

Technical description

AquaCom – RTU level
Rev 1.6



© Copyright 2004 ITT Flygt AB. All rights reserved. No part of this manual may be reproduced or copied without the written permission of ITT Flygt AB.

ITT Flygt AB
Box 2058
291 02 Kristianstad
+46 (0)44-20 59 00

July-2004 Rev 6

This document may change without any further notice.

1 GENERAL.....	4
2 THE AQUACOM PROTOCOL.....	5
2.1 GENERAL	5
2.2 HOW TO COMMUNICATE.....	6
2.2.1 General.....	6
2.2.2 Modems	6
2.2.3 Dialled up lines.....	7
2.2.4 Fixed lines	8
2.3 THE TELEGRAM CONSTRUCTION.....	9
2.3.1 General.....	9
2.3.2 The syntax of a telegram.....	9
2.4 HOW TO CALCULATE THE CHECKSUM.....	11
2.4.1 Example of Checksum calculation.....	11
2.5 TIMEOUTS.	12
3 MESSAGES.....	13
3.1 GENERAL.	13
3.2 MESSAGE TYPES, SUMMARY.....	13
3.3 ANSWER TYPES, SUMMARY.....	14
3.3.1 Connection message.	14
3.4 MESSAGE TYPES FORTH	15
3.4.1 Alarm request (WCA).....	15
3.4.2 Function channels CS→RTU (MCC Forth).....	19
3.4.3 Function channels RTU→CS (MCC Forth).....	29
3.4.4 Analogue channels CS→RTU (MAS Forth).....	30
3.4.5 Analogue channels RTU→CS (MAS Forth).....	31
3.4.6 Alarm code filter CS→RTU (MAT Forth).....	32
3.4.7 Alarm code filter RTU→CS (MAT Forth).....	33
3.4.8 Alarm code priorities CS→RTU (SLT Forth)	34
3.4.9 Alarm code priorities RTU→CS (SLT Forth)	35
3.4.10 Status request (WGS).....	36
3.4.11 Periodical report (WGD).....	39
3.4.12 Historical trend (WGH).....	41
3.4.13 Break (BRK)	45
3.5 MESSAGE TYPES BASIC	46
3.5.2 Function channels CS→RTU (MPA, MPB, MPC, MPD, MPL, MCC).....	47
3.5.3 Analogue channels CS→RTU (MAS Basic).....	48
3.5.4 Break, with acknowledge (EBR).....	48
3.6 ANSWER MESSAGES	49
3.6.1 Telegram OK (SCO).....	50
3.6.2 Error in telegram (SCE).....	50
3.6.3 No Data Available (NDA).....	50
3.6.4 Wait	50
3.7 DETAILED EXAMPLE WITH WGH	51

1 General

This document contains information about telegrams and how to communicate with MacTec standard RTUs.

Knowledge of how communication protocols work and how to implement a protocol in a system is recommended to fully understand this document.

2 The AquaCom protocol

2.1 General

This section describes the principal way to communicate with ITT Flygt RTUs, and the differences between dial-up and fixed line.

In the document you will find different typographical printings, short names etc.

Here is a list of these

<yy> means a variable field. In this case a year.
WCA means a string constant. E.g. in the field stands the characters W, C and A (3 characters).
20H means a hex number. 20H = 32 decimal.

chan	Type	value
------	------	-------

-----repeated-----

means that data in this area is repeated with data.

CS Central System
RTU Remote Terminal Unit

2.2 How to communicate

2.2.1 General

The connection between the central system and the RTUs is utilised with a modem used with fixed or dialled line. A plain RS-232 interface may also be used.

2.2.2 Modems

V.23

The modem should be a V.23, 1200 bps, half duplex, (only A-channel used).

Hayes V.22

Some RTUs supports a standard V.22 modem, 1200 bps.

Hayes V.22bis

Some RTUs supports a standard V.22bis modem, 2400 bps.

Hayes V.28

Some RTUs supports a standard V.28 modem, 14400 bps.

Hayes V.32

Some RTUs supports a standard V.32 modem, 33600 bps.

Hayes V.90

Some RTUs supports a standard V.90 modem, 56600 bps.

Character format is always 8 data bits, 1 stop bit and No parity.

2.2.3 Dialed up lines

With a dialled line, the central system calls the RTU when communication is needed. The RTU calls the central system only when an alarm has occurred.

When an RTU dials in to a central system it or when a central system dials to an RTU the answering telegram in the remote side is an OK-telegram. Due to the fact that some modems establish contact quicker than others it is highly recommended that the answering phase in the remote side is 6*OK with 2.5 second delay between every telegram. The OK-answering sequence is aborted as soon as a telegram or request comes in to the remote side.

2.2.3.1 Example CS dialling to RTU

Dialled RTU with plant number 89 should be prompted for status, (WGS). In this example the RTU modem connects before the CS modem.

```

CS          Dialling
CS←RTU      OK # 218      (Time= 0.0)
CS←RTU      OK # 218      (Time=+2.5)
CS←RTU      OK # 218      (Time=+5.0, now the CS is connected as well)
CS←RTU      OK # 218      (Time=+7.5, CS detects OK)
CS→RTU      86 WGS P 0 ! 1 # 785
CS←RTU      18318 12549 03 48 0 0 0 0 0 14340 5004 5 6 0 0 599 1199 0 0 49754
              2500 2000 1000 750 1 0 30 0 299 242 18507 0 0 1 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 1 1 0 8050 8531 8532 # 7679

CS→RTU      86 WGS P 0 ! 1 # 785
CS←RTU      18318 12549 03 48 0 0 0 0 0 14340 5004 5 6 0 0 599 1199 0 0 49754
              2500 2000 1000 750 1 0 30 0 299 242 18507 0 0 1 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 1 1 0 8050 8531 8532 # 7679

```

Note. Not all types of RTUs support the recommended 6*OK. Instead they send 2*OK with approximately four second in between after the connect message.

2.2.3.2 Caller identification

Some central systems require that the RTU can identify itself with a unique ID and what kind of protocol the RTU is using.

To solve this problem some RTU's are capable of identifying their self with a unique Caller ID.

STX		DEST		ACT		#		CHK		CR
-----	--	------	--	-----	--	---	--	-----	--	----

STX	Start of telegram	1 byte (02H)
DEST	Destination	1 to 3 bytes.
ACT	Type of message	3 bytes
#	End of message	1 byte (23H)
CHK	Checksum	1 to 5 bytes
CR	Carriage Return	1 byte (0DH)

Please observe that some RTU's does not have this caller identification feature. If this is the case, the central system must answer incoming calls with an OK-telegram to start the conversation.

2.2.3.3 Caller identification - detailed example

Dialled RTU with plant number 9 dials in an alarm with code 12.

SOH	Start of telegram 1 byte (01H)	= <SOH> in this section
STX	Start of telegram. 1 byte (02H)	= <STX> in this section
CR	Carriage Return. 1 byte (0DH)	= <CR> in this section
Space	Space 1 byte (20H)	= _ in this section

```

RTU      Dialling
RTU      Connect
CS←RTU   <STX>_9_ACT_#_369_<CR>
CS→RTU   <SOH>_9_WCA_0_3_6_17_13_59_#_1031_<CR>
CS←RTU   <STX>_ALR_9_12_3_6_17_13_58_P_T_0_0_0_0_A_27_#_1867_<CR>
CS→RTU   <SOH>_0_ALO_#_364_<CR>
CS→RTU   <SOH>_9_WCA_0_3_6_17_14_0_#_970_<CR>
CS←RTU   <STX>_ALN_T_0_0_0_0_#_719_<CR>
CS→RTU   <SOH>_9_BRK_#_376_<CR>

```

2.2.4 Fixed lines

With a fixed line the RTUs are connected in a multidrop net where the media and the time it takes to poll the RTUs for alarms is the limit over how many RTU's there can be on one fixed line. A usual limit for a V.23 fixed line is a number between 5-10, the number is depending on modem vendor, length and quality of the line.

2.3 The telegram construction

2.3.1 General

An AquaCom telegram is an ASCII-string with support of IBM extended codes. E.g. codes 01H to FFH is supported.

2.3.2 The syntax of a telegram

2.3.2.1 Telegram from CS

----- TELEGRAM -----

----- MESSAGE -----

SOH		DEST		TYPE		DATA		#		CHK		CR
-----	--	------	--	------	--	------	--	---	--	-----	--	----

SOH	Start of telegram	1 byte (01H)
DEST	Destination	1 to 3 bytes. <i>Note that the ALO telegram differs.</i>
TYPE	Type of message	3 bytes
DATA	Message data	1 to 4000 bytes
#	End of message	1 byte (23H)
CHK	Checksum	1 to 5 bytes
CR	Carriage Return	1 byte (0DH)

Between every "cell" there is an ASCII space character (20H).

2.3.2.2 Telegram from RTU

----- TELEGRAM -----

----- ANSWER -----

STX		DATA		#		CHK		CR
-----	--	------	--	---	--	-----	--	----

STX	Start of telegram	1 byte (02H)
DATA	Answer data	1 to 4000 bytes
#	End of data	1 byte (23H)
CHK	Checksum	1 to 5 bytes
CR	Carriage Return	1 byte (0DH)

Between every "cell" there is an ASCII space character (20H).

2.4 How to calculate the checksum.

The checksum is calculated in following way:

Add together all characters after the STX or SOH to the '#' (Do not add this character), in a 16-bit signed integer.

Make the result from the absolute value of this sum.

Add a space (20H) to the string after the '#' and the characters from the checksum.

Example: The telegram SCO.

