

LMS COMMUNICATION PROTOCOL

COMMUNICATION WITH EXTERNAL PAYMENT SYSTEM

History

Revision	Date	Name	Description
	6.9.2017	M.Kotasek	Released
A	23.10.2017	M.Kotasek	Disabled telegrams X2, XB Added new payment system Com3
B	19.10.2017	J.Havrlant	Added start confirmation on machine for telegrams X2, XB
C	3.2.2019	M.Kotasek	Added more detailed description

1.1 Protocol description

The protocol used is an adapted version of the international standard ISO 1745. This protocol has been especially designed to enable fast, error free data communication in a half duplex, multi-point environment such as RS-485 two wire network.

Frame format

9600bd, 8data, 1stop, no parity

All transmission control functions will be performed by the use of the following transmission control characters.

Master: <SOH><header><STX><information><ETX><CRC><CR>

Slave: <ACK><ENQ><header><STX><information><ETX><CRC><CR>

Or <NACK><CR>

- **SOH** Start of heading **(01 Hex)**

A transmission control character used as the first character of a frame.

- **Header field**

Holding the destination address coded into 2 hex. Characters

- **STX** Start of text **(02 Hex)**

A transmission control character which precedes an information field and which is used to terminate the header field.

- **Information field**

Holding the data to be sent.

- **ETX** End of text **(03 Hex)**

A transmission control character that terminates an information field.

- **CRC** Frame check characters

- **CR** **(0D Hex)**

Response control characters

- **ACK** Acknowledge **(06 Hex)**

A transmission control character sent by the destination to acknowledge the received command.

- **NACK** Negative acknowledge **(15 Hex)**

A transmission control character sent by the destination to indicate an error in the received command.

- **ENQ** Enquiry **(05 Hex)**

Header field

The header field contains the address of the destination. Each address consists of two characters in coded into hex characters in the range <00> to <FF>. Note the address code <00> is used for all slaves, all slaves will respond to that address at the same time so, use this address if only one slave is connected.

Header field = <upper nibble hex character><lower nibble hex character>

Information field

This field is used to send the application data and can be any character of the 7-bit ISO/CCITT standard. Remark: this field cannot contain any protocol control character.

CRC characters

The frame check sequence character is 8 bit byte coded as two hex characters, generated by the addition of all characters in the message starting with the <SOH> or <ENQ> character until the <ETX> character.

<SOH> & <ETX> included.

<CRC> & <ACK> not included.

Data link procedure

The protocol is a 100 % Master-Slave protocol, meaning that only the Master can send data to a slave at any time. The slave can only send data to the master on request of the Master station.

If the Master sends a command on the communication line, all slave stations will synchronise they're receiving buffer on the SOH control character and will receive the message until the ETX and the CRC character is received.

The slave, whose address corresponds with the destination address in the header field, will examine the received message and check it for following points.

- Protocol correctness
- CRC characters
- Control of the information field for a correct slave command.

Slave response

There are to different types of response from a slave module, depending on the command sended to the module.

1/ Command was not a request for data

The slave will respond with a <ACK> character to the master if :

- The destination address in the header corresponds to the slave address
- There are no errors against the protocol format
- There were no communication errors (control of the CRC character).
- There were no command syntax errors.
- There was no problem in executing the command (ex. internal problem).
- The command has been sent when it was allowed (see command syntax).

The slave will respond with a <NACK> character to the master if:

- There was a correct <SOH> & <ETX> detected and the destination address in the header corresponds to the slaves address.

In all other cases the slave will not respond.

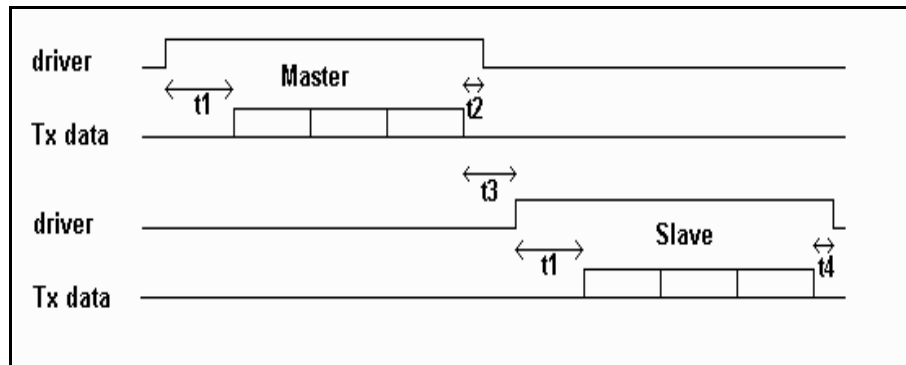
2/ The transmitted command was a request for data from the slave.

