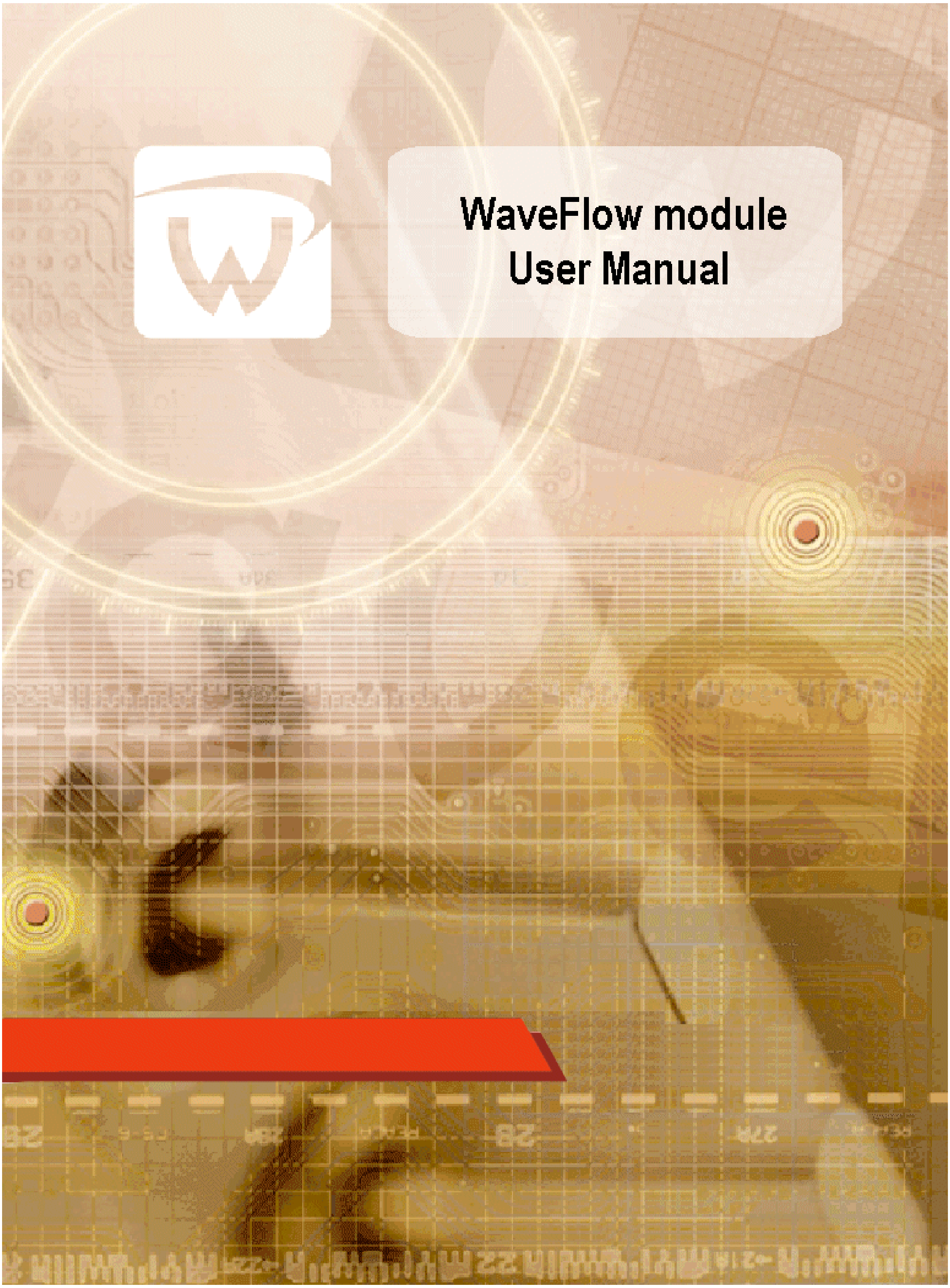


WaveFlow module User Manual

REVISIONS HISTORY

Rev. #	Description	Authot	Date	Comments
1	Original Document	RCS	15/10/04	Version 1
2	Support v2.03 WaveFlow - 4 inputs	RCS	31/03/05	Version 2

SUPPORTED FIRMWARE VERSION

➢ WaveFlow – 4 inputs

Rev. #	Firmware version	Date
1	CSFW063_V00A3_0203	15/09/04

➢ WaveFlow – Standard BackFlow

Rev. #	Firmware version	Date
1	CSFW109_V00A3_0500	24/04/04

➢ WaveFlow – BackFlow – Specific

Rev. #	Firmware version	Date
1	CSFW082_V00A3_010E	26/04/04
2	CSFW082_V00A3_0110	27/04/04

TABLE OF CONTENTS

1 PRESENTATION

This document describes the functionalities of *WaveFlow* radio modules :

- **WaveFlow – 4 inputs :** *this modules allows to count up to four pulse inputs.*
- **WaveFlow – 4800 :** *this module is directly intended to replace the old generation of Is)FlowMeter module. Its functionalities are limited to ensure a whole compatibility with the Is)FlowMeter modules.*
- **WaveFlow – BackFlow – Specific :** *this type of module is intended to be connected to water meters equipped with double pulses heads making it possible to provide information on the direction of water circulation.*
- **WaveFlow – Standard :** *this module has the same functionalities as WaveFlow - BacFlow - Specific, but integrates a more advanced detection model for BackFlow.*
- **WaveFlow – Standard CYBLE 5 fils :** *this module is an alternative of the WaveFlow – Standard, adapted to sensor CYBLE 5 wire.*

This document defines in an exhaustive way the applicatives data relating to serial dialog frames between a *Wavecard* and a host equipment , used to reach the data of the *WaveFlow* radio module.

2 REFERENCE DOCUMENTS

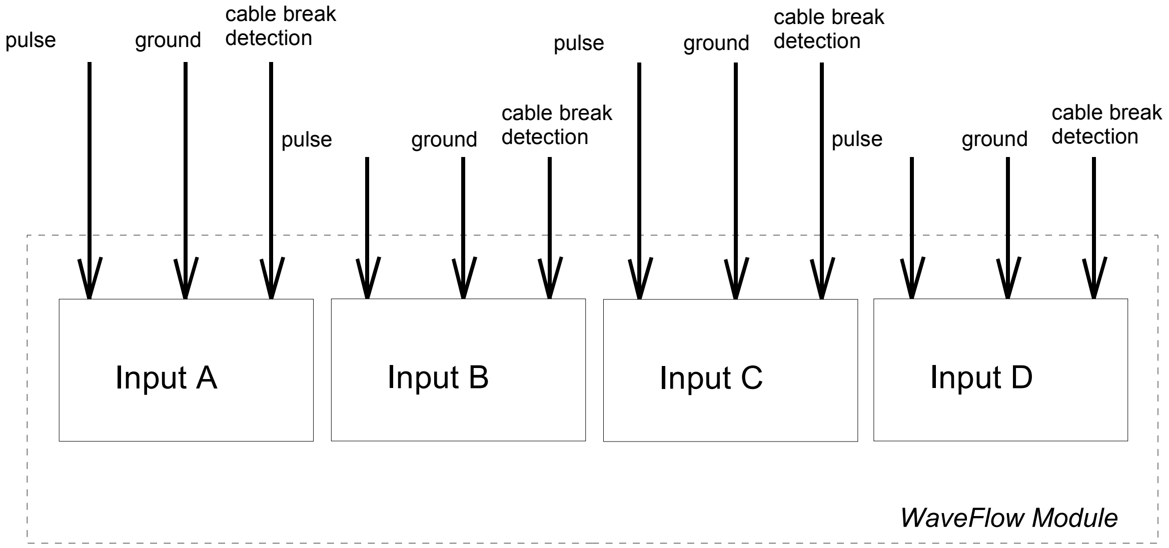
Ref	Title	Reference	Version	Date
DR[1]	WaveCard user handbook			

3 PRESENTATION OF THE WAVEFLOW MODULES FUNCTIONALITIES

3.1 SENSORS INTERFACE

The WaveFlow module manages up to 4 pulse metering inputs.

Each meter input may have up to 3 wires, a pulse wire, a cable break detection wire (optional) and a ground wire.



- **WaveFlow – 4 inputs :** *inputs A to D are used for measurement of pulses coming from water meters.*
- **WaveFlow – 4800 :** *inputs A and B are used for measurement of pulses coming from water meters. The inputs C and D are not used.*
- **WaveFlow – BackFlow Specific :** *inputs A and B are used for counting from water meterd. The inputs C and D are used for backflow detection.*
- **WaveFlow – Standard :** *inputs A and B are used for counting from water meterd. The inputs C and D are used for backflow detection.*
- **WaveFlow – Standard – type CYBLE 5 fils :** *inputs A and B are used for counting from water meterd. The inputs C and D are used for backflow detection.*

3.2 READING AND STORAGE OF THE INDEXES

Compatibility : this functionnalitiy is supported by all the module types

The WaveFlow modules offer the possibility to :

- read the true value of water meters on inputs A and B. In the case of *WaveFlow* modules managing water backflow, this backflow value is subtracted from the effective water consumption.
- transmit the last 24 values memorised
- transmit the current index, the four last indexes stored, and the index at the end of the month.

3.3 PERIODIC INDEX READING (Datalogging)

Compatibility : this functionnalitiy is supported by all the module types

The Datalogging mode enables periodic logging of meter index values at each input. The frequency of these readings may be set in three modes :

- index logging in time steps
- index logging once a week
- index logging once a month

This mode stores up to 24 indexes, with a loop when the table is full. This allows to have always access to the last 24 indexes logged (see chapter 6.4.6).

3.4 AUTOMATIC TRANSMISSION OF FAULTS OR FLOW PROBLEMS

Compatibility : this functionnality is only supported by the following modules,

- ***WaveFlow – 4 inputs***
- ***WaveFlow – Standard***
- ***WaveFlow – Standard – CYBLE 5 fils***
- ***WaveFlow – BackFlow – specific***

The WaveFlow module offers the possibility to automatically transmit radio frames when an occurrence is detected. The following occurrences may provoke an automatic alarm:

- Residual leak detection
- Extreme leak detection
- Water backflow detection *(if this function is available on the meter/pulse emitter pair)*
- Cable break detection
- Reed fault detection *(if this function is available on the meter/pulse emitter pair)*
- End of battery life detection

It is possible to select for each type of occurrence whether or not an alarm frame is to be sent.

The radio address of the receiver module and the repeater path must be preset with a radio signal.

3.5 WAKE-UP SYSTEM MANAGEMENT

Compatibility : this functionnalitiy is supported by all the module types

In order to reduce module power consumption, a wake-up period parameter setting system is incorporated. This system enables modification of the module wake-up period (default setting 1 s) by entering a time and day of the week :

- The wake-up period default value may be modified;
- Two time-windows with different wake-up periods may be defined;
- Each day of the week may be set in one of the following three cases :
  - Wake-up period default setting
  - Wake-up according to predefined time windows
  - No wake-up period (for safety reasons, the module is not disabled on reception and it wakes up every 10 seconds)

The system is disabled by default and must be enabled with a radio signal (see chapter 6.3).

3.6 LEAK DETECTION MANAGEMENT

Compatibility : this functionnalitiy is supported by all the module types

The WaveFlow module detects two types of leaks for each metering input, residual leaks and extreme leaks.

For each of these types of leaks, the module carries out the following operations:

- Leak detection
- Date of leak detection recording
- Recording of detection date and min. (or max.) flow-rate.

If the leak stops:

- Date
- Recording of the leak stoppage date and last flow-rate detected

This data is stored in a circular buffer which may be accessed by radio and contains the last 5 events logged (occurrence or stoppage of leaks).
The module may be programmed to generate an automatic radio frame when a leak is detected.

The leak detection parameters are programmed individually for each pulse input (see chapter 6.5).

3.7 WATER BACKFLOW DETECTION (if supported by the module)

There are two models of detections of the water backflow, each one having a method of measurement, and a different level of details.

3.7.1 Simple water backflow detection principle

> ***Compatibility : this functionnality is only supported by the following module,***
>
> - ***WaveFlow – BackFlow – specific***

The WaveFlow module is able to detect backflow consumption for which the critical flow-rate threshold parameter may be set. When this flow-rate is attained within a given period, a fault detection signal for the current month is transmitted in a flag byte (see chapter 6.6).

In this case, only the water backflow occurrences for the current month are saved together with the effective backflow value.

3.7.2 Advanced water backflow detection principle

> ***Compatibility : this functionnality is only supported by the following module,***
>
> - ***WaveFlow – Standard***
> - ***WaveFlow – Standard – CYBLE 5 fils***

The WaveFlow module is able to detect backflow consumption for which the critical flow-rate threshold parameter may be set. Two detection methods are available for this purpose and may be selected by writing a configuration parameter.

The module carries out the following operations for both methods :

- Water backflow detection,
- Date of *'water backflow' occurrence,*
- Duration, or end of occurrence

This data is stored in a circular buffer which may be accessed by radio and contains the last 4 occurrences logged.

The module may be programmed to generate an automatic radio frame when backflow is detected.

The water backflow detection parameters are programmed individually for each pulse input (see chapter 6.7).

3.8 WIRE CUT DETECTION

Compatibility : this functionnalitiy is supported by all the module types

Cable break fault detection is possible if the cable sensor is of the 3-wire type.
In such a case, the 3rd wire is connected to a module input in the same way as the metering input.

Periodically, the software detects a cable break by measuring the level on this input.
Once a cable break fault has been detected, it is transmitted in a status byte and the date the cable break fault is detected is memorised.

Cable break fault detection is normally enabled but may be disabled with a radio parameter setting signal in an operating mode byte (see chapter 6.8).

3.9 END OF BATTERY LIFE DETECTION

Compatibility : this functionnalitiy is supported by all the module types

To detect the end of battery life, the **WaveFlow** module uses the power metering principle rather than measurement of the battery voltage. Lithium batteries are, in particular during passivation, unsuitable for the voltage measurement method to determine the remaining capacity.

The **WaveFlow** records and evaluates all occurrences (measurements, transmissions) to decrement the power meter according to the battery used. When the meter passes below a predefined threshold, the “end of battery life” is signalled with the STATUS byte.

The initial value of the end-of-life meter is factory-set. It depends on the type and number of batteries used.

When the end of battery life is detected, the detection date is memorised and may be read with a radio command (see chapter 6.9).

