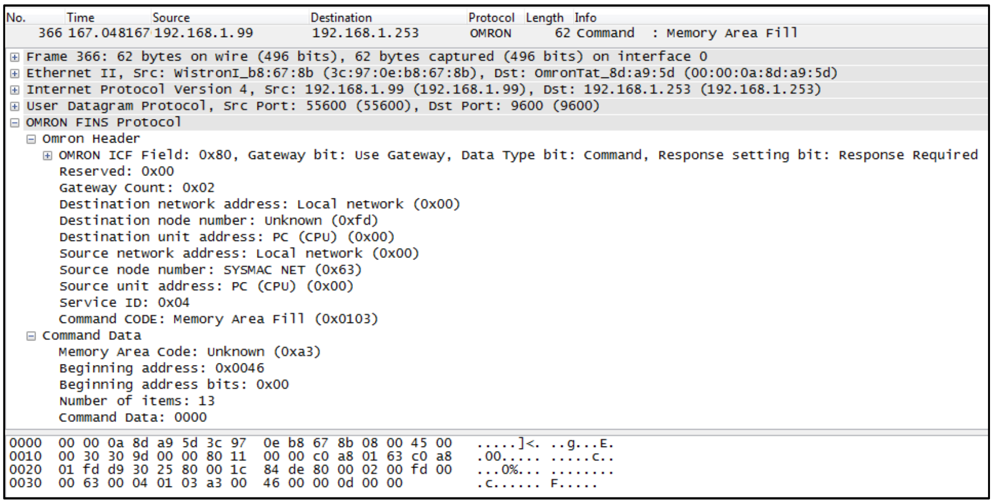
Response from the control system is received in following format (picture 27):

- Command code = 0103_{hex}
- End code = 0000hex = normal completion



PICTURE 27. Memory area fill -response

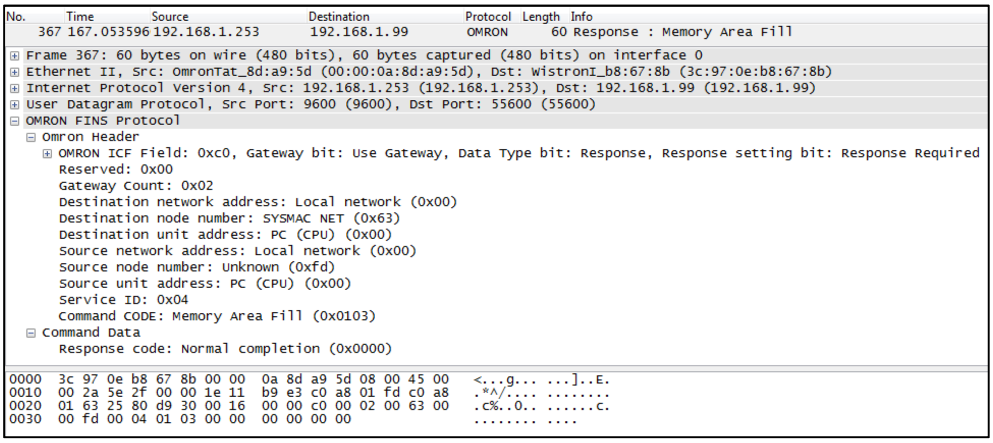
Pictures 28 and 29 presents the memory area fill -command for the statistical trip data in Wireshark Ethernet analyzer.



```
No.   Time              Source          Destination      Protocol  Length  Info
  366 167.048167 192.168.1.99      192.168.1.253    OMRON        62 Command   : Memory Area Fill
⊞ Frame 366: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
⊞ Ethernet II, Src: WistronI_b8:67:8b (3c:97:0e:b8:67:8b), Dst: OmronTat_8d:a9:5d (00:00:0a:8d:a9:5d)
⊞ Internet Protocol Version 4, Src: 192.168.1.99 (192.168.1.99), Dst: 192.168.1.253 (192.168.1.253)
⊞ User Datagram Protocol, Src Port: 55600 (55600), Dst Port: 9600 (9600)
⊟ OMRON FINS Protocol
  ⊟ Omron Header
    ⊞ OMRON ICF Field: 0x80, Gateway bit: Use Gateway, Data Type bit: Command, Response setting bit: Response Required
      Reserved: 0x00
      Gateway Count: 0x02
      Destination network address: Local network (0x00)
      Destination node number: Unknown (0xfd)
      Destination unit address: PC (CPU) (0x00)
      Source network address: Local network (0x00)
      Source node number: SYSMAC NET (0x63)
      Source unit address: PC (CPU) (0x00)
      Service ID: 0x04
      Command CODE: Memory Area Fill (0x0103)
  ⊟ Command Data
      Memory Area Code: Unknown (0xa3)
      Beginning address: 0x0046
      Beginning address bits: 0x00
      Number of items: 13
      Command Data: 0000

0000  00 00 0a 8d a9 5d 3c 97  0e b8 67 8b 08 00 45 00   .....]<. ..g...E.
0010  00 30 30 9d 00 00 80 11  00 00 c0 a8 01 63 c0 a8   .00..... .....c..
0020  01 fd d9 30 25 80 00 1c  84 de 80 00 02 00 fd 00   ...0%... ........
0030  00 63 00 04 01 03 a3 00  46 00 00 0d 00 00         .c...... F.....
```

