

Model 4TX RS485 PROTOCOL

BAUD : 9600
START : 1
DATA : 8
STOP : 1
PARITY : NONE

I. Main Protocol



ACKNOWLEDGE <----- **ADDRESS**
----->
<----- **COMMAND**
DATA[0] ----->
----->
----->
----->
DATA[n] ----->

ADDRESS = 1 byte, 0-127 must add 128 to indicate as address
ACKNOWLEDGE = 1 byte, must be equal to 6
COMMAND = 1 byte, valid commands = (0,1,11,12,13,14,15,16,17,18,30)
DATA[0]-DATA[n]= depends on PARAMETER
n = number of bytes

II. DATA[0] - DATA[n] FORMAT

1. IF **COMMAND = 0** then DATA[0]..DATA[37] (page 0)

FUNCTION : to get display data during normal operation.

DATA[0] - DATA[5] = pH Value, ASCII FORMAT
Possible reading: '-02.00' - '+16.00', 'UNDER ', 'OVER '
DATA[6] - DATA[11] = Temperature Value, ASCII Format
Possible reading: '-010.0' - '+120.0', 'UNDER ', 'OVER '
DATA[12] - DATA[17] = Analog Out (mA), ASCII Format
Possible reading: '+03.00' - '+22.00', 'OFF ', 'FROZEN' &
'ERROR '
'OFF ' = Analog system is OFF.
'FROZEN' = Analog output is frozen.

DATA[18] - DATA[23] = ORP Absolute mV Value
Possible reading : '-02500'-' +02500', 'OVER ', 'UNDER '
DATA[24] - DATA[29] = ORP Relative mV Value
Possible reading : '-06499'-' +06499', 'OVER ', 'UNDER '

DATA[30] = 1 byte, FLAGS, binary coded
Bit 0 = if 1 then RELAY 1 is ON
Bit 1 = if 1 then RELAY 2 is ON
Bit 2 = if 1 then RELAY 3 is ON
Bit 3 = if 1 then RELAY 4 is ON
Bit 4 = if 1 then RELAY 5 is ON
Bit 5 = if 1 then unit is password locked
Bit 6 = Main display ORP unit
= if 0 = Relays & mA output FROZEN/OFF
= if 1 = Relays & mA output normal
Bit 7 = not used
DATA[31] = 1 byte, FLAGS, binary coded
Bit 0 = RELAY 1 ACTION : 0=LO, 1=HI
Bit 1 = RELAY 2 ACTION : 0=LO, 1=HI
Bit 2 = RELAY 3 ACTION : 0=LO, 1=HI
Bit 3 = RELAY 4 ACTION : 0=LO, 1=HI
Bit 4 = Active reading for Relay & mA is pH
Bit 5 = Active reading for Relay & mA is ABS
mV
Bit 6 = Active reading for Relay & mA is REL
mV
Bit 7 = not used

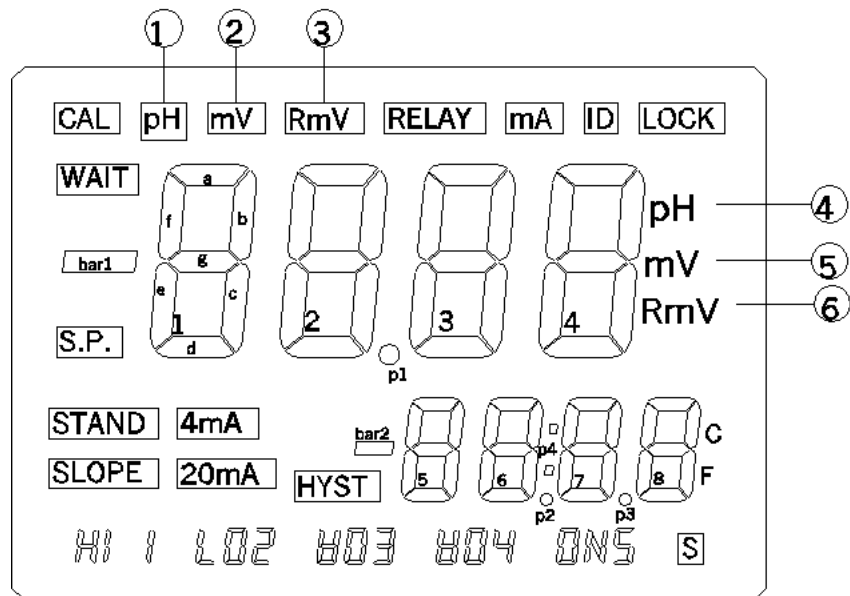


Figure 1.

2. IF **COMMAND** = 1 then DATA[0]..DATA[15]

FUNCTION : To get the current LCD Display

DATA[0] - DATA[3] = the 4 major(see figure 1) digits.
= ASCII FORMAT
DATA[0] is the most significant digit.

