Eilersen
The Weighing Experts

5016 Installation and Communication RS485

5016 Loadcell connection Module

RS485 full-duplex point-to-point communication

Software: LB.150611.2v1

Doc. no.: 5016 InstAndComm 150611-1v0e-eng.doc

Date: 2015-11-28

Rev.: 1v0e

Contact:
Eilersen Electric A/S
Kokkedal Industripark 4
DK-2980 Kokkedal
Denmark
www.eilersen.com
info@eilersen.com
Tel: +45 49 180 100
Fax: +45 49 180 200

Contents

Introduction

The Eilersen Electric 5016 serial loadcell connection module has inputs for 16 loadcells. It has a 4-wire full-duplex RS485 interface for connection to a master system.

Via the serial interface weighings can be trigged, weighing results transmitted, weighing analysis and calibration data can be requested and transmitted.

Installation

To install the system the follow this procedure:

- Connect the loadcells; please see section *– Connect the loadcells*, page 34, below for further details.
- Connect the RS485 communication; please see section *– Connect serial communication*, page 34, below for further details.
- Connect the +24 DC power; please see section,*– Connect power (TBD)* page 34, below for further details.

There is no need for any further setup. Number of loadcells, filters etc. are set via the communication interface.

Communication

Specification

Electrical:	RS485 (4-wire, full-duplex).
Layout:	Point-to-point. (One master, one slave, no multi-drop).
Baudrate:	115200 bps.
Data bits:	8.
Parity:	None.
Stop bits:	1.
Protocol:	Binary with possibly embedded ASCII decimal and hex with checksum as described below.

Protocol

Telegram format

All telegrams are in binary format. Telegrams can contain three different data items

- Request command: An embedded ASCII format command sent from the master to the 5016.
- Response message: An embedded ASCII format response sent from the 5016 to the master.
- Analysis data: A binary format analysis dataset sent from the 5016 to the master.

Telegram format:

<STX><LEN><DATA><CS>

The components are:

<STX>	Start-of-text: Hex 0x02, Decimal: 2.
<LEN>	Number of bytes in the <DATA> section as a 1-digit binary number.
<DATA>	Zero, one or more binary characters. When the data is a request command or a response message the characters are ASCII.
<CS>	Checksum. <DATA>Please see section *– Calculate protocol checksum* , page 33, below for further details and example.

As Request commands and Response messages are ASCII telegrams embedded in the <DATA> portion and the both start with a <LF> character, full binary telegrams (the Analysis data is currently the only full binary telegram) must NOT start with a <LF> character.

Receive algorithm

Please notice that the <LEN>, <DATA> and <CS> fields are binary data and may contain a <STX>. A special algorithm should be used to receive data and to recover from errors in telegrams:

TBD

Flow control

When the master is idle (not transmitting a telegram) it may hold the data transmission from the 5016 by transmitting a single control character:

<NAK> Negative Acknowledged: Hex 0x21, Decimal: 33.

When the hold request is sent, the 5016 will still transmit what is present in the transmission buffer. When including transmission and processing delay this means that the 5016 may transmit 2-5 characters after the hold request is sent.

To resume transmissions from the 5016 the master must sent a single control character:

<ACK> Acknowledge: Hex 0x06, Decimal: 6.

When the resume request is sent, the 5016 will resume character transmission in 2-5 ms.

PLEASE NOTICE that if the 5016 does not receive this character, e.g. due to transmission error, transmission will not resume. To insure transmission is resumed a time-out with retransmission must be implemented in the master.

Request command

All commands sent from a master to the 5016 are built with the following components:

<LF><CMD><PARAMETERS><CS><CR>

The components are:

<LF>	ASCII Linefeed: Hex 0x0a, Decimal: 10.
<CMD>	Printable ASCII upper case letter.
<PARAMETERS>	Zero, one or more ASCII decimal or hex number preceded, separated and proceeded by ASCII ";": Hex 0x3b, decimal 59. Each parameter has fixed length filling with preceding spaces or zeroes. An empty parameter list consists of only a separator ASCII ";": Hex 0x3b, decimal 59.
<CS>	Checksum. Logical XOR of all preceding characters (including <LF>) printed as 2 capital ASCII hex digits. Please see section *– Calculate protocol checksum* , page 33, below for further details and example.
<CR>	ASCII Carriage return: Hex 0x0d, Decimal: 13.

Response message

If a malformed command is received no response is sent. All messages sent from the 5016 to the master are built with the following components:

<LF><RSP><DATA><CS><CR>

The components are:

<LF>	ASCII Linefeed: Hex 0x0a, Decimal: 10.
<RSP>	Printable ASCII lower letter. A copy of the command received changed to lower case. If the message is a status message not sent as a response to any message from the master <RSP> is uniquely identifying the message.
<DATA>	Zero, one or more ASCII decimal or hex number preceded, separated and proceeded by ASCII ";": Hex 0x3b, decimal 59. Each parameter has fixed length filling with preceding spaces (hex 0x20, decimal 32)or zeroes. Negative numbers are transmitted with a preceding minus sign (hex 0x2d , decimal 45) before any zero filling (e.g. "-001234") or after any space filling (e.g. "__-1234"). An empty data list consists of only a separator ASCII ";": Hex 0x3b, decimal 59.
<CS>	Checksum. Logical XOR of all preceding characters (including <LF>) printed as 2 capital ASCII hex digits. Please see section *– Calculate protocol checksum* , page 33, below for further details and example.
<CR>	ASCII Carriage return: Hex 0x0d, Decimal: 13.

Command overview

Command	<CMD> <RSP>	Parameter(s)	Response data	Details page
setFilterMode	F F	Filter number	Filter Number	8
getFilterMode	G G	None	Filter Number	9
setNumberOfUnits	N N	Number of units	Number of units set Number of units supported Number of units detected	10
getNumberOfUnits	M m	None	Number of units set Number of units supported Number of units detected	11
resInit	j	-	Number of units set Number of units supported Number of units detected	11
setParameter	S s	Parameter ID Parameter value	Parameter ID Parameter value	12
getParameter	P p	Parameter ID	Parameter ID Parameter value	13
trigWeighing	T t	Unit Trigger type Measuring time	Unit	14
resWeighing	- r	-	Unit Weighing result	15
trigAnalysis	A a	Trigger type Trigger value Time before Time after	Trigger type	16
resAnalysis	- d	-	Unit Status Index Weight value	18
trigCalibration	C C	Unit Measuring time	Unit	20
resCalibration	d	-	Unit Weight value	21
getAvgWeight	W	Unit	Unit	22

	w		Average weight	
getStatusInfo	I i	Status ID	General status Status ID Status value	23

Commands and responses

setFilterMode

Command: 'F' 0x46 hex 70 decimal

Parameter:

Filter number:

2 decimal digits (00-97/98). 00: Select no filter

Example:

<LF>F;12;4F<CR>

Select filter 12.

