# *Sonata Sprint*

## *Encoder*

# User Guide

**FCC ID: 2A7AA-SONSPR2LCEMM**
**IC: 28664-SON2SPRLCEMM**

**Revision: 1.00**

**Approvals:**

| | Name | Position | Signature |
|---|---|---|---|
| Written by: | Evgeni Kosakovski | Firmware Engineer | |
| Approved by: | | R&D Manager | |
| Approved by: | | Product Manager | |
| Approved by: | | | |

# Federal Communication Commission (FCC) Compliance Notice

**CAUTION**

This device complies with part 15 of the FCC Rules. The User should be aware that changes and modifications to the equipment not expressly approved by Master Meter could void warranty and the user's authority to operate the equipment. Professionally trained personnel should use the equipment.

**ATTENTION**

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions,

may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-- Reorient or relocate the receiving antenna.
-- Increase the separation between the equipment and receiver.
-- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
-- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) This device must accept any interference received, including interference that may cause undesired operation.


**Arad Technologies Ltd.** | st. HaMada, Yokneam Elite , 2069206, Israel | www.arad.co.il

# Industry Canada (IC) Compliance Notice

This device complies with FCC Rules Part 15 and with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent Isotropically radiated power (EIRP) is not more than that necessary for successful communication.

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensitè nècessaire à l'établissement d'une communication satisfaisante.

- This Class B digital apparatus complies with Canadian ICES-003. - Cet appareil numerique de la classe B est conforme a la norme NMB-003 du Canada.

**Radiation Exposure Statement:**

This equipment complies with FCC and IC RF radiation exposure limits set forth for an uncontrolled environment.

Contents

# 1 Introduction

Encoder software requirements specification is a description of a software system to be developed in Encoder module. It lays out functional and non-functional requirements and may include a set of use cases that describe system and user interactions that the software must provide.
Current requirements specification establishes the basis for operation between Arad water measurements from one side and encoder readers 2 or 3 wires from other. Used appropriately, software requirements specifications can help prevent software project failure.
Current document enlists enough and necessary requirements that are required for the Encoder module development include system definition, DFD, communication, etc., and presents the details of the hardware and software interface required to communicate Encoder module with SENSUS pulse readers.

# 2 System Overview

The Sonata Sprint Encoder is a battery-powered sub-system module allowed reading Sonata data through 2W or 3W interface.
It identifies the reader system type (2W or 3W) and converts the serially received data from the Sonata meter to the reader's string formats and transmits it in the Sensus reader type protocol.

# 3 Encoder SW architecture

**3.1** Encoder module is very simple configurable system that:

**3.1.1** Provides a high-resolution pulse output signal.

**3.1.2** Can translate received data from Sonata to electrical pulse for each unit of measurement according to Encoder module configuration. The electrical pulse is transmitted over a two-conductor or three-conductor cable to the remote readout systems.

**3.1.3** Supports communication interface with different pulse Readers.

**3.1.4** The Encoder model is built from a module that only transmits the last string it received from the Sonata meter without any post processing.

**3.2** Encoder module SW architecture is an interrupt-driven SW architecture:

- SPI RX interrupt
- Reader clock interrupts
- Timeouts

**3.3** The main program consists of system initialization and a main loop.

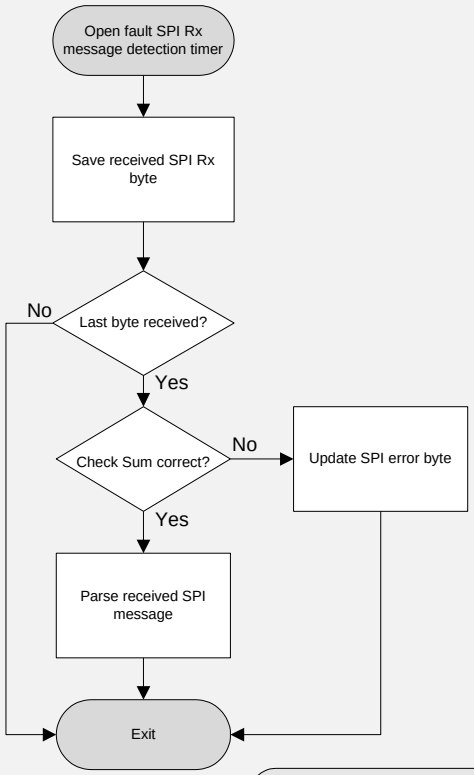
**3.3.1** During the main loop the system waits for SPI RX interrupt or reader interrupt to occur.