3.10 REED FAULT DETECTION *(if supported by the module)*

Compatibility : this functionnality is only supported by the following module,

- ***WaveFlow – Standard***
- ***WaveFlow – BackFlow – specific***

The CYBLE sensor doesn't use dry-contact (reed).

A reed fault is detected when the pulse transmitted by the second reed of the pulse emitter is not detected after several attempts.

Once a reed fault has been detected, *it is transmitted with the corresponding input in a status byte* and the date the reed fault is detected is memorised.

Reed fault detection is normally disabled but may be enabled with a radio parameter setting signal in an operating mode byte (see chapter 6.8).

4 DATA EXCHANGE PRINCIPLE WITH A WAVEFLOW MODULE

The WaveFlow module uses the **WAVENIS®** protocol

The choice of mode used is initiated by the read element which uses a different set of commands (see WaveCard document) when sending commands to the WaveCard.

The following chart indicates the read modes possible as well as their typical applications.

Read mode	Description	Recommendations
peer-to-peer	Individual reading with re-transmission management in case of no reply	Standard use
Polling	This mode enables successive polling of several modules in a single operation . The principle consists of waking up several modules with the 1st radio transmission.	To be used when module reading time is an important factor. Re-transmission not possible.
Broadcast and multicast (*)	This mode enables use of a single frame to address all radio modules within reception range. The multicast mode may only address one group of modules.	This mode enables reading of modules without knowing their radio address. Type of use: detection of radio modules within range of the emitter module (installation phases).

- **Additional functions:**

Additional functions	Compatibility	Description	Recommendations
Repeater	Only used in peer-to-peer mode.	This function enables use of a radio module to relay a frame which was not initially intended for this module. This is a default function of the WaveFlow module i.e. it may be read via several repeaters but may also act as a repeater itself when reading another unit.	This function is used when the caller module and the target Waveflow module are outside radio range. The maximum number of repeaters is limited to 3.

Remark:

- The broadcast and multicast modes are not used as standard on the WaveFlow module (only on request).

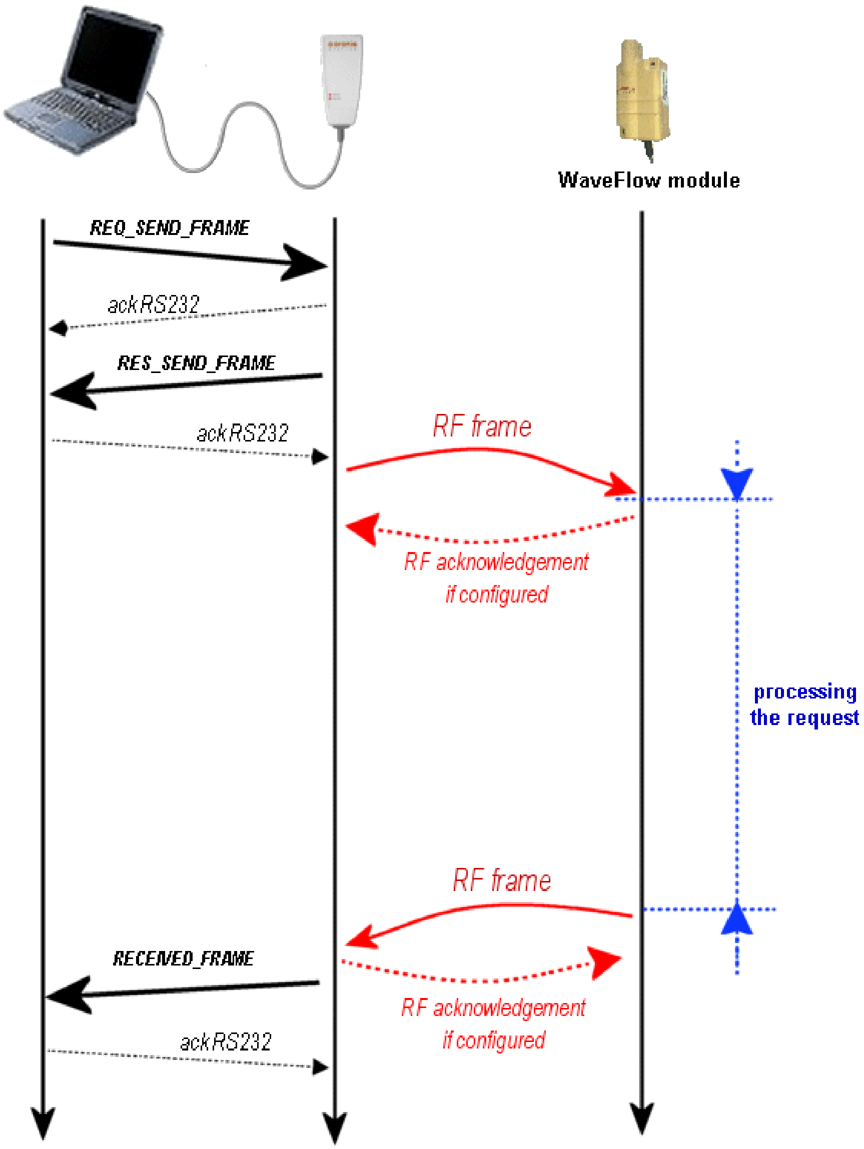
- When used in Polling mode, it is possible to assign a group number to the WaveFlow module with a radio parameter setting.

It is then possible to access all modules with the same group number via the read commands.

- **Example in Point-to-point mode :**

***Remark** : Generally, the exchanges examples given in this document will be in Point-to-point mode, except when the context depends directly on the mode of exchanges.*

This type of radio exchange allows to send a request, then to await a response of the remote equipment.



Note : the commands of Point-to-point exchanges, have the following format: (all the exchanges modes are treated in document [DR1])

CMD	NAME	DESCRIPTION
0x20	REQ_SEND_FRAME	Request to send a radio frame with the waiting for the radio response.
0x30	RECEIVED_FRAME	Received radio frame by the radio board.

The data field of each command must be formatted according to the following table :

CMD	DATA	
	6 bytes	variable (max : 152 bytes)
0x20	**Radio address from equipment to reach**	**Data to transmit**
0x30	**Radio address from transmitter equipment**	**Received Data**

the first byte of the field 'data to transmit' (or 'Received Data') contains an applicative command (or its acknowledgement). That allows to the receptor of the frame to identify the type of requests (or of responses).

	Data to Transmit or Received Data	
	1 byte	151 bytes
REQ_SEND_FRAME	Applicative command	Data relating to the request
RECEIVED_FRAME	Acknowledgement of the applicative command	Data relating to the response

The commands set is available in Appendix A.

ATTENTION, This document describes only the format of the fields 'Data to Transmit', 'Received Data'. These fields are directly dependent on the access to the functionalities of the WaveTherm modules. The other fields of the radio frame depend on the exchanges modes chosen, and are detailed in document [DR1].

5 MODIFICATION OF THE INTERNAL PARAMETERS

5.1 INTERNAL PARAMETERS LIST ACCESSIBLE BY RADIO COMMANDS

5.1.1 Parameters common to all WAVEFLOW versions

The table below describes the internal parameters accessible by standard read and write commands.

N°	Description	Size (in bytes)	Access Right	Default value (Hexa)
0x01	Operation Mode	1	R/W	0x09
0x02	WakeUp system status word	1	R/W	0x00
0x03	Default WakeUp period (in second)	1	R/W	0x01
0x04	Start time for 1st time window	1	R/W	0x07
0x05	WakeUp period for 1st time window (in second)	1	R/W	0x01
0x06	Start time for 2nd time window	1	R/W	0x12
0x07	WakeUp period for 2nd time window (in second)	1	R/W	0x01
0x08	Enalbe time windows by day of the week	1	R/W	0xFF
0x09	Enable WakeUp periods by day of the week	1	R/W	0x00
0x20	Application Status	1	R/W	0x00
0x22	Alarm Configuration Byte	1	R/W	0x00
0x80	Measurement Period (datalogging in time steps)	1	R/W	0x13
0x81	Start hour of the datalogging in time steps	1	R/W	0x08
0x82	Day of the week, or of the month (datalogging)	1	R/W	0x01
0x83	Time of measurement (datalogging once a week, or once a month)	1	R/W	0x08
0x85	Group number to use in polling mode	1	R/W	0x00
0xC4	Measurement Step, exepressed in multiple of minutes	1	R/W	0x3C
0x88	Residual leakage flow (low threshold) for **input A** (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x8A	Residual leakage detection period for **input A** (expressed in multiple of measurement step)	1	R/W	0xA8
0x89	Extreme leakage flow (high threshold) for **input A** (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0xC0	Extreme leakage detection period for **input A** (expressed in multiple of measurement step)	1	R/W	0x03
0x8B	Residual leakage flow (low threshold) for **input B** (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x8C	Extreme leakage flow (high threshold) for **input B** (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0x8D	Residual leakage detection period for **input B** (expressed in multiple of measurement step)	1	R/W	0xA8
0xC1	Extreme leakage detection period for **input B** (expressed in multiple of measurement step)	1	R/W	0x03
0x90	Date of detection of end of battery life	6	R/W	0x00
0x91	Date of wirecut detection on **input A**	6	R/W	0x00
0x92	Date of wirecut detection on **input B**	6	R/W	0x00
0xA2	Battery life duration counter	2	R	0xC15C

0xA3	Pulse Weight for **input A**	1	R/W	0xFF
0xA4	Pulse Weight for **input B**	1	R/W	0xFF
0xB0	Number of repeaters used to transmit an alarm frame	1	R/W	0x00
0xB1	Address of the 1st repeater used to transmit an alarm frame	6	R/W	0x00
0xB2	Address of the 2nd repeater used to transmit an alarm frame	6	R/W	0x00
0xB3	Address of the 3rd repeater used to transmit an alarm frame	6	R/W	0x00
0xB4	Address of the recipient of the alarm frame	6	R/W	0x010A030000BD

ATTENTION, lthe control byte 'Operation Mode', 'Application Status', and 'Alarm Configuration Byte' are common to all Waveflow type, but the meaning of each field may be different according to the module type used.

Please refer to chapter 6.10, for the definition of the 'Alarm Configuration Byte'.

The 'Operation Mode', 'Application Status' are described below.

- Definition of the **Operation Mode** byte :

MSB **LSB**

Operation Mode							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reed Fault detection (1) **0** : deactivated **1** : activated	**Extreme Leak detection** **0** : deactivated **1** : activated	**Residual Leak detection** **0** : deactivated **1** : activated	**WireCut detection** **0** : deactivated **1** : activated	**Datalogging** **00** : deactivated **01** : by time steps **10** : once a week **11** : once a month		Management of the inputs. (2)	

(1) Reed Fault detection : only if supported by the module, else this bit is not used. The Reed fault is supported by the WaveFlow – Standard, and WaveFlow – Specific Backflow.

(2) Management of the inputs :

Module type	**Configuration of bits [b1:b0]**
WaveFlow – 4 inputs	00 : management of one input (A) 01 : management of 2 inputs (A,B) 10 : management of 3 inputs (A, B, C, D) 11 : management of 4 inputs (A, B, C, D)
WaveFlow – Specific Backflow WaveFlow – Standard WaveFlow – Standard CYBLE 5 wires WaveFlow – 4800	00 : management of one input (A) 01 : management of 2 inputs (A,B)

- definition of the ***Application Status*** byte

it is possible to reset the bits by writing the Application Status byte, or by reading the corresponding event table (only for leak and backflow detection).

		MSB							LSB
		Application Status							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WaveFlow 4 inputs		-	*WireCut detection on input D*	*WireCut detection on input C*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Specific BackFlow		*BackFlow detection in the month*	*Reed Fault detection on input B*	*Reed Fault detection on input A*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Standard		*BackFlow detection*	*Reed Fault detection on input B*	*Reed Fault detection on input A*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Standard CYBLE 5 wires		*BackFlow detection*	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow 4800		-	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*

The coding is as follows : *0 : not detected*

1 : detected

5.1.2 Parameters specific to the WaveFlow - 4 inputs

The table below describes the internal parameters exclusively reserved for the WaveFlow - 4 inputs, their access right, via the reading or writing radio requests.

N°	Description	Size (in bytes)	Access Right	Default value
0x98	Residual leakage flow (low threshold) for input C (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x9A	Residual leakage detection period for input C (expressed in multiple of measurement step)	1	R/W	0xA8
0x99	Extreme leakage flow (high threshold) for input C (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0xC2	Extreme leakage detection period for input C (expressed in multiple of measurement step)	1	R/W	0x03
0x9B	Residual leakage flow (low threshold) for input D (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x9D	Residual leakage detection period for input D (expressed in multiple of measurement step)	1	R/W	0xA8
0x9C	Extreme leakage flow (high threshold) for input D (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0xC3	Extreme leakage detection period for input D (expressed in multiple of measurement step)	1	R/W	0x03
0x95	Wirecut detection date on input C	6	R/W	0x00
0x96	Wirecut detection date on input D	6	R/W	0x00
0xA5	Pulse weight on input C	1	R/W	0xFF
0xA6	Pulse weight on input D	1	R/W	0xFF

5.1.3 Parameters specific to the WaveFlow – Specific Backflow

The table below describes the internal parameters exclusively reserved for the WaveFlow – Specific Backflow , their access right, via the reading or writing radio requests.

N°	Description	Size (in bytes)	Access Right	Default value
0x93	Reed fault detection date on input A	6	R/W	0x00
0x94	Reed fault detection date on input B	6	R/W	0x00
0xC5	Backflow detection period on input A (expressed in multiple of hour)	1	R/W	0x01
0xC6	BackFlow threshold on input A (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xC7	Backflow detection period on input B (expressed in multiple of hour)	1	R/W	0x01
0xC8	BackFlow threshold on input B (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xC9	Flag indicating backflow detections per month, on input A **(LSB first)**	2	R/W	0x0000
0xCA	Flag indicating backflow detections per month, on input B **(LSB first)**	2	R/W	0x0000

5.1.4 Parameters specific to the WaveFlow - Standard

The table below describes the internal parameters exclusively reserved for the WaveFlow – Standard , their access right, via the reading or writing radio requests.

N°	Description	Size (in bytes)	Access Right	Default value
0x0A	Extended Operation Mode	1	R/W	0x00
0x93	Reed fault detection date on input A	6	R/W	0x00
0x94	Reed fault detection date on input B	6	R/W	0x00
0xCB	Backflow detection period on input A (expressed in multiple of 10 minutes)	1	R/W	0x01
0xCC	BackFlow threshold on input A (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xCD	Backflow detection period on input B (expressed in multiple of 10 minutes)	1	R/W	0x01
0xCE	BackFlow threshold on input B (expressed in number of pulse per backflow detection period)	1	R/W	0x0A

5.1.5 Parameters specific to the WaveFlow – Standard – CYBLE 5 wire

The table below describes the internal parameters exclusively reserved for the WaveFlow – Standard – CYBLE 5 wire , their access right, via the reading or writing radio requests.

