INDICATOR 3100N

EXCEL PROTOCOL

EXCEL PROTOCOL

The Excel Protocol (EP) is upgraded in order to comply with the new alibi possibilities. If the EP is selected a data string will be send out through the selected com port on every print command. If parameter 25/35 is set to 1, the EP is active.

The transmitted data string contains the following fields:

Field nr.	description	Number of characters/digits send
1	Scale number; any value between 0 and 255 to be entered in parameter 98	3
2	Date; depending on the setting of parameter 19 this will be *dd/mm/yy* (EU) or *mm/dd/yy* (US)	8
3	Time; format *hh:mm*	5
4	Gross weight; this will always be 5 digits, a decimal point, the units (kg or lb) and the positive or negative sign. For example: +0233.5kg or -00136.lb.	9
5	Net weight; this will always be 5 digits, a decimal point, the units (kg or lb), the positive or negative sign and whether it was a calculated net or a measured net. For example: +0233.5kgC or -00136.lb_. The “C” stands for calculated and is send along when a preset tare value was active. If there is no preset tare value active a blanc (space) is put behind the kg (or lb).	10
6	Tare weight; this will always be 5 digits, a decimal point, the units (kg or lb), the positive or negative sign and whether it was a preset tare or a measured tare. For example: +0233.5kgP or -00136.lb_. The “P” stands for preset tare and is send along when a preset tare value was active. If there is no preset tare value active a blanc (space) is put behind the kg (or lb).	10
7	Code; this is the code which can be entered by keypad. If no code is activated this field is left blanc (5 spaces)	5
8	Alibi number > this is a 4 digit number which is generated by the indicator itself. It will start at “0001” and increase with every weighing up to “9999”. When this number is reached it will start at “0001” again.	4
	Total characters send	54

All fields will be separated by a separator sign, semi-colon (;).

The value of characters to be transmitted in one line are 54(fields)+7(;)=61 characters. As ending character(s) we have a <CR> or <LF> or <CR>and<LF> depending on the setting in parameter 24/34. A complete data string could look like following:

001;09/10/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;0024<CR>

or

001;09/01/09;15:42;+00255.lb;+00203.lb_;+00052.lb_;54321;0102<CR><LF>

Excel protocol with ACK/NACK

If the data is to be transmitted wireless (for example by Wifi or Bluetooth) it is hard to check whether the data has been received and whether the received data was correct. This asks for a possibility to work with some kind of acknowledgement. In these cases we can set parameter 25/35 to 6; PC Excel protocol with ack/nack.

The EP with ACK/NACK works similar to the normal EP. The data string contains the same fields. Also the separator is identical (;)

The biggest difference is the fact that the indicator waits for a reply of the PC/Terminal before it continues. Secondly a checksum is calculated over the data string and send along with it. In order to work with this protocol you will need to program an interface which generates the right commands and is capable of saving the data into a file. Underneath you will find some guidelines for making such an interface.

Communication protocol between indicator and PC:

	INDICATOR			**PC**	
Conditions:	Indicator is turned on			PC read out program is working	
1	Press print key for data transmission			PC waits for data	
2	Collecting the weigh data and calculating the checksum (see calculating the checksum)			PC waits for data	
3	Transmitting data and checksum to PC	Data out	→	PC receives data	Data in
4	Indicator waits for reply			Putting data in buffer and calculating checksum	
5	If ACK is received go on to step 6 If NACK is received go back to step 3 and add 1 attempt to the counter If NACK is received 5 times, display “trErr” and wait for any key press to return into the normal weighing mode. If the reaction from the PC takes longer than 3 seconds, a timeout will be generated by the indicator and the display will show	Data in	←	If Checksum=ok, return ACK with dummy character (any HEX value between 21 and FF)and go on to step 6 Checksum=not ok. return NACK with dummy character (any HEX value between 21 and FF), clear buffer and go back to step 1	Data out

	“trErr” and wait for any key press to return into the normal weighing mode.				
6	Display “done” for 1 second and return to the normal weighing mode			Store buffer into CSV file and return to step 1	

Example:

Indicator	TXD	001;09/01/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;002444<CR>
		⇩
PC	RXD	000;09/01/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;002444<CR>
		⇩
Remark:		data was corrupted while transmitted which will result in a wrong checksum. The PC needs to demand for another transmission.
		⇩
PC	TXD	<NACK><dummy><CR> Nack = hex value 15, dummy = hex value 21, CR = hex value 0D
		⇩
Indicator	RXD	<NACK><dummy><CR>
		⇩
Indicator	TXD	001;09/01/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;002444<CR>
		⇩
PC	RXD	001;09/01/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;002444<CR>
		⇩
Remark:		data was correct while transmitted which results in a good checksum. The PC needs to send an acknowledgement and save the weighing in a file.
		⇩
PC	TXD	<ACK><dummy><CR> Ack = hex value 06, dummy = hex value 21, CR = hex value 0D
		⇩
Indicator	RXD	<ACK><dummy><CR>
		⇩
Remark:		indicator will close of the sequence and return to the weigh mode. PC remains in standby mode for next transmission.

Calculating the checksum

The calculation of the checksum will be similar to the way we calculate the checksum with the GW command in the PC protocol. This means:

1. Add all hex values of the characters in the string
2. Remove the most significant digit
3. Invert the hex value resulting from that
4. Convert the hex value to characters which are transmitted along with the data string.

Example:

Data string = 001;09/01/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;0024<CR>

1. Add all hex values of the characters in the string;
   [0]+[0]+[1]+[;]+[0]+[9]+[/]+[0]+[1]+[/]+[0]+[9]+[;]+[1]+[5]+[:]+[4]+[0]+ etc.

This results in:

30+30+31+3B+30+39+2F+30+31+2F+30+39+3B+31+35+3A+34+30+3B+2B+30+31+32+35+2C+35+6B+67+3B+2B+30+31+30+30+2C+35+6B+67+3B+43+3B+2B+30+30+32+35+2C+30+6B+67+50+3B+31+32+33+34+35+3B+30+30+32+34= DBB

2. Remove the most significant digit = BB
3. Inverting the hex value = 44
4. Convert to characters and transmit along with the data string : 44

Complete data string with checksum becomes:

001;09/01/09;15:40;+0125.5kg;+0100.5kgC;+0025.0kgP;12345;002444<CR>