PICTURE 28. Memory area fill -command on Wireshark for trip data.



```
No.   Time              Source          Destination      Protocol  Length  Info
  367 167.053596 192.168.1.253     192.168.1.99     OMRON        60 Response  : Memory Area Fill
⊞ Frame 367: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊞ Ethernet II, Src: OmronTat_8d:a9:5d (00:00:0a:8d:a9:5d), Dst: WistronI_b8:67:8b (3c:97:0e:b8:67:8b)
⊞ Internet Protocol Version 4, Src: 192.168.1.253 (192.168.1.253), Dst: 192.168.1.99 (192.168.1.99)
⊞ User Datagram Protocol, Src Port: 9600 (9600), Dst Port: 55600 (55600)
⊟ OMRON FINS Protocol
  ⊟ Omron Header
    ⊞ OMRON ICF Field: 0xc0, Gateway bit: Use Gateway, Data Type bit: Response, Response setting bit: Response Required
      Reserved: 0x00
      Gateway Count: 0x02
      Destination network address: Local network (0x00)
      Destination node number: SYSMAC NET (0x63)
      Destination unit address: PC (CPU) (0x00)
      Source network address: Local network (0x00)
      Source node number: Unknown (0xfd)
      Source unit address: PC (CPU) (0x00)
      Service ID: 0x04
      Command CODE: Memory Area Fill (0x0103)
  ⊟ Command Data
      Response code: Normal completion (0x0000)

0000  3c 97 0e b8 67 8b 00 00  0a 8d a9 5d 08 00 45 00   <...g... ...]..E.
0010  00 2a 5e 2f 00 00 1e 11  b9 e3 c0 a8 01 fd c0 a8   .*^/.... ........
0020  01 63 25 80 d9 30 00 16  00 00 c0 00 02 00 63 00   .c%..0.. ......c.
0030  00 fd 00 04 01 03 00 00  00 00 00 00               ........ ....
```

PICTURE 29. Memory area fill -response on Wireshark for trip data.

Resetting the alarm information is divided into 2 different FINS –commands, because the maximum number of written memory addresses is 996 on the Ethernet network.

First command resets memory addresses E3_99 to E3_599, then command is in format:

- SID = 05_{hex}
- Command code = 0103_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = 0063_{hex} = E3_99
- Number of items = $0001F5_{hex}$ = 501 words
- Data = 0000_{hex} = 0_{dec}

Second command resets memory addresses E3_600 to E3_1099, then command is in format:

- SID = 06_{hex}
- Command code = 0103_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = 0258_{hex} = E3_600
- Number of items = $0001F4_{hex}$ = 500 words
- Data = 0000_{hex} = 0_{dec}