**3.3.2** If no interrupt occurred and no pulse out command was received the system enters "Power down" mode.

**3.3.3** The system wakes from "Power down" mode by SPI's interrupt or reader's clock interrupt.

**3.3.4** SPI and reader events are processed in ISRs.

**3.4** The following figure shows the Encoder module SPI event handle block.




| SPI ISR handle block | | |
|---|---|---|
| Ver. 1.00 | 03/12/2017 | Evgeni K. |

**3.4.1** Open fault Rx message detection timer.

When byte is received on SPI the system checks if it is a header byte, opens a timer for next byte receive timeout and initiates the timer. This method prevents the system from waiting for bytes for a long time.

If no byte is received for a long time (over 200ms) the SPI error byte is updated and the message is not removed.

**3.4.2** Save received Rx byte

Each byte is saved into Rx buffer.

**3.4.3** Check checksum

When last byte in the message is received, the checksum is validated.

**3.4.4** Update SPI error byte

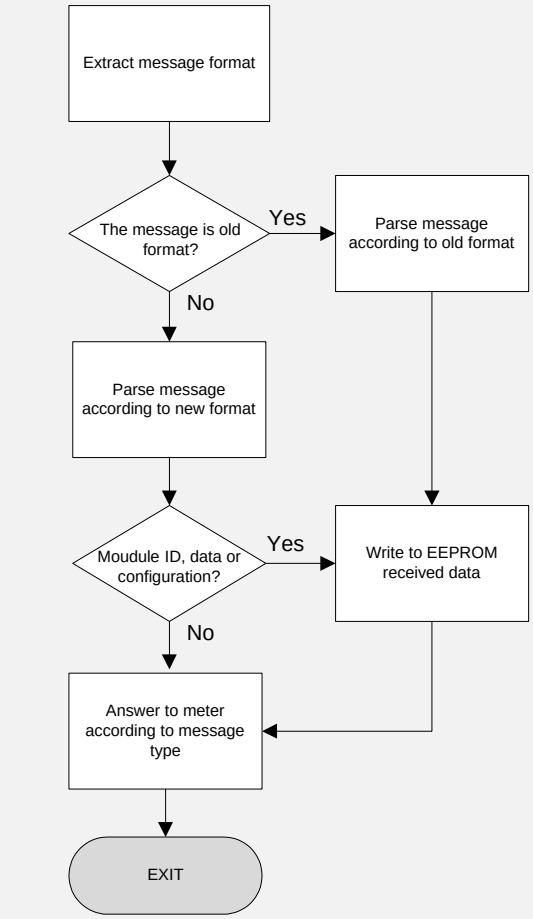
When checksum is not valid, the SPI error byte is updated and the message is not parsed.

**3.4.5** Parse received SPI message

When checksum is valid, the parsing process is called.

The parsing is done in the main loop in order to handle immediately the received buffer as an atomic and un-interfered process. When parsing is performed, no reader event is handled.