<LF>F;00;4C<CR>

Select no filter. Previous filter cancelled.

Response: 'f' 0x66 hex 102 decimal

Response data:

Filter number set:

2 decimal digits (00-97/88, 99). 99: Illegal filter number requested.

Examples:

<LF>f;12;6F<CR>

Filter 12 selected.

<LF>f;00;6C<CR>

No filter selected.

<LF>f;99;6C<CR>

Illegal filter requested. Filter selection not changed.

Usage:

This message is used to switch between a number of preconfigured filters or no filter at all. Please observed that filter number 98 is not a built-in but is reserved for special purposes and should NOT be used for normal operation of the system.

getFilterMode

Command: 'G' 0x47 hex 71 decimal

Parameters:

None

ASCII example:

<LF>G;76<CR>

Embedded example:

0x02 0x06 0x0a 0x47 0x3b 0x37 0x36 0x0d 0x7e

Response: 'g' 0x67 hex 103 decimal

Response data:

Filter number set:

2 decimal digits (00-97/98).

ASCII Examples:

<LF>g;12;6E<CR>

Filter 12 is currently selected.

<LF>g;00;6D<CR>

No filter is currently used.

EmbeddedeExample:

0x02 0x09 0x0a 0x67 0x3b 0x39 0x38 0x3b 0x36 0x43 0x0d 0x1f

Special filter no 98 is currently selected.

Usage:

This message is used to inquire which of the preconfigured filters is currently selected. Please observed that filter number 98 is not a built-in but is reserved for special purposes and should NOT be used for normal operation of the system.

setNumberOfUnits

Command: ‘N’ 0x4e hex 78 decimal

Parameter:

Number of units:

2 decimal digits (08 or 16).

Examples:

<LF>N;08;4C<CR>

Select 8 units.

<LF>N;16;43<CR>

Select 16 units.

Response: ‘n’ 0x6e hex 110 decimal

Response data:

Number of units set:

2 decimal digits (00. 08 or 16), 00: Illegal number of units requested.

Number of units supported:

2 decimal digits (00. 08 or 16), 00: Number of units/units supported not yet detected.

Number of units detected:

2 decimal digits (00-16).

Examples:

<LF>n;08;16;08;63<CR>

8 units selected, 16 units supported; 8 units detected.

<LF>n;00;16;16;64<CR>

Illegal number of units requested. Set number of units not changed. 16 units supported. 16 units detected.

Usage:

This message is used to set the total number of units that the weighing module must be able to measure on. If the weighing module is equipped with more units, then it must simply disable/skip these. If the number of detected units does not match the expected number an error condition must be indicated.

getNumberOfUnits

Command: ‘M’ 0x4d hex 77 decimal

Parameters:

None

Example:

<LF>M;7C<CR>

Response: ‘m’ 0x6d hex 109 decimal

Response data:

Number of units set:

2 decimal digits (08 or 16).

Number of units supported:

2 decimal digits (00. 08 or 16), 00: Number of units/units supported not yet detected.

Number of units detected:

2 decimal digits (00-16).

Examples:

<LF>m;08;16;08;60<CR>

8 units selected. 16 units supported. 8 units detected.

<LF>m;08;00;00;6F<CR>

8 units selected. Number of unit/supported units not yet detected.

Usage:

This message is used to inquire the total number of units that the weighing module must be able to measure on and the number of detected units.. If the number of detected unit does not match the expected number an error condition must be indicated.

resInit

Response: ‘j’ 0x6a hex 109 decimal

Response data:

Number of units set:

2 decimal digits (08 or 16).

Number of units supported:

2 decimal digits (00. 08 or 16), 00: Number of units/units supported not yet detected.

Number of units detected:

2 decimal digits (00-16).

Examples:

<LF>j;08;16;08;67<CR>

8 units selected. 16 units supported. 8 units detected.

<LF>j;08;00;00;68<CR>

8 units selected. Number of unit/supported units not yet detected.

Usage:

This message is sent to inform the master that the systems is ready after reset.

setParameter

Command: 'S' 0x53 hex 83 decimal

Parameters:

Parameter ID:

3 decimal digits (101-999). Please see section *Parameter overview*, page 23, below for further details on valid parameters and their usage.

Parameter value:

10 decimal digits signed number (-999999999-9999999999).

Example:

<LF>S;101;0000000400;56<CR>

Set parameter 101 to a new value of 400.

Response: 's' 0x73 hex 115 decimal

Response data:

Parameter ID:

3 decimal digits (001, 002, 003, 009 or101-999). 001: Invalid parameter number, 002: Parameter value too small, 003: parameter value too big, 009: Parameter value invalid.

Parameter value set:

10 decimal digits (-999999999-9999999999). PLEASE NOTICE that this value may differ slightly from value requested due to rounding or resolution.

Examples:

<LF>s;101;0000000400;76<CR>

Parameter 101 now set to a new value of 400.

<LF>s;001;0000000000;73<CR>

Invalid parameter number set. No parameter values are changed.

<LF>s;009;0000000000;7B<CR>

Invalid parameter value requested. Parameter value is not changed.

Usage:

This message is used to set any parameter to the desired value. Please see section *Parameter overview*, page 23, below for further details on valid parameters and their usage.

getParameter

Command: ‘P’ 0x50 hex 80 decimal

Parameter:

Parameter ID:

3 decimal digits (101-999). Please see section *Parameter overview*, page 23, below for further details on valid parameters and their usage.

Example:

<LF>P;101;6A<CR>

Request parameter 101 value.

Response: ‘p’ 0x70 hex 112 decimal

Response data:

Parameter ID:

3 decimal digits (001 or101-999). 001: Invalid parameter number.

Parameter value:

10 decimal digits (-999999999-9999999999).

Examples:

<LF>p;101;0000000400;75<CR>

Parameter 101 value is 400.

<LF>p;001;0000000099;70<CR>

Invalid parameter number requested.

Usage:

This message is used to read any parameter value. Please see section *Parameter overview*, page 23, below for further details on valid parameters and their usage.

trigWeighing

Command: 'T' 0x54 hex 84 decimal

Parameters:

Unit to trig:

2 decimal digits (01-16).

Type:

1 decimal digit (1-2).

Measuring time in milliseconds:

4 decimal digits (0002-9999).

Example:

<LF>T;05;1;0300;69<CR>

Trig weighing on unit 5 as type 1 with 300 ms measuring time.

