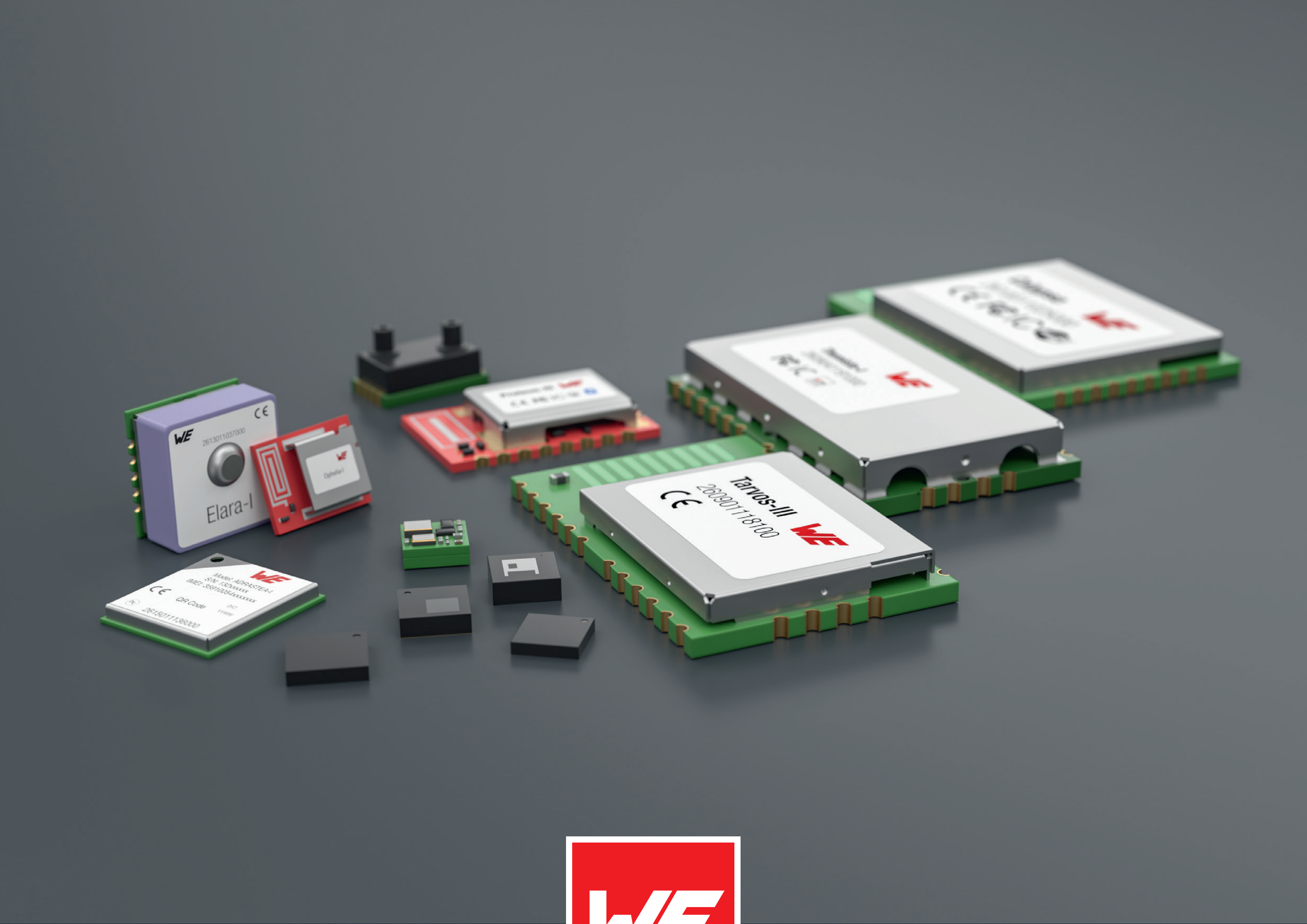

ANR018

GNSS I2C Communication

Version 1.3

July 19, 2023

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

Revision history

Manual version	Notes	Date
1.0	• Initial version	June 2020
1.1	• Figure 7 and 8 updated	July 2020
1.2	• Updated Chapter 3	June 2021
1.3	• Updated `Important notes`, meta data and document style	July 2023

Abbreviations

Abbreviation	Description
ASCII	American Standard Code for Information Interchange
CTS	Clear to send
I^2C	Inter-IC bus
GND	Ground
GNSS	Global Navigation Satellite System
HIGH	Digital (logic) high level
IC	Integrated Circuit
IDC	Insulation Displacement Contact
LOW	Digital (logic) low level
µC	Microcontroller
NMEA	National Marine Electronics Association
OSP	One Socket Protocol
PC	Personal Computer
RTS	Request to send
SCL	Serial Clock
SDA	Serial Data
USB	Universal Serial Bus

Contents

1 Introduction

This application note aims to support the use of the I^{2}C interface on Elara and Erinome GNSS modules. I^{2}C interface is available in these modules as an alternative to UART. The GNSS module functions as Multi-Master.

The first part of the document describes the basics of I^{2}C communication. This technical background information helps to understand the rest of the document. It includes understanding the different roles and phases in the communication, addressing of the bus subscriber and data structure.

The recommended I^{2}C communication hardware setup is described in detail in Chapter 3. The same setup can be used for Elara-I, Elara-II, and Erinome-II modules, operating with 1.8 V power supply. A slightly different setup is required for Erinome-I, operating with 3.3 V power supply.