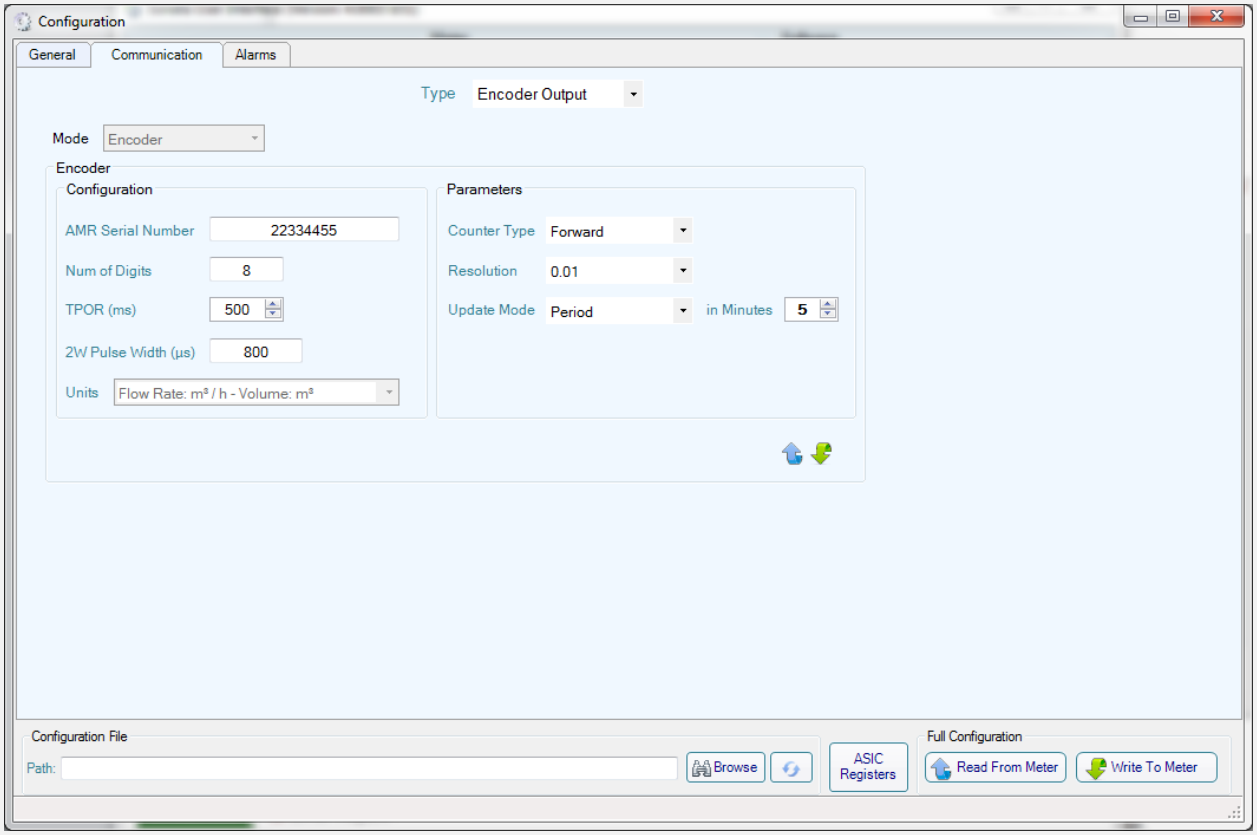
**3.5** The following figure shows parse message flow. Each of the blocks is described briefly in the subparagraphs.




| Parse message flow | | |
|---|---|---|
| Ver. 1.00 | 03/12/2017 | Evgeni K. |

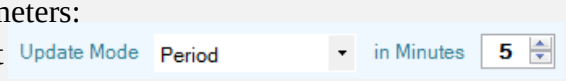
# 4 Encoder module configuration

There are possible to configure Encoder module for operation from GUI.



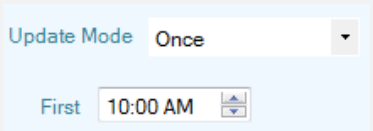

**4.1** Configuration set shall be stored into Sonata meter by press on  button.

**4.2** Sonata shall configure communication to Encoder module by RTC Alarm configuration according to GUI parameters:

**4.2.1** In case of user select 

Sonata RTC Alarm shall be configured for time is defined in “Minutes” field. Communication to Encoder module shall be performed every “Minutes” field time.

**4.2.2** In case of user select




or


Sonata RTC Alarm shall be configured for time is defined in “First” or “Second” field, according to selected option. Communication to Encoder module shall be performed at selected time.

**4.3** Encoder module shall support only **backward variable format**.

**4.4** Counter type:

**4.4.1** Net Unsigned (1 is converted to 99999999).

**4.4.2** Forward (default).

**4.5** Resolution:

**4.5.1** 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000 (default value 1).

**4.6** Update Mode – Sonata period time for send data to Encoder module:

**4.6.1** Period – every predefined time (in Minutes" field, see 4.2.1) Sonata shall send data to Encoder module. (1…59 minutes. Default 5 minutes)

**4.6.2** Once – fixed time when Sonata shall send data to Encoder module once a day (see 4.2.2). Field "First" shall contain time in format: hours and minutes.

**4.6.3** Twice - fixed time when Sonata shall send data to Encoder module twice a day (see 4.2.2). Fields "First" and "Second" shall contain time in format: hours and minutes.

