# COMMUNICATION PROTOCOL FOR
# KELLER DIGITAL MANOMETERS AND TRANSMITTERS

## Customer release

# TABLE OF CONTENTS

# 1Introduction

This document describes the means implemented to allow a supervision master machine to get information such as temperature and pressure, coming from KELLER digital devices. In order to do this, the devices are connected to a bi-directional serial network, using a copper four wire media to spread data and supply the devices. The data transmission between the bus master and the devices is achieved by the help of a communication protocol which is described in this document in a way, that a complete software application can be written.

# 2Physical link

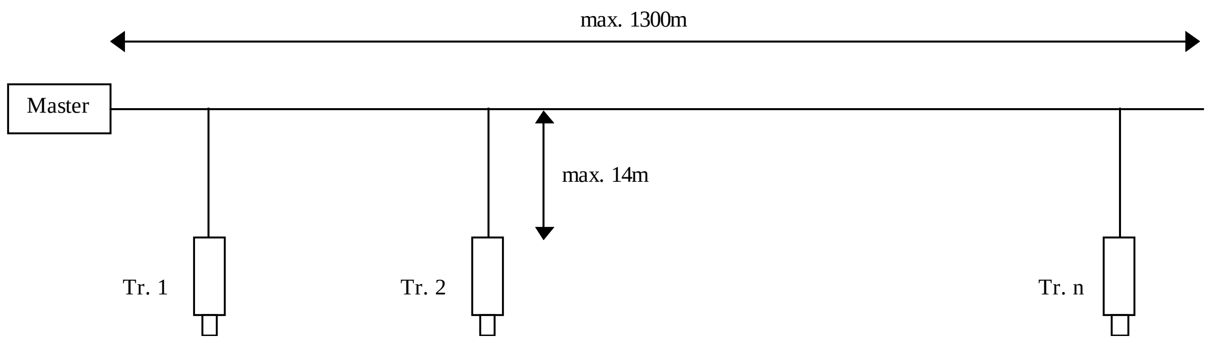
The physical link used in this application is a multi-drop RS485 in 2 wires topology, because on one hand of its good immunity against electrical disturbances, even over long distances and on the other hand because of its electrical characteristics, especially in current, allowing several receivers /transmitters to be connected in parallel (i.e. 31 devices + 1 bus master). The physical media used to transmit data consists of a shielded twisted pair. The power is supplied with a second twisted pair which should be separately shielded from the previous one.

Each device connected to the bus features its electrical interface, designed in a way to allow each of them to transmit and receive information on the same pair at one's turn (half-duplex).

In idle state (in other words, when no message is transmitted over the bus) every device is set to receiving mode.

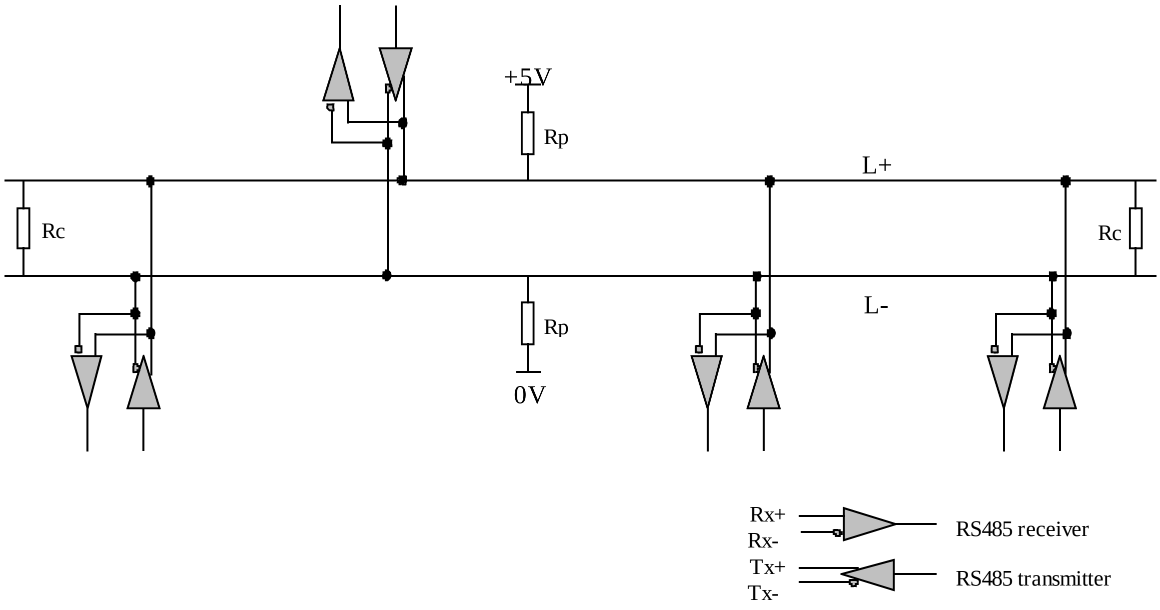
## 2.1Characteristics of the physical link