STX	SCO	#	293	CR
-----	-----	---	-----	----

Detailed view

<STX>	<i>Space</i>	<i>S</i>	<i>C</i>	<i>O</i>	<i>Space</i>	#	<i>Space</i>	2	9	3	<i>Space</i>	<CR>
	32	83	67	79	32							

$$32+83+67+79+32=293$$

(The digits are in decimal)

2.4.1 Example of Checksum calculation

```
Public Function CalcChecksum(strTelegram As String) As Long
    Dim chk As Long
    Dim i As Long
    Dim Char As Byte
    Dim WAIT As Byte: Byte = 19

    chk = 0
    For i = 1 To Len(strTelegram)
        Char = Asc(Mid(strTelegram, i, 1))
        If Char <> WAIT Then
            chk = chk + Char
        End If
    Next i

    chk = chk And 65535
    If (chk >= 32768) Then chk = 65536 - chk

    CalcChecksum = chk
End Function
```

2.5 Timeouts.

Type of communication	Timeout at CS	Timeout at RTU
Normal telegram	4-20 seconds	2 seconds
Hook-On when no telegram	----	30 seconds
Between two characters in the same telegram	----	1 second
Max time for modems to establish a connection	----	20-60 seconds depending on RTU system software

3 Messages

3.1 General.

Message and answer types mean the inner part of a telegram without the destination, checksum and end of telegram character.

3.2 Message types, summary.

AquaCom supports the following message types from CS to the RTU.

Message type	Message	Comment
Alarm request	WCA	
Function channels	MPA, MPB, MPC, MPL, MPD, MCC	MCC is the only one supported in Forth
Analogue channels	MAS	
Alarm code filter	MAT	
Status	WGS	
Periodical report	WGD	
Historical trend	WGH	
Break	BRK	
Break (with ack)	EBR	Not supported in Forth RTU's

Note that not all message types are supported in all RTUs.

3.3 Answer types, summary.

Answers from RTUs to CS can, besides variable data, be fixed answers.

Answer type	Message
Telegram OK	SCO
Error in telegram	SCE
Requested data not available	NDA
Wait	!!

3.3.1 Connection message.

OK

Used only with dialled lines.

The message is sent when

RTU	Dialling
CS → RTU	when the RTU is dialling to transmit an alarm telegram
CS	Dialling
CS ← RTU	when the CS is dialling to request communication

E.g. the answering unit is always sending an OK before communications begins.

3.4 Message types Forth

3.4.1 Alarm request (WCA)

<dest>	WCA	<set>	<YY>	<MM>	<DD>	<hh>	<mm>
--------	-----	-------	------	------	------	------	------

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed					
WCA	3 bytes,	Text					
set	1 byte,	Set time and date (0=normal, 2=Set)					
YY	2 bytes,	Year (00-99)					
MM	1-2 bytes,	Month (1-12)					
DD	1-2 bytes,	Day (1-31)					
hh	1-2 bytes,	Hour (0-24)					
mm	1-2 bytes,	Minutes (0-59)					

NOTE! When the value for parameter **set** is 2, the RTU will adjust the real time clock in the RTU to the received parameters.

3.4.1.1 Dialled line.

When the RTU calls and CS has sent OK, the RTU transmits the alarm message immediately. A WCA-request is not needed to send the first alarm message from the RTU.

3.4.1.2 Example Dialled

Dialled RTU with plant number 9 dials in an alarm with code 12.

```
RTU      Dialling
CS→RTU   OK # 218
CS←RTU   ALR 9 12 1 4 27 10 26 P T 0 0 0 0 A 27 # 1856
CS→RTU   0 ALO # 364
CS→RTU   9 WCA 0 1 4 27 10 27 # 1020
CS←RTU   ALN T 0 0 0 0 # 719
CS→RTU   9 BRK # 376
```

3.4.1.3 Example Fixed

Fixed RTU with plant number 111 and Node ID 5 answer an alarm request with alarm code 11.

```
CS→RTU   5 WCA 0 1 4 27 10 13 # 1011
CS←RTU   ALR 111 11 1 4 27 10 13 A T 0 0 0 0 B 31 # 1922
CS→RTU   5 ALO # 369
CS→RTU   5 WCA 0 1 4 27 10 13 # 1011
CS←RTU   ALN T 0 0 0 0 # 719
```

3.4.1.3.1 Example of alarm section in a Forth CFG-file

```

; ##6
;-----
; Alarmcode Initial_Priority
;-----
28   B ; Substation does not answer
8001 C ; No teleline.
8019 B ; {ND} Missing parameters
8033 C ; Incorrect tel.no
8050 B ; Setpoint changed
8090 A ; {ND} Cold start
8114 B ; {ND} Warm start
8117 B ; {ND} Modem error
8156 H ; {ND} Unsuccessful call CS
8157 H ; {ND} Call ok CS
1    B ; High level
2    B ; Low level
8533 B ; P1, High capacity
8511 B ; P2, High capacity
8514 B ; P1, Low capacity
8534 B ; P2, Low capacity
30   B ; Operation error P1
31   B ; Operation error P2
11   B ; Tripped P1
12   B ; Tripped P2
54   B ; Leakage P1

```

3.4.1.4 Alarm answer - no alarms (ALN).

ALN	T	<v1>	<v2>	<v3>	<v4>
-----	---	------	------	------	------

ALN	3 bytes,	text
T	1 byte,	text
v1	1-11 bytes,	AquaTrain wagon 1 (default zero)
v2	1-11 bytes,	AquaTrain wagon 2 (default zero)
v3	1-11 bytes,	AquaTrain wagon 3 (default zero)
v4	1-11 bytes,	AquaTrain wagon 4 (default zero)

This message is the default message when no alarm condition occurs.

The AquaTrain are functions only available with ITT Flygt MacTec software.

Please note that when the content in an AquaTrain wagon is greater than 65535 a dot is added at the end.

In following example v4 is greater than 65535: ALN T 7 14 21 86723. # 1094

3.4.1.5 Alarm answer - acknowledge.

<dest>	ALO
--------	-----

dest 1-3 bytes, 0 if dialled or
 Node ID, (0-RTU dependant), if fixed
ALO 3 bytes, text.

After a received ALR-message, the alarm must be acknowledged by transmitting an ALO-message. The alarm is then removed from the database in the RTU.

If the RTU is placed on a fixed line, the **destination** will hold the node ID. In a dial up connection this **destination** must be set to zero, (0).

Some older versions of RTUs will go ON-Hook after a received ALO. Newer versions will be waiting for a new telegram (e.g. WCA).

3.4.1.6 Alarm answer - alarm is pending (ALR).

Note: Forth system version 1.?? or greater only

ALR	<lsn>	<code>	<YY>	<MM>	<DD>	<hh>	<mm>	<flank>	
-----	-------	--------	------	------	------	------	------	---------	--

	T	<V1>	<V2>	<V3>	<V4>	<pri>	<ss>
--	---	------	------	------	------	-------	------

ALR	3 bytes,	text
lsn	1-5 bytes,	logical plant number (0-999)
code	1-5 bytes,	alarm code (0-8999)
YY	2 bytes,	year (00-99)
MM	1-2 bytes,	month (1-12)
DD	1-2 bytes,	day (1-31)
hh	1-2 bytes,	hour (0-23)
mm	1-2 bytes,	minute (0-59)
flank	1 bytes	the state of the event / alarm. Valid values are: A = Active / On / Up-flank P = Passive / Off / Down-flank
T	1 byte,	text
v1	1-11 bytes,	AquaTrain wagon 1 (default zero)
v2	1-11 bytes,	AquaTrain wagon 2 (default zero)
v3	1-11 bytes,	AquaTrain wagon 3 (default zero)
v4	1-11 bytes,	AquaTrain wagon 4 (default zero)
pri	1 byte,	Priority
ss	1-2 bytes,	second (0-59)

The AquaTrain are functions only available with ITT Flygt MacTec software.

Please note that when the content in an AquaTrain wagon is greater than 65535 a dot is added at the end.

In following example v4 is greater than 65535, (16bit): ALN T 8 16 24 86979. # 1113

3.4.2 Function channels CS→RTU (MCC Forth)

<dest>	MCC	<lsn>	<YY>	<MM>	<DD>	<hh>	<mm>	<ss>	
--------	-----	-------	------	------	------	------	------	------	--

	<chan>	<type>	<value>
--	--------	--------	---------

----- Repeated (<= 200) -----

3.4.2.1 HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
prod	3 bytes,	Name of setpoint telegram.
lsn	1-5 bytes,	logical plant number (0-999).
YY	2 bytes,	year (00-99).
MM	2 bytes,	month (01-12).
DD	2 bytes,	day (01-31).
hh	2 bytes,	hour (00-23).
mm	2 bytes,	minute (00-59).
ss	2 bytes,	second (00-59).

3.4.2.2 DATA

chan	1-3 bytes,	channel number, (1-255).
type	1 byte,	type of data that follow. A = Analogue (0-200). D = Integer (0-32767). T = Text (Data between character ' (27H)). 0-4 = Float (Data transmitted with 0-4 decimals).
value	1-128 bytes,	data as described above.