**4.7** AMR Serial Number – up to 8 digits ID number (default same as meter ID)

- Only numeric numbers (in backwards mode).
- Only 8 least significant numbers (in backwards mode).

**4.8** Number of digits - 1- 8 digits from the right most position to be sent to the 2/3W reader (default 8 digits).

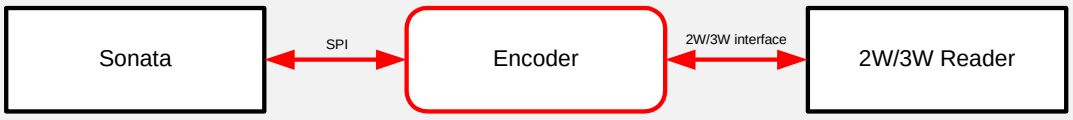
**4.9** TPOR - Time that the reader waits until the master stops the start sync (see TouchRead Interface) (0…1000ms. Default 500ms).

**4.10** 2W Pulse Width – (60…1200ms. Default 800ms).

**4.11** Units – flow units and volume units the same as in Sonata water meter (read only).

**4.12** Encoder module doesn't support alarms in backward format. Therefore we can't have option for Alarms indication on module side.

# 5 Communication definition




| Sonata - Encoder Interfaces | | |
|---|---|---|
| Ver. 1.00 | 23/11/2017 | Evgeni K. |

## 5.1 Sonata↔ Encoder Communication

**5.1.1** Sonata water meter communicate with Encoder module through SPI protocol: 500kHz, No Data control). Using other settings will produce unpredictable results, and can easily render the connected Sonata water meter unresponsive.

**5.1.2** After Sonata restart current configuration shall be sent to Encoder module with the first communication request within 1 minute of Sonata operation.

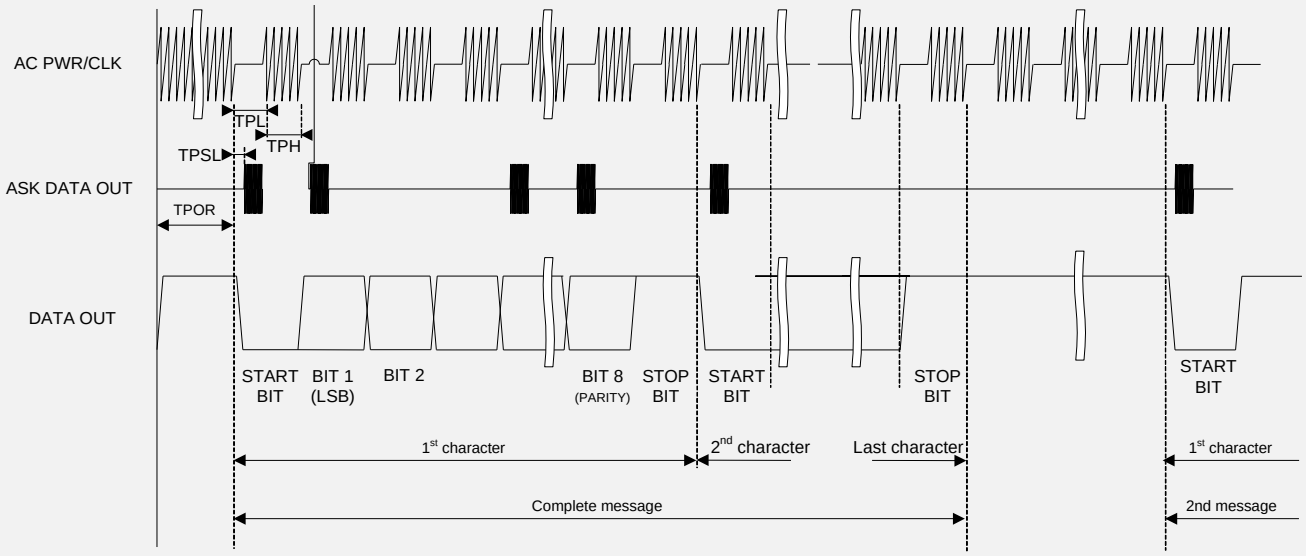
**5.1.3** In case of Encoder module doesn't receive configuration by 3 times, Sonata shall execute Encoder module Reset through "Reset" pin for 200ms and shall try to send configuration again.