The slave will respond with the data requested after the <ACK> character. The message will be conform to the protocol. The address of the slave, will be used as the destination address. So if address "0" is used the slave will use his true address number.

Communication Timing

The Timer has been specially designed to respond quickly to the commands sent by the master, so special care should be taken in writing the communication driver.

The time to take the line and drop the line depends on the RS-485 converter on the master side. Please read the instructions of your converter carefully.



Communication Timing.

timing :

- t_1 : Time between activating the line and the first character depends on the transmission speed; 1.2 times a character time.
- t_2 : Time to drop the line after the last character : max. xxxx us
- t_3 : Response time from the slave after the last character from the master was received.
- t_4 : Time to drop the line by the slave

1.2 Commands and syntax for washers

General to all commands

The commands are structured in a way that they are readable. This makes debugging easy.

A command consists of two characters; the first indicating the command group; the second character being an index digit in the same group.

Values are sent as 8 bit, 16 bit or 32 bit variable in hexadecimal format and sent always the length of 2,4 or 8 characters. A hexadecimal number has the ':' character as a field separator before the first hex character. The value will be sent using the following format ex for 16 bit integer <:HHLL> HH indicating the upper byte LL the lower byte of a 16 bits value.

- In the examples we use slave address "1".
- The non readable control characters are typed as<..>.
- The two characters that represents the CRC are typed as <CRC1><CRC2>
- Data fields are typed as <Data1>

S1 - Status request

- Command <SOH>01<STX>S1<ETX><CRC1><CRC2><CR>
- Response

<ACK><ENQ>01<STX>s1:<Data1>:<Data2>:<Data3>:<Data4>:<Data5>:<Data6>:<Data7>:<Data8>:<Data9>:<Data10>:<Data11><ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	Machine Status1
Data2	Byte (8 bit)	Error nr on display (higher byte)
Data3	Byte (8 bit)	Selected Program nr More information about default wash cycles can be found in Programming manual
Data4	Byte (8 bit)	Block nr in execution
Data5	Unsigned Long (32 bit)	Estimated rest time in seconds
Data6	Byte (8 bit)	Time counting in hold mode
Data7	Byte (8 bit)	Error nr on display (lower byte) More information about errors can be found in Programming manual
Data8	Unsigned Long (32 bit)	Error counter since machine installation
Data9	Integer (16 bit)	Software version and controller type
Data10	Byte (8 bit)	Machine type selected: RX65N,RX80N,RX105N,RX135N,RX180N,RX240N,RX280N,RX350N,RX520N,RX65M,RX80M,RX105M,RX135M,RX180M,RX240M,RX280M,RX350M,RX180H,RX240H,RX280H,FX65N,FX80N,FX105N,FX135N,FX180N,FX240N,FX280N,FX65M,FX80M,FX105M,FX135M,FX180M,FX240M,FX280M,FX65H,FX80H,FX105H,FX135H,FX180H,FX240H,FX280H
Data11	Byte (8 bit)	Brand Type: PRIMUS, LAVAMAC, UNIMAC, SQ, IPSO, HUEBSCH, PRIMUS_US, UNIMAC_US, SQ_US, IPSO_US, HUEBSCH_US, NOBRAND

- Data1: Machine Status 1

Code	Status
0	Machine is busy with power-up
1	Machine Free
2	Machine busy with wash
3	Machine in pause mode
4	Machine in stop / continue mode
5	Machine Done and in error mode
6	Machine Done (waiting for door open)
7	Machine in external communication locking (for program data downloading)
8	Machine is in Menu mode
9	Machine is busy with restarting from power-down
10	Machine Free with error
11	machine in delayed start (not used for payment systems)
12	ready for payment, machine is free (idle) and request for payment is shown on display
13	payment command (or deposit) received by machines