Response: 't' 0x74 hex 116 decimal

Response data:

Unit trigged:

2 decimal digits (00-16), 00: Invalid unit or measuring time requested.

Examples:

<LF>t;05;7B<CR>

Unit 5 trigged for weighing.

<LF>t;00;7E<CR>

Invalid unit or measuring time requested. No weighing trigged.

Usage:

This message is used to trig a weighing starting when the trigger is received. When the measuring time has elapsed a *resWeighing* telegram is sent. The type is only for informational purposes. It is returned in a *resAnalysis* or *dataAnalysis* telegram if a weighing is trigged during analysis.

resWeighing

Response: ‘r’ 0x72 hex 114 decimal

Response data:

Unit that has completed a weighing:

2 decimal digits (01-16).

Weighing result:

10 digit signed decimal number. Resolution is internal loadcell resolution. 9999999999: Error during weighing. No valid result available.

Examples:

<LF>r;13;0000027376;46<CR>

Unit 13 weighing has completed with a result of 27376.

<LF>r;07;-000009257;50<CR>

Unit 7 weighing has completed with a result of -9257.

<LF>r;03;9999999999;40<CR>

Unit 3 weighing has completed with an error, no valid result.

Usage:

This message is sent when a previous trigged weighing has finished, when the measuring time has elapsed.

trigAnalysis

Command: 'A' 0x41 hex 65 decimal

Parameters:

Trigger type:

1 decimal digit (0-3; 6-8):

0: Cancel analysis:

Cancel any on-going analysis. If no analysis is started the command is ignored. This can be done before or after the analysis is actually trigged.

1: Trig on weight: Transmit resAnalysis telegrams when trigged.

6: Trig on weight: Transmit dataAnalysis telegrams when trigged.

An analysis is trigged on the first unit where the internal loadcell value raises with the *Trigger value* parameter.

2: Trig on trigger: Transmit resAnalysis telegrams when trigged.

7: Trig on trigger: Transmit dataAnalysis telegrams when trigged.

An analysis is trigged on the unit number in the *Trigger value* parameter when a trigWeighing is received.

3: Trig instantly: Transmit resAnalysis telegrams.

8: Trig instantly: Transmit dataAnalysis telegrams.

An analysis is trigged on the unit number in the *Trigger value* parameter instantly.

Other values are invalid.

Trigger value:

8 decimal digits. When trig on weight is selected, this is the raise in the internal loadcell value needed to trig an analysis. When trig on trigger or trig instantly is selected, this is the unit to trig an analysis for.

Time before:

4 decimal digits (0000-2500). Time in milliseconds. The analysis contains data from this time before the analysis is trigged.

Time after:

4 decimal digits (0000-9999). Time in milliseconds. The analysis contains data from this time after the analysis is trigged. Please notice the sum of *Time before* and the *Time After* parameters must not exceed 10000.

Examples:

<LF>A;0;00000000;0000;0000;40<CR>

Cancel on-going analysis, if any.

<LF>A;1;00010000;1000;8000;49<CR>

Trig an analysis on the first unit where the internal loadcell values raises 10000. Send data from 1000 ms before and 8000 ms after the trigger point.

<LF>A;2;00000012;0000;5000;44<CR>

Trig an analysis on unit when a trigWeighing is received on unit 12. Send data from the trigger point and 5000 after.

<LF>A;3;00000003;1000;9000;48<CR>

Trig an analysis now on unit 3. Send data 1000 ms old and 9000 ms from now.

Response: 'a' 0x61 hex 97 decimal

Response data:

Trigger type set:

1 decimal digit (0-3, 9): 0: Any ongoing analysis cancelled. 1: Trig on weight analysis started. 2: Trig on trigger analysis started. 9: Invalid trigger type or value requested.

Examples:

<LF>a;2;59<CR>

Analysis on trigger started.

<LF>a;9;52<CR>

Invalid trigger or measuring time requested. No weighing trigged.

Usage:

This message is used to start an analysis. The analysis is trigged a resAnalysis telegram is sent for every loadcell sample (sample time 2 ms) in the time interval requested. Alternatively a number of dataAnalysis telegrams are sent, each containing data for 16 loadcell samples. As the interval may start before the trigger point and the transmission time of the resAnalysis or the dataAnalysis telegrams may be longer than the sample time then transmission of resAnalysis or the dataAnalysis telegrams may end some time after the analysis has actually ended.

Please also notice that only **ONE** analysis can run at a time.

resAnalysis

Response: ‘b’ 0x62 hex 98 decimal

Response data:

Unit:

2 decimal digits (01-16). Unit triggered for analysis.

Status:

1 capital hex digit (1-F):

Hex representation of a 4 bit binary value:

Bit 0: Weighing of type 1running, a *trigWeighing* is received with type 1 and the measuring time has not yet elapsed.

Bit 1: Weighing of type 2running, a *trigWeighing* is received with type 2 and the measuring time has not yet elapsed.

Bit 3: Error during this sample.

Index:

4 decimal digits (0001-5000). First telegram in analysis has index 0001, index incremented for every telegram.

Weight value:

10 digit signed decimal number. Resolution is internal loadcell resolution. 9999999999: Error during this sample.

Examples:

<LF>b;13;0;0001;0000012876;46<CR>

Unit 13 first sample, no weighing triggered, internal loadcell value is 12876.

<LF>b;07;1;0876;-000316423;70<CR>

Unit 7 sample 876, weighing triggered, internal loadcell value is -316423.

<LF>b;03;8;0122;9999999999;69<CR>

Unit 3 sample 122. Error no valid loadcell value calculated.

Usage:

A number of these messages can be sent when an analysis is trigged.

dataAnalysis

The dataAnalysis telegram is a full binary telegram with no embedded ASCII and is built with the following components

<ID><UNIT><NO><IX><SEC00><SEC01> … <SEC15>

The components are:

<ID>	1 byte: ASCII 'D': Hex 0x44, Decimal: 68.
<UNIT>	1 byte: Binary 8 bit number in the range 1-16. Unit triggered for analysis.
<NO>	1 byte: Binary 8 bit number in the range 1-16. Indicating the number of valid sections. Normal when an analysis is trigged all but the last telegram will have this value set to 16, in the last telegram the number may be smaller to indicate that only some of the sections contains valid analysis data.
<IX>	2 bytes: Binary 16 bit number (LSB first) in the range 0001-5000. First telegram in analysis has index 0001, index incremented with 16 for every telegram with 16 valid sections. The number will the always be the index of the first section in the telegram.
<SEC00-15>	4 bytes per section totaling to 64 bytes: The first byte in each section is a status indicator: Bit 0: Weighing of type 1running, a *trigWeighing* is received with type 1 and the measuring time has not yet elapsed. Bit 1: Weighing of type 2running, a *trigWeighing* is received with type 2 and the measuring time has not yet elapsed. Bit 3: Error during this sample. The 3 last bytes of each section is the weight in internal resolution, LSB first. Please notice that this might be a negative number (2-complement), and sign must be converted correctly when copying to a 4 byte variable

Example (TBD)

Usage:

A number of these messages can be are sent when an analysis is trigged.

trigCalibration

Command: 'C' 0x43 hex 67 decimal

Parameters:

Unit to calibrate:

2 decimal digits (01-16). More units can be calibrating simultaneously.