**5.1.4** After configuration request performs success Sonata shall began sending data to Encoder module.

## 5.2 Encoder ↔ Sensus Reader (TouchRead) Interface

**5.2.1** The interface specification for the TouchRead mode is defined in terms of operation in a standard circuit.

**5.2.2** Encoder module shall communicate with readers through Sensus 2W or 3W protocol. There are TouchRead Interface timing diagram for Sensus 2W or 3W communication.



| Sym | Description | Min | Max | Default | Units |
|---|---|---|---|---|---|
| TPOR | Power ON to meter ready (Note 1) | | 500 | 500 | msec |
| TPL | Power/Clock low time | 500 | 1500 | | µsec |
| | Power/Clock low time jitter (Note 2) | | ±25 | | % |
| TPH | Power/Clock high time | 1500 | Note 3 | | µsec |
| TPSL | Delay, Clock to Data Out | | 250 | | µsec |
| | Power/Clock Carrier Frequency | 20 | 30 | | kHz |
| | Ask Data Out Frequency | 40 | 60 | | kHz |
| TRC | Reset command. Time for Power/Clock low to force register reset | | 200 | | msec |
| TRR | Meter Re-Read Time (Note 1) | | 200 | | msec |

Notes:

1. During TPOR power/clock pulses can be present but are ignored by the register. Some registers may not repeat message without reset command
2. The register clock jitter is specified because some registers may be sensitive to large variations in clock low time.
3. The register shall be static device. The register shall remain in the current state as long as the Power/Clock signal remains high.

**5.2.3** Supported readers:

**2W**

1. TouchReader II Sensus M3096 – 146616D
2. TouchReader II Sensus M3096 – 154779D
3. TouchReader II Sensus 3096 – 122357C
4. Sensus AutoGun 4090-89545 A
5. VersaProbe NorthROP Grumman VP11BS1680
6. Sensus RadioRead M520R C1-TC-X-AL

**3W**

1. VL9 ,Kemp-Meek Mineola, TX (Tap)
2. Master Meter MMR NTAMMR1 RepReader
3. Sensus AR4002 RF

## 5.3 Encoder Power mode

**5.3.1** When occurred timeout is indicated no activity of readers (200msec), SPI or Readers the system enters power down mode.

**5.3.2** The system can wake up from power down mode only when SPI is received or Reader clock is received.

**5.3.3** The power down mode of the system is HALT mode (minimal power consumption).

**5.3.4** Before entering power down mode SPI module is configured as EXTI in order to enable wake up from HALT mode when SPI message is received.

**5.3.5** PB0 is configured to EXTI in order to wake up from HALT mode when Reader's clock is received.

**5.3.6** The GPIO is configured for minimal power consumption during power down mode.

**5.3.7** Entering power down mode is executed from main loop after the timeout timer, timer 2 has elapsed.

### 5.4 Backward compatibility message

Message from meter:

| Byte Num | (0:3) | (4:7) |
|---|---|---|
| 0 | 'S' | |
| 1 | ID [0]-0x30 | ID [1]-0x30 |
| 2 | ID [2]-0x30 | ID [3]-0x30 |
| 3 | ID[4]-0x30 | ID [5]-0x30 |
| 4 | ID[6]-0x30 | ID [7]-0x30 |
| 5 | Acc[0]-0x30 | Acc [1]-0x30 |
| 6 | Acc [2]-0x30 | Acc [3]-0x30 |
| 7 | Acc [4]-0x30 | Acc [5]-0x30 |
| 8 | Acc [6]-0x30 | Acc [7]-0x30 |
| 9 | Check sum for(i=1;i<9;a^= message[i++]); | |
| 10 | 0x0D | |