After some simple modifications, the evaluation board can be used to test I^{2}C communication with the GNSS module. The modifications are also explained in detail in chapter 3.

Chapter 4 presents a specific application example, where the GNSS evaluation board communicates via I^{2}C with the Aardvark I2C/SPI host adapter from Total Phase [1]. This chapter guides the user through the software and hardware configuration to successfully establish I^{2}C communication with the GNSS module. The application note is concluded by a troubleshooting section which helps the user to debug the process.

Module	Article number	Power supply	Input logic level	Output logic level
Elara-I	2613011037000	1.8 V	3.3 V	1.8 V
Elara-II	2613021137000	1.8 V	3.3 V	1.8 V
Erinome-I	2614011037000	3.3 V	3.3 V	1.8 V
Erinome-II	2614021137000	1.8 V	3.3 V	1.8 V

Table 1: GNSS module power supply and logic levels

[1] *https://www.totalphase.com/products/aardvark-i2cspi/* .

2 I^{2}C Digital interface

The GNSS module supports standard I^{2}C (Inter-IC) bus interface as an alternative to the recommended UART interface. I^{2}C is a serial 8-bit protocol with two-wire interface, which supports communication between different ICs. For example between the µC and other peripheral devices such as a GNSS module or sensors.

2.1 General characteristics

A serial data line (*SDA*) and a serial clock line (*SCL*) are required for the communication between the devices connected via I^{2}C bus. Both *SDA* and *SCL* lines are bidirectional. The output stages of devices connected to the bus must have an open-drain or open-collector. Hence, the *SDA* and *SCL* lines shall be connected to a positive supply voltage using pull-up resistors.
In the I^{2}C protocol, the communication is realized through master-slave principle. The master device generates the clock pulses, a start command and a stop command for the data transfer. Each connected device on the bus is addressable via a unique address. Master and slave can act as a transmitter or a receiver depending upon whether the data needs to be transmitted or received.

The GNSS module supports the role of I^{2}C multi-master which means it will act as a I^{2}C master and send data to a specific slave address whenever it has data available and the bus is idle. In case the GNSS module has no data to send, it will switch into the I^{2}C slave role to be able to receive data from another master that addresses the GNSS module via the bus.

2.2 *SDA* and *SCL* logic levels

Logic level is the positive supply voltage to which *SDA* and *SCL* lines are to be pulled up (through internal or external pull-up resistors).

The GNSS modules support specified logic levels (product specific, e.g. 1.8 V) which must be either used by every device on the I^{2}C bus or a logic level conversion must be implemented in-between the GNSS module and the other bus participants. This conversion IC must be selected in a manner that it supports the requirements of I^{2}C set by the GNSS module, i.e. open-drain / open-collector and at least 400kHz clock speed.

2.3 Communication phase

2.3.1 Idle state

During the idle state, the bus is free and both *SDA* and *SCL* lines are in logic high ’1’ state. No master has requested the bus by sending START(S).

2.3.2 START (S) and STOP (P) condition

Data transfer on the bus starts with a START command, which is generated by the active master. A start condition is defined as a high-to-low transition on the *SDA* line while the *SCL* line is held high.
The bus is considered busy after the start condition, so no other master may access it. Slaves are not allowed to perform any bus activity unless addressed by a read access from a master.

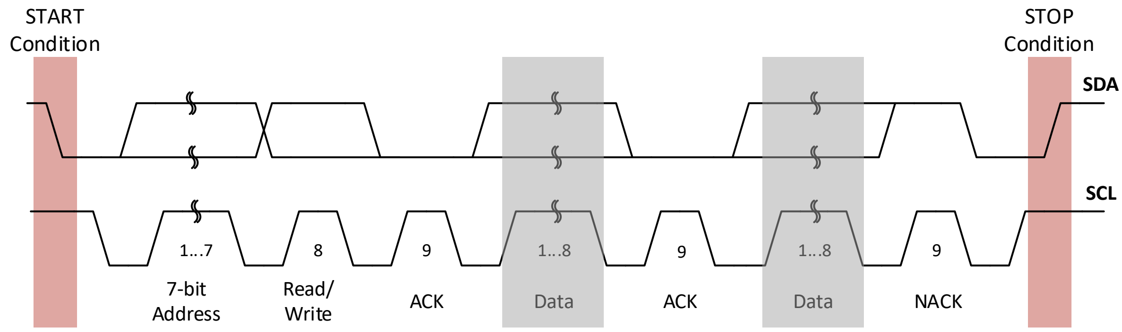

Figure 1: Data transfer from GNSS to slave

Data transfer on the bus is terminated with a STOP command, which is also generated by the active master. A low-to-high transition on the *SDA* line, while the *SCL* line being high is defined as a STOP condition.
After the stop condition, the bus is again considered free and is in idle state. Figure 2 shows the I^2C bus START and STOP conditions.

2.3.3 Data validity

After the START condition, one data bit is transmitted with each clock pulse. The transmitted data is only valid when the *SDA* line data is stable (HIGH or LOW) during the high period of the clock pulse. HIGH or LOW state of the data line can only change when the clock pulse is in low state.