Measuring time in milliseconds:

4 decimal digits (0002-9999).

Example:

<LF>C;05;0500;72<CR>

Calibrate on unit 5 with 500 ms measuring time.

Response: 'c' 0x63 hex 99 decimal

Response data:

Unit calibrating:

2 decimal digits (00-16), 00: Invalid unit or measuring time requested.

Examples:

<LF>c;07;6E<CR>

Unit 7 calibrating.

<LF>c;00;69<CR>

Invalid unit or measuring time requested. No calibration started.

Usage:

This message is used to trig a calibration starting when the reading is steady (timeout 10 seconds). When the measuring time has elapsed a *resCalibration* telegram is sent.

resCalibration

Response: ‘d’ 0x64 hex 100 decimal

Response data:

Unit that has completed a calibration:

2 decimal digits (01-16)

Calibration result:

10 digit signed decimal number. Resolution is internal loadcell resolution. 9999999999: Error during calibration or timeout. No valid result available.

Examples:

<LF>d;13;0000027376;50<CR>

Unit 13 Calibration has completed with a result of 27376.

<LF>d;07;-000009257;46<CR>

Unit 7 calibration has completed with a result of -9257.

<LF>d;03;9999999999;56<CR>

Unit 3 weighing has completed with e.g. a timeout, no valid result.

Usage:

This message is sent when a calibration has finished, when the measuring time has elapsed or after a 10 second timeout.

getAvgWeight

Command: 'W' 0x57 hex 87 decimal

Parameters:

Unit:

2 decimal digits (01-16).

Example:

<LF>W;02;5F<CR>

Read average weight for unit 2.

Response: 'w' 0x77 hex 119 decimal

Response data:

Unit:

2 decimal digits (00-16), 00: Invalid unit requested.

Average weight:

10 digit signed decimal number. Resolution is internal loadcell resolution. 9999999999: Error during weighing. No valid result available.

Examples:

<LF>w;13;0000027376;43<CR>

Unit 13 weighing has completed with a result of 27376.

<LF>w;07;-000009257;55<CR>

Unit 7 weighing has completed with a result of -9257.

<LF>w;03;9999999999;45<CR>

Unit 3 weighing has completed with an error, no valid result.

Usage:

This message is used get the last average weight reading for continuous weight display. The averaging measuring time can be set as a parameter, please see section *Parameter overview*, page 24, below .

getStatusInfo

Command: 'I' 0x49 hex 73 decimal

Parameter:

Status ID:

3 decimal digits (101-999). Please see section *Status overview,* page 25, below for further details on valid parameters and their usage.

Example:

<LF>I;102;70<CR>

Request parameter 102 value.

Command: 'i' 0x69 hex 105 decimal

Response data:

General status:

2 capital hex digits (00-FF):

Hex representation of a 8 bit binary value:

Bit 0: System is operational; no errors.

Bit 1: Units not yet detected.

Bit 2: Unit error – one or more units in an error condition

Bit 3: *Reserved for future use.*

Bit 4: Power low. Supply more the 10% below specified value.

Status ID:

3 decimal digits (001 or101-999). 001: Invalid status ID requested.

Status value:

10 digit signed decimal number or 10 hex digits (-999999999-9999999999 or 0000000000-FFFFFFFFF), depending on the status ID. Please see section *Status overview*, page 25, below for further details.

Examples:

<LF>i;01;102;000000FFFF;51<CR>

Systems is operational; no errors. Status ID 102 value is FFFF.

<LF>i;04;001;0000000000;56<CR>

One or more unit is in an error condition. Invalid parameter number requested.

Usage:

This message is used to read general status and specific status info. Please see section *Status overview*, page25, below for further details on valid Status ID's and their usage.

Please notice, that if the general status changes to or from an error condition a response telegram is sent without a request command.

Parameter overview

Parameters can be set and read with the commands above *setParameter*, page 12 and *getParameter*, page 13. The following parameters are used:

Parameter ID	Parameter
101	Averaging measuring time. Measuring time in milliseconds. Default at power-on: 400.
102	Steady limit for calibration Internal loadcell value for steady detection during calibration. Default at power-on: 50
103	Communication delay Minimum time in milliseconds from start of transmission of one telegram to start of transmission of next telegram. Default at power-on: 0

Status overview

Status information can be read with the command above *getStatusInfo*, page 23.

The *General status* uses the following bits

Bit	General status
0	System is operational; no errors.
1	Units not yet detected.
2	Unit error – one or more units in an error condition.
3	*Reserved for future use*
4	Power low. Supply more the 10% below specified value.

The following status information can be read:

Status ID	Format	Status Info
101	Decimal	Number of units detected.
102	Hex 4 digits 2 bytes	Mask of detected units, e.g.: FFFF: Unit 1-16 detected. 00FF: Unit 1-8 detected.
103	Hex 4 digits 2 bytes	Unit error mask: 0000: No unit errors. 0505: Unit 1, 3, 9, 12 in error condition.
201-216	Hex 2 digits 1 bytes	Unit 1-16 error code: Bit 5: New unit installed. Bit 6: Unit controller not communicating. Bit 7: Unit not communicating.
221-236	Decimal	Unit 1-16 loadcell production year.
241-256	Decimal	Unit 1-16 serial number.
261-276	Decimal	Unit 1-16 maximum capacity in internal loadcell units.
281-296	Decimal	Unit 1-16 internal loadcell resolution: -3: 0,001 gram -2: 0,01 gram -1: 0,1 gram 0: 1 gram 1: 10 gram 2: 10 gram 3: 1000 gram