N°	Description	Size (in bytes)	Access Right	Default value
0xCB	Backflow detection period on input A (expressed in multiple of 10 minutes)	1	R/W	0x01
0xCC	BackFlow threshold on input A (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xCD	Backflow detection period on input B (expressed in multiple of 10 minutes)	1	R/W	0x01
0xCE	BackFlow threshold on input B (expressed in number of pulse per backflow detection period)	1	R/W	0x0A

5.1.6 Parameters specific to the WaveFlow - 4800

The WaveFlow module - 4800 does not have specific parameters; it **only uses the common ressources** of the WaveFlow.

5.2 PRINCIPLE OF READING AND WRITING OF INTERNAL PARAMETERS

Document [DR1] details the exchanges modes, and their associated requests; with an aim of sending data to a distant module.
This chapter details the data field in order to configure the internal parameters of the WaveFlow modules.

	DATA Field	
	1 byte	Max = 151 bytes
REQ_SEND_FRAME	Applicative Commands	Data
RECEIVED_FRAME	Acknowledgement of the applicative commands	Data

There are two commands used to configure the internal parameters of the WaveFlow modules, and each one has a corresponding acknowledgement command.

Applicatives Commands	Description
0x10	Request of parameter(s) reading
0x90	Acknowledgement of the request of parameter(s) reading
0x11	Request of parameter(s) writing
0x91	Acknowledgement of the request of parameter(s) writing

Remark : *In the command byte coding, the Response frame type are taking the Request command byte value with the MSB bit set to 1.*

It is possible to access up to 9 parameters simultaneously for writing or reading *(all for reading, or all for writing).*

➢ Format of access for parameter(s) reading

- contents of request REQ_SEND_FRAME

Data Field (max : 152 bytes)						
Applicative command	**Number of parameters to be read**	**Number of the 1st parameter**	**Size of the 1st parameter**	**Number of the 2nd parameter**	**Size of the 2nd parameter**	**...**
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

- contents of response RECEIVED_FRAME

Data field (max : 152 bytes)								
Acknowledgement of the applicative command	**Number of parameters read**	**Number of the 1st parameter**	**Size of the 1st parameter**	**Data of the 1st parameter**	**...**	**Number of the nth parameter**	**Size of the nth parameter**	**Data of the nth parameter**
1 byte	1 byte	1 byte	1 byte	variable	...	1 byte	1 byte	variable

$n_{max} = 9$

When a parameter is not known by the system, or the size is configured with a wrong value, the corresponding data byte in response is set to 0xFF.

➢ Format of access for parameter (s) writing

- Contents of request REQ_SEND_FRAME

Data field (max : 152 bytes)								
Applicative command	**Number of parameters read**	**Number of the 1st parameter**	**Size of the 1st parameter**	**Data of the 1st parameter**	**...**	**Number of the nth parameter**	**Size of the nth parameter**	**Data of the nth parameter**
1 byte	1 byte	1 byte	1 byte	variable	...	1 byte	1 byte	variable

n_{max} = 9

- Contents of response RECEIVED_FRAME

Data Field (max : 152 bytes)						
Acknowledgement of the applicative command	**Number of parameters written**	**Number of the 1st parameter**	**Update Status**	**Number of the 2nd parameter**	**Update Status**	**...**
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	...

'Update Status' :	*0x00 : update OK*
	0xFF : update error

Attention, some of the parameters are limited, i.e. their values should not be written out of the limits.

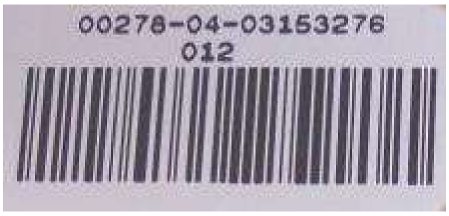
If a value is written out of the limits, the value will be written but not managed, the status of writing will be OK.

Example: parameter 0x04 (hour of the first time-window) this parameter is set up from 0 to 23. Thus if value 40 is set, it will be written, but is not managed by the system.

5.3 CODING OF THE WAVEFLOW MODULE RADIO ADDRESS

A barcode label is applied to each module, indicating the WaveFlow module radio address. This address may be given in two forms:

- either with direct display of the radio address: 12 digits indicating the hexadecimal radio address of the module;
- or in the form of a serial number: in this case, the radio address is coded in the first 15 digits of the serial number; the other digits represent the CRC (algorithm available on request – contact technicalsupport@coronis-systems.com).
  To find the radio address in a serial number, proceed as follows:



00278-04-03153276
012

➢ ***Based on the serial number:*** The radio address (hexadecimal) is compiled as follows.

<table>
<tr><td colspan="3">Address indicated on the barcode (without CRC): 00278-04-03153276</td></tr>
<tr><td colspan="3">The chain of characters is split into 3 sections (as indicated below)</td></tr>
<tr><td>00278</td><td>04</td><td>03153276</td></tr>
<tr><td>Conversion
Decimal to Hexadecimal
(on 2 bytes)</td><td>Conversion
Decimal to Hexadecimal
(on 1 byte)</td><td>Conversion
Decimal to Hexadecimal
(on 3 bytes)</td></tr>
<tr><td>0116</td><td>04</td><td>301D7C</td></tr>
<tr><td colspan="3">A combination of these 3 parts provides the radio address (hexadecimal) of the module: 011604301D7C</td></tr>
</table>

6 WAVEFLOW MODULE FUNCTIONS

6.1 PARAMETER SETTING OF THE WAVEFLOW MODULE

some of the parameters of module information are accessible by standard internal parameters read and write commands, described in chapter 5.2; whereas other parameters of information are accessible by specific applicative commands.

6.1.1 Reading of the module type

The module type is obtained by interrogating the WaveTherm module, with the GET_TYPE command.

Applicative Commands	Description
0x20	Request to read the module type
0xA0	Response to the request to read the module type

Remark : *In the command byte coding, the Response frame type are taking the Request command byte value with the MSB bit set to 1.*

- Contents of request REQ_SEND_FRAME

Data Field (max : 152 bytes)	
Applicative command	**-**
1 bytce	-
0x20	-

- Contents of response RECEIVED_FRAME

Data Field (max : 152 bytes)				
Acknowledgement of the applicative command	**Module Type**	**RSSI level ***	**Current Awakening period (in second)**	**Equipment Type**
1 byte	1 byte	1 byte	1 byte	1 byte
0xA0	0x16 : WaveFlow		Default value = 0x01	0x16 for WaveFlow

**** the RSSI level (Received Signal Strength Indicator) indicates the reception level of the received frame. This parameter can be used for the installations but is not useful from an application point of view.***

The equipment type indicates if a Wavenis module is integrated in more complex equipment. The equipment type, and module type have the same value.

Remark : *The RSSI measurement WaveFlow - 4800 is different from the same measurement done on a Is)FlowMeter; that is due to a difference on the hardware of the two boards.*

6.1.2 Reading of the firmware version

The firmware version of the module is obtained by interrogating the distant module with GET_FIRMWARE_VERSION command.

Applicative command	Description
0x28	Request of reading the firmware version (GET_FIRMWARE_VERSION)
0xA8	Response to the request of reading the firmware version

- Contents of request REQ_SEND_FRAME

Data Field (max : 152 bytes)	
Applicative command	**Data**
1 byte	-
0x28	-

- Contents of response RECEIVED_FRAME

Data Field (max : 152 bytes)			
Acknowledgement of the applicative command	**Character 'V' in ASCII format**	**Mode of transmission**	**Firmware version**
1 byte	1 byte	2 bytes	2 bytes
0xA8	0x56	Default value = 0x00A3	

Remark *: Possible values for the mode of transmission,*

	Value
868 MHz single channel 4800 baud	0x0012
868 MHz frequency hopping 9600 baud	0x00A3
868 MHz single channel 9600 baud, with channel selection	0x00A2

6.1.3 Reading the number of inputs to be used

The number of inputs to be used is coded in the bits b1 and b0 of the Operation Mode byte.

Applicative Command	Internal parameter associated	Description
0x10	0x01	Request to read the number of inputs to be used
0x90	0x01	Response

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)			
Aplicative command	**Number of parameters to be read**	**Number of the first parameter**	**Length of the first parameter**
1 byte	1 byte	1 byte	1 byte
0x10	0x01	0x01	0x01

- RECEIVED_FRAME response

Data Field (max : 152 bytes)				
Acknowledgement of the applicative command	**Number of parameters read**	**Number of the first parameter**	**Length of the first parameter**	**Value of the first parameter**
1 byte	1 byte	1 byte	1 byte	1 byte
0x90	0x01	0x01	0x01	**(1)**

(1) *The value returned is the 'Operation Mode' byte. The number of inputs to be used is coded in the bits b1, and b0.*

WaveFlow managing 2 inputs	WaveFlow – 4 inputs
Bit b0 **0** : 1 input used for water counting (A) **1** : 2 inputs used for water counting (A-B)	**Bits [b1:b0]** **00** : 1 input used for water counting (A) **01** : 2 inputs used for water counting (A-B) **10** : 3 inputs used for water counting (A-B-C) **11** : 4 inputs used for water counting (A-B-C-D)

6.1.4 Initialization of the inputs to be used

The number of inputs to be used is coded in the bits b1 and b0 of the Operation Mode byte.

Applicative Command	Internal parameter associated	Description
0x10	0x01	Request to initialize the number of inputs to be used
0x90	0x01	Response

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)				
Applicative command	**Number of parameter to be written**	**Number of the first parameter**	**Length of the first parameter**	**Value of the first parameter**
1 byte	1 byte	1 byte	1 byte	1 byte
0x11	0x01	0x01	0x01	**Operation Mode**

- RECEIVED_FRAME response

Data Field (max : 152 bytes)			
Acknowledgement of the applicative command	**Number of parameter written**	**Numéro du 1er paramètre**	**Writing status**
1 byte	1 byte	1 byte	1 byte
0x91	0x01	0x01	0x00 : update OK 0xFF : update Error

6.1.5 Definition of the pulse weight

The pulse weight is used to convert the water consumption read in number of pulses, in volume unit (litre for example).

The WaveFlow modules provides up to 4 parameters (according to the module type) to store the pulse weight associated with the water meter.

ATTENTION, the pulse weight will only be stored for informative purpose. It will not be used by the module to convert automatically the readings.

Because of the wide range of pulse weight, all operations are processed in number of pulses, it is up to the user to convert the information in volume unit.

Applicative command	Associated internal parameters	Description
0x10	0xA3 ; 0xA4 0xA5 ; 0xA6	Request to read the pulse weight parameters
0x90	0xA3 ; 0xA4 0xA5 ; 0xA6	Response to the reading of the pulse weight
0x11	0xA3 ; 0xA4 0xA5 ; 0xA6	Request to set the pulse weight
0x91	0xA3 ; 0xA4 0xA5 ; 0xA6	Response to the pulse weight configuration

ATTENTION, the parameters 0xA5, and 0xA6 (pulse weight of inputs C, and D) are only existing on the WaveFlow – 4 inputs. For other module types, these two parameters are not used.

- **Reading and Writing the pulse weight**

The pulse weight parameters are reached by standard reading, and writing commands (described in chapter 5.2).

- 0xA3 *pulse weight of the input A*
- 0xA4 *pulse weight of the input B*
- 0xA5 *pulse weight of the input C (if supported)*
- 0xA6 *pulse weight of the input D (if supported)*

- **Definition of the pulse weight parameters**

MSB							LSB
b7	b6	b5	b4	b3	b2	b1	b0
Volume Unit				Pulse Weight			

- ***Pulse Weight :*** The range is from 0 to 15. In standard application, the zero value is prohibited.

- ***Volume Unit :*** this unit is expressed by 10^n (where n is the value contained by the bits [b7:b4].
  In order to standardize the information, the minimum unit is the millitre (ml).
  I.e. If [b7:b4] = 0 ; then the unit equal to 10^0 = 1 millilitre. (see table next page).

This table summarizes the unit used :

Unit used (hexadecimal) [b7:b4]	Interpreted value	Unit
0	10^0	Millilitre (ml)
1	10^1	Centilitre (cl)
2	10^2	Decilitre (dl)
3	10^3	Litre (l)
4	10^4	Decalitre (dal)
5	10^5	Hectolitre (hl)
6	10^6	M^3 (≈ kilolitre)
7	10^7	-
8	10^8	-
9	10^9	-
A	10^{10}	-
B	10^{11}	-
C	10^{12}	-
D	10^{13}	-
E	10^{14}	-
F	10^{15}	-

6.1.6 Reading of the date and time of the module

Applicative command	Description
0x12	Request to read the date and time of the module
0x92	Response to request to read the date and time of the module

Remark : *In the command byte coding, the Response frame type are taking the Request command byte value with the MSB bit set to 1.*

- Contents of request REQ_SEND_FRAME

Data Field (max : 152 bytes)	
Applicative command	**Data**
1 byte	-
0x12	-

- Contents of response RECEIVED_FRAME

Data Field (max : 152 bytes)						
Acknowledgement of the applicative command	**Day**	**Month**	**Year**	**Day of the week**	**Hour**	**Minute**
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte
0x92			**(1)**	**(2)**		

(1) *Year = current year - 2000*

(2) *day of the week : value from 0 to 6*

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

6.1.7 Setting the date and time of the module

Applicative command	Description
0x13	Request to set the date and time of the module
0x93	Response to request to set the date and time of the module

Remark : *In the command byte coding, the Response frame type are taking the Request command byte value with the MSB bit set to 1.*

- Contents of request REQ_SEND_FRAME

Data Field (max : 152 bytes)						
Applicative command	**Day**	**Month**	**Year**	**Day of the week**	**Hour**	**Minute**
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte
0x13			**(1)**	**(2)**		

(1) *Year = current year - 2000*

(2) *day of the week : value from 0 to 6*

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

- Contents of response RECEIVED_FRAME

Data Field (max : 152 bytes)		
Acknowledgement of the applicative command	**Update Status**	**-**
1 byte	1 byte	-
0x93	0x00 : update OK 0xFF : update error	-

6.2 INDEXES MANAGEMENT

Compatibility: All modules support the indexes management.