S3 – Advance status request

- Command <SOH>01<STX>S3<ETX><CRC1><CRC2><CR>
- Response

<ACK><ENQ>01<STX>s3:<Data1>:<Data2>:<Data3>:<Data4>:<Data5>:<Data6>:<Data7>:<Data8>:
:<Data9>:<Data10>:<Data11><ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	Selected Program nr
Data2	Byte (8 bit)	Running wash step
Data3	Unsigned Long (32 bit)	Estimated rest time in seconds
Data4	Byte (8 bit)	Weight of linen (for machines with weighing system)
Data5	Byte (8 bit)	Requested water temp.
Data6	Byte (8 bit)	Real water temp.
Data7	Byte (8 bit)	Binary information about active soap outputs
Data8	Byte (8 bit)	Reserve (0)

C3 - External start release

- Command “<SOH>01<STX>C3:Data1<ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	Program number

Program number

- 0 : The program number selected on the machine is used
- 1-15: Sets program 1 to 15
- Response

<ACK><ENQ>01<STX>c3:<Data1> <ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	Return code

Return Code

- 210 : No errors
- 211 : Illegal Program selected
- 212 : Wrong external payment mode (check machine configuration)
- 213 : Machine is not "Free" (Check machine status first)

X1 – Status check

- Command <SOH>01<STX>X1 <ETX><CRC1><CRC2><CR>
- Response

<ACK><ENQ>01<STX>x1:<Data1>:<Data2>:<Data3>:<Data4>:<Data5>:<Data6>:<Data7><ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	MachineStatus1 (like S1 telegram)
Data2	Byte (8 bit)	Decimal point position
Data3	Byte (8 bit)	Selected Program nr
Data4	Long (32 bit)	Estimated rest time in seconds
Data5	Long (32 bit)	Basic Program Price (without Modifiers)
Data6	Long (32 bit)	Modifiers Price
Data7	Byte (8 bit)	Error nr on display

X2 – Program start release command

- Command <SOH>01<STX>X2 <ETX><CRC1><CRC2><CR>
Payment system sends command X2 when money are paid.
- Response – the same like for X1 telegram

XB – Start machine

- Command <SOH>01<STX>XB:<Data1> <ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Int (16 bit)	Program to start - 1

- Response
<ACK><ENQ>01<STX>xb:<Data1><ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	Door state: 0.. Opened 1 .. Closed

XC –Advance command

- Command <SOH>01<STX>XC <ETX><CRC1><CRC2><CR>
Command XC enable advance function on machine and advances to next step.
Multiple XC advance commands will terminate running cycle (jump to the end of cycle).

- Response
 <ACK><ENQ>01<STX>xc<Data1><ETX><CRC1><CRC2><CR>

C4 – Deposit from external payment system – new suggestion

- Command “<SOH>01<STX>C4:<Data1>:<Data2><ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Int (16 bit)	Deposit, integral part
Data2	Byte (8 bit)	Deposit, decimal fraction

Example:

- payment 150\$: C4:0096:00
- payment 15.0\$: C4:000F:00
- payment 1.50\$: C4:0001:42
- payment 0.15\$: C4:0000:0F

- Response

<ACK><ENQ>01<STX>c4:<Data1>:<Data2><ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Int (16 bit)	Registered deposit added to internal machine escrow, integral part
Data2	Byte (8 bit)	Registered deposit added to internal machine escrow, decimal fraction

C5 – Payment audit data

- Command “<SOH>01<STX>C5:Data1<ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	Byte (8 bit)	Request to clear resettable counters

Data 1 - Request to clear resettable counters:

- 0.. resettable counters are not reset
- 1.. resettable counters are reset to zero

- Response

<ACK><ENQ>01<STX>c5:<Data1>:<Data2> <ETX><CRC1><CRC2><CR>

Field	Type	Name
Data1	long (32 bit)	Coin 1 no resettable counter
Data2	long (32 bit)	Coin 2 no resettable counter
Data3	Int (16 bit)	Coin 1 resettable counter
Data4	Int (16 bit)	Coin 2 resettable counter
Data5	long (32 bit)	Network vend, integral part
Data6	Byte (8 bit)	Network vend, decimal fraction
Data7	long (32 bit)	Resettable network vend, integral part
Data8	Byte (8 bit)	Resettable network vend, decimal fraction
Data9	long (32 bit)	Total vend, integral part
Data10	Byte (8 bit)	Total vend, decimal fraction
Data11	long (32 bit)	Total resettable vend, integral part
Data12	Byte (8 bit)	Total resettable vend, decimal fraction