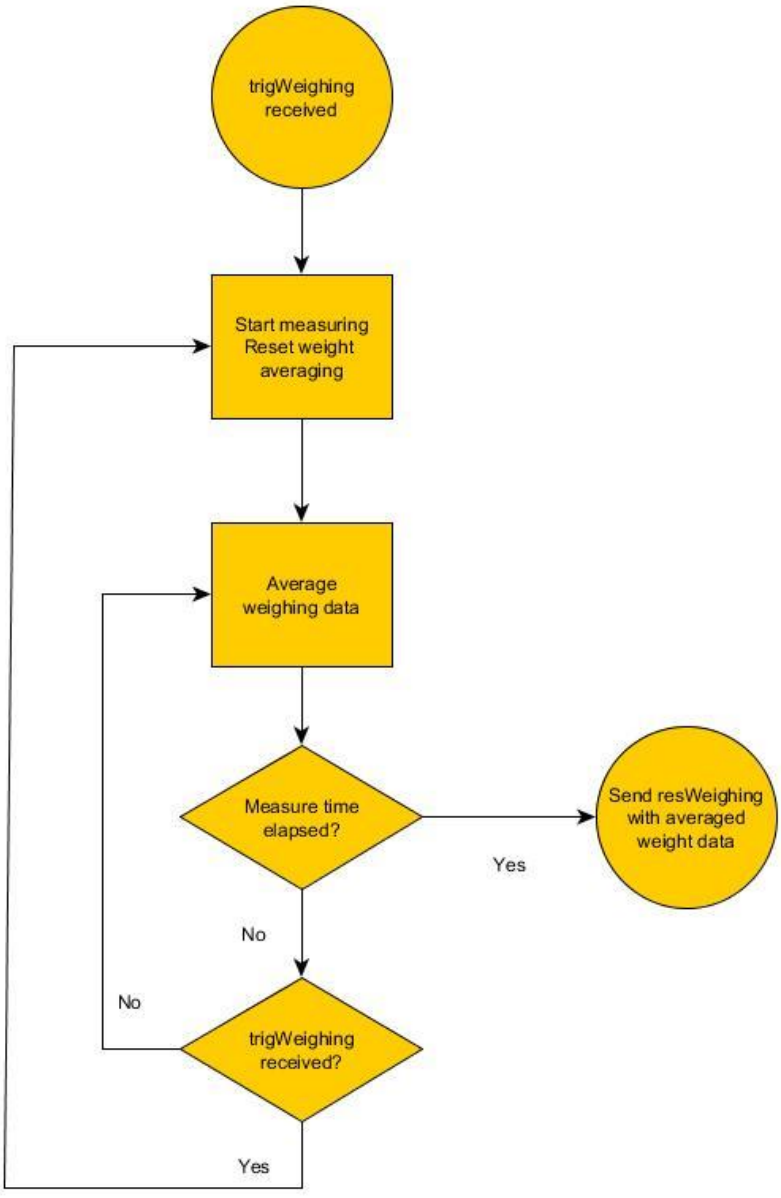
How to

– Perform a normal weighing operation

A normal weighing operation is started by a *trigWeighing* telegram, please see section trigWeighing, page 14, above for further details. The telegram identifies the unit and the measuring time. A new weighing is started immediately on the unit requested. Any number of weighings can be trigged simultaneously on different units. If a new *trigWeighing* telegram is received on a unit before a previous trigged weighing is finished, the old weighing is cancelled, no result is sent, and a new weighing is trigged immediately. When the measuring time has elapsed a *resWeighing* telegram is sent from the 5016, it is not necessary to send any kind of command to request the result. Please section resWeighing, page 15, above for further details.

Flowchart

Similar state machines are running simultaneously on all units:



- Perform a weighing analysis

A weighing analysis is started by a *trigAnalysis* telegram, please see section *trigAnalysis*, page 16, above, for further details. Three types of analysis can be started.

- Trig on weight

  An analysis is trigged on the first unit where the internal loadcell value raises with a value defined in the request.

- Trig on trigger

  An analysis is trigged on the unit defined in the request when a weighing is trigged on this unit.

- Trig instantly

  An analysis is trigged instantly on a unit defined in the request.

Please notice that only **ONE** analysis can run at a time. If a new analysis request Is received, while an analysis is started but not yet trigged, the previous analysis will be cancelled and the new analysis started. If a new analysis request is received when a previous analysis has been trigged but is not yet finished an error is raised and the new analysis will not be started.

When an analysis is trigged, a number of *resAnalysis* telegrams will be transmitted. Please see section *resAnalysis*, page 18, for further details.

Please notice that a normal weighing cycle can run simultaneously with an analysis. The analysis will contain information on whether/when a normal weighing is trigged on the same unit.

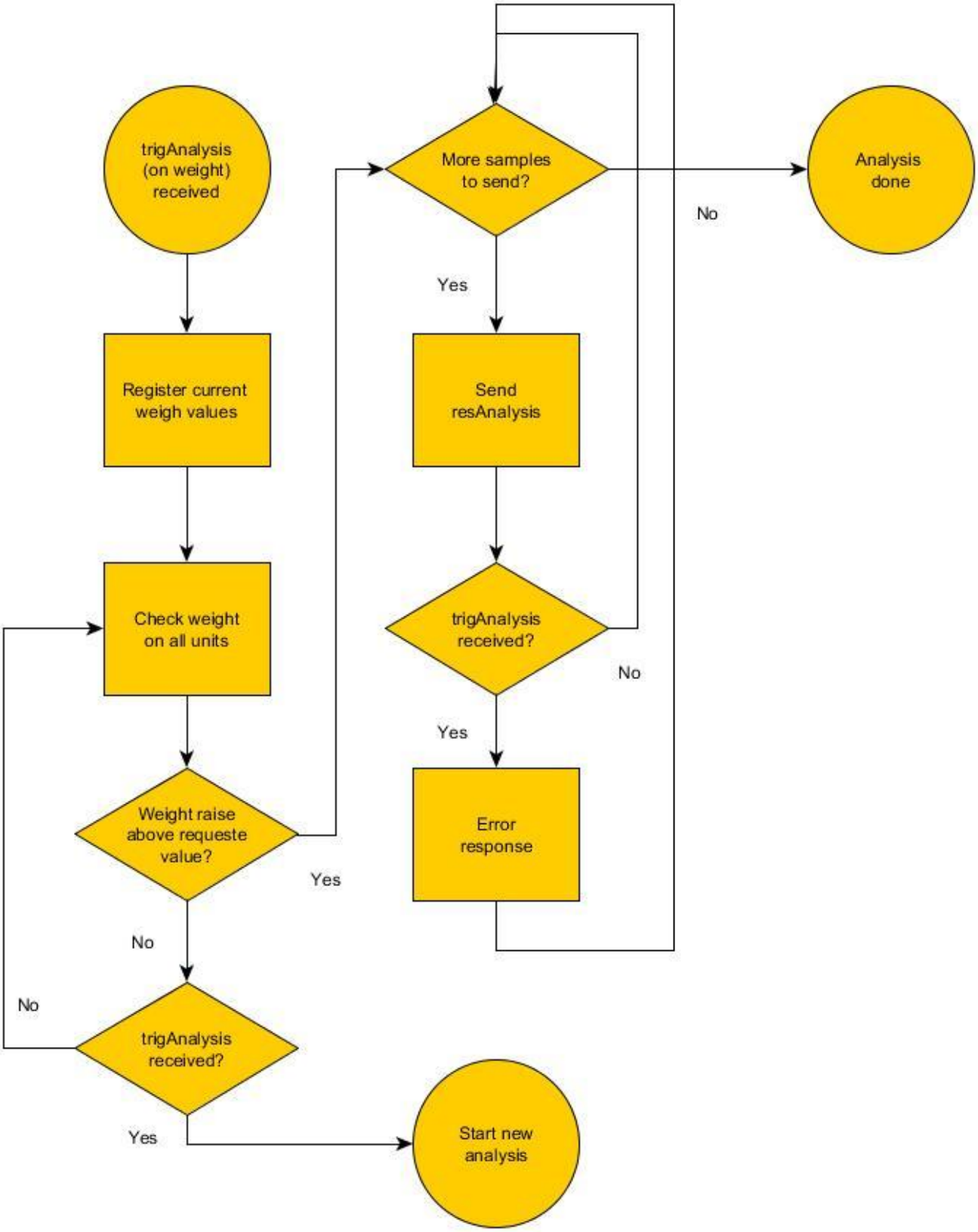
The request specifies an amount of time before the trigger point and an amount of time after the trigger point. All sampled (2 ms sample time) weight values in this interval will be transmitted in separate telegrams, along with information on unit, sequence number, and information on whether a normal weighing is trigged on the sample.