3.4.2.3 Example - Fixed communication

Fixed RTU with plant number 1 and Node ID 1 accepts a MCC telegram

```
CS➔RTU 1 MCC 1 01 04 30 11 02 59 3 T '123456789' 4 T '234567890' 9 D 2 10
T '_' 11 T '_' 12 T '_' 13 T 'My_Station_name' 14 D 0 20 D 0 23 D 13 24
D 1 25 D 10 26 D 10 27 D 30 28 D 30 29 D 30 31 2 3.00 32 2 0.00 33
2 2.80 34 2 0.50 35 2 0.10 36 D 0 38 1 10.0 39 1 8.0 40 1 4.0 41 1 1.0
43 1 10.0 44 1 8.0 45 1 4.0 46 1 1.0 48 4 1.2345 50 1 1.0 51 1 0.0 53 0
2 54 0 2 55 D 0 56 2 2.00 57 2 1.00 58 2 2.20 59 2 1.20 60 2 0.00 61 D
5 62 D 10 63 D 0 64 0 0 65 D 0 67 2 1.80 68 2 1.00 69 0 5 70 2 1.00 71
2 0.00 72 2 10.00 73 2 10.00 74 2 10.00 75 2 0.00 76 2 0.00 77 2 0.00
78 2 0.00 79 2 0.00 80 2 0.00 82 1 50.0 83 1 10.0 85 1 50.0 86 1 10.0
95 2 0.20 96 1 11.0 97 1 22.0 98 1 33.0 99 1 44.0 100 1 55.0 101 1
66.0 102 1 77.0 103 1 88.0 104 1 99.0 105 1 100.0 106 1 110.0 107 1
120.0 108 1 130.0 109 1 140.0 110 1 150.0 111 1 160.0 112 1 170.0
113 1 200.0 114 1 190.0 115 1 180.0 125 0 750 127 0 750 251 0 1 #
24268
CS←RTU SCO # 293
```

Comment: The result could be a SCE telegram as well if the RTU does not accept the contents in the send telegram. A SCE answer for a MCC telegram might contain the channel number the RTU rejected.

Example:

```
CS←RTU SCE 48 # 423
```

3.4.2.3.1 Example of MCC section in a Forth CFG-file

```

;-----
; Number of channels for setup (0-199)
;-----
88
;-----
; Data for setup
;
; Channel Type (A/B/C/D/S/T/I/n) Min Max Text Setpoint Time
;-----
; ##0

;G * R_4126 *
3 C 16.000 16.000 "Telephone number 1 MTC-COM/PAGER " 0.000 00:00:00
4 C 16.000 16.000 "Telephone number 2 MTC-COM/PAGER " 0.000 00:00:00
251 0 0.000 30.000 "Response delay substation. (s) " 0.000 00:00:00
9 D 0.000 3.000 "PSys (0=MC num,1=MC text,2=Telia,3=Euro) " 2.000 00:00:00
10 C 16.000 16.000 "Paging - telno. to PAD/SMSC " 0.000 00:00:00
11 C 16.000 16.000 "Paging - TX number " 0.000 00:00:00
12 C 16.000 16.000 "Paging - password " 0.000 00:00:00
13 C 16.000 16.000 "Station name " 0.000 00:00:00
14 D 0.000 4.000 "Program mode RTU (0=Run) " 0.000 00:00:00
20 I 000 016 "Digital inputs " 0.000 00:00:00
;G Alarm log: ###
23 I 000 004 "Alarmdistribution (ABCP) " 13.000 00:00:00
24 D 0.000 2.000 "Alarm status (0=Local 1=Remote) " 0.000 00:00:00
25 S 0.000 5999.000 "Digital alarm delay (mm:ss) " 10.000 00:00:00
26 S 0.000 5999.000 "Analogue alarm delay (mm:ss) " 10.000 00:00:00
27 S 0.000 5999.000 "Alarm delay power fail (mm:ss) " 30.000 00:00:00
28 S 0.000 480.000 "Max work time (hh:mm) " 0.000 00:00:00
29 S 0.000 900.000 "Warning time (mm:ss) " 0.000 00:00:00
;G LEVEL (CALIB.)
31 2 0.000 99.990 "Max level (m) " 3.000 00:00:00
32 2 -99.990 99.990 "Min level (m) " 0.000 00:00:00
33 2 0.000 99.990 "High level (m) " 0.000 00:00:00
34 2 0.000 99.990 "Low level (m) " 0.000 00:00:00
35 2 0.000 10.000 "Level alarm hysteresis (m) " 0.000 00:00:00
36 S 0.000 1080.000 "Sensor control (hh:mm) " 0.000 00:00:00
;G CURRENT P1
38 1 0.000 999.900 "Max current P1 (A) " 0.000 00:00:00
39 1 0.000 999.900 "High current P1 (A) " 0.000 00:00:00
40 1 0.000 999.900 "Low current P1 (A) " 0.000 00:00:00
41 1 0.000 99.900 "Current alarm hysteresis P1 (A) " 0.000 00:00:00
;G CURRENT P2
43 1 0.000 999.900 "Max current P2 (A) " 0.000 00:00:00
44 1 0.000 999.900 "High current P2 (A) " 0.000 00:00:00
45 1 0.000 999.900 "Low current P2 (A) " 0.000 00:00:00
46 1 0.000 999.900 "Current alarm hysteresis P2 (A) " 0.000 00:00:00
;G ENERGY
48 4 0.0000 99.9999 "Energy (kWh/pulse) " 1.2345 00:00:00

```

22 - Messages

AquaCom Technical description

```

;G RAIN
50 1 0.000 99.900 "Rain/pulse (mm) " 1.000 00:00:00
51 1 0.000 99.000 "Rain alarm (mm/5min) " 0.000 00:00:00
;G PUMP CONTROL...
53 0 0.000 2.000 "Max running pumps " 2.000 00:00:00
54 0 0.000 2.000 "Number of alternating pumps " 2.000 00:00:00
55 D 0.000 4.000 "Control option (0 = Alternation) " 0.000 00:00:00
56 2 0.000 99.990 "Start level 1 (m) " 0.000 00:00:00
57 2 0.000 99.990 "Stop level 1 (m) " 0.000 00:00:00
58 2 0.000 99.990 "Start level 2 (m) " 0.000 00:00:00
59 2 0.000 99.990 "Stop level 2 (m) " 0.000 00:00:00
60 2 0.000 9.990 "Random start span (m) " 0.000 00:00:00
61 S 0.000 900.000 "Start delay (mm:ss) " 10.000 00:00:00
62 S 0.000 900.000 "Stop delay (mm:ss) " 5.000 00:00:00
63 S 0.000 3599.000 "Hi level run (mm:ss) " 0.000 00:00:00
64 0 0.000 99.000 "No of pumpstops flushvalve " 0.000 00:00:00
65 S 0.000 3599.000 "Flushtime (mm:ss) " 0.000 00:00:00
;G CAPACITY...
67 2 0.000 99.990 "Upper level cap. calc. (m) " 0.000 00:00:00
68 2 0.000 99.990 "Lower level cap. calc. (m) " 0.000 00:00:00
69 0 0.000 20.000 "No. of values in cap. calc. (0-20) " 5.000 00:00:00
70 2 0.500 1.000 "Pump ratio: 2 " 1.000 00:00:00
71 2 0.000 99.990 "Level 1 (m) " 0.000 00:00:00
72 2 0.000 999.990 "Area 1 (m2) " 0.000 00:00:00
73 2 0.000 99.990 "Level 2 (m) " 0.000 00:00:00
74 2 0.000 999.990 "Area 2 (m2) " 0.000 00:00:00
75 2 0.000 99.990 "Level 3 (m) " 0.000 00:00:00
76 2 0.000 999.990 "Area 3 (m2) " 0.000 00:00:00
77 2 0.000 99.990 "Level 4 (m) " 0.000 00:00:00
78 2 0.000 999.990 "Area 4 (m2) " 0.000 00:00:00
79 2 0.000 99.990 "Level 5 (m) " 0.000 00:00:00
80 2 0.000 999.990 "Area 5 (m2) " 0.000 00:00:00
;G CALC CAP. P1
82 1 0.000 9999.900 "Nominal capacity P1 (l/s) " 0.000 00:00:00
83 1 0.000 9999.900 "Capacity divergation limit P1 (l/s) " 0.000 00:00:00
;G CALC CAP. P2
85 1 0.000 9999.900 "Nominal capacity P2 (l/s) " 0.000 00:00:00
86 1 0.000 9999.900 "Capacity divergation limit P2 (l/s) " 0.000 00:00:00

```

;G FLOWS...