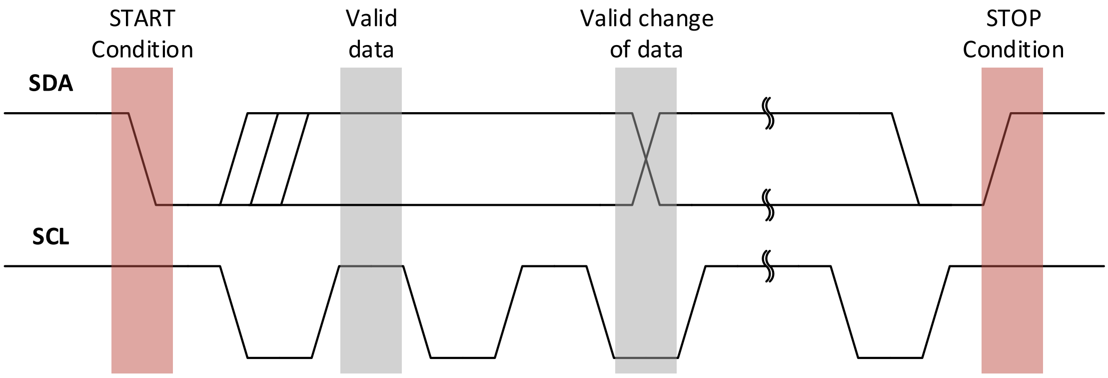

Figure 2: Data validity, START and STOP condition

2.3.4 Data format

Data transmission on the *SDA* line is always done in bytes, with each byte being 8-bits long. Data is transmitted with the most significant bit (MSB) first.
If the slave cannot receive or transmit another complete byte of data, it can force the master into a wait state by holding *SCL* LOW. Data transfer continues when the slave is ready which is indicated by releasing the *SCL* pin.

2.3.5 Acknowledge and No-Acknowledge

Each byte transmitted on the data line must be followed by an acknowledge bit (ACK). The receiver of the data (master or slave) generates an acknowledge signal to indicate that the data byte was received successfully and the receiver is ready to receive next data byte. This behaviour is shown in figure 1.

After one byte is transmitted, the master generates an additional acknowledge clock pulse to continue the data transfer. The transmitter releases the *SDA* line during this clock pulse so that the receiver can pull the *SDA* line to low state in such a way that the *SDA* line remains stable low during the entire high period of the clock pulse. It is considered as an Acknowledge signal.

If the receiver does not want to receive any further byte, it will not pull down the *SDA* line and it remains in stable high state during the entire clock pulse. It is considered as a No-Acknowledge (NACK) signal. After that, the master can generate either a stop condition to terminate the data transfer or a repeated start condition to initiate a new data transfer.

If there is no slave reaction following the master's message, the NACK signal will occur. This is caused by the pull-up resistor forcing a HIGH logic level on the *SDA* bus line. As already explained, a HIGH logic level during the clock pulse is interpreted as NACK by the master.

2.3.6 Addressing the GNSS module

The GNSS module supports 7-bit addressing. Users must make sure that each device on the I²C bus has a unique address. The master selects the slave by sending a slave address after each START condition.

When the GNSS module takes the role of a master/transmitter it will always try to send data to a fixed 7-bit slave address 1100010b (0x62).

The 7-bit slave/receiving address of the GNSS module is 1100000b (0x60). This address is used by the GNSS module only to receive data from a master.

The R/W bit determines the data direction. A ’0’ (LOW logic level) indicates a write operation (transmission from master to slave) and a ’1’ (HIGH) indicates a read operation (data request from slave).

In the master role the GNSS module only uses "write" operation as indicated by the R/W bit in the first byte of a transfer. In the slave role it only supports "write" operations from the master that is sending data to the GNSS module.

2.3.7 GNSS communication protocol

The GNSS module uses the NMEA protocol (default) or the OSP binary protocol (optional, can be enabled temporarily by the user). This is valid for any of its host communication interfaces. Details regarding the protocols can be found in the product specific user manual. The first protocol byte is directly following the 7-bit address and 1 R/W bit.

Users need to ensure that the bus capacity is adequate for the bus data transfer load peaks and that resulting latencies are not detrimental to system performance. This means that only a limited number of participants can be part of the shared bus.
Depending on the enabled messages, the GNSS module will send more or less data to the slave that is addressed by the 0x62 address.

2.4 I²C timing parameters

The clock is fixed to 400 kHz during master operation of the module and sending data (fast mode). During slave mode (receiving data from a master) fast mode (400kHz) as well as normal mode (100kHz) are supported by the GNSS module. The bus contention/arbitration timeout is 30 ms and cannot be changed.
Old data in the GNSS module buffer is discarded when a slave is not acknowledging the reception of the messages from the GNSS module. For optimal operation, the GNSS module assumes unrestricted outflow of serial messages and no periodic inflow of serial messages into the GNSS module.

When a message is sent to the GNSS module the reaction to this message may be delayed until all pending messages in the buffer are sent by the GNSS module and acknowledged by a slave.

3 Hardware Setup

The I^{2}C (Inter-IC) bus interface can be selected as the communication interface in the GNSS module through */CTS* and */RTS* pins. During power up, the module recognizes the I^{2}C bus interface through the */CTS* and */RTS* pin connections as per table 2.

Interface	/CTS	/RTS
I^{2}C	Open	External pull-down

Table 2: I^{2}C Interface Setting

By default the evaluation board is implemented with UART interface. To communicate with the module through I^{2}C bus interface, modifications on the evaluation board are required by the user. Our Erinome-I GNSS module operates with a power supply of 3.3 V. This makes the hardware setup slightly different from the other GNSS modules Elara-I, Elara-II and Erinome-II. Details follow in the next sections.