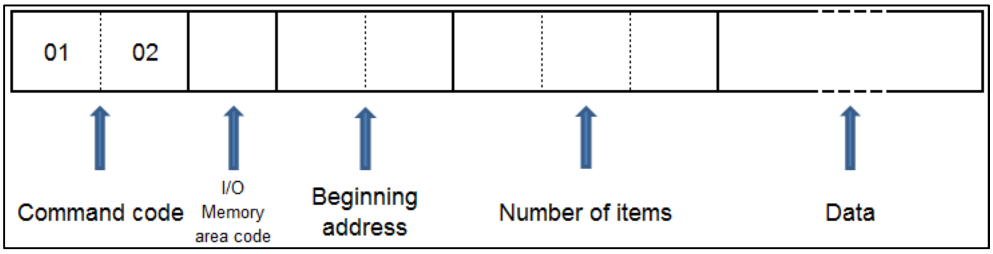
6.2.3 Memory area write

Memory area write –command writes specified data to the specified memory addresses. This is used to allow maintenance page access for the GUI viewer.

Allowing maintenance page access is done by writing value $FFFF_{hex}$ to the memory address E3_90. This is executed with following command sent from the software application (picture 30):

- SID = 07_{hex}

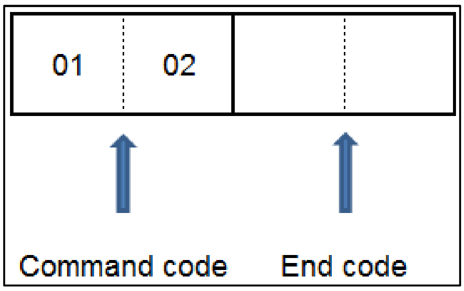
- Command code = 0102_{hex}
- I/O memory area code = $A3_{hex}$ = E3 (EM area, bank3)
- Beginning address = $005A_{hex}$ = E3_90
- Number of items = 000001_{hex} = 1 word
- Data = $FFFF_{hex}$



PICTURE 30. Memory area write -command

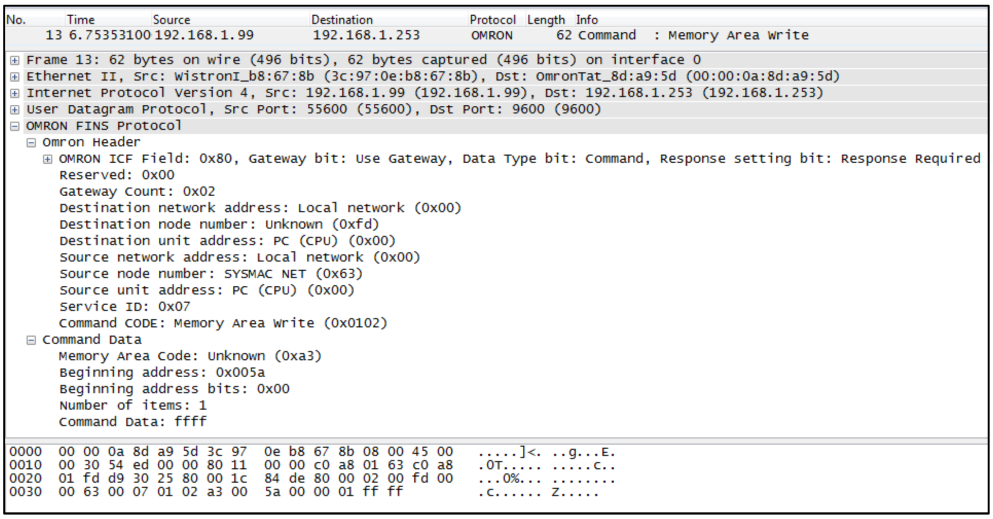
Response from the control system is received in following format (picture 31):