Such an electrical link allows a network length of about 1300 meters, and derivations of about 14 meters to connect the devices to the bus (see figure below). In electrically harsh environments, the length of such derivations should be as reduced as possible. The cable used must comply with the EIA RS 485 norm. Its characteristic impedance should therefore be 120 Ω.




The electrical connection to earth of the different shieldings must be achieved with good engineering practice. It's especially important to design an equipotential earth circuit, also for high frequencies if possible, and to connect at least each end of the main cable to this earth.

In order to avoid reflections, the network impedance is adapted by connecting a resistive load Rc of the same value as the characteristic impedance of the cable (120 Ω) at each end of it. It can also be necessary to polarize the transmission lines. This is realized by connecting the L+ and L- lines respectively to +5V and 0V with a bias resistor of 470 Ω. This biasing must be made only once in the network, for example near the bus master connection. The values of the bias resistors are mentioned as an indication, and are subject to modification according to possible change in technical requirements.




Bus topology (synoptic)

# 3Logic link

The logic link defines the rules used in exchanging information over such a bus. It is the part of the protocol managed by computer programs. This communication type is a master/slave half duplex one. This means that only one device is allowed to transmit a message over the bus at the same time. All the other devices at this time are set to reception mode. The bus master is the supervision computer, which can consist of a PC including a RS485 interface or any other computing device equipped with this interface. The slave devices are the digital KELLER devices. Every data exchange performed over the bus is under bus master control. The exchanged data is shifted out in messages transmitted as blocks. Each message has a particular field allowing the receiver slave device to be recognized : The address.

This structure allows two types of data transmission :

a) Broadcasting : This kind of communication allows the bus master to send information to every slave device at the same time, by sending only one message. The address used for this operation concerns every slave device : this address is „00h". In this particular type of communication, there is of course no response or acknowledgement of the devices. So the master has only one way to check, if there was 100% the correct reception of the data by the devices. This possibility is given by special reading features (apart from reading diagnostic counters, this feature is not implemented in our system). This is the reason why this kind of transmission is not very often used or only for certain special operations.

b) Data exchange : This kind of communication allows the bus master to communicate with one and only one device at the same time. In normal operating conditions, an exchange consist of the transmission of two messages : A request sent by the master, immediately followed by the corresponding response coming from the concerning device. Only the master computer is allowed to initiate any exchange of data on the bus. In this mode, which is the normal operating one, the bus master sends a request to a particular device, e.g. a pressure reading request. This message is received by every device, but only the addressed device will answer to the request. This answer must be sent and it has to begin before a predefined time-out. Otherwise the device could be declared as in failure (by the master after several unsuccessful attempts). In normal operating mode, the master polls every device in turn. This way of proceeding allows to know the status of each device at each data exchange.

## 3.1Message transmission format

The information is sent serially through the bus. Each message is composed with bytes transmitted in an asynchronous way, according to the standard format, and in the following order :

- 1 start bit
- 8 data bits with the least significant one sent in first
- no parity bit
- 1 stop bit
- 9600 baud

The byte is the basic datum transmittable over the bus.

## 3.2Structure of the messages

### 3.2.1Messages coming from the master

Each message sent by the master is composed as follows :

| | | | | | |
|---|---|---|---|---|---|
| Device address | 0 | Function code | Parameters (optional) | CRC16 H | CRC16 L |

- DEVICE ADDRESS :
This is the address code mentioned above, used by the addressed device to recognize itself, thus leading it to perform the requested function, and to acknowledge the reception of the master message by sending in turn the correct response before the predefined time-out.

This address is in fact the number of the slave device to which the message is intended. This number is choosen as follows :

Address „00h“ is reserved for the broadcasting mode.
Address 1 through 249 (including these values) can be used for bus operation.
Address 250 is reserved for non-bus operation. Each device always reacts on this address.
Address 251 through 255 (including these values) are reserved for further developments.

- FUNCTION CODE :
This code indicates the function to be executed by the receiving device. It can be a request for data or a command. The function is coded by the least 7 bits. Therefore the function codes can be from 0 to 127. The implemented functions are described in the bus function documentation of the different devices.