95	2	0.000	99.990	"Overflow range	(m) "	0.000	00:00:00
96	1	0.000	9999.900	"Overflow increment 1	(l/s) "	0.000	00:00:00
97	1	0.000	9999.900	"Overflow increment 2	(l/s) "	0.000	00:00:00
98	1	0.000	9999.900	"Overflow increment 3	(l/s) "	0.000	00:00:00
99	1	0.000	9999.900	"Overflow increment 4	(l/s) "	0.000	00:00:00
100	1	0.000	9999.900	"Overflow increment 5	(l/s) "	0.000	00:00:00
101	1	0.000	9999.900	"Overflow increment 6	(l/s) "	0.000	00:00:00
102	1	0.000	9999.900	"Overflow increment 7	(l/s) "	0.000	00:00:00
103	1	0.000	9999.900	"Overflow increment 8	(l/s) "	0.000	00:00:00
104	1	0.000	9999.900	"Overflow increment 9	(l/s) "	0.000	00:00:00
105	1	0.000	9999.900	"Overflow increment 10	(l/s) "	0.000	00:00:00
106	1	0.000	9999.900	"Overflow increment 11	(l/s) "	0.000	00:00:00
107	1	0.000	9999.900	"Overflow increment 12	(l/s) "	0.000	00:00:00
108	1	0.000	9999.900	"Overflow increment 13	(l/s) "	0.000	00:00:00
109	1	0.000	9999.900	"Overflow increment 14	(l/s) "	0.000	00:00:00
110	1	0.000	9999.900	"Overflow increment 15	(l/s) "	0.000	00:00:00
111	1	0.000	9999.900	"Overflow increment 16	(l/s) "	0.000	00:00:00
112	1	0.000	9999.900	"Overflow increment 17	(l/s) "	0.000	00:00:00
113	1	0.000	9999.900	"Overflow increment 18	(l/s) "	0.000	00:00:00
114	1	0.000	9999.900	"Overflow increment 19	(l/s) "	0.000	00:00:00
115	1	0.000	9999.900	"Overflow increment 20	(l/s) "	0.000	00:00:00

;G SERVICE INTERVAL

125	0	0.000	99999.000	"P1 service interval	(h) "	2.778	00:00:00
127	0	0.000	99999.000	"P2 service interval	(h) "	2.778	00:00:00

3.4.2.3.2 Example of Setpoint, (MCC), printout from AquaView

Setpoints

```

* R_4126 *
(003) Telephone number 1 MTC-COM/PAGER      123456789
(004) Telephone number 2 MTC-COM/PAGER      234567890
(251) Response delay substation.            (s) 1
(009) PSys (0=MC num,1=MC text,2=Telia,3=Euro) 2
(010) Paging - telno. to PAD/SMSC
(011) Paging - TX number
(012) Paging - password
(013) Station name                          My Station name
(014) Program mode RTU                      (0=Run) 0
(020) Digital inputs                        0000000000000000

```

Alarm log:

```

(023) Alarmdistribution                     (ABCP) 1101
(024) Alarm status                         (0=Local 1=Remote) 1
(025) Digital alarm delay                  (mm:ss) 00:10
(026) Analogue alarm delay                 (mm:ss) 00:10
(027) Alarm delay power fail               (mm:ss) 00:30
(028) Max work time                       (hh:mm) 00:30
(029) Warning time                        (mm:ss) 00:30

```

LEVEL (CALIB.)

```

(031) Max level                           (m) 3.00
(032) Min level                           (m) 0.00
(033) High level                          (m) 2.80
(034) Low level                           (m) 0.50
(035) Level alarm hysteresis               (m) 0.10
(036) Sensor control                      (hh:mm) 00:00

```

CURRENT P1

```

(038) Max current P1                      (A) 10.0
(039) High current P1                     (A) 8.0
(040) Low current P1                      (A) 4.0
(041) Current alarm hysteresis P1         (A) 1.0

```

CURRENT P2

```

(043) Max current P2                      (A) 10.0
(044) High current P2                     (A) 8.0
(045) Low current P2                      (A) 4.0
(046) Current alarm hysteresis P2         (A) 1.0

```

ENERGY

```

(048) Energy                             (kWh/pulse) 1.2345

```

RAIN

```

(050) Rain/pulse                          (mm) 1.0
(051) Rain alarm                          (mm/5min) 0.0

```

PUMP CONTROL...

(053) Max running pumps	2
(054) Number of alternating pumps	2
(055) Control option	(0 = Alternation) 0
(056) Start level 1	(m) 2.00
(057) Stop level 1	(m) 1.00
(058) Start level 2	(m) 2.20
(059) Stop level 2	(m) 1.20
(060) Random start span	(m) 0.00
(061) Start delay	(mm:ss) 00:05
(062) Stop delay	(mm:ss) 00:10
(063) Hi level run	(mm:ss) 00:00
(064) No of pumpstops flushvalve	0
(065) Flushtime	(mm:ss) 00:00

CAPACITY...

(067) Upper level cap. calc.	(m) 1.80
(068) Lower level cap. calc.	(m) 1.00
(069) No. of values in cap. calc.	(0-20) 5
(070) Pump ratio: 2	1.00
(071) Level 1	(m) 0.00
(072) Area 1	(m2) 10.00
(073) Level 2	(m) 10.00
(074) Area 2	(m2) 10.00
(075) Level 3	(m) 0.00
(076) Area 3	(m2) 0.00
(077) Level 4	(m) 0.00
(078) Area 4	(m2) 0.00
(079) Level 5	(m) 0.00
(080) Area 5	(m2) 0.00

CALC CAP. P1

(082) Nominal capacity P1	(l/s) 50.0
(083) Capacity divergation limit P1	(l/s) 10.0

CALC CAP. P2

(085) Nominal capacity P2	(l/s) 50.0
(086) Capacity divergation limit P2	(l/s) 10.0

FLOWS...

(095) Overflow range	(m) 0.20
(096) Overflow increment 1	(l/s) 11.0
(097) Overflow increment 2	(l/s) 22.0
(098) Overflow increment 3	(l/s) 33.0
(099) Overflow increment 4	(l/s) 44.0
(100) Overflow increment 5	(l/s) 55.0
(101) Overflow increment 6	(l/s) 66.0
(102) Overflow increment 7	(l/s) 77.0
(103) Overflow increment 8	(l/s) 88.0
(104) Overflow increment 9	(l/s) 99.0
(105) Overflow increment 10	(l/s) 100.0
(106) Overflow increment 11	(l/s) 110.0
(107) Overflow increment 12	(l/s) 120.0
(108) Overflow increment 13	(l/s) 130.0
(109) Overflow increment 14	(l/s) 140.0
(110) Overflow increment 15	(l/s) 150.0
(111) Overflow increment 16	(l/s) 160.0
(112) Overflow increment 17	(l/s) 170.0
(113) Overflow increment 18	(l/s) 200.0
(114) Overflow increment 19	(l/s) 190.0
(115) Overflow increment 20	(l/s) 180.0

SERVICE INTERVAL

(125) P1 service interval	(h) 750
(127) P2 service interval	(h) 750

3.4.2.3.3 Example - Dissection of telegram contents and CFG file

```
CFG 3 C 16.000 16.000 "Telephone number 1 MTC-COM/PAGER" 0.000 00:00:00
CS (003) Telephone number 1 MTC-COM/PAGER 123456789
MCC 3 T '123456789'
```

Comment: The C type in the CFG file is send as T.

Note 1: Empty strings are send as '_'

Note 2: Space in strings are send as underscore 'TEXT_WITH_SPACE'

Note 3: Min and Max should show how long texts the RTU accepts.

Note 4: Some RTU's are capable of sending and receiving data in UTF8 format.

Please note that it is only the data part of the telegram that is send as UTF8.

In order to send/ receive a UTF8 MCC telegram from a Forth RTU the ATF/CFG file must look something like this:

```
-----
;
; Type of controler      1, 11, 21 = MPA
;                        2, 12, 22 = MPB
;                        3, 13, 23 = MPC
;                        4, 14, 24 = MPD
;                        5, 15, 25 = MCC-180, MRD
;                        6, 16, 26 = MCC-180 with MAS-180
;                        7, 17, 27 = MPL, MTCCOM-4
;                        100, 100, 100 = Station built with Maestro 1.01+
;                        101, 101, 101 = Station built with Maestro 1.10+ (events)
-----
```

101 001

It is the last number, (010), that determines if it is possible to fetch alarm priorities or not from the RTU.

Position 1 = Possible to fetch alarm priorities

Position 2 = Possible to fetch alarm code filters

Position 3 = Possible to send/receive UTF8 in a MCC telegram

```
CFG 9 D 0.000 3.000 "PSys (0=MC num,1=MC text,2=Telia,3=Euro)" 2.000 00:00:00
CS (009) PSys (0=MC num,1=MC text,2=Telia,3=Euro) 2
MCC 9 D 2
```

Comment: The D type in the CFG file is send as D.

```
CFG 20 I 000 016 "Digital inputs" 0.000 00:00:00
CS (020) Digital inputs 0000000000000000
MCC 20 D 0
```

Comment: The I type, (binary), in the CFG file is send as D.

Note 1: Min and Max should show how many bits to use.

CFG 25 S 0.000 599.0 "Digital alarm delay (mm:ss)" 10.000 00:00:00

CS (025) Digital alarm delay (mm:ss) 00:10

MCC 25 D 10

Comment: The S type, (Time), in the CFG file is send as D. Please note that there are no difference in the telegram if the time is send in minutes or in seconds. The "formula" is 60*"Large time" + small time.

CFG 31 2 0.000 99.990 "Max level (m)" 3.000 00:00:00

CS (031) Max level (m) 3.00

MCC 31 2 3.00

Comment: The 2 type means two decimals.

CFG 38 1 0.000 999.900 "Max current P1 (A)" 0.000 00:00:00

CS (038) Max current P1 (A) 10.0

MCC 38 1 10.0

Comment: The 1 type means one decimal.

CFG 48 4 0.0000 99.9999 "Energy (kWh/pulse)" 1.2345 00:00:00

CS (048) Energy (kWh/pulse) 1.2345

MCC 48 4 1.2345

Comment: The 4 type means four decimals.

CFG 53 0 0.000 2.000 "Max running pumps " 2.000 00:00:00

CS (053) Max running pumps 2

MCC 53 0 2

Comment: The 0 type means no decimals. This is actually the same as the D type

3.4.3 Function channels RTU→CS (MCC Forth)

<dest>	MCC	<lsn>	<YY>
--------	-----	-------	------

3.4.3.1 HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
prod	3 bytes,	Name of setpoint telegram.
lsn	1-5 bytes,	must be 0, (zero).
YY	2 bytes,	must be 1, (one).