- Command code = 0102_{hex}
- End code = 0000hex = normal completion

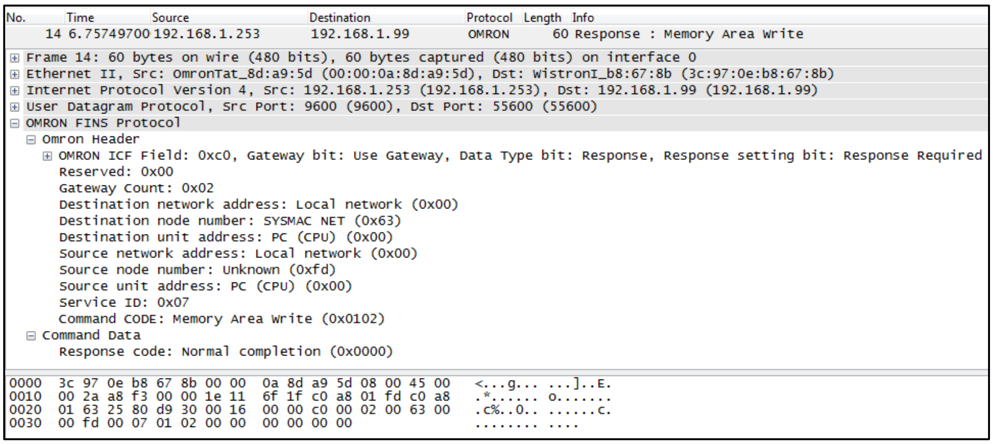


PICTURE 31. Memory area write -response

Pictures 32 and 33 shows the memory area write -command in Wireshark Ethernet analyzer.



PICTURE 32. Memory area write -command on Wireshark



PICTURE 33. Memory area write -response on Wireshark

7 CONCLUSIONS

Overall the user survey was a success. It shows that the machine parameters used in the software application are correct and valued by the users. The amount of ideas, given as open comments, shows that the topic was important to the respondents and they were highly motivated to give their own ideas on this matter. Open comments are in many way more valuable than numerical results. Open comments given in this survey should be used for future development after the application base line is taken into use.

In the market comparison three different applications were introduced. Truconnect® and Green zoneTM are more concentrated on productivity data where as WinEEM4 is mainly focused on operation data and fault diagnostics. According to the user survey combining productivity and operation data with specific fault diagnostic information will give the best outcome for the user.

This new maintenance assisting software application answers to all development key points which were set when this work started. Machine parameters were updated to serve modern-day needs and machine types and their necessity was confirmed in the user survey.

Alarm diagnostics were improved adding specific alarm description, conditions for alarm activation and troubleshooting information individually for each alarm which will fasten the fault diagnostics and therefore decrease machine down-time.

Communication will be changed to Ethernet which prepares the machine for all future data transfer possibilities. Simplest and most cost effective way for adding wireless communication between machine and a maintenance laptop or mobile device can be done by adding WLAN access point on the machine. The range of the access point signal can be limited so that it doesn’t cause long range disturbances in other WLAN networks. Also adding GPRS -modem to the machine will open similar possibilities for wireless communication as WLAN, but some additional costs will come from the transferred data, depending on the used tele operator. Also adding an additional on-board data logger to the machine should be considered. This would help with the integration to an external servers and cloud services.

Drafts from the new user interface were presented in this work. These drafts will work as guideline but the final layout will be determined later.

In conclusion for future development adding a real-time monitoring for certain machine parameters should be considered. Parameters like tire pressure, oil level and fuel level would help the maintenance person quickly to check the daily inspections.

LIST OF REFERENCES

Konecranes, http://www.konecranes.com/service/truconnect

Bromma, http://www.brommagreenzone.com

Omron 2008, Web interface manual V100-E1-01

Omron 2010, Reference manual for communications commands W342-E1-15

APPENDICES

Appendix 1. User survey form

1. Name:
____

2. Company:
Cargotec / Other
3. Department:
Engineering / Commissioning / Product Support / Service / Management / Other: __________
4. How often do you use the program or report ?
Daily / Weekly / Monthly / Yearly / Never