### 5.5 Encoder interface configuration

| Byte Num | | |
|---|---|---|
| 1 | Bits:<br>0 - Enable External Power<br>1 - 0 Fix format<br>1 Variable format | Default is 0<br>No external power and Variable format |
| 2 | TPOR | In 10 ms steps |
| 3 | 2W clock freq | In Khz |
| 4-5 | Vsense threshold | Switch to external power when Vsense exceeds threshold |
| 6 | 2W pulse width in 5*us | 0 means 0us<br>10 means 50us<br>100 means 500us |
| 7-8 | Battery Access threshold<br>In thousands of accesses. | TBD |
| 9 | Decimal point location | |
| 10 | Num of digits | 0-8 |
| 11 | Manufacturer Id | |
| 12 | Volume Unit | See Appendix A |
| 13 | Flow Unit | See Appendix A |
| 14-15 | Bitwise:<br>0 - send Alarm<br>1 - send Unit<br>2 -send flow<br>3 -send volume | |
| 16 | Flow Type | C |
| 17 | Volume Type | B |
| 18-30 | Meter ID Main | Forward (8 LSB in Fix mode) |
| 31-42 | Meter ID (secondary) | Backward Flow (8 LSB in Fix mode) |

## 5.6 Encoder Message formatting

### 5.6.1 Fixed Length Format

RnnnniiiiiiiiCR

R[Encoder Data][ Meter ID 8 LSB(Configuration)]CR

The fixed length format is of the form:

Where: "R" is the leading character.
"nnnn" is a four character meter reading.
"iiiiiiii" is an eight character identification number.
"CR" is the carriage return character (ASCII value 0Dh)

Valid characters for "n" are "0-9" and "?"

Valid characters for "i" are: 0-9, A-Z, a-z, ?

In case of fix format the module will:

1. Convert the Meter counter sent to the module to ASCII (0 to 9999)
2. Take the 8 LSB from the Meter ID Main or Meter ID (secondary)

### 5.6.2 Variable Length Format

The variable length format consists of a leading character "V", a series of fields, and a terminator character "CR". The general form:

V;IMiiiiiiiiiii;RBmmmmmmm,uv;Aa,a,a;GCnnnnn,ufCR

1. Take the 12 LSB chars from the Meter ID Main or Meter ID (secondary)
2. Convert the meter counter field of the Encoder Data and convert to ASCII (0 to 99999999) , the number of digits depends on configuration
3. Send the Alarm Byte from the Encoder Data , if exists
4. Send unit Byte from the Encoder Data , if exists
5. Convert the meter Flow field of the Encoder Data and convert from float to ASCII, the number of digits are 4 and decimal point and sign if required.
6. Concatenate all with appropriate headers and separators
7. Add CR.

| **Totalizer** | 0 | 1 | 2 | 3 | . | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sensus** | **0** | **0** | **0** | **0** | **0** | **1** | **2** | **3** | | |
| **Encoder Data-volume** | **123** | | | | | | | | | |

Number of digit = 8
Resolution = 1
Decimal point location = 0 (no decimal point)

| **Totalizer** | 0 | 1 | 2 | 3 | . | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sensus** | 0 | 0 | 1 | 2 | 3 | . | 4 | 5 | | |
| **Encoder Data-volume** | **12345** | | | | | | | | | |

Number of digit = 7 (max because of decimal point)
Resolution = 1
Decimal point location = 2

| **Totalizer** | 0 | 1 | 2 | 3 | . | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sensus** | 1 | 2 | 3 | 4 | 5 | . | 6 | 7 | | |
| **Encoder Data-volume** | **1234567** | | | | | | | | | |

Number of digit =7 (max because of decimal point)

Resolution =x0.01
Decimal point location = 2

| **Totalizer** | 0 | 0 | 1 | 2 | . | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sensus** | 0 | 0 | 0 | 1 | 2 | 3 | 4 | | | |
| **Encoder Data-volume** | | | **1234** | | | | | | | |

Number of digit = 7
Resolution = x0.01
Decimal point location = 0

| **Totalizer** | 0 | 1 | 2 | 3 | . | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sensus** | 0 | 0 | 0 | 0 | 0 | 1 | 2 | | | |
| **Encoder Data-volume** | | | **12** | | | | | | | |

Number of digit = 7
Resolution =x10
Decimal point location = 0

## 5.7 Field definition

**5.7.1** The message format is identified according to the first message byte.

1. 0x55 indicated a new format message.
2. 0x53 ('S') indicates an old format message

**5.7.2** There are several optional subfields presented below. These are enclosed in brackets "[,]". If more than one subfield is defined for a field the subfields must appear in the order presented.

**5.7.3** The Module converts the data from the Meter to one of the two formats according to configuration (Fix or variable).