6.2.1 Compatibility with the Is)FlowMeter module

The WaveFlow module maintains a compatibility with the applicatives commands of the Is)FlowMeter module which managed only two indexes without water backflow detection.
Additional applicatives commands are thus implemented.

The table below gives the list of the applicatives commands, relating to the indexes management.

Applicative Command	Description	Is)FlowMeter	WaveFlow
0x01	Request for immediate reading of indexes	X	X
0x81	Response to the immediate reading	X	X
0x02	Request to write new indexes	X	X
0x82	Response to the writing new indexes request	X	X
0x03	Request to read the datalogging storage table (24 values) for inputs A and B	X	X
0x83	Response to the reading of the datalogging storage table	X	X
0x05	Request for a global reading of indexes		X
0x85	Response to the global reading		X
0x06	Request for an extended reading of the indexes	X	X
0x86	Response to the extended reading	X	X

Remark : *In the command byte coding, the Response frame type are taking the Request command byte value with the MSB bit set to 1.*

6.2.2 Immediate Reading of the indexes

Compatibility : the immediate reading is supported by all modules.

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)
Applicative Command
1 byte
0x01

- RECEIVED_FRAME response

Data Field (max : 152 bytes)				
Acknowledgement of the applicative command	**Operation Mode**	**Application Status**	**Index A**	**Index B**
1 byte	1 byte	1 byte	4 bytes (MSB First)	4 bytes (MSB First)
0x81				**(1)**

***(1)** indexes relating to unused inputs are automatically set to zero.*

The A and B indexes contains the information of the effective water consumption.

For the WaveFlow modules managing the water backflow, the value of these backflow indexes is substracted from the effective water consumption.

6.2.3 Global reading of the indexes

Compatibility : this functionnality is not supported by the WaveFlow – 4800 to maintain compatibility with the Is)FlowMeter module.

In addition, the response to a global reading is different according to the module used.

The global reading allows to read the index of each input., in a single frame.

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)
Applicative command
1 byte
0x05

- RECEIVED_FRAME response

Data Field (max : 152 bytes)						
Acknowledgement of the applicative command	**Operation Mode**	**Application Status**	**Index A**	**Index B**	**Index C (*)**	**Index D (*)**
1 byte	1 byte	1 byte	4 bytes (MSB first)	4 bytes (MSB first)	4 bytes	4 bytes
0x85						

The indexes A and B indicates the effective water consumption.
(*) The indexes C and D are different according to the type of module used :

Module type	Description of Index C and D
WaveFlow – 4 inputs	indicates the effective water consumption on C and D inputs (coded in MSB first)
WaveFlow – Standard WaveFlow – Standard – CYBLE 5 wire WaveFlow – specific backflow	indicates the effective backflow (index C for backflow of input A ; and index D for backflow of input B). (coded in LSB first)

6.2.4 Immediate writing of the indexes

Compatibility : this functionnality is supported by all module types. However, the command used to set the indexes, has different format according to the module type used.

The immediate writing of the indices allows to initialize the index of each input(except inputs relating with water backflow).

WaveFlow measures, and count pulses coming from the pulses transmitter. It is thus necessary to establish the link between the index of the meter given in volume unit (litre, for example), and the number of pulses counted by the WaveFlow.

Exemple : *Ithe index of the water meter indicates 1000 litre.*

- if the pulse transmitter is from type ***k = 0,1*** *(1 pulse per decilitre)*
10 000 pulses represent 1000 litres, so the value to be written into the WaveFlow will be 10 000.

- if the pulse transmitter is from type ***k = 1*** *(1 pulse per litre)*
the number of pulses represent the water volume, so the value to be written into the WaveFlow will be 1000.

- if the pulse transmitter is from type ***k = 10*** *(1 pulse per decalitre)*
100 pulses represent 1000 litres, so the value to be written into the WaveFlow will be 100.

- if the pulse transmitter is from type ***k = 100*** *(1 pulse per hectolitre)*
10 pulses represent 1000 litres, so the value to be written into the WaveFlow will be 10.

In order to know through a radio command, the pulse weight of the meter connected to the WaveFlow, the module integrates specific parameters for each input allowing to store the corresponding pulse weight.

These parameters are purely informative, and are not used by the applicative part of WaveFlow. These parameters must be to configure at the same time as the initialization of the indexes, according to the pulse weight of the water meter. See chapter 6.1.5.

➢ REQ_SEND_FRAME request

Data Field (max : 152 bytes)					
Applicative command	**Writing type**	**Index A**	**Index B**	**Index C**	**Index D**
1 byte	1 byte	4 bytes (MSB first)	4 bytes (MSB first)	4 bytes (MSB first)	4 bytes (MSB first)
0x02	(*)				

- Writing type : *indicates which index has to be written.*

Writing type byte							
b7	b6	b5	b4	b3	b2	b1	b0
Not used	Not used	Not used	Not used	**Index D** 0: skip 1: write	**Index C** 0: skip 1: write	**Index B** 0: skip 1: write	**Index A** 0: skip 1: write

- Index C ; Index D : *this two indexes has only to be initialized for the WaveFlow – 4 inputs.*

For modules managing backflow, when an input is initialized (A or B) the corresponding backflow index is automatically set to zero.

- RECEIVED_FRAME response

Data Field (max : 152 bytes)	
Acknowledgement of the applicative command	**Writing status**
1 byte	1 byte
0x82	0x00 : writing OK 0xFF : writing error

6.2.5 Extended reading of the indexes

Compatibility : the extended reading is supported by all module types

The extended reading is the reading mode the most detailled, it returns the following pieces of information :

- Control parameters : 'Operation Mode' and 'Application Status' ;
- the immediate indexes (effective water consumption) ;
- the value of the indexes at the last end of month ;
- the four last logged value in the datalogging table ;
- the date of the last logged value in the datalogging mode ;
- and the datalogging measurement period (period used when datalogging in time steps, refer to 6.4.3)

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)
Applicative command
1 byte
0x06

- RECEIVED_FRAME response

Data Field (max : 152 bytes)							
Ack. of applicative command	**Operation Mode**	**Application Status**	**Immediate indexes area**	**indexes of end of month**	**4 last logged indexes**	**Date of last logged value**	**Datalog. Measurement period**
1 byte	1 byte	1 byte	Variable **(*)**	Variable **(*)**	Variable **(*)**	6 bytes	1 byte
0x86							Refer to 6.4.3

(*) the format and the size of the fields 'Immediate indexes area' ; 'indexes of end of month' ; and '4 last logged indexes' depends of the number of inputs activated.

The table below describes the format and the size of these fields :

Number of inputs	Immediate indexes area	indexes of end of month	4 last logged indexes
1	Size : 4 bytes [b3:b0] : index A	Size : 4 bytes [b3:b0] : index A – end of month	Size : 16 bytes [b15:b0] : 4 last index A
2	Size : 8 bytes [b3:b0] : index A [b7:b4] : index B	Size : 8 bytes [b3:b0] : index A – end of month [b7:b4] : index B – end of month	Size : 32 bytes [b15:b0] : 4 last index A [b31:b16] : 4 last index B
3	Size : 12 bytes [b3:b0] : index A [b7:b4] : index B [b11:b8] : index C	Size : 12 bytes [b3:b0] : index A – end of month [b7:b4] : index B – end of month [b11:b8] : index C – end of month	Size : 48 bytes [b15:b0] : 4 last index A [b31:b16] : 4 last index B [b47:b32] : 4 last index C
4	Size : 16 bytes [b3:b0] : index A [b7:b0] : index B [b11:b8] : index C [b15:b12] : index D	Size : 16 bytes [b3:b0] : index A – end of month [b7:b0] : index B – end of month [b11:b8] : index C – end of month [b15:b12] : index D – end of month	Size : 64 bytes [b15:b0] : 4 last index A [b31:b16] : 4 last index B [b47:b32] : 4 last index C [b63:b48] : 4 last index D

6.3 WAKE-UP SYSTEM MANAGEMENT

Compatibility : cette fonction est supportée par tous les modules

In order to reduce module power consumption, a wake-up period parameter setting system is incorporated. This system enables modification of the module wake-up period (default setting 1 s) by entering a time and day of the week :

- The wake-up period default value may be modified;
- Two time-windows with different wake-up periods may be defined;
- Each day of the week may be set in one of the following three cases :
  - Wake-up period default setting
  - Wake-up according to predefined time windows
  - No wake-up period (for safety reasons, the module is not disabled on reception and it wakes up every 10 seconds)

6.3.1 - Description of the parameters used

Parameter number	Description	Size (in bytes)	Access rights	Default value
0x01	Operating mode	1	R/W	Depending of the module used (see 5.1.1)
0x02	Wake-up system status word	1	R/W	0x00
0x03	Default wake-up period (in s)	1	R/W	0x01
0x04	Start time for 1st time window (in hour)	1	R/W	0x07
0x05	Wake-up period - 1st time window (in s)	1	R/W	0x01
0x06	Start time for 2nd time window (in hour)	1	R/W	0x12
0x07	Wake-up period - 2nd time window (in s)	1	R/W	0x01
0x08	Enable time windows by day of week	1	R/W	0xFF
0x09	Enable wake-up periods by day of week	1	R/W	0x00

6.3.2 - Choice of wake-up mode

These modes are directly dependant on the *'wake-up system status word'* configuration and the values of parameters associated with each mode.

Wake-up system status word							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	-	**Enable day-of-week selection** 0 : disabled 1 : enabled	**Enable time windows** 0 : disabled 1 : enabled

Wake-up system status word	Wake-up mode
0x00	Case n°1 : Periodic wake-up, without distinction of day of the week
0x01	Case n°2 : Periodic wake-up in specific time windows for certain days of the week, periodic wake-up for the other days
0x02	Case n°3 : Periodic wake-up for certain days of the week, periodic wake-up disabled for the other days
0x03	Case n°4 : Periodic wake-up in specific time windows for certain days of the week, periodic wake-up for some days and periodic wake-up disabled for the remaining days

Remark: before enabling a specific wake-up mode, the parameters associated with this mode must first be set.

6.3.3 - Set a new wake-up period

The WaveTherm module wake-up default setting is every second. The wake-up period may be easily modified by entering a new value in the '*default wake-up period parameter*'.

Attention, the value associated with this parameter may not exceed 10 seconds.

Attention, an erroneous value of this parameter involves a wake-up every second, the maximum value is 0x0A (10 seconds).

6.3.4 - Set a fixed wake-up period for certain days of the week

The wake-up system parameters may be set to allow disabling of WaveTherm module periodic wake-up for certain days of the week.

In practice, when periodic wake-up is disabled, the WaveLog polls every 10 seconds.

The parameter setting procedure is as follows :

- ***disable periodic wake-up for certain days****, with the 'Enable periodic wake-up for certain days of the week' parameter.*
- ***Enable selection of the days of the week****, with the 'wake-up system status word' parameter;*
  'wake-up system status word' = 0x02

In this way, on days when periodic wake-up is disabled, the module polls every 10 seconds, whereas for the rest of the week the module wakes up at the default period setting.

Enable wake-up periods by day of week							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	**Sunday** 0 : enabled 1 : disabled	**Saturday** 0 : enabled 1 : disabled	**Friday** 0 : enabled 1 : disabled	**Thursday** 0 : enabled 1 : disabled	**Wednesday** 0 : enabled 1 : disabled	**Tuesday** 0 : enabled 1 : disabled	**Monday** 0 : enabled 1 : disabled

Attention, the coding of each bit is reversed, compared to the parameter ' Enable time windows by day of the week'.

6.3.5 - Set day/night system parameter without distinction of days of the week

The wake-up system parameters may be set to enable configuration of the time windows with different wake-up periods.

Whether the user wants a distinction of the days of the week, or not; the parameter ' Enable time windows by day of the week' must be suitably configured. Thus in the case of a time windows activation without distinction of the days of week, every day of the week must be enabled (' Enable time windows by day of the week' = 0x7F).

The time windows function as follows,

- Set the start time for the first time window and its wake-up period ;
- Set the start time for the second time window and its wake-up period ;
- Select the days of the week during which the time windows are enabled ;

'Enable time windows by day of the week' = 0x7F

- Validate the time window mode with the 'wake-up system status word'.

'Wake-up system status word' = 0x01

Note: The format of the parameters 'Start time for time windows' is expressed in hour, and its value lies between 0 and 23.

For example, if the start time of time windows at 12h00; the value 0x0C should be configured.

The format of the wake-up periods of each time windows, is the same one as the default wake-up period; i.e. it is expressed in seconds, and cannot exceed value 0x0A.

The format of the applicatives commands for reading and writing internal parameters, is described in chapter 5.2.

Enable time windows by day of week							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	**Sunday** 0 : enabled 1 : disabled	**Saturday** 0 : enabled 1 : disabled	**Friday** 0 : enabled 1 : disabled	**Thursday** 0 : enabled 1 : disabled	**Wednesday** 0 : enabled 1 : disabled	**Tuesday** 0 : enabled 1 : disabled	**Monday** 0 : enabled 1 : disabled

Attention, the coding of each bit is reversed, compared to the parameter ' Enable wake-up periods by day of the week'.

6.3.6 - Set the day/night system parameters according to day of the week

The day/night system according to the day of the week parameter setting procedure is the same as that described in the previous chapter with the exception that the “*Enable time window according to the day of the week*” parameter is only set for days required.

For example, we wish to enable the time window from Monday to Wednesday.

The *'Enable time windows according to day of the week' parameter is set to* 0x07.
In this way, the module wakes up during these time windows for a period set in the associated parameters with a specific start time for each window from Monday to Wednesday.

For the other days of the week, the wake-up mode depends on the '*wake-up system status word*' :

- '*Wake-up system status word*' = 0x01

  the bit 1 is not enabled, thus the parameter 'Enable wake-up periods by day of week' is not used ; so the rest of the week, the module uses the default wake-up period.

- '*Wake-up system status word*' = 0x03

  the bit 1 is enabled, thus the parameter 'Enable wake-up periods by day of week' is used, so the wake-up for the rest of the week will depend on this parameter.

  Days with wake-up period enabled : default wake-up period

  days with wake-up period disabled : periodic wake-up disabled (polling every 10 sec.)

The format of the applicatives commands for reading and writing internal parameters, is described in chapter 5.2.

6.4 PARAMETER SETTING OF THE DATALOGGING MODE

Compatibility: this functionnality is supported by all the modules.