3.1 Hardware Setup - 1.8 V

Common I^{2}C hardware setup for Elara-I, Elara-II and Erinome-II evaluation board is shown in the figure 3.

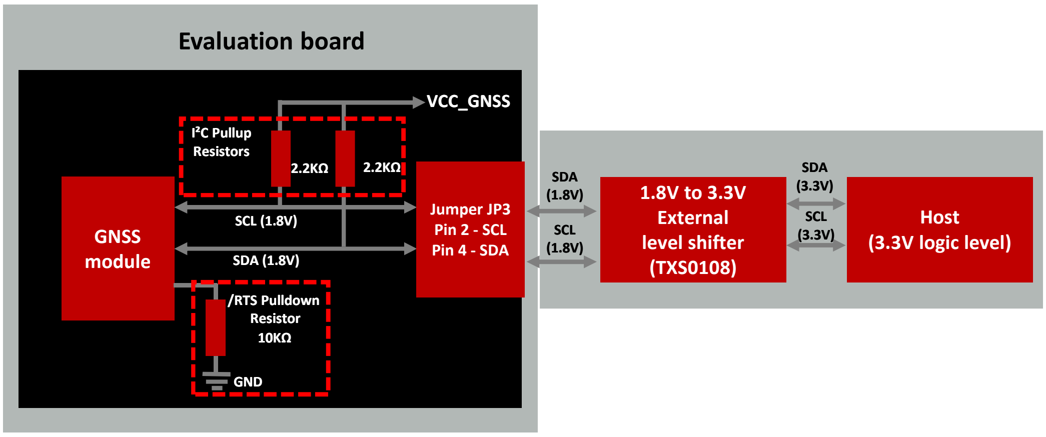

Figure 3: I^{2}C Hardware setup - 1.8 V

The block diagram illustrates the pull-up resistors assembly for the SDA and SCL bus lines as well as the pull-down resistor on the */RTS* pin for booting up in I^{2}C interface mode. It also shows that the I^{2}C bus access on evaluation board jumper JP3-2 and JP3-4 can be used to connect the bus to a host.

The I^{2}C bus outputs on jumper JP3 are 1.8 V logic levels. Therefore, a suitable level shifter is needed for further logic level translation (i.e. when the host does not support 1.8 V logic level). For a logic level translation to 3.3 V, the level shifter TXS0108 by Texas Instruments is used in the tested hardware setup. Further information of the level shifter can be found at *https://www.ti.com/product/TXS0108E*.

Please note that the level shifter shall use an open drain circuit and support I²C communication.