Statistics
5. Engine Hours [h]:
1 2 3 4 5
not relevant … … … definately needed
6. Total hours (in hybrid machines different than engine hours) [h]:
1 2 3 4 5
not relevant … … … definately needed
7. Driving hours [h]:
1 2 3 4 5
not relevant … … … definately needed
8. Driving hours according to driving direction [h]:
1 2 3 4 5
not relevant … … … definately needed
9. Standing hours [h]:
1 2 3 4 5
not relevant … … … definately needed
10. Hoisting/Lowering hours [h]:
1 2 3 4 5
not relevant … … … definately needed
11. Driving distance [km]:
1 2 3 4 5
not relevant … … … definately needed
12. Driving distance according to driving direction [km]:
1 2 3 4 5
not relevant … … … definately needed
13. Number of containers [pcs]:
1 2 3 4 5
not relevant … … … definately needed
14. Number of containers according to pick mode [Single, Twin..]:
1 2 3 4 5
not relevant … … … definately needed
15. Number of containers according to container lenght [20ft, 40ft...]:
1 2 3 4 5
not relevant … … … definately needed
16. Fuel consumption [l/h]:
1 2 3 4 5
not relevant … … … definately needed
17. Energy consumption [kWh]:
1 2 3 4 5
not relevant … … … definately needed
18. Number of stability alarms [pcs]:
1 2 3 4 5
not relevant … … … definately needed
19. Timestamp for the stability alarms:
1 2 3 4 5
not relevant … … … definately needed