- PARAMETERS :
Every function can have some associated parameters. The number and the meaning of the bytes required for these parameters of course depends on the function itself. It must be noticed, that some of the functions don’t require any parameter. In this case, the useful part of the message is constituted only by the function code.

- CRC16 :
This code allows the receiving device to detect any error present in the message. Its length is 16 bits and it is sent in two bytes, beginning with the most significant one. The CRC16 is a control code defined and described in the CCITT recommendation documents, used as a reference in telecom systems. It is computed using every previous byte of the message. After receiving a message, the device compares the CRC16 included in it to the one it has computed itself, the equality of them makes then sure the integrity of the message. The CRC16 computing method that we use is described in chapter 4.

Remark : The length of a master message is always greater or equal 4 bytes.

### 3.2.2Messages coming from the devices

The messages sent as a response to the master have the following shape :

| | | | | | |
|---|---|---|---|---|---|
| Device address | X | Function code | Data (optional) | CRC16 H | CRC16 L |

- DEVICE ADDRESS :
It is a copy of the devices address present in the request to which this message replies.

- FUNCTION CODE:
It is a copy of the function code present in the request to which this message replies. If the most significant bit of the returned function code is zero, the function was correctly executed by the device. Otherwise an exception error occurred (see item 3.3.2.2).

- DATA :
This field is optional. It contains, if required, the data to transmit to the bus master.

- CRC16 :
This code is computed and used in the same way as the CRC16 present in the message coming from the master.

Remark : The length of a slave message is always greater or equal 5 bytes.

## 3.3Constitution of exchanges

### 3.3.1General rules

The message exchanges taking place on this bus have to respect the following rules :

- The order in which the devices are polled by the master is free.
- At each address only one device connected to the same bus can correspond, otherwise malfunction may occur.
- A message consists of at least 4 bytes. These bytes are transferred without any interruption during the transmission.
- The requested device is answering earliest 50ms after the reception of a message but it has to answer at least after 500ms maximum. Otherwise the message was not valid or the device is not present on the bus.


### 3.3.2Error handling

During a data exchange between the bus master and a given device, two main types of error can occur : The transmission error on one hand, and the exception error on the other hand.

### *3.3.2.1Transmission error*

This error occurs when a message has not been received correctly, regardless of the reason for it. It is detected by the receiver either because of the difference between the CRC16 present in the message and the CRC16 computed by the receiver or simply because of an error in the reception of one byte of the message, such as overrun or framing error. The reception of a message consisting of less than 4 bytes (Slave) respectively 5 bytes (Master). A message which is longer than the devices receiver buffer is also interpreted as a transmission error.

In case of a transmission error, the receiver will remain silent, discard the wrong message and will go back to receiving state without doing anything else. This is valid for both broadcasting and exchange mode.

In a general way, the bus master is the only device that monitors the data exchange on the bus. Therefore it decides of every action to be performed in such a case. This means to say how many times, how and when retrying to send a request, according to the time-out rules described in § 3.3.1. For example, it is able to send 4 requests in a row to a particular device which doesn't answer and then declares it in failure if no valid exchange could be achieved.

### *3.3.2.2Exception error*

This error occurs when a message sent by the bus master has been successfully received (correct CRC16), but one of the data inside the message makes no sense. Therefore the addressed device is not able to proceed, and sends back a so called exception message.

This error also can occur when the device was disconnected from power supply immediately before the masters request. In this case, the device always answers with an exception error and showing up with the initialization code until the master has sent the „confirmation for initialization" function. Only the confirmation command can stop the device always answering with the initialization exception error.

The structure of the answer is arranged as follows :

The second byte of the message is as well a copy of the function code present in the request, except from the first bit which is always set. This means for the bus master, that the device has detected an exception error, and was therefore maybe not able to proceed.