The Datalogging mode enables periodic logging of indexes of each input . The frequency of these readings may be set in three modes:

- index logging in time steps
- index logging once a week
- index logging once a month

This periodic reading allows to store globally up to 24 indexes, when the storage table is full, the most recent measurements crush the oldest values.

6.4.1 Description of the parameters used

According to the datalogging type (time steps, once a week, once a month), some specific parameters must be used.

Parameter number	Description	Size (in bytes)	Access Right	Default value
0x01	Operating mode	1	R/W	Depending of the module used (see 6.1.5)
0x80	Measurement period (datalogging in time steps)	1	R/W	0x13
0x81	Start hour of the datalogging in time steps	1	R/W	0x08
0x82	Day of the week, or of the month	1	R/W	0x01
0x83	Time of measurement (datalogging once a week, or once a month)	1	R/W	0x08

These specific parameters must be initialized before activating the datalogging mode.

6.4.2 Activating the datalogging mode

The datalogging mode is activated (or deactivated) by setting the bits [b3:b2] in the *Operating Mode* byte.

MSB **LSB**

Operating Mode							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reed Fault detection **0** : deactivated **1** : activated	**Extreme Leak detection (High Threshold)** **0** : deactivated **1** : activated	**Residual Leak detection (Low Threshold)** **0** : deactivated **1** : activated	**WireCut detection** **0** : deactivated **1** : activated	**Datalogging** **00** : deactivated **01** : time steps **10** : once a week **11** : once a month		**Selection of the number of input to be used (see 5.1.1)**	

Attention, stopping then restarting the datalogging mode induce the re-initialization of the storage table. In this case, the logged values will be lost.

6.4.3 Index logging in time steps

This type of datalogging is used to log the index value for each input at periods ranging from one minute to over thirty hours. The time of the first logging may be set with a parameter.

When the datalogging mode in time steps is enabled, the system only logs the memorised index values as soon as the preset time is attained; and this until the datalogging mode is disabled.

The parameters to be used, are the following :

- measurement period of the datalogging in time steps (parameter 0x80)
- start hour of the datalogging in time steps (parameter 0x81)

> ***These parameters must by initialized before activating the datalogging mode.***
>
> ***The format of the applicatives commands for reading and writing internal parameters, is described in chapter 5.2.***

➢ Measurement period of the datalogging in time steps (parameter 0x80)

Measurement period of the datalogging in time steps (parameter 0x80)							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
[b7:b2] : measurement period expressed in time units min : once a minute max : 63 times 30 minutes = 31 h 30 min						**[b1:b0] : time unit** 00 : 1 minute 01 : 5 minutes 10 : 15 minutes 11 : 30 minutes	

➢ Start hour of the datalogging in time steps (parameter 0x81)

This parameter allows to synchronize the periodic measurement. It is expressed in multiples of hour, and its value must set from 0 to 23.
When the datalogging mode in time steps is enabled, the system only logs the memorised index values as soon as the preset time is attained

> ***Example :*** *a user wants to log the indexes every 4 hours. And the first value mus be log at 19:00.*
>
> - *Parameter 'measurement period of the datalogging in time steps' (0x80) = 0x23*
> - *Parameter 'start hour of the datalogging in time steps ' (0x81) = 0x13*
> - *bits [b3:b2] of 'Operating Mode' byte (0x01) set to '01'*

6.4.4 Index logging once a week

This type of datalogging is used to log the index value for each input once a week. The time and day of the week logging is carried out may be set with a parameter.

The parameters to be used, are the following :

- Time of measurement (datalogging once a week, or once a month) (parameter 0x83)
- Day of the week, or of the month (datalogging once a week, or once a month) (parameter 0x82)

> ***These parameters must by initialized before activating the datalogging mode.***
>
> ***The format of the applicatives commands for reading and writing internal parameters, is described in chapter 5.2.***

- **Time of measurement (datalogging once a week, or once a month) (parameter 0x83)**

This parameter allows to synchronize the periodic measurement. It is expressed in multiples of hour, and its value must set from 0 to 23.

- **Day of the week, or of the month (datalogging once a week, or once a month) (parameter 0x82)**

Value	Day of the week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

> ***Example :*** *a user wants to log the indexes every monday, at 12:00 am.*
>
> - *Time of measurement (datalogging once a week, or once a month) (0x83) = 0x0C*
> - *Day of the week, or of the month (0x82) = 0x01*
> - *bits [b3:b2] of 'Operating Mode' byte (0x01) set to '10'*

6.4.5 Index logging once a month

This type of datalogging is used to log the index value for each input once a month. The time and day (from 1 to 28) logging is carried out may be set with a parameter.

The parameters to be used, are the following :

- Time of measurement (datalogging once a week, or once a month) (parameter 0x83)
- Day of the week, or of the month (datalogging once a week, or once a month) (parameter 0x82)

> ***These parameters must by initialized before activating the datalogging mode.***
>
> ***The format of the applicatives commands for reading and writing internal parameters, is described in chapter 5.2.***

- **Time of measurement (datalogging once a week, or once a month) (parameter 0x83)**

This parameter allows to synchronize the periodic measurement. It is expressed in multiples of hour, and its value must set from 0 to 23.

- **Day of the week, or of the month (datalogging once a week, or once a month) (parameter 0x82)**

Here, the format is different from the datalogging once a week. Indeed, the day of measurement is set from 0 to 28. And, the system does not manage changes in the number of days in a month.

> ***Example*** *: a user wants to log the indexes the 5th day of the month, at 12:00.*
>
> - *Time of measurement (datalogging once a week, or once a month) (0x83) = 0x0C*
> - *Day of the week, or of the month (0x82) = 0x05*
> - *bits [b3:b2] of 'Operating Mode' byte (0x01) set to '11'*

6.4.6 Reading the logged values

The reading of the logged values is accomplished by sending the following commands :

Applicative Command	Description
0x03	Request to read the datalogging storage table for inputs A and B (24 indexes)
0x83	Response to the request to read the datalogging storage table for inputs A and B
0x07	Request to read the datalogging storage table for inputs C and D (24 indexes)
0x87	Response to the request to read the datalogging storage table for inputs C and D

The format is as follow :

- Contents of the request REQ_SEND_FRAME

Data Field (max : 152 bytes)
Applicative Command
1 byte
0x03 or 0x07

- Contents of the response RECEIVED_FRAME

Data Field (max : 152 bytes)					
Acknowledgement of the applicative command	**Operating Mode**	**Application Status**	**Storage table of the indexe**	**Date of the last logged index**	**Measurement period**
1 byte	1 byte	1 byte	96 bytes	6 bytes	1 byte
0x83 or 0x87			**(3)**	**(2)**	**(1)**

(1) Measurement period : *the format of this byte is given in chapter 6.4.3.*

ATTENTION : The 'Measurement Period' field is updated only when datalogging in time steps is enabled. In the other modes, this field is not significant.

(2) Date of the last logged temperature : *the format of this field is equivalent to the standard date and time format (see chapter 6.1.6).*

(3) Storage table of the indexes : *the format is different according to the number of inputs used, see next page.*

(3) The storage table of the indexes *is defined as follows : if T is the measurement period of the datalogging.*

Number of inputs used	Number of values per counter	Structure of the storage table		Applicative command to be used
1	24 indexes A	Byte 1 to 96	Index A (instant t) index A (instant t – 1T) index A (instant t – 2T) ... index A (instant t – 23T)	Read by command 0x03
2	12 indexes A	Byte 1 to 48	Index A (instant t) index A (instant t – 1T) ... index A (instant t – 11T)	Read by command 0x03
	12 indexes B	Byte 49 to 96	Index B (instant t) index B (instant t – 1T) ... index B (instant t – 11T)	Read by command 0x03
3	12 indexes A	Byte 1 to 48	Index A (instant t) index A (instant t – 1T) index A (instant t – 2T) ... index A (instant t – 11T)	Read by command 0x03
	12 indexes B	Byte 49 to 96	Index B (instant t) index B (instant t – 1T) ... index B (instant t – 11T)	Read by command 0x03
	12 indexes C	Byte 1 to 48	Index C (instant t) index C (instant t – 1T) ... index C (instant t – 11T)	Read by command 0x07 the bytes 49 to 96 are set to 0xFF
4	12 indexes A	Byte 1 to 48	Index A (instant t) index A (instant t – 1T) index A (instant t – 2T) ... index A (instant t – 11T)	Read by command 0x03
	12 indexes B	Byte 49 to 96	Index B (instant t) index B (instant t – 1T) ... index B (instant t – 11T)	Read by command 0x03
	12 indexes C	Byte 1 to 48	Index C (instant t) index C (instant t – 1T) ... index C (instant t – 11T)	Read by command 0x07
	12 indexes D	Byte 49 to 96	Index D (instant t) index D (instant t – 1T) ... index D (instant t – 11T)	Read by command 0x07

6.5 CONFIGURATION OF THE LEAKAGE DETECTION

Compatibility : This functionnality is supported by all the module types.

The WaveFlow module detects two types of leaks for each metering input, residual leaks and extreme leaks. For each of these types of leaks, the module carries out the following operations :

- Leak detection
- Date of leak detection recording
- Recording of detection date and min. (or max.) flow-rate.

If the leak stops :

- Date
- Recording of the leak stoppage date and last flow-rate detected

6.5.1 Description of the parameters used

Parameter number	Description	Size (in bytes)	Access Right	Default value
0x01	Operation Mode	1	R/W	0x09
0xC4	Measurement Step, exepressed in multiple of minutes	1	R/W	0x3C
0x88	Residual leakage flow (low threshold) for **input A** (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x8A	Residual leakage detection period for **input A** (expressed in multiple of measurement step)	1	R/W	0xA8
0x89	Extreme leakage flow (high threshold) for **input A** (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0xC0	Extreme leakage detection period for **input A** (expressed in multiple of measurement step)	1	R/W	0x03
0x8B	Residual leakage flow (low threshold) for **input B** (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x8C	Extreme leakage flow (high threshold) for **input B** (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0x8D	Residual leakage detection period for **input B** (expressed in multiple of measurement step)	1	R/W	0xA8
0xC1	Extreme leakage detection period for **input B** (expressed in multiple of measurement step)	1	R/W	0x03
0x98	Residual leakage flow (low threshold) for input C (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x9A	Residual leakage detection period for input C (expressed in multiple of measurement step)	1	R/W	0xA8
0x99	Extreme leakage flow (high threshold) for input C (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0xC2	Extreme leakage detection period for input C (expressed in multiple of measurement step)	1	R/W	0x03
0x9B	Residual leakage flow (low threshold) for input D (expressed in number of pulse, per measurement step)	1	R/W	0x0F
0x9D	Residual leakage detection period for input D (expressed in multiple of measurement step)	1	R/W	0xA8

0x9C	Extreme leakage flow (high threshold) for input D (expressed **LSB FIRST** in number of pulse, per measurement step)	2	R/W	0x03E8
0xC3	Extreme leakage detection period for input D (expressed in multiple of measurement step)	1	R/W	0x03

6.5.2 Principle of the water flow measurement

On an input, a water volume is periodically measured (in number of pulses), according to the parameter 'measurement step' (0xC4).

On each measurement step, the previous water volume value is substracted from the current one. Thus, it gives the water flow espressed in number of pulses per measurement step.

> ***Measurement Step (0xC4) :*** *its value is used for leakage detection on all the inputs. It is expressed in multiple of minutes (from 0 to 255).*

6.5.3 Residual leak detection

Detection is enabled when the module detects that the instantaneous flow-rate (by default calculated every hour) is systematically higher than that set by the user (parameter *residual leak threshold*)for a given detection period (parameter *residual leak detection period*).

- ***Residual leak threshold*** : Detection threshold, expressed in number of pulses per measurement step. To link this value to the right pulse weight, please refer to chapter 6.1.5.
- ***Residual leak detection period*** : minimum time during which the threshold value must be exceeded before leak detection is validated (expressed in multiple of measurement step)

The parameters relative to this detection, has to be configured before activating the detection.

The residual leak detection is activated by setting the bit 5 of the *Operation Mode* parameter.

- Definition of the **Operation Mode** byte :

MSB **LSB**

Operation Mode							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reed Fault detection (1) **0** : deactivated **1** : activated	**Extreme Leak detection** **0** : deactivated **1** : activated	**Residual Leak detection** **0** : deactivated **1** : activated	**WireCut detection** **0** : deactivated **1** : activated	**Datalogging** **00** : deactivated **01** : by time steps **10** : once a week **11** : once a month		Management of the inputs. (see chapter 5.1.1)	

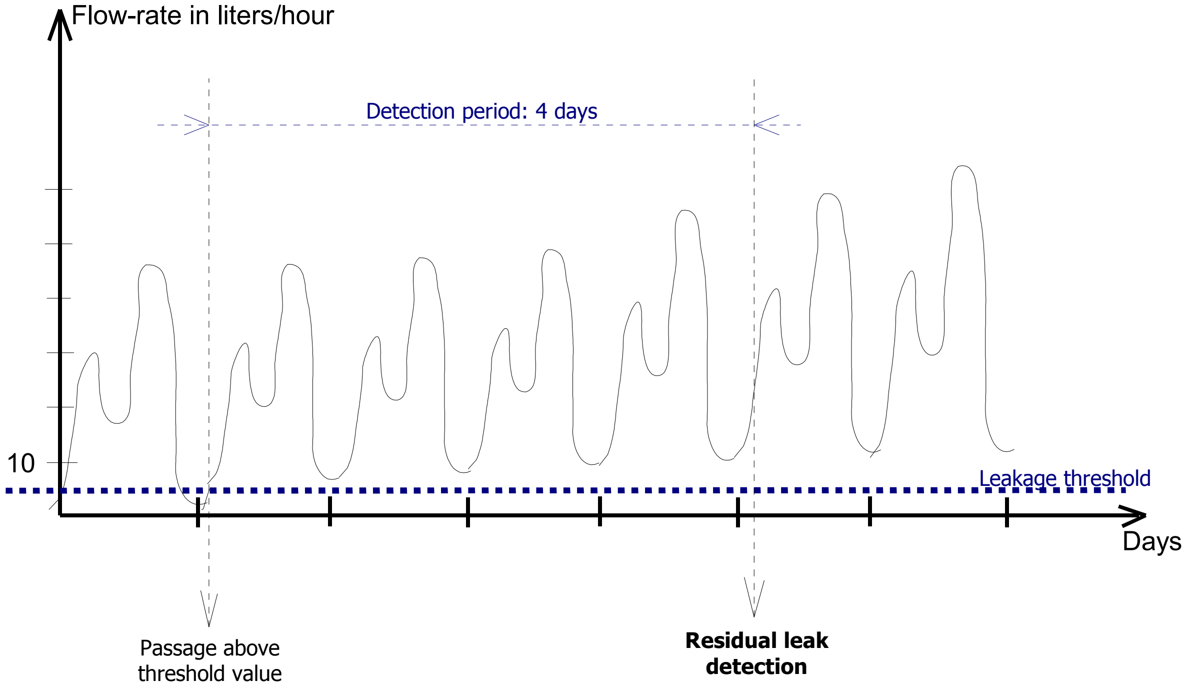
(1) Reed Fault detection : only if supported by the module, else this bit is not used. The Reed fault is supported by the WaveFlow – Standard, and WaveFlow – Specific Backflow.