20. Timestamp for use of the hoist bypass:

1	2	3	4	5
not relevant	…	…	…	definately needed

21. Is it required to reset statistic information?

Yes / No

22. Any other requirements for statistics?

____

Alarms

23. Alarm timestamp:

1	2	3	4	5
not relevant	…	…	…	definately needed

24. Alarm with additional error code:

1	2	3	4	5
not relevant	…	…	…	definately needed

25. Alarm priority:

1	2	3	4	5
not relevant	…	…	…	definately needed

26. Alarm list type:

Chronological / Frequency / Other:__________

27. Is it required to reset alarm information?

Yes / No

28. Any other requirements for alarms?

____

GUI viewer

29. Remote monitoring of the driver display:

1	2	3	4	5
not relevant	…	…	…	definately needed

30. Remote operation of the driver display:

1	2	3	4	5
not relevant	…	…	…	definately needed

31. Remote access to the maintenance menu's in the driver display:

1	2	3	4	5
not relevant	…	…	…	definately needed

32. Any other requirements for HMI viewer?

____

Others

33. Printable report (e.g. *.pdf or *.txt):

1	2	3	4	5
not relevant	…	…	…	definately needed

34. Other comments:

____

Appendix 2. Control system memory areas

Variable	Data type	Address	Comment
MachineType	UINT	E3_1	Machine type
MachineNumber	UINT	E3_2	Machine number
EngineHoursTotal	UDINT	E3_3	Engine hours (total)
HybridHoursTotal	UDINT	E3_5	Hybrid hours (total)
TravelingHoursTotalFwd	UDINT	E3_7	Traveling hours forward (total)
TravelingHoursTotalBwd	UDINT	E3_9	Traveling hours backward (total)
HoistLowerTotalHours	UDINT	E3_11	Hoisting and Lowering hours (total)
StandingTotalHours	UDINT	E3_13	Standing hours (total)
Traveling DistanceTotalFwd	UDINT	E3_15	Traveling distance forward (total)
Traveling DistanceTotalBwd	UDINT	E3_17	Traveling distance backward (total)
20ftCountTotal	UDINT	E3_19	Number of 20ft containers (total)
40ftCountTotal	UDINT	E3_21	Number of 40ft containers (total)
TwinCountTotal	UDINT	E3_23	Number of twin picks (total)
NumberStabilityAlarms	UDINT	E3_25	Total number of stability alarms
StabilityAlarm1_YM	WORD	E3_27	Stability alarm timestamp - Y & M
StabilityAlarm1_DH	WORD	E3_28	Stability alarm timestamp - D & H
StabilityAlarm1_MS	WORD	E3_29	Stability alarm timestamp - M & S
StabilityAlarm2_YM	WORD	E3_30	Stability alarm timestamp - Y & M
StabilityAlarm2_DH	WORD	E3_31	Stability alarm timestamp - D & H
StabilityAlarm2_MS	WORD	E3_32	Stability alarm timestamp - M & S
StabilityAlarm3_YM	WORD	E3_33	Stability alarm timestamp - Y & M
StabilityAlarm3_DH	WORD	E3_34	Stability alarm timestamp - D & H
StabilityAlarm3_MS	WORD	E3_35	Stability alarm timestamp - M & S
StabilityAlarm4_YM	WORD	E3_36	Stability alarm timestamp - Y & M
StabilityAlarm4_DH	WORD	E3_37	Stability alarm timestamp - D & H
StabilityAlarm4_MS	WORD	E3_38	Stability alarm timestamp - M & S
StabilityAlarm5_YM	WORD	E3_39	Stability alarm timestamp - Y & M
StabilityAlarm5_DH	WORD	E3_40	Stability alarm timestamp - D & H
StabilityAlarm5_MS	WORD	E3_41	Stability alarm timestamp - M & S
StabilityAlarm6_YM	WORD	E3_42	Stability alarm timestamp - Y & M
StabilityAlarm6_DH	WORD	E3_43	Stability alarm timestamp - D & H
StabilityAlarm6_MS	WORD	E3_44	Stability alarm timestamp - M & S
StabilityAlarm7_YM	WORD	E3_45	Stability alarm timestamp - Y & M
StabilityAlarm7_DH	WORD	E3_46	Stability alarm timestamp - D & H
StabilityAlarm7_MS	WORD	E3_47	Stability alarm timestamp - M & S
StabilityAlarm8_YM	WORD	E3_48	Stability alarm timestamp - Y & M
StabilityAlarm8_DH	WORD	E3_49	Stability alarm timestamp - D & H
StabilityAlarm8_MS	WORD	E3_50	Stability alarm timestamp - M & S
StabilityAlarm9_YM	WORD	E3_51	Stability alarm timestamp - Y & M
StabilityAlarm9_DH	WORD	E3_52	Stability alarm timestamp - D & H
StabilityAlarm9_MS	WORD	E3_53	Stability alarm timestamp - M & S
StabilityAlarm10_YM	WORD	E3_54	Stability alarm timestamp - Y & M
StabilityAlarm10_DH	WORD	E3_55	Stability alarm timestamp - D & H
StabilityAlarm10_MS	WORD	E3_56	Stability alarm timestamp - M & S

FuelConsumption	UINT	E3_70	Fuel consumption l/h (trip)
EnergyConsumption	UINT	E3_71	Energy consumption kW (trip)
EngineHoursTrip	UINT	E3_72	Engine hours (trip)
HybridHoursTrip	UINT	E3_73	Hybrid hours (trip)
TravelingHoursTripFwd	UINT	E3_74	Traveling hours forward (trip)
TravelingHoursTripBwd	UINT	E3_75	Traveling hours backward (trip)
HoistLowerTripHours	UINT	E3_76	Hoisting and Lowering hours (trip)
StandingTripHours	UINT	E3_77	Standing hours (trip)
Traveling DistanceTripFwd	UINT	E3_78	Traveling distance forward (trip)
Traveling DistanceTripBwd	UINT	E3_79	Traveling distance backward (trip)
20ftCountTrip	UINT	E3_80	Number of 20ft containers (trip)
40ftCountTrip	UINT	E3_81	Number of 40ft containers (trip)
TwinCountTrip	UINT	E3_82	Number of twin picks (trip)
MaintenceViewAccess	UINT	E3_90	Allows maintenance page access
TotalNumberOfAlarms	UINT	E3_99	Total number of alarms
AlarmID	UINT[100]	E3_100	Alarm id from row 0 to 99
AdditionalCode1	UINT[100]	E3_200	Additional error code1 from row 0 to 99
AdditionalCode2	UINT[100]	E3_300	Additional error code2 from row 0 to 99
NumberOfAlarms	UINT[100]	E3_400	Number of alarms from row 0 to 99
LastYearMonth	WORD[100]	E3_500	Last year and month of alarms from row 0 to 99
LastDayHour	WORD[100]	E3_600	Last day and hour of alarms from row 0 to 99
LastMinuteSecond	WORD[100]	E3_700	Last minute and second of alarms from row 0 to 99
FirstYearMonth	WORD[100]	E3_800	First year and month of alarms from row 0 to 99
FirstDayHour	WORD[100]	E3_900	First day and hour of alarms from row 0 to 99
FirstMinuteSecond	WORD[100]	E3_1000	First minute and second of alarms from row 0 to 99