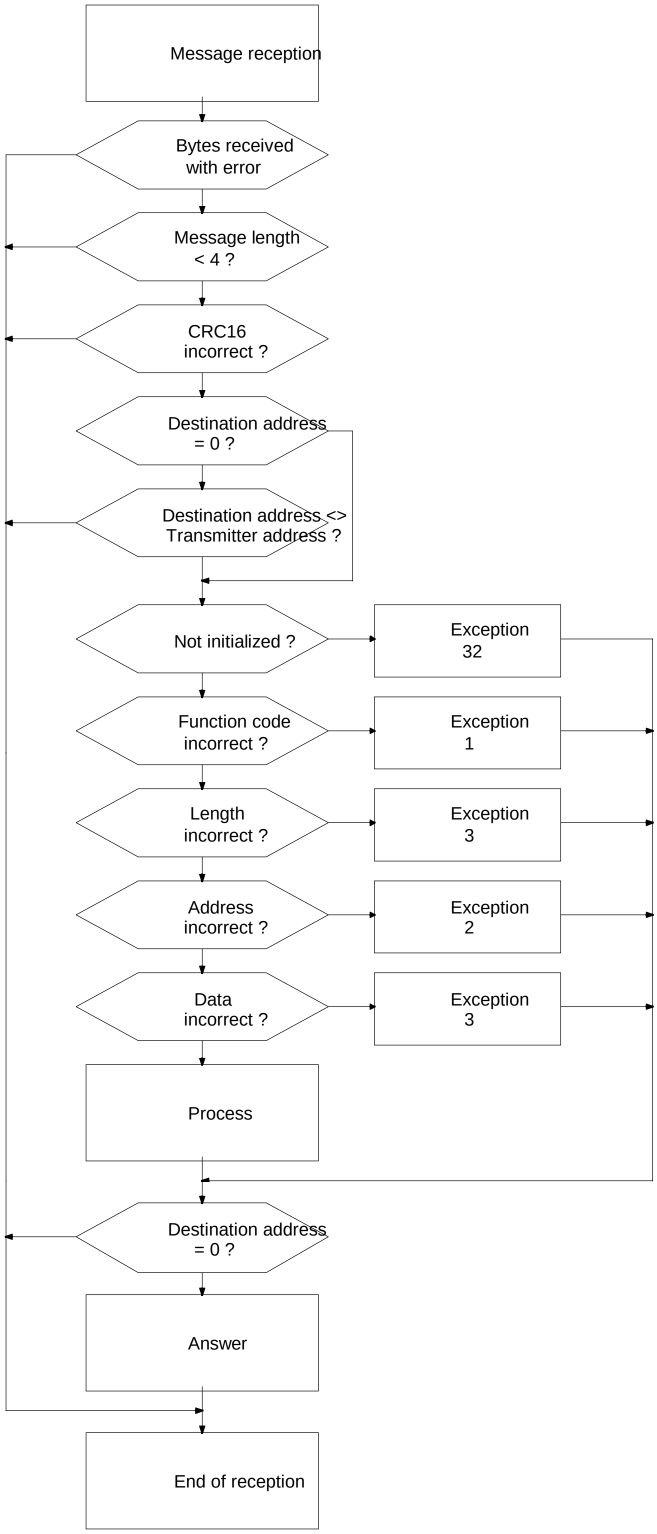
| | | | | | |
|---|---|---|---|---|---|
| Device address | 1 | Function code | Exception code | CRC16 H | CRC16 L |

The following code is the exception code indicating the type of error detected. The used codes are as follows :

| | |
|---|---|
| - unknown function code | 1 |
| - incorrect address | 2 |
| - incorrect data | 3 |
| - initialization | 32 |

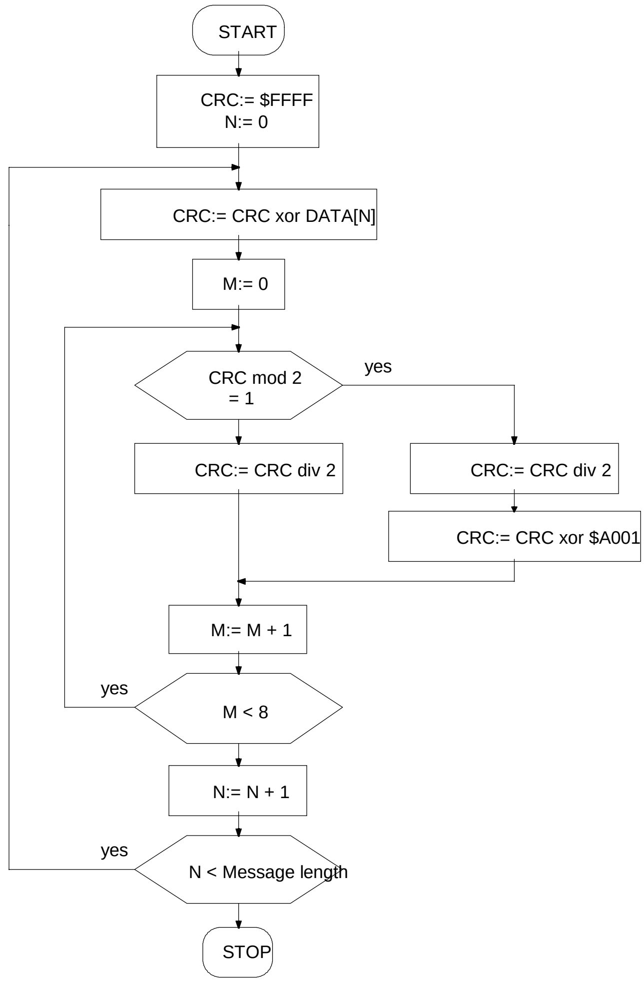
## 3.3.3Reception of a message by a device