3.2 Evaluation Board Modification - 1.8 V

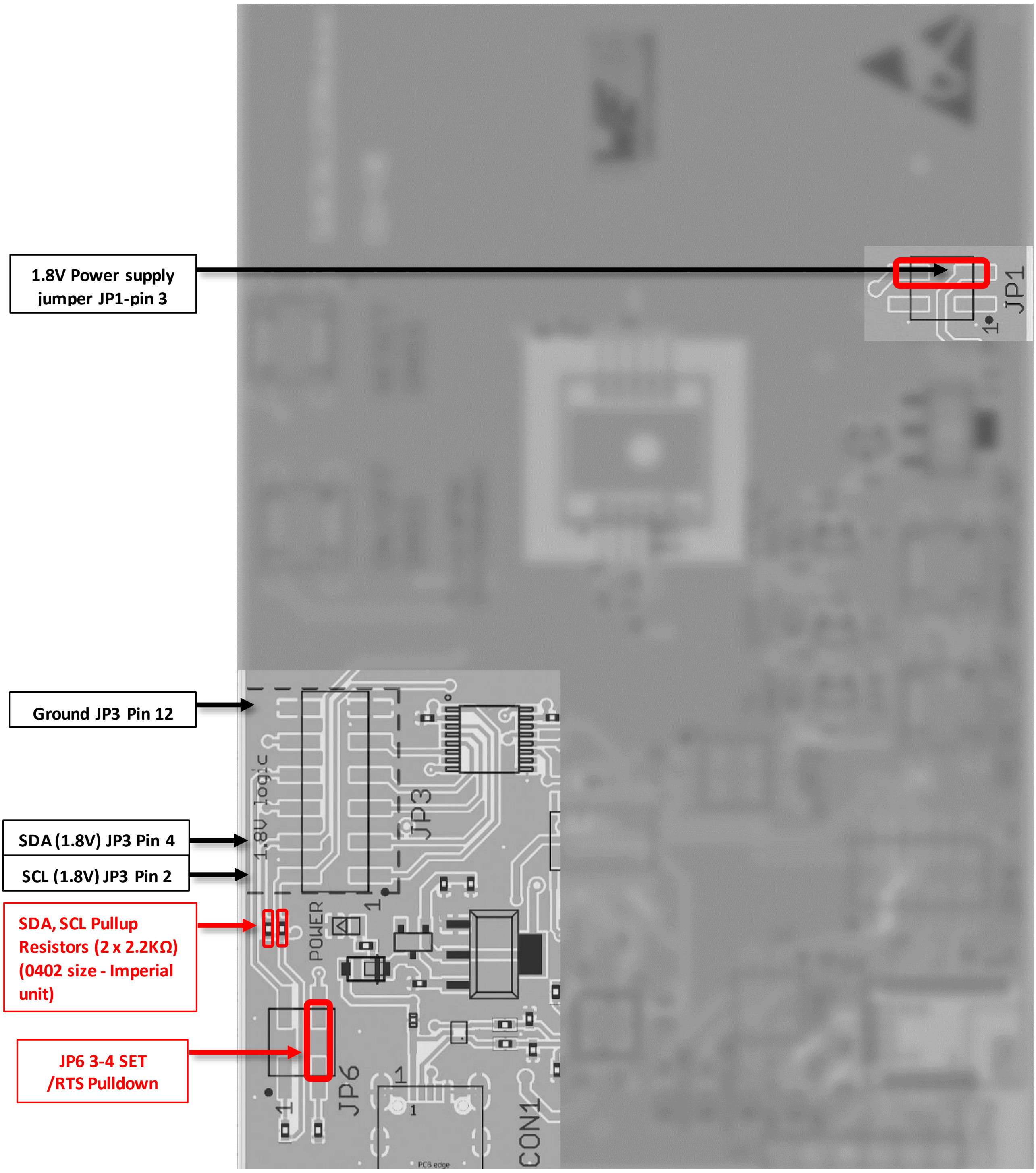

Figure 4: I²C evaluation board modification-1.8 V

Figure 4 illustrates the necessary modification to be done on the evaluation board for I²C communication.

The highlighted area in figure 4 is common for the evaluation boards of Elara-I, Elara-II and Erinome-II modules.

The following changes must be done:

- Solder 2.2kΩ pull-up resistors on the SDA and SCL bus;
- Connecting 10 kΩ pull-down resistor on */RTS* line by switching JP6 from default position (1,2) to (3,4).

> Apart from the hardware modifications listed above and from what shown in Figure 4, other jumpers shall be kept in default position, please refer to module specific evaluation board manual.

The I²C bus (1.8 V logic level) can be accessed through the jumper JP3.

Jumper JP3 (1.8 V logic level)	
I²C SCL	Pin 2
I²C SDA	Pin 4
Ground	Pin 12

Table 3: I²C Jumper JP3 Connection-1.8 V

The 1.8 V reference supply can be accessed through JP1 Pin-3.

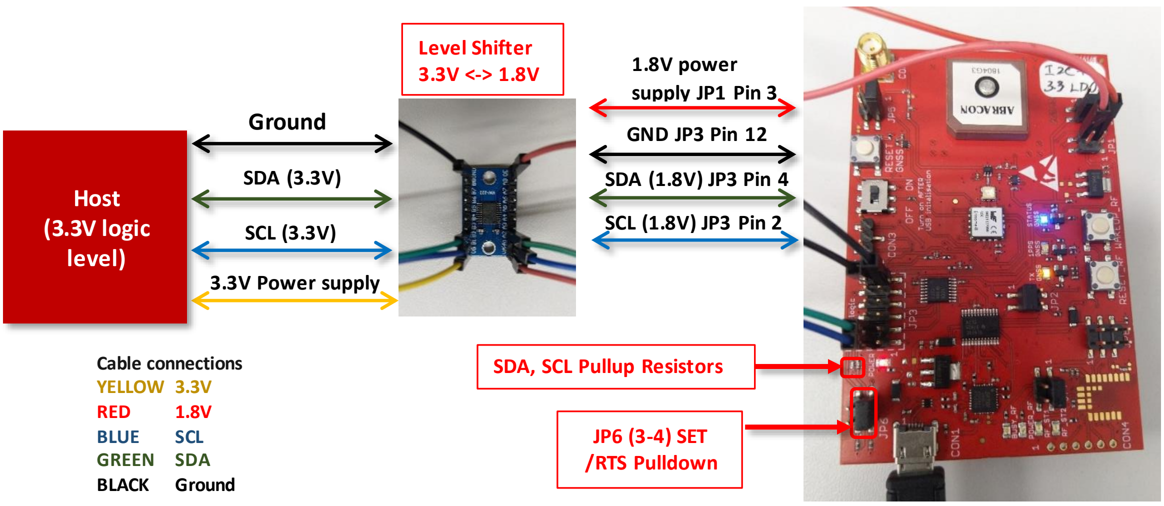

Figure 5: 1.8 V - GNSS evaluation board connection to 3.3 V Host

3.3 Hardware Setup - Erinome-I

In the figure 6 the I^{2}C hardware setup for Erinome-I evaluation board is shown.