- **Example :** The measurement step is set to measure the flow-rate in litres/hour and the residual leakage detection parameter is then set as follows:
  - ***Residual leak threshold :*** *5 litres per hour*

  the value of the parameter depends of the pulse weight :

    - *if k=1 , then Residual Leak Threshold = 0x05 (for 5 pulses)*
    - *if k=0,1 , then Residual Leak Threshold = 0x32 (for 50 pulses)*

  - ***Residual leak detection period :*** 4 days. *(= 0x60)*



Remark : *it is advised to configure the detection period value to several days (or a week) in order to avoid alarms when opening a tap.*

6.5.4 Extreme leak detection

Detection is enabled when the module detects a flow-rate higher than that set by the user in the *Extreme Leak Threshold* parameter for a given detection period (parameter *Extreme Leak Detection Period*).

The parameters relative to this detection, has to be configured before activating the detection.
The residual leak detection is activated by setting the bit 6 of the *Operation Mode* parameter.

- **Extreme Leak Threshold** : Detection threshold. Expressed in number of pulses per measurement step.

> ***Attention, this parameter is coded in LSB first, except for the WaveFlow – 4800 where it is coded in MSB First, to maintain compatibility with the Is)Flowmeter.***

- **Extreme Leak Detection Period** : minimum time during which the threshold value must be exceeded before leak detection is validated. Expressed in multiple of *Measurement Step*.

6.5.5 Description of data relating to leak detection

The WaveFlow stores in an internal table, the pieces of information relative to the occurrence, or the disappearance of the leaks.
The table is a circular buffer which can store up to 5 events, and has the following structure :

Status	Flow-rate	Date
1 byte	2 bytes	6 bytes
Status_Evt 0	Debit_Evt 0	Date_Evt 0
Status_Evt 1	Debit_Evt 1	Date_Evt 1
Status_Evt 2	Debit_Evt 2	Date_Evt 2
Status_Evt 3	Debit_Evt 3	Date_Evt 3
Status_Evt 4	Debit_Evt 4	Date_Evt 4

> ***Remark : Data are stored in a circular buffer which may be accessed by radio and contains the last 5 events logged(occurrence or stoppage of leaks).***

- **Status :** indicates the event type (occurrence or disappearance) and the corresponding input.

Status							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Corresponding input (*) **00** : input A **10** : input B **01** : input C **11** : input D		-	-	-	-	**Leak type** 0 : Extreme leak 1 : Residual leak	**Event Type** 0 : disappearance 1 : occurence

(*) For modules managing 2 inputs (water backflow type) the bit 6 is unused.

- **Flow-rate :** according to the event type, the flow-rate has different meaning,

  ➔ ***Occurrence of a residual leak :*** *minimum flow-rate value which is higher than the specified threshold, for the given Residual Leak Detection Period;*
  ➔ ***Disappearance of a residual leak :*** *minimum flow-rate value higher than the threshold, logged just before the disappearance of the leak;*
  ➔ ***Occurrence of an extreme leak :*** *maximal flow-rate value logged on the specified Extreme Leak Detection Period.*
  ➔ ***Disappearance of an extreme leak :*** *the value is not significant.*

- **Date :** the format of the date is strictly identical to the module date and time, described in chapter 6.1.6.

6.5.6 Reading the pieces of information relative to the leak detection

When a leak is detected, a bit is set in the *Application Status* parameter, and the pieces of information relative to the leak are stored in an internal table.
It is possible to reset the *Application Status* parameter, by writing it, or by reading the event table.

➢ **Application Status**

MSB **LSB**

Application Status							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
See chapter 5.1.1	*See chapter 5.1.1*	*See chapter 5.1.1*	**Extreme Leak**	**Residual Leak**	**WireCut on input B**	**WireCut on input A**	**End of Battery Life**
			0 : not detected **1 :** detected	**0 :** not detected **1 :** detected	**0 :** not detected **1 :** detected	**0 :** not detected **1 :** detected	**0 :** not detected **1 :** detected

When one of the bits relating to the leaks detection is set to one, it is possible to read the entire event table.

Remark : Events are significant, only if the date is valid. The fields not used are filled with 0xFF.

The commands used to read the event table are as follows :

Applicative Command	Description
0x04	Request to read the leak event table.
0x84	Response

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)
Applicative command
1 byte
0x04

- RECEIVED_FRAME response

Data Field (max : 152 bytes)					
Ack. of applicative command	**Most Recent Leak Event**	**Leak Event (n-1)**	**Leak Event (n-2)**	**Leak Event (n-3)**	**Leak Event (n-4)**
1 byte	9 bytes	9 bytes	9 bytes	9 bytes	9 bytes
0x84		**(*)**	**(*)**	**(*)**	**(*)**

***(*) :** when no event has been detected, the table is filled with 0xFF.*

6.6 SIMPLE WATER BACKFLOW DETECTION MODEL

Compatibility : this functionnality is only supported by the WaveFlow – Specific Backflow

The WaveFlow module is able to detect backflow consumption for which the critical flow-rate threshold parameter may be set. When this flow-rate is attained within a given period, a fault detection signal for the current month is transmitted in the ***'Flag indicating backflow detections per month'*** parameter.

In this case, only the water backflow occurrences for the current month are saved together with the effective backflow value.

This detection is activated by setting a value different of zero, in the ***'Backflow Detection Period'*** parameter (0xC5 et 0xC7).

Water backflow detection is enabled by default.

The flow-rate of the backflow is calculated according to the parameter***'Backflow Detection Period'***.

6.6.1 Description of the parameters used

Parameter number	Description	Size (in bytes)	Access Right	Default value
0x20	Application Status	1	R/W	0x00
0xC5	Backflow detection period on **input A** (expressed in multiple of hour)	1	R/W	0x01
0xC6	BackFlow threshold on **input A** (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xC7	Backflow detection period on **input B** (expressed in multiple of hour)	1	R/W	0x01
0xC8	BackFlow threshold on **input B** (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xC9	Flag indicating backflow detections per month, on **input A** **(LSB first)**	2	R/W	0x0000
0xCA	Flag indicating backflow detections per month, on **input B** **(LSB first)**	2	R/W	0x0000

6.6.2 Principle of the simple water backflow detection model

Water backflow is measured periodically according to the Detection Period parameter (0xC5, and 0xC7) expressed in hours.

A point T is given for each measurement; the module calculates the flow-rate according to the values detected at the T points (T-1).

This water backflow rate is processed and, if it exceeds the ***'BackFlow Threshold'*** parameter, water flow is signalled in the *Application Status* byte and the month this backflow occurred is memorised in the ***'Flag indicating backflow detections per month'*** parameter.

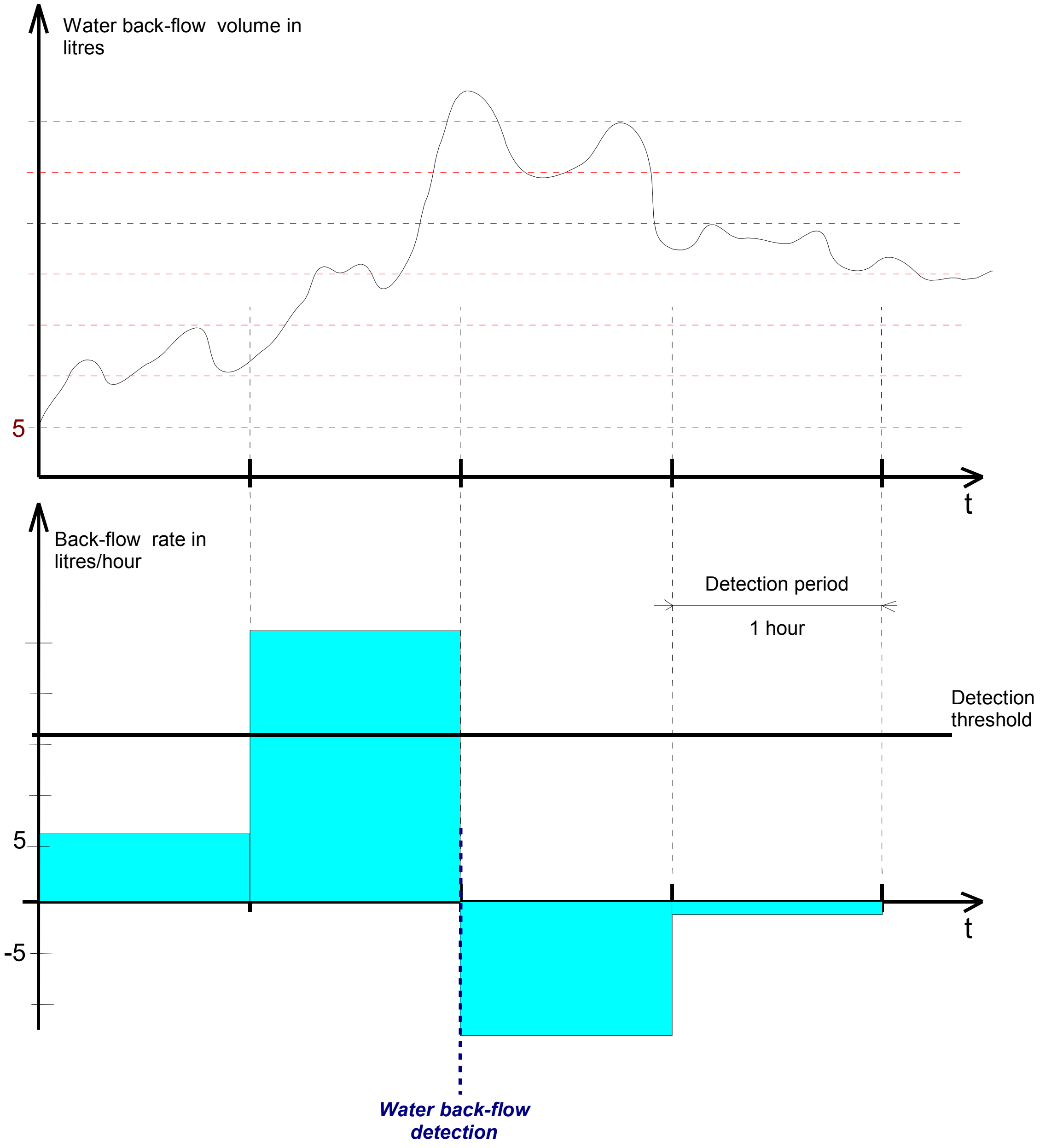
- **Example :** The pulse weight is set (see chapter 6.1.5) to measure the water volume in litres and the water backflow parameter is then set as follows:

  - ***Backflow Detection period :*** measurement in litres/hour, thus must be set to 0x01.
  - ***BackFlow threshold :*** threshold set to 16 litres/hour.

  Attention, the value of the parameter depends of the pulse weight :

  - if k=0,1 , then threshold = 160 pulses (=0xA0)
  - if k=1 , then threshold = 16 pulses (=0x10)

During each detection period, the module measures the flow-rate for the previous period in litres/ hour (see illustration).



6.6.3 Description of the Flag indicating backflow detections per month

When a backflow is detected, the least significant bit of the monthly flags parameter relating to the input where the backlow was detected, is set to 1.

during the transition from a month to another, all the bits of the monthly flags parameter are shifted left (from LSB to MSB).

Flag indicating backflow detections per month															
Most Significant Byte								Least Significant Byte							
B7	B6	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0
-	-	-	Month -12	Month -11	Month -10	Month -9	Month -8	Month -7	Month -6	Month -5	Month -4	Month -3	Month -2	Month -1	Current month

This flag is reseted by sending a write command on the parameter, with data 0x0000.

Attention, when reading these parameters, the contents is received with the Least Significant Byte first.

6.7 ADVANCED WATER BACKFLOW DETECTION MODEL

Compatibility : this functionnality is only supported by the following modules,

- ***WaveFlow – Standard***
- ***WaveFlow – Standard – CYBLE 5 wires***

The WaveFlow module is able to detect backflow consumption for which the critical flow-rate threshold parameter may be set. Two detection methods are available for this purpose and may be selected by writing a configuration parameter.

The module carries out the following operations for both methods:

- Water backflow detection,
  - for the first method, the backflow is detected as soon as the WaveFlow module measures a continuous water backflow volume higher than the set threshold value; the water volume measured in the positive direction is deducted from the backflow taken into account.
  - for the second method, the water backflow is detected as soon as the WaveFlow module measures a continuous water backflow rate higher than the set threshold value.
- Date of *'water backflow' occurrence,*
- Duration, or end of occurrence:
  - Date of end of 'backflow' occurrence for the first method;
  - Log occurrence duration for the second method.

6.7.1 Description of the parameters used

Parameter number	Description	Size (in bytes)	Access right	Default value
0x01	Operation Mode	1	R/W	0x09
0x20	Application Status	1	R/W	0x00
0x0A	Extended Operation Mode	1	R/W	0x00
0xCB	Backflow detection period on input A (expressed in multiple of 10 minutes)	1	R/W	0x01
0xCC	BackFlow threshold on input A (expressed in number of pulse per backflow detection period)	1	R/W	0x0A
0xCD	Backflow detection period on input B (expressed in multiple of 10 minutes)	1	R/W	0x01
0xCE	BackFlow threshold on input B (expressed in number of pulse per backflow detection period)	1	R/W	0x0A

6.7.2 Water backflow detection method with measurement of water volume

With this method, the water backflow volume is measured using a preset measurement unit. The water volume measured in the positive flow direction (not the backflow water) is subtracted from the water backflow volume.

Water backflow is measured as soon as it occurs, but the occurrence is only logged when the water backflow volume exceeds a preset threshold value (*backflow threshold* parameter).

The table is updated during the entire backflow duration and the maximum water volume recorded is saved as well as the date of end of detection.

End of detection takes place when the water backflow volume becomes stable.