3.4.3.2 Example Dialled RTU

Dialled RTU with plant number 9 receives a request for MCC settings

```
CS          Dialling
CS←RTU      OK # 218
CS→RTU      9 MCC 0 1 # 525
CS←RTU      3 T 'T817' 4 T " 7 D 2 8 T " 9 T " 10 T " 11 T " 12 D 0 18 D 1024 21 D
13 22 D 1 23 D 0 24 D 0 25 D 0 26 D 0 27 D 0 29 2 3.00 30 2 0.00 31
2 0.00 32 2 0.00 33 2 0.00 34 D 0 36 3 0.100 38 1 1.0 39 1 0.0 41 0 4
42 0 4 43 D 0 44 2 0.00 45 2 0.00 46 2 0.00 47 2 0.00 48 2 0.00 49 2
0.00 50 2 0.00 51 2 0.00 52 2 0.00 53 D 0 54 D 0 55 D 0 56 0 0 57 D 0
59 2 0.00 60 2 0.00 61 0 5 62 2 1.00 63 2 1.00 64 2 1.00 65 2 0.00 66 2
0.00 67 2 0.00 68 2 0.00 69 2 0.00 70 2 0.00 71 2 0.00 72 2 0.00 73 2
0.00 74 2 0.00 76 1 0.0 77 1 0.0 79 1 0.0 80 1 0.0 82 1 0.0 83 1 0.0 85
1 0.0 86 1 0.0 95 2 0.00 96 1 0.0 97 1 0.0 98 1 0.0 99 1 0.0 100 1 0.0
101 1 0.0 102 1 0.0 103 1 0.0 104 1 0.0 105 1 0.0 106 1 0.0 107 1 0.0
108 1 0.0 109 1 0.0 110 1 0.0 111 1 0.0 112 1 0.0 113 1 0.0 114 1 0.0
115 1 0.0 126 0 3 128 0 3 130 0 3 132 0 3 251 0 0 # 28910
```

Comment: In order to fetch a MCC telegram from a Forth RTU you must first send one. This means that it is not possible to fetch settings from a cold started RTU.

Important 1: Please note that the RTU only sends the data part of the telegram.

Important 2: Up till now, (2002-08-30), it is only possible to fetch the last sent setpoint telegram from the RTU. This means that if we have 100 channels to send, but we only send the first 50 channels it is NOT possible to fetch the second half of the setpoint telegram. This problem will hopefully be solved in the future, but today you must send what you would like to receive.

3.4.4 Analogue channels CS→RTU (MAS Forth)

<dest>	MAS	<input>	<signal>	<high>	<low>	<hyst>	<dec>
--------	-----	---------	----------	--------	-------	--------	-------

----- repeated (<= 96) -----

3.4.4.1 HEADER

dest 1-3 bytes, Logical plant number, (1-999), if dialled or
Node ID, (0-RTU dependant), if fixed
MAS 3 bytes, text.

3.4.4.2 DATA

input 1-2 bytes, input number (1-96).
signal 1 bytes, type of signal (0=0-20mA, 4=4-20mA).
high 1-3 bytes, level high-alarm
low 1-3 bytes, level low-alarm
hyst 1-3 bytes, hystereses before alarm can be reactivated
dec 1 byte, Number of decimals for this signal

3.4.4.3 Example Fixed

Fixed RTU with plant number 102 and Node ID 2 accepts a MAS telegram

CS→RTU 2 MAS 1 4 2.80 0.50 0.10 2 2 4 0.0 0.0 0.0 1 3 4 0.0 0.0 0.0 1 4 0 0 0 0
 0 5 0 0 0 0 0 6 0 0 0 0 0 # 4294
CS←RTU SCO # 293

The result could be a SCE telegram as well if the RTU did not accept the contents in the send telegram.

3.4.4.4 Example of MAS section in a Forth CFG-file

```
; ##1
;-----
; Number of inputs on MAS-180 (Only used with station type 6, 100 and 101)
;-----
3
;-----
; Input Dummy(0) Min Max Text High Low Hystereses Decimals
; The dummy must always be 0 with station type 100 and 101
;-----
1    0   0.00    9999.00   "Level        (m)"   9999.00   0.00   0.00   2
2    0   0.00    9999.00   "P1 Current (A)"   9999.00   0.00   0.00   1
3    0   0.00    9999.00   "P2 Current (A)"   9999.00   0.00   0.00   1
```

3.4.5 Analogue channels RTU→CS (MAS Forth)

<dest>	MAS	<input>	<signal>
--------	-----	---------	----------

----- repeated (<= 96) -----

3.4.5.1 HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
MAS	3 bytes,	text.

3.4.5.2 DATA

input	1-2 bytes,	input number must be 0, (zero).
signal	1 bytes,	type of signal must be set to 1, (one). Note that 1 is not a valid value when sending the MAS telegram to the RTU.

3.4.5.3 Example Fixed

Fixed RTU with plant number 102 and Node ID 2 gets a request for MAS settings

CS→RTU 2 MAS 0 1 # 532

CS←RTU 1 4 2.80 0.50 0.10 2 2 4 0.0 0.0 0.0 1 3 4 0.0 0.0 0.0 1 4 0 0 0 0 5 0 0
0 0 0 6 0 0 0 0 # 3955

Comment: In order to fetch a MAS telegram from a Forth RTU you must first send one. This means that it is not possible to fetch settings from a cold started RTU.

Important: Please note that the RTU only sends the data part of the telegram.

3.4.5.4 Example of MAS section in a Forth CFG-file

```
; ##1
;-----
; Number of inputs on MAS-180 (Only used with station type 6, 100 and 101)
;-----
3
;-----
; Input Dummy(0) Min Max Text High Low Hystereses Decimals
; The dummy must always be 0 with station type 100 and 101
;-----
1  0  0.00  9999.00  "Level      (m)"  9999.00  0.00  0.00  2
2  0  0.00  9999.00  "P1 Current (A)"  9999.00  0.00  0.00  1
3  0  0.00  9999.00  "P2 Current (A)"  9999.00  0.00  0.00  1
```

3.4.6 Alarm code filter CS→RTU (MAT Forth)

<dest>	MAT	<old>	<new>
--------	-----	-------	-------

----- repeated (<=20) -----

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
MAT	3 bytes,	text.
old	1-5 bytes,	old alarm code (1-8999).
new	1-5 bytes,	new alarm code which should be generated instead of old.

3.4.6.1 Example Fixed

Fixed RTU with plant number 102 and Node ID 2 receive telegram for alarm code filter.

CS→RTU 2 MAT 7 7 81 81 82 9003 # 1194

CS←RTU SCO # 293

The result could be a SCE telegram as well if the RTU did not accept the contents in the send telegram.

3.4.6.2 Example Dialled

Dialled RTU with plant number 9 receives a telegram for alarm code filter.

CS Dialling

CS←RTU OK # 218

CS→RTU 9 MAT 7 9001 # 700

CS←RTU SCO # 293

CS→RTU 9 BRK # 376

The result could be a SCE telegram as well if the RTU did not accept the contents in the send telegram.

3.4.6.3 Example of alarm code filter file

7 7

Comment: There is no actual information in the CFG file for this kind of telegram. The information is usually stored in a file called ANLxxxx.LKF, where xxxx is the logical plant number.

It is not necessary to implement this telegram.

3.4.7 Alarm code filter RTU→ CS (MAT Forth)

<dest>	MAT	<lsn>	<YY>
--------	-----	-------	------

3.4.7.1 HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
MAT	3 bytes,	text. (Name of telegram)
lsn	1-5 bytes,	must be 0, (zero).
YY	2 bytes,	must be 1, (one).

3.4.7.2 Example Dialled RTU

Dialled RTU with plant number 9 receives a request for alarm code filter settings

```
CS      Dialling
CS←RTU  OK # 218
CS→RTU  9 MAT 0 1 # 557
CS←RTU  7 9001 # 353
```

Comment: In order to fetch a MAT telegram from a Forth RTU the ATF/CFG file must look something like this:

```
-----
;
;
; Type of controler   1,  11,  21 = MPA
;                    2,  12,  22 = MPB
;                    3,  13,  23 = MPC
;                    4,  14,  24 = MPD
;                    5,  15,  25 = MCC-180, MRD
;                    6,  16,  26 = MCC-180 with MAS-180
;                    7,  17,  27 = MPL, MTCCOM-4
;                    100, 100, 100 = Station built with Maestro 1.01+
;                    101, 101, 101 = Station built with Maestro 1.10+ (events)
;
-----

101 010
```

It is the last number, (010), that determines if it is possible to fetch alarm priorities or not from the RTU.

Position 1 = Possible to fetch alarm priorities

Position 2 = Possible to fetch alarm code filters

Position 3 = Possible to send/receive UTF8 in a MCC telegram

3.4.8 Alarm code priorities CS→RTU (SLT Forth)

<dest>	SLT	<code>	<prio>
--------	-----	--------	--------

-- repeated for all alarms ---

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
SLT	3 bytes,	text.
code	1-5 bytes,	old alarm code (1-8999).
prio	1 byte,	the alarm priority as A-Z. Ordinary priorities are: A = Alarms that will be sent to pager. B = Normal alarms. C = Alarms that you do not care about. D = Time controlled A-alarms H = Events are treated as an alarm, The RTU must be configured to handle events before this feature will work.