The device receiving a message processes it usually as follows:



# 4Appendix

## 4.1Calculation of CRC-16 checksum



## 4.2Character set

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 32 | | 33 | ! | 34 | " | 35 | # | 36 | $ |
| 37 | % | 38 | & | 39 | ' | 40 | ( | 41 | ) |
| 42 | * | 43 | + | 44 | , | 45 | - | 46 | . |
| 47 | / | 48 | 0 | 49 | 1 | 50 | 2 | 51 | 3 |
| 52 | 4 | 53 | 5 | 54 | 6 | 55 | 7 | 56 | 8 |
| 57 | 9 | 58 | : | 59 | ; | 60 | < | 61 | = |
| 62 | > | 63 | ? | 64 | @ | 65 | A | 66 | B |
| 67 | C | 68 | D | 69 | E | 70 | F | 71 | G |
| 72 | H | 73 | I | 74 | J | 75 | K | 76 | L |
| 77 | M | 78 | N | 79 | O | 80 | P | 81 | Q |
| 82 | R | 83 | S | 84 | T | 85 | U | 86 | V |
| 87 | W | 88 | X | 89 | Y | 90 | Z | 91 | [ |
| 92 | \ | 93 | ] | 94 | ^ | 95 | _ | 96 | ` |
| 97 | a | 98 | b | 99 | c | 100 | d | 101 | e |
| 102 | f | 103 | g | 104 | h | 105 | i | 106 | j |
| 107 | k | 108 | l | 109 | m | 110 | n | 111 | o |
| 112 | p | 113 | q | 114 | r | 115 | s | 116 | t |
| 117 | u | 118 | v | 119 | w | 120 | x | 121 | y |
| 122 | z | 123 | { | 124 | \| | 125 | } | 126 | ~ |
| 127 | | 128 | | 129 | | 130 | ‚ | 131 | ƒ |
| 132 | „ | 133 | … | 134 | † | 135 | ‡ | 136 | ˆ |
| 137 | ‰ | 138 | Š | 139 | ‹ | 140 | Œ | 141 | |
| 142 | | 143 | | 144 | | 145 | ‘ | 146 | ’ |
| 147 | “ | 148 | ” | 149 | • | 150 | – | 151 | — |
| 152 | ˜ | 153 | ™ | 154 | š | 155 | › | 156 | œ |
| 157 | | 158 | | 159 | Ÿ | 160 | | 161 | ¡ |
| 162 | ¢ | 163 | £ | 164 | ¤ | 165 | ¥ | 166 | ¦ |
| 167 | § | 168 | ¨ | 169 | © | 170 | ª | 171 | « |
| 172 | ¬ | 173 | - | 174 | ® | 175 | ¯ | 176 | ° |
| 177 | ± | 178 | ² | 179 | ³ | 180 | ´ | 181 | µ |
| 182 | ¶ | 183 | · | 184 | ¸ | 185 | ¹ | 186 | º |
| 187 | » | 188 | ¼ | 189 | ½ | 190 | ¾ | 191 | ¿ |
| 192 | À | 193 | Á | 194 | Â | 195 | Ã | 196 | Ä |
| 197 | Å | 198 | Æ | 199 | Ç | 200 | È | 201 | É |
| 202 | Ê | 203 | Ë | 204 | Ì | 205 | Í | 206 | Î |
| 207 | Ï | 208 | Ð | 209 | Ñ | 210 | Ò | 211 | Ó |
| 212 | Ô | 213 | Õ | 214 | Ö | 215 | × | 216 | Ø |
| 217 | Ù | 218 | Ú | 219 | Û | 220 | Ü | 221 | Ý |
| 222 | Þ | 223 | ß | 224 | à | 225 | á | 226 | â |
| 227 | ã | 228 | ä | 229 | å | 230 | æ | 231 | ç |
| 232 | è | 233 | é | 234 | ê | 235 | ë | 236 | ì |
| 237 | í | 238 | î | 239 | ï | 240 | ð | 241 | ñ |
| 242 | ò | 243 | ó | 244 | ô | 245 | õ | 246 | ö |
| 247 | ÷ | 248 | ø | 249 | ù | 250 | ú | 251 | û |
| 252 | ü | 253 | ý | 254 | þ | 255 | ÿ | | |