Please notice that the amount of time before the trigger point is limited to 2500ms, and the total time (sum of time before and time after the trigger point) is limited to 10000ms.

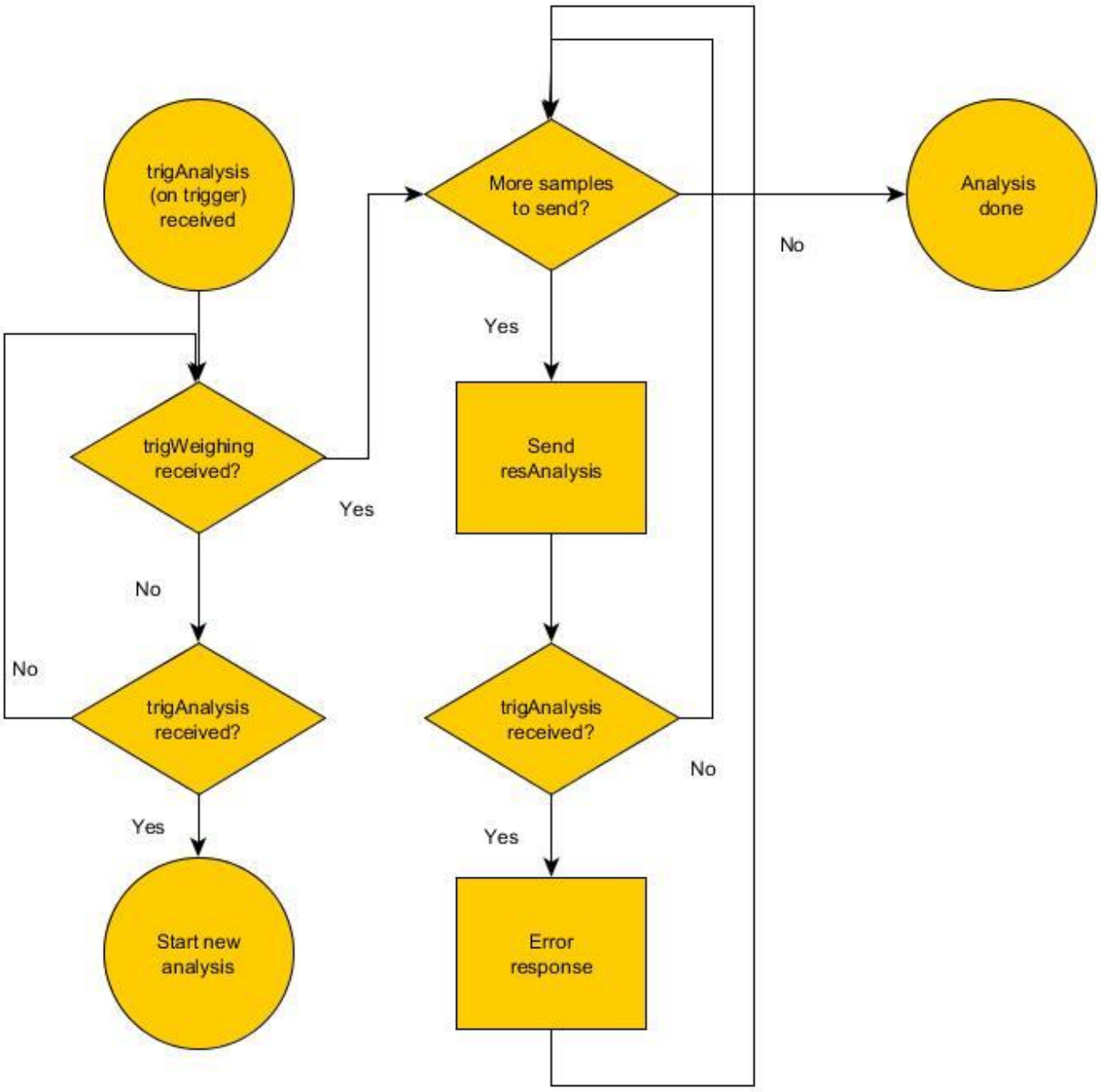
As the interval may start before the trigger point and the transmission time of the resAnalysis telegram may be longer than the sample time then transmissions may end some time after the analysis has actually ended.