1.3 Payment sequences for washers

1.3.1 Start on machine – one price for all programs

- Payment mode in Price menu must be set to Comm1
- Customer loads machine (and selects wash program) and goes to payment system
- Payment system checks machine status with telegram S1
- If machine is Free then payment system can accept new payment
- When correct price is paid then payment system sends command C3 (with parameter 00) to release machine
- Customer has to (select wash program on machine) and press start button to activate wash process

1.3.2 Program selected on machine, start from payment system – individual price for each program

- Payment mode in Price menu must be set to Comm2
- Customer loads machine, selects wash program and goes to payment system
- Payment system checks machine status with telegram X1 and read selected wash program
- If machine is Free then payment system can accept new payment corresponding to selected wash program
- When correct price is paid then payment system sends command X2 to start selected program on machine
- Confirmation screen of remote start (Picture no.1) is appear and selected program is started after confirmation with “Start” button



Picture no.1

1.3.3 Program selected on payment system, start machine from payment system – individual price for each program

- Payment mode in Price menu must be set to Comm2
- Customer loads machine and goes to payment system
- Customer selects requested wash program on payment system
- Payment system checks machine status with telegram S1
- Customer pays correct price
- Payment system sends command XB (with selected program)
- Payment system checks answer to telegram XB to see door state, if door is not closed then payment systems shows warning
- After telegram XB, the confirmation screen of remote start is appear (Picture no.1) and selected program is started after confirmation with “Start” button

1.3.4 Program selected and started from machine, money paid on payment system– individual price for each program

- Payment mode in Price menu must be set to Comm1
- Program prices are set to 100 (telegram E5 for MCB/MCG, telegram G6 for MCX)
- Customer loads machine, selects wash program and goes to payment system
- Customer pays correct price
- Payment system checks program prices and paid money and for cheaper programs set price to 0
- Customer goes to machine and can start wash programs for which price was set to 0 (wash programs with price 100 will show request “PAY” on display)
- After wash cycle start, payment system sets price for all wash programs to 100

1.3.5A Program selected and started from machine, money paid on payment system– individual price for each program

- Payment mode in Price menu must be set to Comm3
- Customer loads machine, selects wash program (and modifiers) and goes to payment system
- Payment system checks that machine status is FREE with telegram X1
- Payment system checks selected program number, basic program price and modifiers price, decimal point position (from telegram X1)
- When total payment is satisfied then payment system sends telegram C4 with received deposit
- Payment system checks with telegram X1 that machine state was changed to PAID
- If machine status stays FREE then payment systems repeats telegram C4 (previous telegram was lost)
- After change of machine status in state PAID, payment system regularly checks machine status with telegram X1. If status changed to FREE and new price is received (program changed or modifier selected) then payment system requests from customer additional money.
- When additional money are paid then payment system sends telegram C4 with received deposit and again check status with telegram X1. State must be changed to PAID, otherwise C4 telegram is repeated
- After payment customer starts selected program on machine by pressing Start button
- Audit data can be checked with telegram C5

1.3.5B Program selected and started from machine, money paid on payment system– individual price for each program

- Payment mode in Price menu must be set to Comm3
- Customer loads machine, selects wash program (and modifiers) and goes to payment system
- Payment system checks that machine status is FREE with telegram X1
- Payment system checks selected program number, basic program price and modifiers price and decimal point position (from telegram X1)
- When total payment is satisfied then payment system sends telegram C5 and remember actual network vend value
- Payment system sends C4 with received deposit
- Payment system checks with telegram C5 that machine network vend value was correctly increased
- If machine network vend value was not increased then payment systems repeats telegram C4 (previous telegram was lost)
- Payment system regularly checks machine status with telegram X1. If status changed to FREE and new price is received (program changed or modifier selected) then payment system requests from customer additional money.
- When additional money are paid then payment system sends telegram C5 and remember actual network vend value
- Then payment system sends C4 with received deposit and again check network vend value with telegram C5
- After payment customer customer starts selected program on machine by pressing Start button