Appendix 3. FINS command codes

Type	Command code	Name	Function
I/O memory area access	01 01	Memory area read	Reads the contents of consecutive I/O memory area words.
	01 02	Memory area write	Writes the contents of consecutive I/O memory area words.
	01 03	Memory area fill	Writes the same data to the specified range of I/O memory area words.
	01 04	Multiple memory area read	Reads the contents of specified noncon-secutive I/O memory area words.
	01 05	Memory area transfer	Copies the contents of consecutive I/O memory area words to another I/O memory area.
Parameter area access	02 01	Parameter area read	Reads the contents of consecutive parameter area words.
	02 02	Parameter area write	Writes the contents of consecutive parameter area words.
	02 03	Parameter area fill	Writes the same data to the specified range of parameter area words.
Program area access	03 06	Program area read	Reads the user memory area
	03 07	Program area write	Writes the user memory area
	03 08	Program area clear	Clears the user memory area
Operating mode changes	04 01	Run	Changes the CPU Unit’s operating mode to RUN or MONITOR.
	04 02	Stop	Changes the CPU Unit’s operating mode to PROGRAM.
Machine configu-ration reading	05 01	CPU unit data read	Reads CPU Unit data.
	05 02	Connection data read	Reads the model numbers of the device corresponding to addresses.
Status reading	06 01	CPU unit status read	Reads the status of the CPU Unit.
	06 20	Cycle time read	Reads the maximum, minimum, and average cycle time.
Time data access	07 01	Clock read	Reads the present year, month, date, minute, second, and day of the week.
	07 02	Clock write	Changes the present year, month, date, minute, second, or day of the week.
Message display	09 20	Message read/clear	Reads and clears messages, and reads FAL/FALS messages.
Access rights	0C 01	Access right acquire	Acquires the access right as long as no other device holds it.
	0C 02	Access right forced ac-quire	Acquires the access right even if another device already holds it.
	0C 03	Access right release	Releases the access right that has been acquired.
Error log	21 01	Error clear	Clears errors or error messages.
	21 02	Error log read	Reads the error log.

	21 03	Error log clear	Clears the error log pointer.
FINS write access log	21 40	FINS write access log read	The CPU Unit automatically keeps a log of any access for FINS write commands. This command reads this log.
	21 41	FINS write access log clear	Clears the FINS write access log.
File memory	22 01	File name read	Reads file memory data.
	22 02	Single file read	Reads a specified length of file data from a specified position within a single file.
	22 03	Single file write	Writes a specified length of file data from a specified position within a single file.
	22 04	File memory format	Formats (initializes) the file memory
	22 05	File delete	Deletes specified files stored in the file memory.
	22 07	File copy	Copies files from one file memory to another file memory in the same system.
	22 08	File name change	Changes a file name.
	22 0A	Memory area-file transfer	Transfers or compares data between the I/O memory area and the file memory.
	22 0B	Parameter area-file transfer	Transfers or compares data between the parameter area and the file memory.
	22 0C	Program area-file transfer	Transfers or compares data between the UM (User Memory) area and the file memory.
	22 15	Directory create/delete	Creates or deletes a directory.
	22 20	Memory cassette transfer	Transfers and verifies data between a Memory Cassette and the CPU Unit.
Debugging	23 01	Forced set/reset	Force-sets or force-resets bits, or releases force-set status.
	23 02	Forced set/reset cancel	Cancels all bits that have been force-set or force-reset.
Serial Gateway functions	28 03	Convert to CompoWay/F command	Encapsulates a CompoWay/F command in a FINS command and sends it to a serial port.
	28 04	Convert to Modbus-RTU command	Encapsulates a Modbus-RTU command in a FINS command and sends it to a serial port.
	28 05	Convert to Modbus-ASCII command	Encapsulates a Modbus-ASCII command in a FINS command and sends it to a serial port.
	Any Any	Convert to host link fins command	Sends any FINS command to a PLC connected to the serial port of a Serial Communications Board or Unit.