3.4.8.1 Example Fixed

Fixed RTU with plant number 102 and Node ID 2 receive telegram for alarm priorities.

```
CS→RTU  2 SLT 1 B 2 B 3 B 4 A 5 A 7 B 11 B 12 B 15 B 16 B 17 B 18 B 28 B 30 B
          31 B 34 A 35 B 36 B 40 B 54 B 55 B 72 B 8001 C 8019 B 8033 C 8050 B
          8090 A 8114 B 8117 B 8156 H 8157 H 8194 B 8200 B 8201 B 8202 B
          8203 B 8204 B 8205 B 8210 B 8211 B 8505 B 8511 B 8514 B 8531 B
          8532 B 8533 B 8534 B 8540 B 8542 B # 14331
CS←RTU  SCO # 293
```

The result could be a SCE telegram as well if the RTU did not accept the contents in the send telegram.

3.4.8.2 Example Dialled

Dialled RTU with plant number 9 receives a telegram for alarm priorities.

```
CS      Dialling
CS←RTU  OK # 218
CS→RTU  9 SLT 1 B 2 B 3 B 4 A 5 A 7 B 11 B 12 A 13 A 14 A 28 B 30 A 31 A 32 A
          33 A 34 A 40 A 72 B 8001 C 8019 B 8033 C 8050 B 8090 A 8114 B
          8117 B 8156 H 8157 H 8194 B 8200 B 8201 B 8202 B 8203 B 8204 B
          8205 B 8210 B 8211 B 8505 B 8510 B 8511 B 8512 B 8513 B 8514 B
          8515 B 8516 B 8517 B 8518 B 8519 B 8531 B 8532 B 8540 B 8542 B #
          15425
CS←RTU  SCO # 293
CS→RTU  9 BRK # 376
```

The result could be a SCE telegram as well if the RTU did not accept the contents in the send telegram.

3.4.9 Alarm code priorities RTU→CS (SLT Forth)

<dest>	SLT	<lsn>	<YY>
--------	-----	-------	------

3.4.9.1 HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
SLT	3 bytes,	text. (Name of telegram)
lsn	1-5 bytes,	must be 0, (zero).
YY	2 bytes,	must be 1, (one).

3.4.9.2 Example Dialled RTU

Dialled RTU with plant number 9 receives a request for alarm priority settings

CS Dialling

CS←RTU OK # 218

CS→RTU 9 SLT 0 1 # 557

CS←RTU 1 B 2 B 3 B 4 A 5 A 7 B 11 B 12 A 13 A 14 A 28 B 30 A 31 A 32 A 33 A
 34 A 40 A 72 B 8001 C 8019 B 8033 C 8050 B 8090 A 8114 B 8117 B
 8156 H 8157 H 8194 B 8200 B 8201 B 8202 B 8203 B 8204 B 8205 B
 8210 B 8211 B 8505 B 8510 B 8511 B 8512 B 8513 B 8514 B 8515 B
 8516 B 8517 B 8518 B 8519 B 8531 B 8532 B 8540 B 8542 B # 15061

Comment: In order to fetch a SLT telegram from a Forth RTU the ATF/CFG file must look something like this:

```

;-----
;
; Type of controler   1,   11,   21 = MPA
;                   2,   12,   22 = MPB
;                   3,   13,   23 = MPC
;                   4,   14,   24 = MPD
;                   5,   15,   25 = MCC-180, MRD
;                   6,   16,   26 = MCC-180 with MAS-180
;                   7,   17,   27 = MPL, MTCCOM-4
;                   100, 100, 100 = Station built with Maestro 1.01+
;                   101, 101, 101 = Station built with Maestro 1.10+ (events)
;-----

```

101 100

It is the last number, (100), that determines if it is possible to fetch alarm priorities or not from the RTU.

Position 1 = Possible to fetch alarm priorities

Position 2 = Possible to fetch alarm code filter

Position 3 = Possible to send/receive UTF8 in a MCC telegram

3.4.10 Status request (WGS)

<dest>	WGS	P	<mash>	<dir>	<pict>
--------	-----	---	--------	-------	--------

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
WGS	3 bytes,	text.
P	1 bytes,	text.
mash	1-3 bytes,	machine number to make a remote control of.
dir	1 byte,	command. F = Forward, start machine. B = Backward, reverse the machine. N = Zero, stop the machine. M = Motor protection ! = Ignore. RTU will ignore the command.
pict	1-3 bytes,	picture number from CS. If not used send 1, (one).

3.4.10.1 Status answer

<ver>	<prod>	<Nal>	<alp>	<rem>	<r1>	<r2>	<r3>	<r4>	
		<mp1>	<mp2>		<mp128>				

ver	3-5 bytes,	Forth system version, some older basic RTUs send their version number * 100.
prod	1-3 bytes,	type of RTU or compile ID, sometimes not used.
Nal	1-3 bytes,	number of alarms following last measuring point.
alp	1-3 bytes,	position where the first alarm is to be found.
rem	1 byte,	indicates if RTU is controlled locally = 0 or is in remote control = 1.
r1 - r4	1-5 bytes,	reserved for future use.
mp1-mp128	1-12 bytes,	data from the measuring points 1 to 128 for requested picture.

3.4.10.5 Example of command contents for status picture in CFG-file

```

4
;-----
; Machine number "Name" Forward Reverse
;-----
1  "P1" 1 0 ; {Start P1 ,Stop P1 , , }
2  "P2" 1 0 ; {Start P2 ,Stop P2 , , }
3  "RES" 1 1 ; {Reset Inhibit pumps , , ,Reset Lockout Trip Condition }
0  "AUTO" 1 0 ; {Resume to Auto mode , , , }

```

Explanation: The first column determines what machine number it is.

The second column is the presentation name used in the central system, and then the fun begins. The digits 1 0 are supposed to determine whether it is possible to Start or Reverse the object.

But since it is possible to change the texts shown in AquaView for remote controlling an object, these two digits is more or less obsolete today.

```

1  "P1" 1 0
2  "P2" 1 1

```

If the example above looked like this, following combinations would be possible.

```

1 F    1 N        0 M
2 F    2 N    2 B    0 M

```

Please note that M, (Motor protection), is sent as number zero.

However if there are texts in the CFG/ATF file like this

```

1  "P1" 1 0 ; {Start P1 ,Stop P1 , , }
2  "P2" 1 0 ; {Start P2 ,Stop P2 , , }

```

Following combinations would be possible

```

1 F    1 N
2 F    2 N

```

But if there are texts in the CFG/ATF file like this

```

1  "P1" 1 0 ; {Start pump,Stop pump, ,Reset motor protection}
2  "P2" 1 0 ; {Start pump,Stop pump, ,Reset motor protection@0}

```

then following combinations would be possible

```

1 F    1 N    1 M
2 F    2 N    0 M

```

Comment: If there are no remote controlling texts in the CFG/ATF file use zero as machine number for Motor protection. If there are texts use them instead but keep your eyes open for the @-sign which will “re-direct” the machine number.

3.4.11 Periodical report (WGD)

<dest>	WGD	<YY>	<MM>	<DD>	<hh>	<mm>	<aa>	<bb>	<cc>
--------	-----	------	------	------	------	------	------	------	------

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
WGD	3 bytes,	text.
YY	2 bytes,	year to set the real time clock.
MM	2 bytes,	month to set the real time clock.
DD	2 bytes,	day to set the real time clock.
hh	2 bytes,	hour to set the real time clock.
mm	2 bytes,	minute to set the real time clock.
aa	2 bytes,	requested year for report.
bb	2 bytes,	requested month for report.
cc	2 bytes,	requested day for report.

The answer could be "periodical data" or NDA, (No Data Available).

3.4.11.1 Periodical data

<seg1>	<seg2>	<seg3>	<seg4>	<sum>
--------	--------	--------	--------	-------

----- Repeated for all measuring points (1-128) -----

seg1	1-12 bytes,	data from segment1 (default 00-06).
seg2	1-12 bytes,	data from segment2 (default 06-09).
seg3	1-12 bytes,	data from segment3 (default 09-16).
seg4	1-12 bytes,	data from segment4 (default 16-00).
sum	1-12 bytes,	total value from seg1 to seg4.

Fixed RTU with plant number 111 and node ID 5 receives a report request for 2001-05-01.

```
CS←RTU    0 0 0 501 501 0 0 0 1506 1506 0 0 0 3513 3513 0 0 0 1 1 0 0 0 1 1 0 0  
           0 39 39 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 # 4984
```

Dialled RTU with plant number 9 receives a telegram for a report request for 2001-05-01.

```
CS←RTU  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 # 5632
```

CS→RTU 9 BRK # 376

```

; ##2
;-----
; Number of points in Report
;-----
10
;-----
; Point Text_number Wagon Min Max Max_input_value
; Text number < 1000 located in RAPPTXT.CFG
; Text number >=1000 located in ANL???.RFG
;-----
1 1001 1 0 1 3600 ; {DC} RpP1RunTime
2 1002 2 0 1 3600 ; {DC} RpP2RunTime
3 1003 4 0 1 1 ; {DC} RpP1Starts
4 1004 5 0 1 1 ; {DC} RpP2Starts
5 1005 6 0 1 1 ; {DC} RpW1OutFlVol
6 1006 7 0 1 1 ; {DC} RpW1OFlow#
7 1007 3 0 1 3600 ; {DC} RpW1OFlowTime
8 1008 8 0 1 1 ; {DC} RpW1OFlowVol
9 1009 9 0 0.1 1 ; RpRain
10 1010 10 0 0.001 1 ; RpEnergy

```

$$\text{Scaled Value} = ((\text{Raw Value from RTU} * (\text{Max} - \text{Min})) / \text{Max input value}) + \text{Min}$$

Example of RpPlRunTime when the Raw Value From RTU=7200

Scaled Value= $((7200*(1-0))/3600)+0=2$ hours running time

3.4.12 Historical trend (WGH).

<dest>	WGH	<YY>	<MM>	<DD>	<hh>	<mm>
--------	-----	------	------	------	------	------

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
WGH	3 bytes,	text.
YY	2 bytes,	year of last stored block.
MM	2 bytes,	month of last stored block.
DD	2 bytes,	day of last stored block.
hh	2 bytes,	hour of last stored block.
mm	2 bytes,	minute of last stored block.