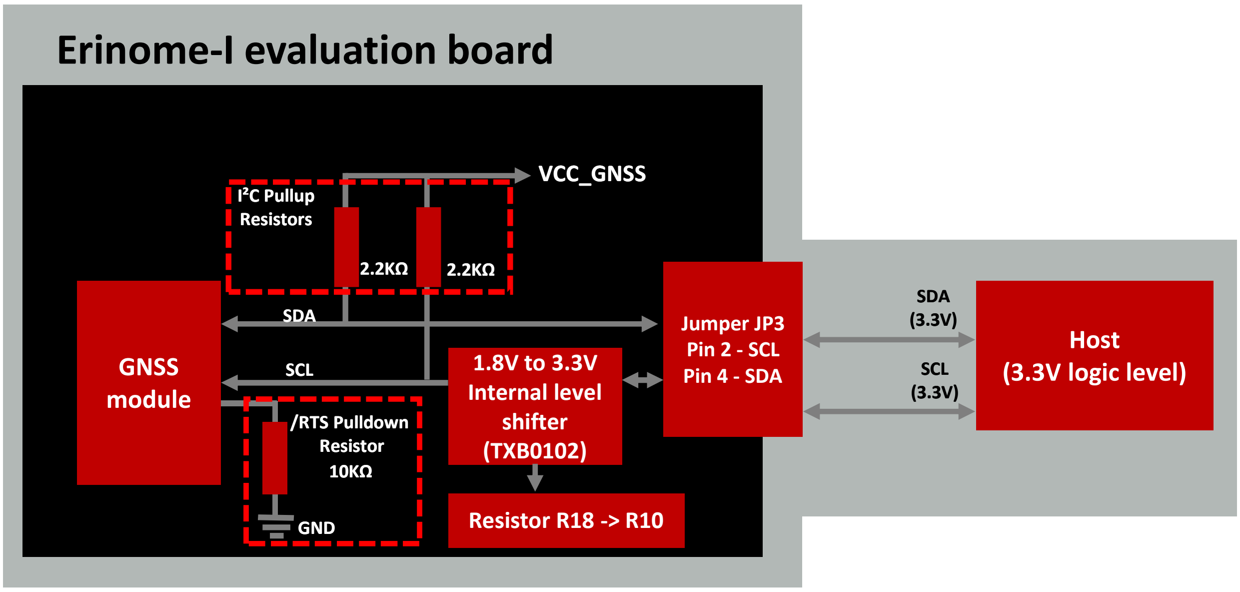

Figure 6: I^{2}C Hardware setup - Erinome-I

Erinome-I module operates with input logic level of 3.3 V, VCC of 3.3 V and output logic level of 1.8 V. For detailed information please refer to the product specific user manual.

Erinome-I evaluation board has an internal level shifter circuit implemented in TX data line (I^{2}C SCL bus) for UART interface operation. This has to be adapted to I^{2}C hardware setup. The internal level shifter used in the evaluation board is TXB0102, which uses push-pull switching circuit for UART operation. Further information of the level shifter can be found at *https://www.ti.com/product*

The pull-up and pull-down resistor assembly needed for the I^{2}C communication and level shifter circuit adaptation are illustrated in the block diagram.

In this setup, the I^{2}C bus outputs on jumper JP3 use 3.3 V logic level. Therefore, an additional external level shifter is not needed for 3.3 V logic level operation.

> **!** For a logic level translation to other logic level, please note that the level shifter shall use an open drain circuit and support I^{2}C communication.