Flowchart

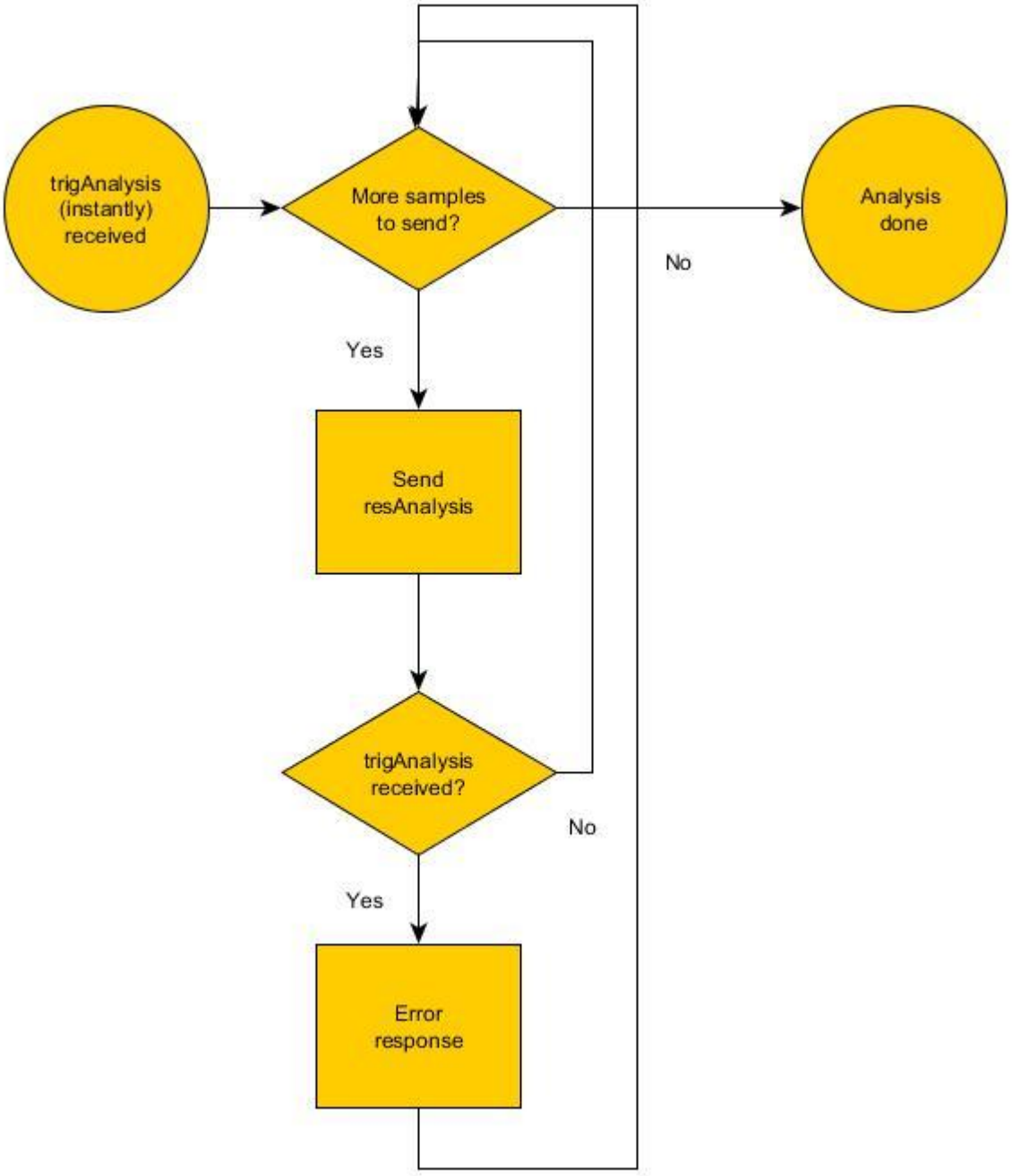
An analysis trigged by a weigh raise follows this state machine:



An analysis trigged by a normal weighing trigger follows this state machine:



An analysis trigged instantly follows this state machine:



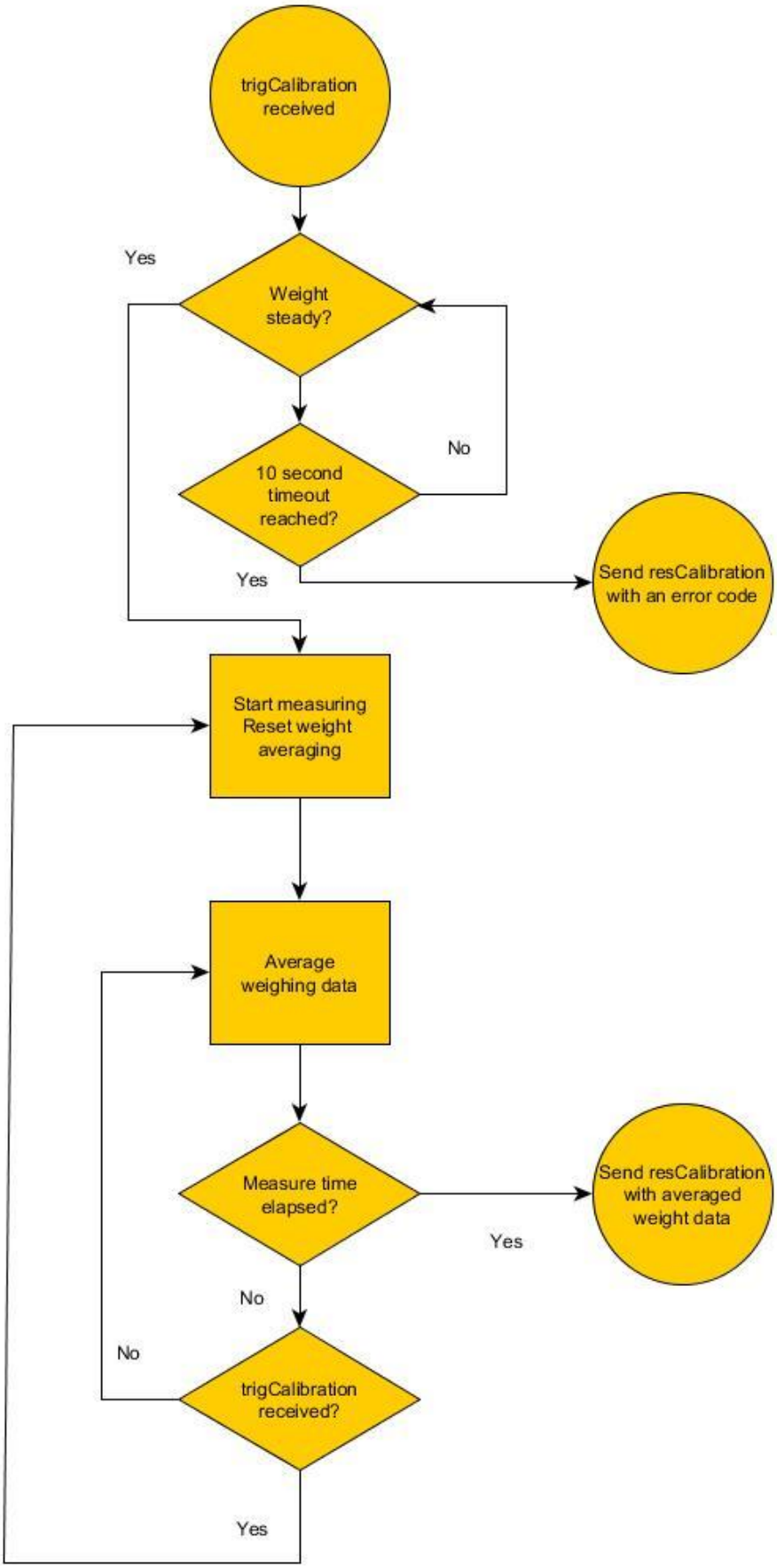
– Perform a static calibration

A static calibration operation is started by a *trigCalibration* telegram, please see section *trigCalibration*, page20, above for further details. The telegram identifies the unit and the measuring time. A new weighing is started when the weight reading is steady on the unit requested. Any number of calibrations can be trigged simultaneously on different units. If a new *trigCalibration* telegram is received on a unit before a previous trigged calibration is finished, the old calibration is cancelled, no result is sent, and a new calibration is started. When the measuring time has elapsed a *resCalibration* telegram is sent from the 5016, it is not necessary to send any kind of command to request the result. Please section *resCalibration*, page 21, above for further details.

If the weight is not steady within 10 seconds, to start the calibration a *resCalibration* telegram is sent with an error code.

Flowchart

Similar state machines are running simultaneously on all units:



– Calculate protocol checksum

The checksum is calculated as bitwise logical XOR of all preceding characters (including <LF>) printed as 2 capital ASCII hex digits.

Truth table:

Bit 1	Bit 2	Logical XOR
0 (FALSE)	0 (FALSE)	0 (FALSE)
0 (FALSE)	1 (TRUE)	1 (TRUE)
1 (TRUE)	0 (FALSE)	1 (TRUE)
1 (TRUE)	1 (TRUE)	0 (FALSE)

Example

Message preceding checksum:

<LF>w;07;-000009257;

Checksum calculation:

ASCII	Hex value	Upper nibble		Lower nibble	
		Hex	Binary	Hex	Binary
<LF>	0x0a	0	0000	A	1010
W	0x77	7	0111	7	0111
;	0x3b	3	0011	B	1011
0	0x30	3	0011	0	0000
7	0x37	3	0011	7	0111
;	0x3b	3	0011	B	1011
-	0x2d	2	0010	D	1101
0	0x30	3	0011	0	0000
0	0x30	3	0011	0	0000
0	0x30	3	0011	0	0000
0	0x30	3	0011	0	0000
0	0x30	3	0011	0	0000
9	0x39	3	0011	9	1001
2	0x32	3	0011	2	0010
5	0x35	3	0011	5	0101
7	0x37	3	0011	7	0111
Logical XOR	0x6E	6	0110	E	1110

– Use weighing data values

All weighing data are sent in internal loadcell units. Internal loadcell units are in 10^x gram where x ranges from -3 to +3, giving internal loadcell resolution from 0.001 to 1000 gram. The internal resolution is fixed in the loadcell and relates to the loadcell capacity. Loadcell resolution is fixed at a value giving a loadcell value of 300000 to 5000000 for maximum loadcell loads.

E.g. a 10 kg loadcell has an internal resolution of 0.01 gram. Maximum load is 10000 gram giving an internal load output off 1000000. A load of 7.850 kg will give a loadcell output of 785000.

The internal resolution of the loadcells connected to each unit can be read with a *getStatusInfo* telegram. Please see the details in section getStatusInfo, page23 , above.

Examples

Loadcell resolution	Load	Weighing data value
0.01	4.45621 kg	445621
0.1	4.5 kg	45000
1	4.5 kg	4500

– Connect the loadcells (TBD)

TBD

– Connect serial communication (TBD)

TBD

– Connect power (TBD)

TBD

Trouble shooting (TBD)

Problem	Solution
No connetion to device	**Power** *TBD* **Communication** TBD
No connection to loadcell	TBD

Appendices (TBD)

Appendix A – Status indicators

Indicator	Use

Appendix B – Filter specification

Sampling time		2 ms	
Filter No	Taps	Cut off frequency	Total filter time
1	7	120 Hz	14 ms
2	9	100 Hz	18 ms
3	9	120 Hz	18ms
4	12	80 Hz	24 ms
5	12	100 Hz	24 ms
6	15	80 Hz	30 ms
7	17	60 Hz	34 ms
8	21	60 Hz	42 ms
9	25	40 Hz	50 ms
10	32	40 Hz	64 ms
11	50	20 Hz	100 ms
12	64	20 Hz	128 ms
13	67	15 Hz	134 ms
14	85	15 Hz	170 ms
15	100	10 Hz	200 ms
16	150	7 Hz	300 ms
17	150	8 Hz	300 ms
18	150	9 Hz	300 ms
19	175	6 Hz	350 ms
20	175	7 Hz	350 ms
21	175	8 Hz	350 ms
22	175	9 Hz	350 ms
23	200	5 Hz	400 ms
24	200	6 Hz	400 ms
25	200	7 Hz	400 ms
26	200	8 Hz	400 ms
27	225	5 Hz	450 ms
28	225	6 Hz	450 ms
29	225	7 Hz	450 ms
30	250	4 Hz	500 ms
31	250	5 Hz	500 ms
32	250	6 Hz	500 ms

Revision History

Date	Author	Rev.	*Update*
2015-06-11	JK	1v0	*Initial document created from 5016 InstAndComm 141114-1v0e-eng.* *All ASCII telegrams no embedded in a binary format.* *Full binary dataAnalysis telegram added for faster transfer of analysis data* *Documentation to do added.*
2015-07-10	JK	1v0a	*resInit telegram added*
2015-09-07	JK	1v0b	*Binary examples added*
2015-09-29	JK	1v0c	*resInit changde to 'j'*
2015-11-06	JK	1v0d	*Trigger type added to trigWeighing and dataAnalysis telegrams*
2015-11-28	JK	1v0e	*Filter specification for build-in filters added*

To do in documentation

Flow chart for binary receive and error recovery algorithm.

Binary example telegram with and without embedded ASCII.

Checksum calculation and example for binary telegrams.

ACK/NAK flow diagram.

dataAnalysis telegram example.

dataAnalysis telegrams added to Analysis flowchart.

Troubleshooting guide.

Hardware description.

Contact

With further questions or improvement suggestions please contact us:

Eilersen
The Weighing Experts

Eilersen Electric A/S
Kokkedal Industripark 4
DK-2980 Kokkedal
Denmark
www.eilersen.com
info@eilersen.com
Tel: +45 49 180 100
Fax: +45 49 180 200