The answer could be "historical trend data" or "No Data Available" (NDA).

3.4.12.1 Historical trend data.

<YY>	<MM>	<DD>	<hh>	<mm>	<v1>	<v2>	<v3>		<v32>	!
------	------	------	------	------	------	------	------	-------	-------	---

----- Repeated for every 5min segment -----

YY	2 bytes,	year for the block.
MM	1-2 bytes,	month for the block.
DD	1-2 bytes,	day for the block.
hh	1-2 bytes,	hour for the block.
mm	1-2 bytes,	minute for the block.
v1-v32	1-5 bytes,	data from measuring point 1 to 32. The actual numbers is depending of the type of RTU and version.
!	1 bytes,	text. Separates the blocks.

The length of the telegram must be calculated as follow.

The time must not exceed 6 hours and the whole telegram must be smaller than 4000 characters.

3.4.12.2 Example Dialed

Dialed RTU with plant number 9 receives a telegram for a trend request with last logged block for 2001-04-30 23:30.

```

CS          Dialling
CS←RTU      OK # 218
CS→RTU      9 WGH 01 04 30 23 30 # 1039
CS←RTU      1 4 30 23 35 64786 0 0 0 0 0 0 0 ! 1 4 30 23 40 64786 0 0 0 0 0 0 0 ! 1
4 30 23 45 64786 0 0 0 0 0 0 0 ! 1 4 30 23 50 64786 0 0 0 0 0 0 0 ! 1 4
30 23 55 64786 0 0 0 0 0 0 0 ! 1 4 30 00 64786 0 0 0 0 0 0 0 ! 1 5 1 0 5
64786 0 0 0 0 0 0 0 ! 1 5 1 0 10 64786 0 0 0 0 0 0 0 ! 1 5 1 0 15 64786
0 0 0 0 0 0 0 ! 1 5 1 0 20 64786 0 0 0 0 0 0 0 ! 1 5 1 0 25 64786 0 0 0 0
0 0 0 ! 1 5 1 0 30 64786 0 0 0 0 0 0 0 ! 1 5 1 0 35 64786 0 0 0 0 0 0 0 !
1 5 1 0 40 64786 0 0 0 0 0 0 0 ! 1 5 1 0 45 64786 0 0 0 0 0 0 0 ! 1 5 1 0
50 64786 0 0 0 0 0 0 0 ! 1 5 1 0 55 64786 0 0 0 0 0 0 0 ! 1 5 1 1 0
64786 0 0 0 0 0 0 0 ! 1 5 1 1 5 64786 0 0 0 0 0 0 0 ! 1 5 1 1 10 64786 0
0 0 0 0 0 0 ! 1 5 1 1 15 64786 0 0 0 0 0 0 0 ! 1 5 1 1 20 64786 0 0 0 0 0
0 0 ! 1 5 1 1 25 64786 0 0 0 0 0 0 0 ! 1 5 1 1 30 64786 0 0 0 0 0 0 0 ! 1
5 1 1 35 64786 0 0 0 0 0 0 0 ! 1 5 1 1 40 64786 0 0 0 0 0 0 0 ! 1 5 1 1
45 64786 0 0 0 0 0 0 0 ! 1 5 1 1 50 64786 0 0 0 0 0 0 0 ! 1 5 1 1 55
64786 0 0 0 0 0 0 0 ! 1 5 1 2 0 64786 0 0 0 0 0 0 0 ! 1 5 1 2 5 64786 0 0
0 0 0 0 0 ! 1 5 1 2 10 64786 0 0 0 0 0 0 0 ! 1 5 1 2 15 64786 0 0 0 0 0 0
0 ! 1 5 1 2 20 64786 0 0 0 0 0 0 0 ! 1 5 1 2 25 64786 0 0 0 0 0 0 0 ! 1 5
1 2 30 64786 0 0 0 0 0 0 0 ! 1 5 1 2 35 64786 0 0 0 0 0 0 0 ! 1 5 1 2 40
64786 0 0 0 0 0 0 0 ! 1 5 1 2 45 64786 0 0 0 0 0 0 0 ! 1 5 1 2 50 64786
0 0 0 0 0 0 0 ! 1 5 1 2 55 64786 0 0 0 0 0 0 0 ! 1 5 1 3 0 64786 0 0 0 0 0
0 0 ! 1 5 1 3 5 64786 0 0 0 0 0 0 0 ! 1 5 1 3 10 64786 0 0 0 0 0 0 0 ! 1 5
1 3 15 64786 0 0 0 0 0 0 0 ! 1 5 1 3 20 64786 0 0 0 0 0 0 0 ! 1 5 1 3 25
64786 0 0 0 0 0 0 0 ! 1 5 1 3 30 64786 0 0 0 0 0 0 0 ! 1 5 1 3 35 64786
0 0 0 0 0 0 0 ! 1 5 1 3 40 64786 0 0 0 0 0 0 0 ! 1 5 1 3 45 64786 0 0 0 0
0 0 0 ! 1 5 1 3 50 64786 0 0 0 0 0 0 0 ! # 6922

```

CS→RTU 9 WGH 01 05 01 03 50 # 1038
 CS←RTU 1 5 1 3 55 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 0 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 5
 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 10 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 15 64786
 0 0 0 0 0 0 0 0 ! 1 5 1 4 20 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 25 64786 0 0 0 0
 0 0 0 0 ! 1 5 1 4 30 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 35 64786 0 0 0 0 0 0 0 0 !
 1 5 1 4 40 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 45 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4
 50 64786 0 0 0 0 0 0 0 0 ! 1 5 1 4 55 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 0
 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 5 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 10 64786 0
 0 0 0 0 0 0 0 0 ! 1 5 1 5 15 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 20 64786 0 0 0 0 0
 0 0 ! 1 5 1 5 25 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 30 64786 0 0 0 0 0 0 0 0 ! 1
 5 1 5 35 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 40 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5
 45 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 50 64786 0 0 0 0 0 0 0 0 ! 1 5 1 5 55
 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 0 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 5 64786 0 0
 0 0 0 0 0 0 ! 1 5 1 6 10 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 15 64786 0 0 0 0 0 0
 0 0 ! 1 5 1 6 20 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 25 64786 0 0 0 0 0 0 0 0 ! 1 5
 1 6 30 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 35 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 40
 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 45 64786 0 0 0 0 0 0 0 0 ! 1 5 1 6 50 64786
 0 0 0 0 0 0 0 0 ! 1 5 1 6 55 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 0 64786 0 0 0 0 0
 0 0 ! 1 5 1 7 5 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 10 64786 0 0 0 0 0 0 0 0 ! 1 5
 1 7 15 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 20 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 25
 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 30 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 35 64786
 0 0 0 0 0 0 0 0 ! 1 5 1 7 40 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 45 64786 0 0 0 0
 0 0 0 0 ! 1 5 1 7 50 64786 0 0 0 0 0 0 0 0 ! 1 5 1 7 55 64786 0 0 0 0 0 0 0 0 !
 1 5 1 8 0 64786 0 0 0 0 0 0 0 0 ! 1 5 1 8 5 64786 0 0 0 0 0 0 0 0 ! 1 5 1 8 10
 64786 0 0 0 0 0 0 0 0 ! # 6479
 CS→RTU 9 WGH 01 05 01 08 10 # 1039
 CS←RTU 1 5 1 8 15 64786 0 0 0 0 0 0 0 0 ! 1 5 1 8 20 64786 0 0 0 0 0 0 0 0 ! 1 5 1 8
 25 64786 0 0 0 0 0 0 0 0 ! # 4220
 CS→RTU 9 WGH 01 05 01 08 20 # 1040
 CS←RTU NDA # 275
 CS→RTU 9 BRK # 376

Comment: Please note that the last block for a day is the one with time stamp 00:00.

2001-04-30 23:50-23:55	1 4 30 23 55 64786 0 0 0 0 0 0 0 0
2001-04-30 23:55-00:00	1 4 30 0 0 64786 0 0 0 0 0 0 0 0
2001-05-01 00:00-00:05	1 5 1 0 5 64786 0 0 0 0 0 0 0 0

3.4.12.3 Example of WGH section in a Forth CFG-file

```

; ##3
;-----
; Number of points for Historical trend
;-----
8
;-----
; Point Text_number Wagon Min Max Max_input_value
; Text number < 1000 located in RAPPTXT.CFG
; Text number >=1000 located in ANL???.RFG
;-----

1      1015      1      0      0.01      10      0 ; {DC} TrW1Lev
2      1016      2      0      0.1       6      1 ; {DC} TrP1CapCalc
3      1017      3      0      0.1       6      1 ; {DC} TrP2CapCalc
4      1018      4      0      0.1       6      1 ; {DC} TrP3CapCalc
5      1019      5      0      0.1       6      1 ; {DC} TrP4CapCalc
6      1020      6      0      0.1       6      1 ; {DC} TrInFl
7      1021      7      0      0.1       6      1 ; {DC} TrW1OFlow
8      1022      8      0      0.1       1      1 ; TrRain

```

Comment: The scaling works like this:

Scaled_Value= ((Raw_Value_from_RTU*(Max-Min))/Max_input_value) + Min

3.4.13 Break (BRK)

<dest>	BRK
--------	-----

dest 1-3 bytes, Logical plant number, (1-999), if dialled or
Node ID, (0-RTU dependant), if fixed

BRK 3 bytes, text.