3.4 Evaluation Board Modification - Erinome-I

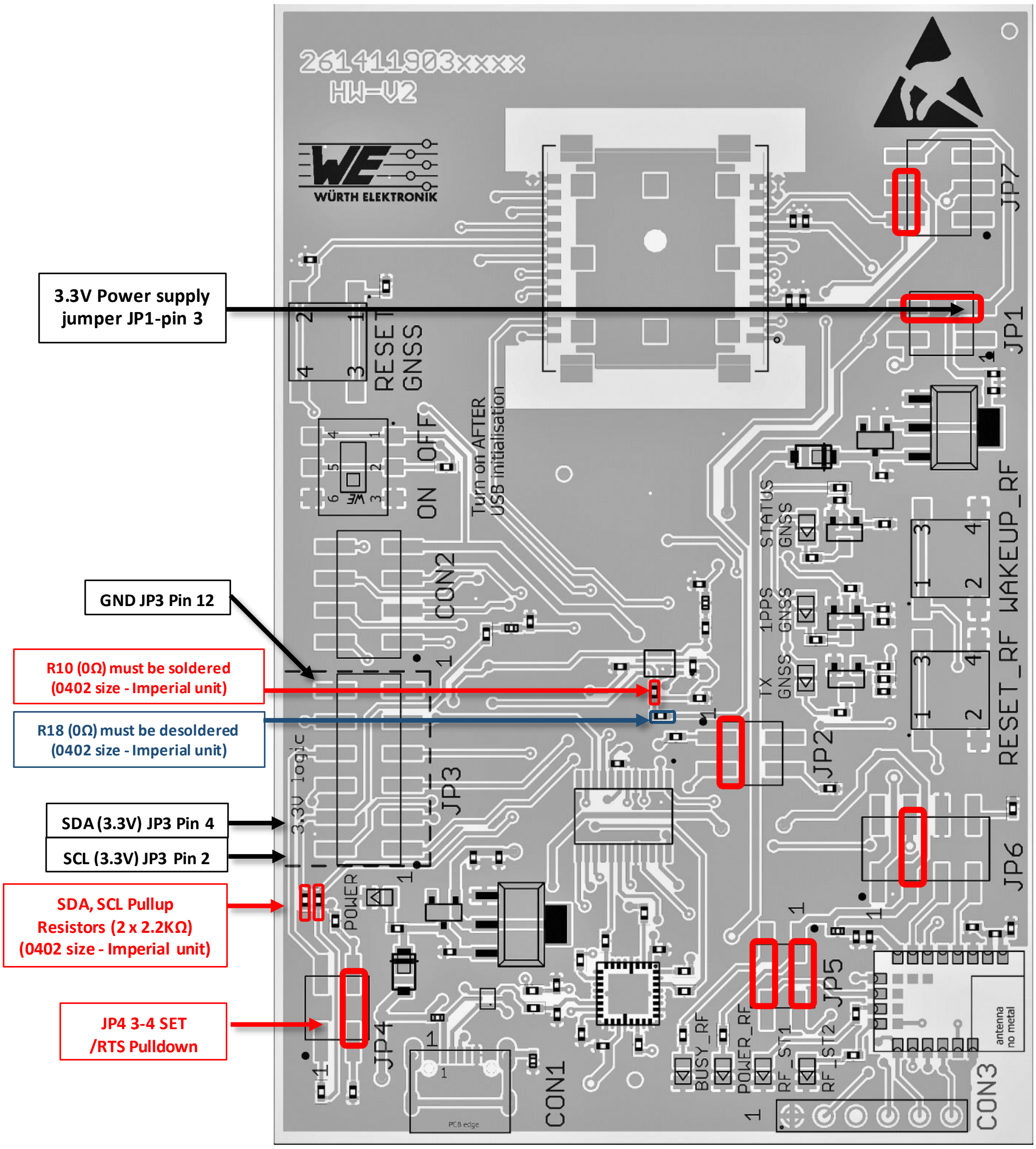

Figure 7: I^{2}C evaluation board modification - Erinome-I

Figure 7 illustrates the necessary modification, including correct jumper settings, to be executed on the Erinome-I evaluation board for I^{2}C communication.

Following modifications must be done:

- Solder 2.2kΩ pull-up resistors on the SDA and SCL bus;
- Connecting 10 kΩ pull-down resistor on */RTS* line by switching jumper JP4 from default position (1,2) to (3-4);

- Desolder R18 (0 Ω) resistor;
- Solder R10 (0 Ω) resistor.

> Apart from the hardware modifications listed above, jumpers shall be set according to Figure 7.

The I²C bus (3.3 V logic level) can be accessed through the jumper JP3.

Jumper JP3 (3.3 V logic level)	
I²C SCL	Pin 2
I²C SDA	Pin 4
Ground	Pin 12

Table 4: I²C Jumper JP3 Connection- Erinome-I

The 3.3 V reference supply can be accessed through JP1 Pin-3.

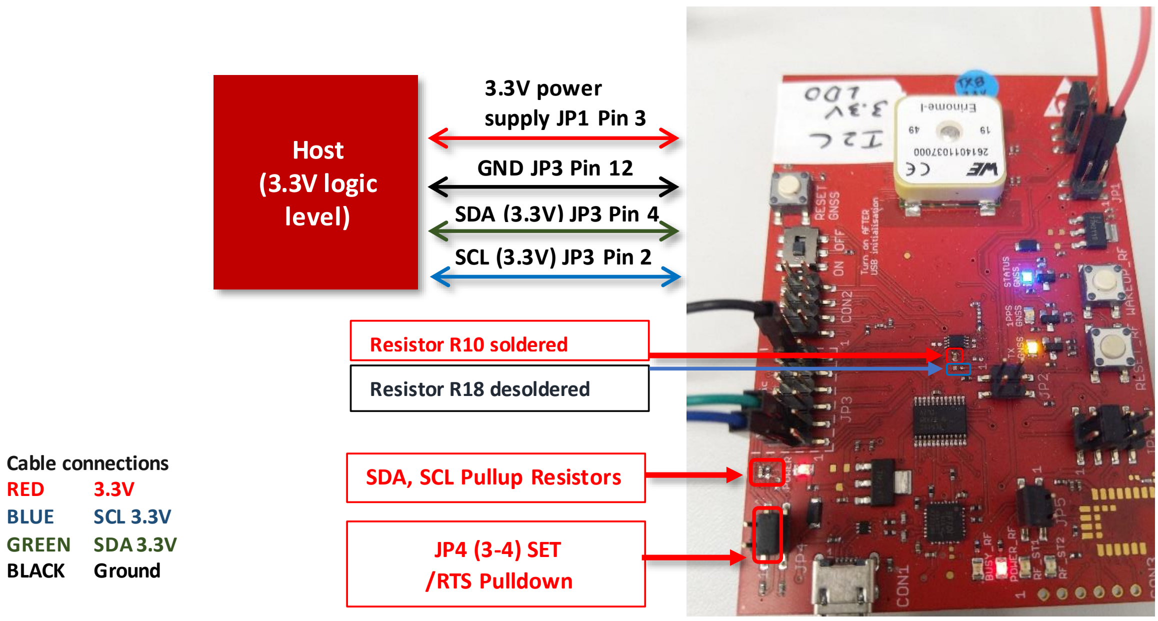

Figure 8: 3.3 V - Erinome-I evaluation board connection to 3.3 V Host

For further information about I²C communication with the Erinome-I, including an application example with Aardvark and its matching PC software, please refer to our dedicated application note ANR018 [1].

4 Application Example

4.1 Total Phase Aardvark host adapter

Aardvark is a bus to USB adapter with a matching Windows PC Software. It allows to access I^2C as well as SPI bus either as master or slave. In case of I^2C slave and master operation, addresses can be configured manually. This allows selecting the address required for communication with the GNSS module. For detailed information please refer to the Aardvark user manual at *https://www.totalphase.com/support/articles/200468316-Aardvark-I2C-SPI-Host-Adapter-User-Manual*.

Figure 9: Aardvark bus to USB adapter

In this example, Aardvark takes the role of the host and is used to communicate with the GNSS modules. Aardvark adapter has USB-B 2.0 connector for the PC connection and ribbon cable connector for the bus connection. Aardvark adapter has a standard 10-wire 1.25mm pitch ribbon cable with a standard 2.54mm pitch IDC type ribbon cable connector.