The next table defines supported length formats:

| **Output message Format** | **Form** | **Where** | **Configuration** |
|---|---|---|---|
| Fixed Length Format | RnnnniiiiiiiiCR | R leading character<br>n – meter reading<br>i – meter ID<br>CR – ASCII 0Dh | meter reading units |
| Variable Length Format | V;IMiiiiiiiiiii;<br>RBmmmmmmm,ffff,uv;<br>Aa,a,a;<br>GCnnnnnn,uf<br>CR | V - leading character<br>I - Identification field. i - up to 12 characters<br>M - Manufacturer Id<br>RB - Current Volume<br>A - Alarm field. a - alarm types up to 8 alarm code subfields are permitted.<br>GC - Current Flowrate<br>m - up to 8 digits<br>f - mantissa<br>uv - volume units (see Units table)<br>nnnnnn - 4-6 characters:<br>4-numbers, 1 decimal point, 1 sign character<br>uf - flow units (see Units table) | |

The fields:
f (mantissa), a (alarm) ,u (units) are optional.
Valid characters: "0-9", "A-Z", "a-z", "?" is valid as an error indicator.

## 5.8 Parse message according to old format

**5.8.1** In old format the message contains meter ID and Volume date.

**5.8.2** The message is parsed according to the ICD.

## 5.9 Write to EEPROM received parameters

**5.9.1** When module ID, data message or Configuration message is received, the parameters of the message are written into the EEPROM.

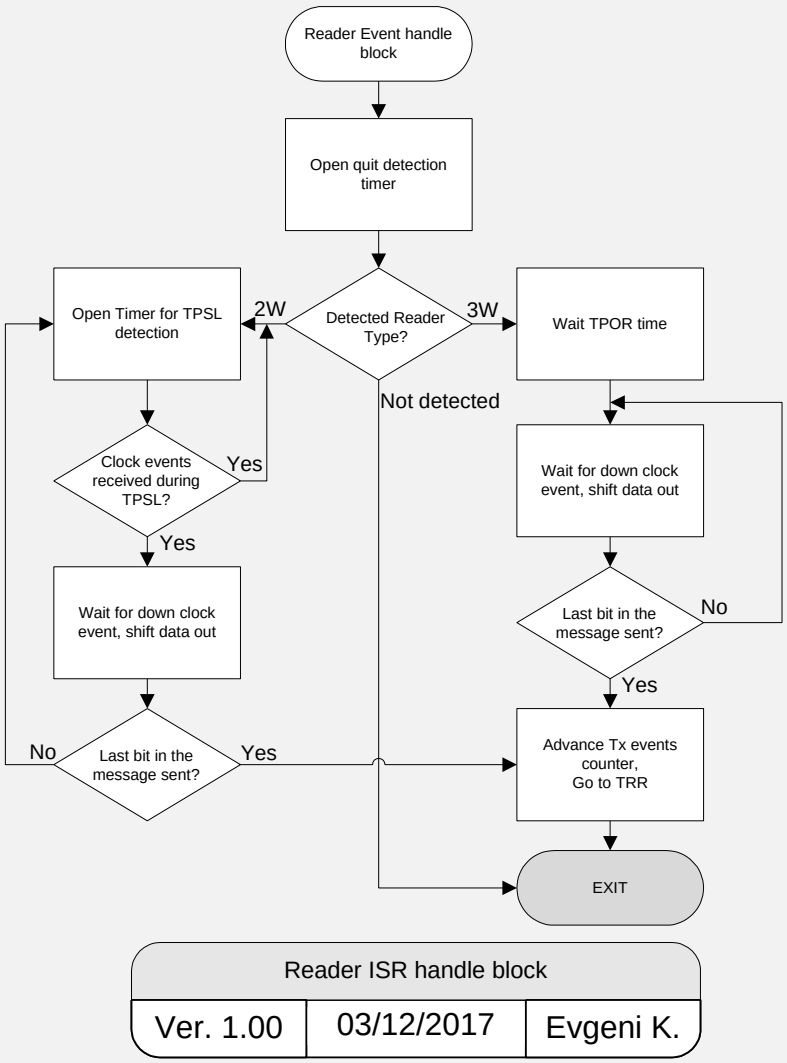
**5.9.2** This writing to EEPROM prevents the system from losing data when system reset occurs.