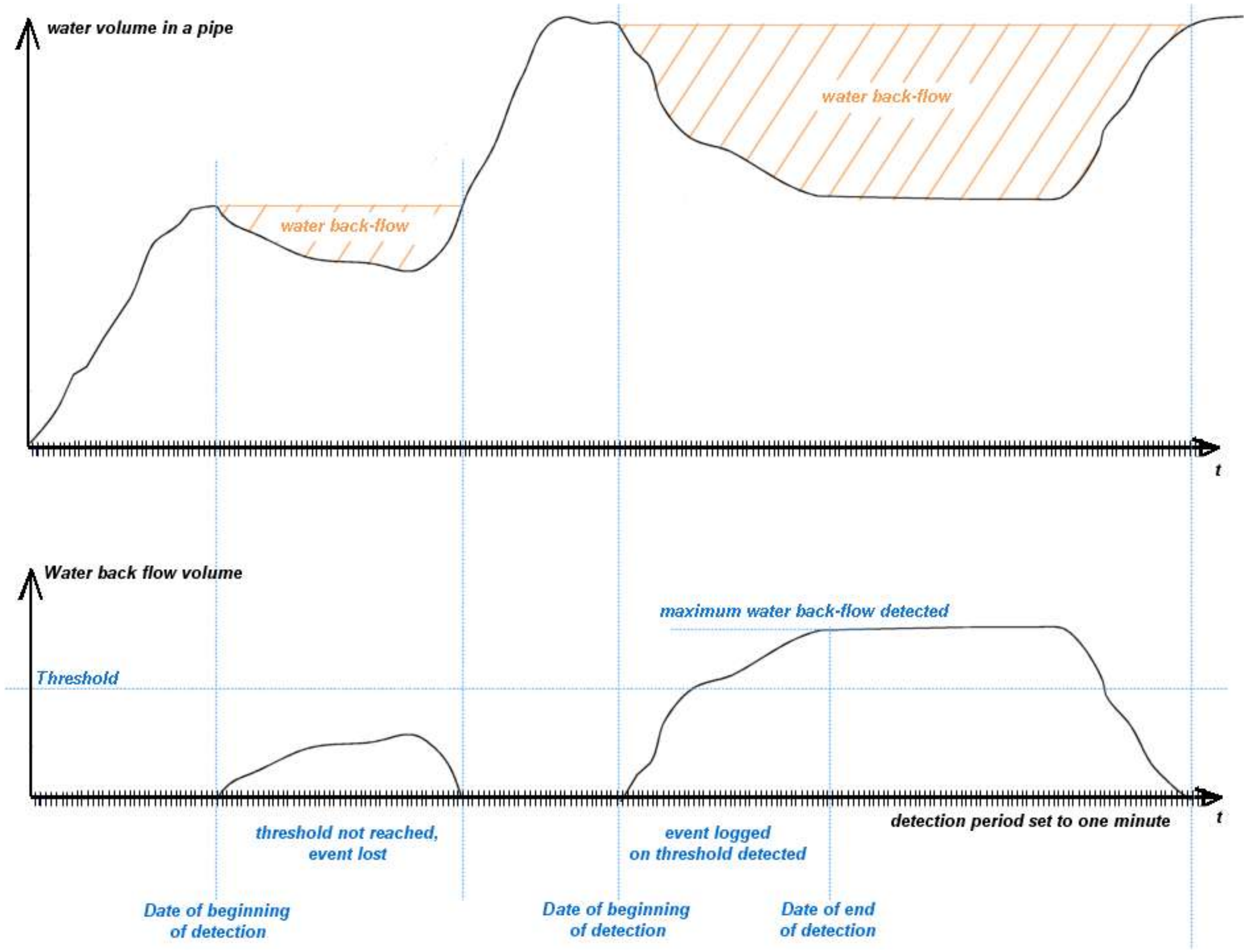
➢ Parameters :

- **Backflow Threshold :** this is the water backflow volume which triggers logging of the occurrence. It is expressed in number of pulses (see chapter 6.1.5 for pulse weight).

- **Backflow detection period :** this period indicates the water backflow volume measurement granularity. This value is set at 1 minute and may not be modified.

> *Attention: do not confuse this fixed detection period with the flow-rate detection method for which the parameters may be modified.*

➢ Example :



6.7.3 Format of the event table of backflow detection method with measurement of water volume

The module stores in an internal table the information relative to the occurrence of a backflow. This table can store up to 4 events.

The structure is as follows :

Number of the inputs	Water volume detected as backflow	Detection date	Date of end of detection
1 byte	2 bytes	6 bytes	6 bytes
Input_Evt 0	Volume_Evt 0	Date_Evt 0	Date_end_Evt 0
Input_Evt 1	Volume_Evt 1	Date_Evt 1	Date_end_Evt 1
Input_Evt 2	Volume_Evt 2	Date_Evt 2	Date_end_Evt 2
Input_Evt 3	Volume_Evt 3	Date_Evt 3	Date_end_Evt 3

This data is stored in a circular buffer which may be accessed by radio and contains the last 4 occurrences logged. The module may be programmed to generate an automatic radio frame when backflow is detected.

The water backflow detection parameters are programmed individually for each pulse input.

- **Number of the inputs :** 0 : input A
  1 : input B

- **Water Volume detected as backflow :** indicates the maximum water volume logged during the detection of the backflow.

- **Detection date :** date logged at the backflow occurence.

- **Date of end of detection :** date of disappearance of the backflow

Date					
Day	**Month**	**Year**	**Day of the Week**	**Hour**	**Minute**
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte
		(*)	**(*)**		

(*) ***Year*** *= (current year) - 2000*

Day of the Week : coded from 0 to 6

0 : Sunday
1 : Monday
2 : Tuesday
3 : Wednesday
4 : Thursday
5 : Friday
6 : Saturday

6.7.4 Water backflow detection method with measurement of water flow-rate

With this method, the water backflow rate is measured during a predefined detection period, based on '*backflow detection period'* parameter.

The detection period is divided into 10 measurement steps. At each step, the average flow-rate for the previous measurement period is calculated (ie the last 10 measurement steps).

This system enables detection of very short flow peaks which would be missed with other measurement methods. It also enables conservation of a low measurement granularity while expressing the flow-rate in number of pulses per measurement period.

backflow is detected when the flow rate exceeds a preset threshold value. The occurrence is then logged in a table. The table is updated during the entire water backflow period in order to log the maximum backflow rate detected and the end of detection date.

End of detection takes place as soon as the water backflow rate returns below the preset threshold value.

- Parameters :

  - **Backflow Threshold :** this is the water backflow rate which triggers detection and logging of the occurrence. It is expressed in number of pulses, per backflow detection period (see chapter 6.1.5, for pulse weight).

  - **Backflow detection period :** this period is expressed in 10-minute steps. It is used to set the measurement granularity; in this way, a detection period is sub-divided into 10 measurement steps. Consequently, the minimum measurement step is 1 minute.

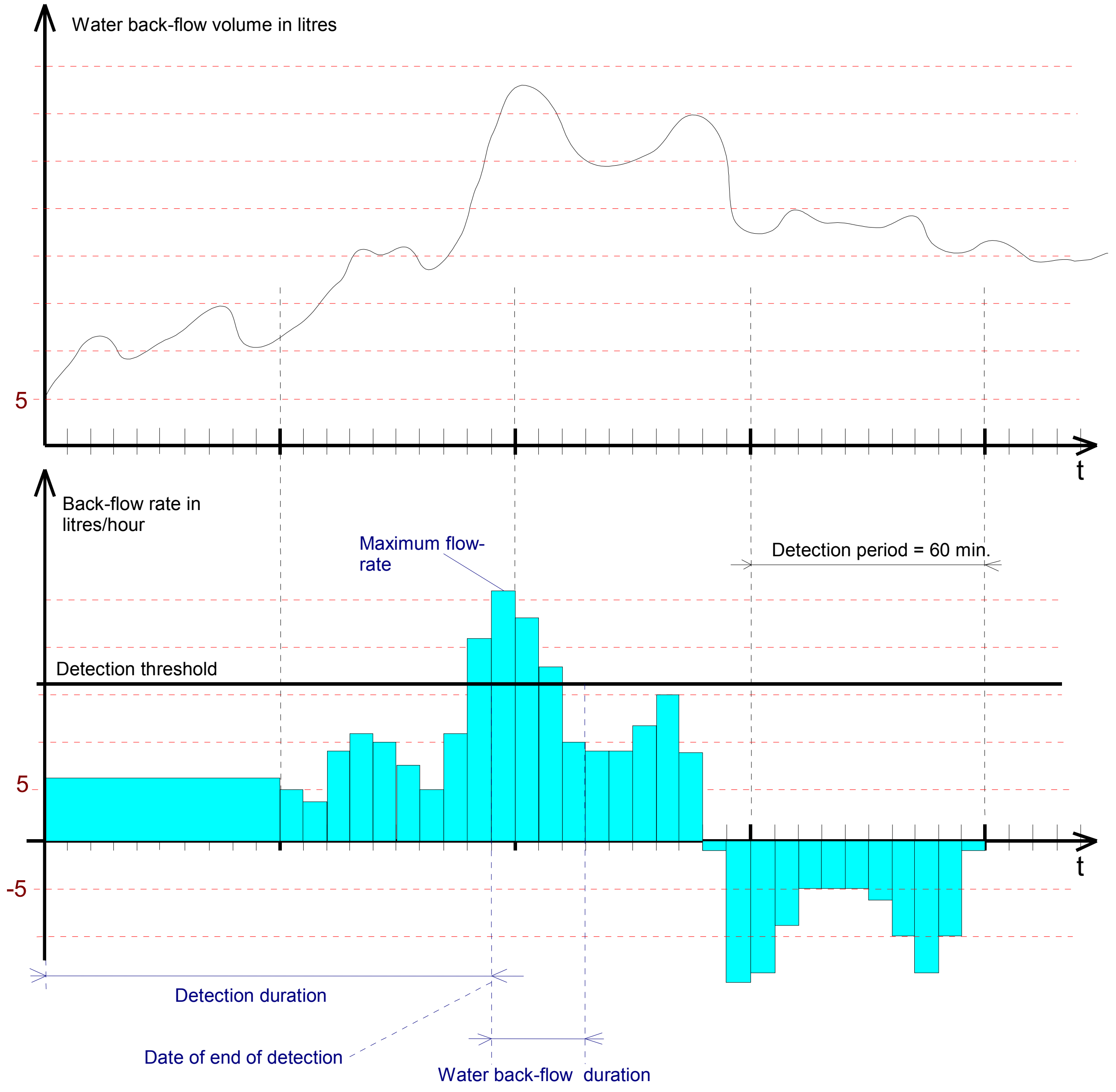
- **Example :** The measurement weight is set (see chapter 6.1.5) to measure the water volume in litres and the associated backflow parameters are set as follows:

  - ***Backflow detection period :*** if a measurement scale in litres/hour is required; the parameter is simply set to a value of 6.

  - ***Backflow Threshold :*** the threshold is set to 16 litres/hour.

> *Attention, the value of the parameter depends on the pulse weight of the sensor.*
> *Example : if k = 0,1 , then threshold = 160 pulses (= 0xA0)*
> *if k = 1 , then threshold = 16 pulses (= 0x10)*

At each measurement step (a tenth of the detection period), the flow-rate is measured for the previous detection period; in this way the flow-rate in litres/hour is obtained for each measurement step.



6.7.5 Format of the event table of backflow detection method with measurement of water flow-rate

The module stores in an internal table the information relative to the occurrence of a backflow. This table can store up to 4 events.

The structure is as follows :

Number of the inputs	Maximum Flow rate	Detection duration	Water backflow duration	Not used	Date of end of detection
1 byte	2 bytes	2 bytes	2 bytes	2 bytes	6 bytes
Input_Evt 0	Flow_Evt 0	Duration_detect_Evt 0	Duration_pr_Evt 0	-	Date_Evt 0
Input_Evt 1	Flow_Evt 1	Duration_detect_Evt 1	Duration_pr_Evt 1	-	Date_Evt 1
Input_Evt 2	Flow_Evt 2	Duration_detect_Evt 2	Duration_pr_Evt 2	-	Date_Evt 2
Input_Evt 3	Flow_Evt 3	Duration_detect_Evt 3	Duration_pr_Evt 3	-	Date_Evt 3

This data is stored in a circular buffer which may be accessed by radio and contains the last 4 occurrences logged. The module may be programmed to generate an automatic radio frame when backflow is detected.

The water backflow detection parameters are programmed individually for each pulse input.

- **Number of the inputs :** 0 : input A
  1 : input B
- **Flow rate :** maximum flow rate logged during the backflow detection.
- **Detection duration :** time elapsed to detect the backflow (in minutes).
- **Water backflow duration :** duration of the occurrence of the backflow, the value is updated while the flow rate is higher than the threshold (expressed in minutes).
- **Date of end of detection :** date indicating when the backflow rate has returned below the threshold.

Date of end of detection					
Day	**Month**	**Year**	**Day of the Week**	**Hour**	**Minute**
1 byte	1 byte	1 byte	1 byte	1 byte	1 byte
		(*)	**(*)**		

(*) ***Year*** *= (current year) - 2000*
Day of the Week : coded from 0 to 6

0 : Sunday
1 : Monday
2 : Tuesday
3 : Wednesday
4 : Thursday
5 : Friday
6 : Saturday

6.7.6 Configuration of the backflow detection

The backflow detection is configured by following these two steps :

- **Selection of the detection method :** by writing the *'extended operation mode'* parameter

MSB **LSB**

Extended Operation Mode							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
-	**-**	**-**	**-**	**-**	**-**	**-**	**Backflow detection method**

Backflow detection method : ***0 :*** *backflow detection method with measurement of water volume*
1 : *backflow detection method with measurement of water flow-rate*

- **Configuration of the parameters** relative to the inputs concerned, and to the method used.

Attention, the format of the 'Backflow Threshold' parameter is not the same, depending of the method used :

- Method with measurement of water volume : *the threshold is expressed in number of pulses.*

- Method with measurement of water flow-rate : *the threshold is expressed in number of pulses per measurement period.*

6.7.7 Reading the information relative to the backflow detection

The occurrence of a water backflow is signaled by setting the bit 7 of the *Application Status* (parameter 0x20), to 1.