Is transmitted to the RTU when the communication should stop.

No answer is returned from the RTU.

To make sure that the RTU really hears the BRK-telegram and hangs up the BRK telegram is usually send three times in a row before the central system hangs up the modem.

It is also important to send BRK to a fixed RTU when disconnecting WGS since some RTUs are “remote controlled” until BRK. This only applies if some command has been sent in the WGS request.

3.5 Message types Basic

This section tries to explain the minor dialectal differences there are between an old Basic RTU and a newer Forth.

3.5.1.1 Alarm answer - alarm is pending (ALR).

Note: For all versions.

ALR	<lsn>	<code>	<YY>	<MM>	<DD>	<hh>	<mm>	<flank>	
-----	-------	--------	------	------	------	------	------	---------	--

	T	<V1>	<V2>	<V3>	<V4>	<pri>
--	---	------	------	------	------	-------

ALR	3 bytes,	text
lsn	1-5 bytes,	logical plant number (0-999)
code	1-5 bytes,	alarm code (0-8999)
YY	2 bytes,	year (00-99)
MM	1-2 bytes,	month (1-12)
DD	1-2 bytes,	day (1-31)
hh	1-2 bytes,	hour (0-23)
mm	1-2 bytes,	minute (0-59)
flank	1 bytes	the state of the event / alarm. Valid values are: A = Active / On / Up-flank P = Passive / Off / Down-flank
T	1 byte,	text
v1	1-5 bytes,	AquaTrain wagon 1 (default zero)
v2	1-5 bytes,	AquaTrain wagon 2 (default zero)
v3	1-5 bytes,	AquaTrain wagon 3 (default zero)
v4	1-5 bytes,	AquaTrain wagon 4 (default zero)
pri	1 byte,	Priority

3.5.2 Function channels CS→RTU (MPA, MPB, MPC, MPD, MPL, MCC)

<dest>	<prod>	<lsn>	<YY>	<MM>	<DD>	<hh>	<mm>	<ss>	
	<chan>	<type>	<value>						

----- Repeated (<= 200) -----

3.5.2.1 HEADER

dest	1-3 bytes,	Logical plant number, (1-999), if dialled or Node ID, (0-RTU dependant), if fixed
prod	3 bytes,	product, MPA, MPB, MPC, MPD, MPL or MCC.
lsn	1-5 bytes,	logical plant number (0-999).
YY	2 bytes,	year (00-99).
MM	2 bytes,	month (01-12).
DD	2 bytes,	day (01-31).
hh	2 bytes,	hour (00-23).
mm	2 bytes,	minute (00-59).
ss	2 bytes,	second (00-59).

3.5.2.2 DATA

chan	1-3 bytes,	channel number, (1-200).
type	1 byte,	type of data that follow. A = Analogue (0-200). D = Integer (0-32767). T = Text (Data between character ' (27H)). 1-4= Float (Data transmitted with 1-4 decimals).
value	1-128 bytes,	data as described above.

3.5.3 Analogue channels CS→RTU (MAS Basic)

<dest>	MAS	<input>	<signal>	<high>	<low>	<hyst>	<htr>
--------	-----	---------	----------	--------	-------	--------	-------

----- repeated (<= 96) -----

3.5.3.1 HEADER

dest 1-3 bytes, Logical plant number, (1-999), if dialled or
Node ID, (0-RTU dependant), if fixed
MAS 3 bytes, text.

3.5.3.2 DATA

input 1-2 bytes, input number (1-96).
signal 1 bytes, type of signal (0=0-20mA, 4=4-20mA).
high 1-3 bytes, level high-alarm (0-200).
low 1-3 bytes, level low-alarm (0-200).
hyst 1-3 bytes, hystereses before alarm can be reactivated (0 -100%).
htr 1 byte, defines the signal for historical trend (0=No, 1=Yes).

3.5.4 Break, with acknowledge (EBR)

<dest>	EBR
--------	-----

dest 1-3 bytes, Logical plant number, (1-999), if dialled or
Node ID, (0-RTU dependant), if fixed
EBR 3 bytes, text.

Is transmitted to the RTU when the communication should stop.

This version is different to BRK. EBR expects a answer from the RTU.

If a SCO is not received before timeout or a wrong telegram is received, EBR must be retransmitted (maximum 6 times).

3.6 Answer messages

Answer messages is transmitted from the RTUs at following conditions.

- Telegram OK.
- Error in telegram.
- Requested data missing.
- Wait

These answer telegrams are used with different telegrams. With some telegram there are other answers as well.

3.6.1 Telegram OK (SCO)

SCO

SCO 3 bytes, text.

3.6.2 Error in telegram (SCE)

SCE

SCE 3 bytes, text.

3.6.3 No Data Available (NDA)

NDA

NDA 3 bytes, text.

3.6.4 Wait

!!

Wait 1 byte, text.

This is a special telegram that consist of only one byte, (13H).

The Wait telegrams purpose is to prevent timeouts. Some telegrams can take quite some time to put together in the RTU. In the mean while the RTU should send Wait telegrams to prevent that timeouts occur in the central system. **This means that it is very important that the central system accepts Wait as a valid telegram and that the timeout is reset.**

Please note that it is allowed for RTU's to add Wait telegrams inside another telegram, but the Wait signs should not be a part of the checksum calculation.

Example WGS answer without extra Wait:

32000 654321 03 11 0 0 0 0 1 2 3 4 5 6 7 8 9 10 8050 8531 8532 # 2925

Example WGS answer with extra Wait:

32000 65!!4321 03 11!! 0 0 0 0 0!! 1 2 3 4 5!! 6 7 8 9 1!!0 8050 853!!1 8532 # 2!!925

Note that the checksums are identical.

3.7 Detailed example with WGH

Since some characters in AquaCom are “non-printable”, this section will try to make it easier to understand how it all fits together since the examples in earlier in this document does not show this kind of characters or the Wait-telegram.

SOH	Start of telegram 1 byte (01H)	= <SOH> in this section
STX	Start of telegram. 1 byte (02H)	= <STX> in this section
CR	Carriage Return. 1 byte (0DH)	= <CR> in this section
Wait	Wait 1 byte (13H)	= <DC3> in this section
Space	Space 1 byte (20H)	= _ in this section

CS→RTU <SOH>_500_WGH_99_12_31_21_40_#_1147_<CR>
 CS←RTU

```
<DC3><DC3><STX>_99_12_31_21_45_1000_1619_1559_1049_0_0_!_99_
_12_31_21_50_700_1559_1559_962_0_0_!_99_12_31_21_55_604_1559_1
559_767_0_0_!_99_12_31_22_0_896_1559_1559_601_0_0_!_99_12_31_2
2_5_1196_1559_1559_600_0_0_!_99_12_31_22_10_1496_1559_1559_600
_0_0_!_99_12_31_22_15_1796_1559_1559_600_0_0_!_99_12_31_22_20_
1896_1559_1559_793_0_0_!_99_12_31_22_25_1606_1559_1559_956_0_0
_!_99_12_31_22_30_1320_1559_1559_958_0_0_!_99_12_31_22_35_1005
_1559_1499_931_0_0_!_99_12_31_22_40_705_1559_1439_841_0_0_!_99
_12_31_22_45_601_1559_1439_715_0_0_!_99_12_31_22_50_891_1559_1
439_601_0_0_!_99_12_31_22_55_1191_1559_1439_600_0_0_!_99_12_31
_23_0_1491_1559_1439_600_0_0_!_99_12_31_23_5_1791_1559_1439_60
0_0_0_!_99_12_31_23_10_1899_1559_1439_787_0_0_!_99_12_31_23_15
_1611_1559_1439_955_0_0_!_99_12_31_23_20_1311_1559_1439_958_0
_0_!_99_12_31_23_25_1025_1515_1439_933_0_0_!_99_12_31_23_30_710
_1439_1439_842_0_0_!_99_12_31_23_35_598_1439_1439_712_0_0_!_99
_12_31_23_40_886_1439_1439_600_0_0_!_99_12_31_23_45_1186_1439_1
439_600_0_0_!_99_12_31_23_50_1486_1439_1439_600_0_0_!_99_12_31_2
3_55_1786_1439_1439_600_0_0_!_99_12_31_0_0_1902_1439_1439_727_0
_0_!_0_1_1_0_5_1615_1439_1439_837_0_0_!_0_1_1_0_10_1315_1439_1439
_838_0_0_!_0_1_1_0_15_1015_1439_1391_813_0_0_!_0_1_1_0_20_729_14
39_1319_724_0_0_!_0_1_1_0_25_596_1439_1319_658_0_0_!_0_1_1_0_30
_882_1439_1319_600_0_0_!_0_1_1_0_35_1182_1439_1319_600_0_0_!_0_1
_1_0_40_1482_1439_1319_600_0_0_!_0_1_1_0_45_1782_1439_1319_600_0
_0_!_0_1_1_0_50_1904_1439_1319_722_0_0_!_0_1_1_0_55_1620_1439_131
9_836_0_0_!_0_1_1_1_0_1320_1439_1319_838_0_0_!_0_1_1_1_5_1020_13
91_1319_814_0_0_!_0_1_1_1_10_720_1319_1319_722_0_0_!_0_1_1_1_15
_592_1319_1319_660_0_0_!_0_1_1_1_20_862_1319_1319_600_0_0_!_0_1_1
_1_25_1177_1319_1319_600_0_0_!_#_11019_<CR>
```




www.flygt.com