## 5.10 Reader event handle block

**5.10.1** When Reader Clock is received, the system handles reader's ISR event.

**5.10.2** All the processes are done in the ISR in order to be synchronized with the reader.

**5.10.3** If no clock is detected for 200ms, the system goes to power down mode.



### 5.11 Open quite detection timer

**5.11.1** When reader clock is received, a Quite Detection timer is opened.

**5.11.2** When there are no clock events for 200ms, the system goes to power down mode.

### 5.12 Detect reader type

**5.12.1** First 3 clock events are used for clock detection type.

**5.12.2** The detection is done by the measuring the frequency of the Reader's clock.

**5.12.3** The clock frequency for 2w reader is: 20 kHz - 30 kHz.

**5.12.4** The clock frequency for 3w reader is less than 2 kHz.

### 5.13 Open timer for TPSL detection

**5.13.1** When 2w reader is detected, a timer is opened for detection of TPSL time of quite before transmitting each byte.

**5.13.2** In 2w reader's protocol, each bit is transmitted in interval or quite.

### 5.14 Wait for down clock event, shift data out

- In 2w connection. After TPSL time is detected the bit is transmitted according to 2w protocol.

  '0' is transmitted as pulse of 50 kHz for 300 µs

  '1' is transmitted as '0' for 300 µs
- In 3w connection. After TPOR time of delay the bit is transmitted according to 3w protocol.

  '0' is transmitted as '1'

  '1' is transmitted as '0'

Each bit is transmitted after clock down event.

### 5.15 Advance TX events counter, go to TRR

After each message transmission, the counter of TX events is updated. The counter is used for indicating battery access exceed error when number of readings exceed battery access value. After each transmission, for TRR time, the system is not receiving reader's clock events.

### 5.16 Message format/ Encoder configuration

Message from meter to Encoder:

| | **Header** | **Addr [7:6]** | **Type [5:0]** | **Len** | **Data** | | **End** |
|---|---|---|---|---|---|---|---|
| Get Encoder Access | 55 | X | 12 | 0 | Null | | CSum |
| Get Encoder Status | 55 | X | 13 | 0 | Null | | CSum |
| Clear Encoder Status | 55 | X | 14 | 0 | Null | | CSum |
| Encoder Data | 55 | X | 15 | 4-10 | **Byte** | **Meter Data** | CSum |
| | | | | | 1-4 | Meter volume (singed Int) | |
| | | | | | 5 | Alarm | |
| | | | | | 6-9 | Flow (float) | |
| Encoder Configuration | 55 | X | 16 | | **Error! Reference source not found.** | | CSum |

**Len** – data length; **CSum -** check sum over all frame [55...Data] or AA.

Encoder reply to meter:

| | Header | Addr | Type | Len | Data | End |
|---|---|---|---|---|---|---|
| Get Encoder Access | 55 | ~~X~~ | ~~9~~ | ~~2~~ | <number of reads from interface> | Module ID |
| Get Status | 55 | ~~X~~ | ~~10~~ | ~~1~~ | **Bitwise**<br>0 OK<br>1 WatchDog occurred<br>2 UART Error<br>4 Exceed read number<br>8 Encoder Interface errors | Module ID |
| All Commands | 55 | ~~X~~ | ~~X~~ | ~~0~~ | | Module ID |

# 6 Glossary

| Term | Description |
|---|---|
| CSCI | Computer Software Configuration Interface |
| EEPROM | Electronically Erasable PROM |
| GUI | Graphical User Interface |
| ISR | Interrupt Service Routine |
| SRS | Software Requirements Specification |
| WD | Watch-Dog |

# 7 Appendix

## 7.1 Measurement Units

| Character | Units |
|---|---|
| $m^3$ | Cubic Meters |
| $ft^3$ | Cubic Feet |
| US Gal | US Gallons |
| l | Litres |


**Arad Technologies Ltd.** | st. HaMada, Yokneam Elite , 2069206, Israel | www.arad.co.il

## 8 External Documents

| Name and Location |
|---|
| 2W-SENSUS |
| 3W-SENSUS |
| |
| |

## 9 Revision History:

| Revision | Section affected | Date | Changed by | Change Description |
|---|---|---|---|---|
| 1.00 | All | 04/12/2017 | Evgeni Kosakovski | Document creation |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

**~ End of Document ~**