DATA[4] - DATA[7] = the 4 minor(see figure 1) digits.
= ASCII FORMAT
DATA[4] is the most significant digit.

DATA[8] - DATA[9] = reserved

DATA[10] = 1 byte, binary coded.

Bit 0 = if 1 then RELAY 1 is ON (see **DATA[11]**)
Bit 1 = if 1 then RELAY 2 is ON (see **DATA[11]**)
Bit 2 = if 1 then RELAY 3 is ON (see **DATA[11]**)
Bit 3 = if 1 then RELAY 4 is ON (see **DATA[11]**)
Bit 4 = if 1 then RELAY 5 is ON (**ON5**)
Bit 5 = if 1 then unit is password locked
Bit 6 = if 1 all data are valid.
Check this flag first before updating the display of your program.

Bit 7 = not used

DATA[11] = 1 byte, binary coded.

Bit 0	= if 1 then RELAY 1 is ACTIVE HIGH(HI1), 0=LOW(LO1)
Bit 1	= if 1 then RELAY 2 is ACTIVE HIGH(HI2), 0=LOW(LO2)
Bit 2	= if 1 then RELAY 3 is ACTIVE HIGH(HI3), 0=LOW(LO3)
Bit 3	= if 1 then RELAY 4 is ACTIVE HIGH(HI4), 0=LOW(LO4)
Bit 4	= reserved
Bit 5	= reserved
Bit 6	= reserved
Bit 7	= not used

DATA[12] = 1 byte, binary coded.

Bit 0	= annunciator (6) (RmV) (see figure 1)
Bit 1	= annunciator (4) (pH) (see figure 1)
Bit 2	= major digit's decimal point (p1) (88.88) (see figure 1)
Bit 3	= minor digit's decimal point (p2) (88.88) (see figure 1)
Bit 4	= minor digit's decimal point (p3) (888.8) (see figure 1)
Bit 5	= colon at minor digit (88:88) (see figure 1)
Bit 6	= degree C (°C)
Bit 7	= not used

DATA[13] = 1 byte, binary coded.

Bit 0	= major digit's sign bar (-88.88) (see figure 1)
Bit 1	= annunciator 3 (RmV) (see figure 1)
Bit 2	= annunciator 1 (pH) (see figure 1)
Bit 3	= WAIT annunciator
Bit 4	= HYST (hysterisis) annunciator
Bit 5	= 20mA annunciator
Bit 6	= SLOPE annunciator
Bit 7	= not used

DATA[14] = 1 byte, binary coded.

Bit 0	= RELAY annunciator
Bit 1	= annunciator 2 (mV) (see figure 1)
Bit 2	= CAL annunciator
Bit 3	= S.P. (set point) annunciator
Bit 4	= 4mA annunciator
Bit 5	= STAND annunciator
Bit 6	= minor sign bar (-8888) annunciator
Bit 7	= not used

DATA[15] = 1 byte, binary coded.

Bit 0	= annunciator 5 (mV) (see figure 1)
Bit 1	= LOCK annunciator
Bit 2	= mA annunciator
Bit 3	= reserved
Bit 4	= reserved
Bit 5	= ID annunciator
Bit 6	= reserved
Bit 7	= not used

3. IF **COMMAND = 11 to 18** then DATA[0]

COMMAND = 11

FUNCTION : same as pressing the **MODE key** on the model 4TX POT keypad.

COMMAND = 12

FUNCTION : same as pressing the **CAL key** on the model 4TX POT keypad but bypassing the 2 sec. delay requirement.

COMMAND = 13

FUNCTION : same as pressing the **UP key** on the model 4TXPOT Keypad.

COMMAND = 14

FUNCTION : same as pressing the **DOWN key** on the model 4TX POT keypad.

COMMAND = 15

FUNCTION : same as pressing the **ENTER key** on the model 4TX POT keypad.

COMMAND = 16

FUNCTION : same as pressing the **MODE key for 2 secs.** on the Model 4TX POT keypad.

COMMAND = 17

FUNCTION : same as pressing the **WASH key** on the model 4TX POT keypad.

COMMAND = 18

FUNCTION : same as pressing the **WASH key for 2 secs.** on the Model 4TX POT keypad.

Data format for command 11 to 17:

DATA[0] = 1 Byte
= if 6 then sent key is executed
= if 18 then unit is busy, sent key
is not executed

4. IF **COMMAND = 30** then DATA[0]..DATA[9]
FUNCTION : to get the page no. & model no.

DATA[0] = reserved

DATA[1]-DATA[9] = ASCII format

Some examples: 'EN4TXPOT', 'EN6308OT', 'CH6308OT'..etc.

EN= English

CH= Chinese

4TXPOT = model

© May 14, 2003