MSB LSB

		Application Status							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WaveFlow Standard		*BackFlow detection*	*Reed Fault detection on input B*	*Reed Fault detection on input A*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Standard CYBLE 5 wires		*BackFlow detection*	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*

The coding is as follows : *0 : not detected*
1 : detected

This status indicates if a backflow occurs or not, but no information on the inputs concerned, or the type of event. These pieces of information are stored in the events table.

The events table is read by the following commands :

Applicative Command	Description
0x08	Reading the backflow events table
0x88	Response to the reading of the backflow events table

- REQ_SEND_FRAME request

Data Field (max : 152 bytes)
Applicative command
1 byte
0x08

- RECEIVED_FRAME response

Data Field (max : 152 bytes)				
Acknowledgement of the applicative command	**Most recent event relative to the backflow detection**	**Backflow Event (n-1)**	**Backflow Event (n-2)**	**Backflow Event (n-3)**
1 byte	15 octets	15 bytes	15 bytes	15 bytes
0x88		**(*)**	**(*)**	**(*)**

***(*) :** If no event has been detected, then the fields are set to 0x00.*

6.8 WIRECUT AND REED FAULT DETECTION

Compatibility :

WireCut Detection : this functionnality is supported by all the modules,

Reed Fault Detection : this functionnality is only supported by the following modules,

- ***WaveFlow – Standard***
- ***WaveFlow – specific backflow***

6.8.1 Principle of the detections

➢ **Principle of the WireCut detection**

Cable break fault detection is possible if the cable sensor is of the 3-wire type.
In such a case, the 3rd wire is connected to a module input in the same way as the metering input.

Periodically, the software detects a cable break by measuring the level on this input.
Once a cable break fault has been detected, it is transmitted in the 'Applicaton Status' byte and the date the cable break fault is detected is memorised.

Cable break fault detection is normally enabled but may be disabled with a radio parameter setting signal in an operating mode byte.

➢ **Principle of the Reed fault detection**

A reed fault is detected when the pulse transmitted by the second reed of the pulse emitter is not detected after several attempts.
Once a reed fault has been detected, *it is transmitted with the corresponding input in the 'Application Status' byte* and the date the reed fault is detected is memorised.

Reed fault detection is normally disabled but may be enabled with a radio parameter setting signal in an operating mode byte.

6.8.2 Description of the parameters used

Parameter number	Description		Size (in bytes)	Access Right	Default value
0x01	Operation Mode		1	R/W	0x09
0x20	Application Status		1	R/W	0x00
0x91	Date of the wirecut detection on input A		6	R/W	0x00
0x92	Date of the wirecut detection on input B		6	R/W	0x00
0x93	Date of the Reed fault detection on input A	(if supported)	6	R/W	0x00
0x94	Date of the Reed fault detection on input B	(if supported)	6	R/W	0x00
0x95	Date of the wirecut detection on input C	(if supported)	6	R/W	0x00
0x96	Date of the wirecut detection on input D	(if supported)	6	R/W	0x00

6.8.3 Activation of the WireCut, or Reed Fault detections

Detections are activated (or deactivated) by modifying the bits 4, and 7 of the *Operation Mode* parameter.

MSB **LSB**

Operation Mode							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reed Fault detection (*) **0** : deactivated **1** : activated	**Extreme Leak detection** **0** : deactivated **1** : activated	**Residual Leak detection** **0** : deactivated **1** : activated	**WireCut detection** **0** : deactivated **1** : activated	**Datalogging** **00** : deactivated **01** : by time steps **10** : once a week **11** : once a month		Management of the inputs. Refer to chapter 5.1.1	

(*) Reed Fault detection : only if supported by the module, else this bit is not used.

6.8.4 Reading the pieces of information relative to the detections

When a wirecut, or a redd fault is detected, a bit is set in the Application Status byte, and the detection date is stored in an internal parameter (the date format is described in chapter 6.1.6).

It is possible to reset the bits of the Application Status byte, by using the internal parameter write command.

The format of the 'Application Status' byte depend of the module used :

MSB LSB

	Application Status							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WaveFlow 4 inputs	-	*WireCut detection on input D*	*WireCut detection on input C*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Specific BackFlow	*BackFlow detection in the month*	*Reed Fault detection on input B*	*Reed Fault detection on input A*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Standard	*BackFlow detection*	*Reed Fault detection on input B*	*Reed Fault detection on input A*	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow Standard CYBLE 5 wires	*BackFlow detection*	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*
WaveFlow 4800	-	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	*End of Battery life*

The coding is as follows : *0 : not detected*

1 : detected

6.9 END OF BATTERY LIFE DETECTION

Compatibility : this functionnality is supported by all the module.

To detect the end of battery life, the **WaveFlow** module uses the power metering principle rather than measurement of the battery voltage. Lithium batteries are, in particular during passivation, unsuitable for the voltage measurement method to determine the remaining capacity.

The **WaveFlow** records and evaluates all events (measurements, transmissions) to decrement the power meter according to the battery used. When the meter passes below a predefined threshold, the “end of battery life” is signalled with the *Application Status* byte.

The initial value of the end-of-life meter is factory-set. It depends on the type and number of batteries used.

When the end of battery life is detected, the detection date is memorised and may be read with a radio command.

6.9.1 Description of the parameters used

Parameter number	Description	Size (in bytes)	Access Right	Default value
0x20	Application Status	1	R/W	0x00
0x90	Date of detection of end of battery life	6	R/W	0x00
0xA2	Battery life duration counter	2	R	0xC15C

When the end of battery life is detected, the bit b0 of the *Application Status* is set to 1 ; and the detection date is stored. This date can be read by a standard read command of internal parameter (format of the date described in chapter 6.1.6).

MSB LSB

Application Status							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
This bits depends of the module used see chapter 5.1.1			*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*WireCut detection on input B*	*WireCut detection on input A*	***End of Battery life***

The coding is as follows : 0 : not detected

1 : detected

6.10 MANAGEMENT OF ALARMS FRAMES

Compatibility : this functionnality is only supported by the following modules,

- ***WaveFlow – 4 inputs***
- ***WaveFlow – Standard***
- ***WaveFlow – Standard – CYBLE 5 wires***
- ***WaveFlow – Specific Backflow***

The WaveFlow – 4800 doesn't use alarms frames to keep the compatibility with the Is)FlowMeter.

The WaveFlow module offers the possibility to automatically transmit radio frames when an occurrence is detected. The following occurrences may provoke an automatic alarm:

- Extreme Leak detection (High threshold);
- Residual Leak detection (Low threshold);
- Wirecut detection;
- End of battery life detection;
- Reed fault detection *(if supported by the module)*;
- BackFlow detection *(if supported by the module)*.

It is possible to select for each type of occurrence whether or not an alarm frame is to be sent.

6.10.1 Description of the parameters used

Parameter number	Description	Size (in bytes)	Access Right	Default value
0x01	Operating Mode	1	R/W	Depending of the module used (see 5.1.1)
0x20	Application Status	1	R/W	0x80
0x22	Alarm Configuration byte	1	R/W	0x00
0xB0	Number of repeaters used to transmit an alarm frame	1	R/W	0x00
0xB1	Address of the 1st repeater used to transmit an alarm frame	6	R/W	0x00
0xB2	Address of the 2nd repeater used to transmit an alarm frame	6	R/W	0x00
0xB3	Address of the 3rd repeater used to transmit an alarm frame	6	R/W	0x00
0xB4	Address of the recipient of the alarm frame	6	R/W	0x010A030000BD

6.10.2 - Configuration of the route to reach the alarm frames recipient

The route can be configured by a standard write command of the concerned parameters, or in an automatic way.

Indeed when a distant module (WaveCard, or WavePort type) send the Alarm Configuration command (0x23), the WaveTherm module stores the radio address of the transmitter, and the relay route (if used) as the recipient of alarm frames.

6.10.3 - Configuration of the alarms to be sent

Applicative Command	Description
0x23	Request to configure the alarms to be sent
0xA3	Acknowledgement of the request

The alarms to be sent are configured by writing to the parameter 0x22 (*Alarm Configuration Byte*), or by sending the following command :

Data Field (max : 152 bytes)								
Applicative command	**Alarms Configuration byte**							
1 byte	b7	b6	b5	b4	b3	b2	b1	b0
0x23	**Depends of the module used**							

When sending this command, the internal parameter relative to the configuration of the alarm (0x22) will be automatically updated.

The structure of the 'Alarm Configuration Byte' depends of the module used, as described below :

		Alarm Configuration Byte							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WaveFlow – Standard CYBLE 5 wires		-	-	-	Backflow detection	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*End of Battery life*	WireCut detection
WaveFlow – Standard		-	-	-	Backflow detection	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*End of Battery life*	WireCut or Reed fault detection
WaveFlow – specific backflow		-	-	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*End of Battery life*	WireCut or Reed fault detection
WaveFlow – 4 inputs		-	-	-	-	*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	*End of Battery life*	WireCut detection

0 : alarm frames disabled
1 : alarm frames enabled

- **In case of error :** Following the alarms configuration request, the WaveFlow returns an acknowledgement. In the case of error, the acknowledgement frame contains a status byte :

Data Field (max : 152 bytes)	
Acknowledgement of the applicative command	**Status**
1 byte	1 byte
0xA3	**0xFF : updating error**

- **In case of success :** It returns the following parameters : Operating Mode ; Application Status ; and indexes corresponding to the inputs.

Data Field (max : 152 bytes)						
Acknowledgement of the applicative command	**Data**					
	Operating Mode	**Application Status**	**Index A**	**Index B**	**Index (*)**	**Index (*)**
1 byte	1 byte	1 byte	4 bytes	4 bytes	4 bytes	4 bytes
0xA3			**MSB First**	**MSB First**	**MSB First**	**MSB First**

(*) The information given by the two index fields, depend of the module used :

- WaveFlow – 4 inputs :	*Index C*	*;*	*index D*
- WaveFlow – specific backflow :	*BackFlow index A*	*;*	*BackFlow index B*
- WaveFlow – Standard :	*BackFlow index A*	*;*	*BackFlow index B*
- WaveFlow – Standard – CYBLE 5 Wires :	*BackFlow index A*	*;*	*BackFlow index B*

The indexes corresponding to unused inputs are set to zero.

6.10.2 Triggering an alarm frame

Applicative Command	Description
0x40	Alarm frame
0xC0	Acknowledgement of the alarm frame

After detection of a fault, if the configuration mode authorises transmission of alarms, the module transmits an alarm frame (applicative command 0x40).

⚠ ***Attention, an alarm frame only has one type of detection. When several alarms are detected, the WaveTherm module emits the frames one after the other.***

An alarm frame will be transmitted after the previous frame has been acknowledged.

Data Field (max : 152 bytes)			
Applicative Command	**Alarm Status**	**Date**	**Flow**
1 byte	1 byte	6 bytes	2 bytes
0x40	**Depend of the module used**		

- ***Flow :*** *this field is used only when the alarm type is a leakage (extreme or residual) detection.*
- ***Date*** *: the date format is given in chapter 6.1.6*
- ***Alarm Status*** *: The structure of the 'Alarm Status' depends of the module used, as described below.*

		Alarm Status							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
WaveFlow – Standard CYBLE 5 wires		*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	WireCut detection	*End of Battery life*	*BackFlow*	-	*Input concerned by the alarm frame* *01 : input A* *10 : input B*	
WaveFlow – Standard		*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	WireCut detection	*End of Battery life*	*BackFlow*	*Reed fault*	*Input concerned by the alarm frame* *01 : input A* *10 : input B*	
WaveFlow – specific backflow		*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	WireCut detection	*End of Battery life*	-	*Reed fault*	*Input concerned by the alarm frame* *01 : input A* *10 : input B*	
WaveFlow – 4 inputs		*High Threshold (Extreme leak)*	*Low threshold (Residual leak)*	WireCut detection	*End of Battery life*	-	-	*Input concerned by the alarm frame* *01 : input A* *10 : input B* *00 : input C* *11 : input D*	

0 : not detected

1 : detected

The remote device must send an acknowledgement frame (command 0xC0) to confirm reception of the alarm frame and end dialogue.

Data Field (max : 152 bytes)	
Acknowledgement of the applicative command	**Alarm Status**
1 byte	1 byte
0xC0	**Same as received in the alarm frame**

If the WaveTherm module does not receive this acknowledgment, it re-transmits the alarm frame several time, with a delay between each retransmission. The delay, and the number of retries depend of the equipment type used :

	4 - inputs	**Specific backflow**	**Standard**	**Standard – CYBLE 5 wires**
Number of retries of alarm sending	3 times	3 times	7 times	7 times
Delay between each retransmission	1 minute	1 minute	1st retry : 1 minutes 2nd retry : 15 minutes 3rd retry : 45 minutes 4th retry : 90 minutes 5th retry : 180 minutes 6th retry : 360 minutes 7th retry : 720 minutes	1st retry : 1 minutes 2nd retry : 15 minutes 3rd retry : 45 minutes 4th retry : 90 minutes 5th retry : 180 minutes 6th retry : 360 minutes 7th retry : 720 minutes

7. APPENDIX A : SET OF THE APPLICATIVE COMMANDS

Applicatives Commands	Description
INTERNAL PARAMETERS SETTING	
0x10	Request of parameter(s) reading
0x90	Acknowledgement of the request of parameter(s) reading
0x11	Request of parameter(s) writing
0x91	Acknowledgement of the request of parameter(s) writing
MODULES INFORMATION	
0x20	Request to read the module type
0xA0	Response to the request to read the module type
0x28	Request of reading the firmware version (GET_FIRMWARE_VERSION)
0xA8	Response to the request of reading the firmware version
0x12	Request to read the date and time of the module
0x92	Response to request to read the date and time of the module
0x13	Request to set the date and time of the module
0x93	Response to request to set the date and time of the module
READING THE INDEXES AND TABLES	
0x01	Request for immediate reading of indexes
0x81	Response to the immediate reading
0x02	Request to write new indexes
0x82	Response to the writing new indexes request
0x03	Request to read the datalogging storage table (24 values) for inputs A and B
0x83	Response to the reading of the datalogging storage table
0x04	Request to read the leakage event table
0x84	Response to the reading of the leakage event table
0x05	Request for a global reading of indexes
0x85	Response to the global reading
0x06	Request for an extended reading of the indexes
0x86	Response to the extended reading
0x07	Request to read the datalogging storage table (24 values) for inputs C and D
0x87	Response to the reading of the datalogging storage table
0x08	Request to read the backflow event table
0x88	Response to the reading of the backflow event table
ALARM FRAMES MANAGEMENT	
0x23	Request to configure the alarms to be sent
0xA3	Acknowledgement of the request