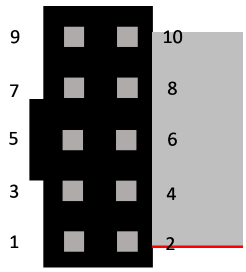

Figure 10: Aardvark connector pinout

Function	Pin number
SCL	1
GND	2
SDA	3

Table 5: Aardvark pins used for I^2C communication

The Control Center Serial Software is used to configure the Aardvark for I²C communication. This software tool, usable with the Aardvark hardware only, can be downloaded from Total Phase. The USB drivers for Aardvark are also available through Total Phase.

> We cannot offer any support that extends this description. In case of further questions regarding the Aardvark tool, please get in direct contact with the manufacturer Total Phase.

Aardvark input logic level is 3.3 V. Therefore, corresponding logic level translation is necessary to establish communication between GNSS modules and Aardvark. For the modules with 1.8 V power supply (Elara-I, Elara-II and Erinome-II) the external level shifter as shown in the hardware setup is implemented. The evaluation board of TXS0108 level shifter is used in this application.

In case of Erinome-I evaluation board, SCL and SDA bus connection with 3.3 V logic level from the jumper JP3 can be directly connected to Aardvark.

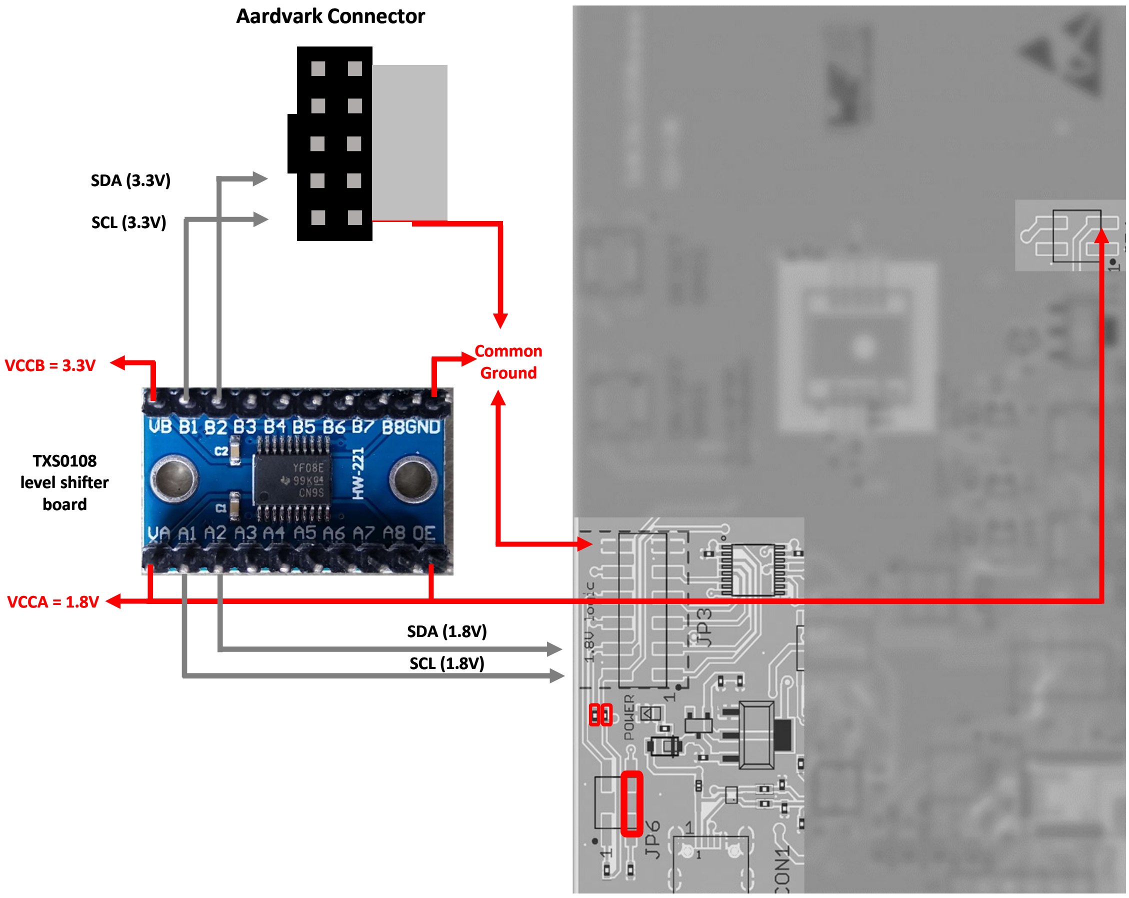

Figure 11: Aardvark Host to GNSS evaluation board - 1.8 V

Connection	Level shifter pin	GNSS evaluation board
1.8 V supply	VA	JP1 - pin 3
1.8 V supply (same as VA Pin)	OE	JP1 - pin 3
3.3 V supply	VB	Host
SCL 1.8 V to level shifter	A1	JP3 - pin 2
SDA 1.8 V to level shifter	A2	JP3 - pin 4
Common ground connection	GND	JP3 - pin 12

Table 6: GNSS evaluation board to TXS0108 level shifter connection

Connection	Level shifter pin	Aardvark pin
SCL 3.3 V to Aardvark	B1	1
SDA 3.3 V to Aardvark	B2	3
Common ground connection	GND	2

Table 7: TXS0108 level shifter to Aardvark connection

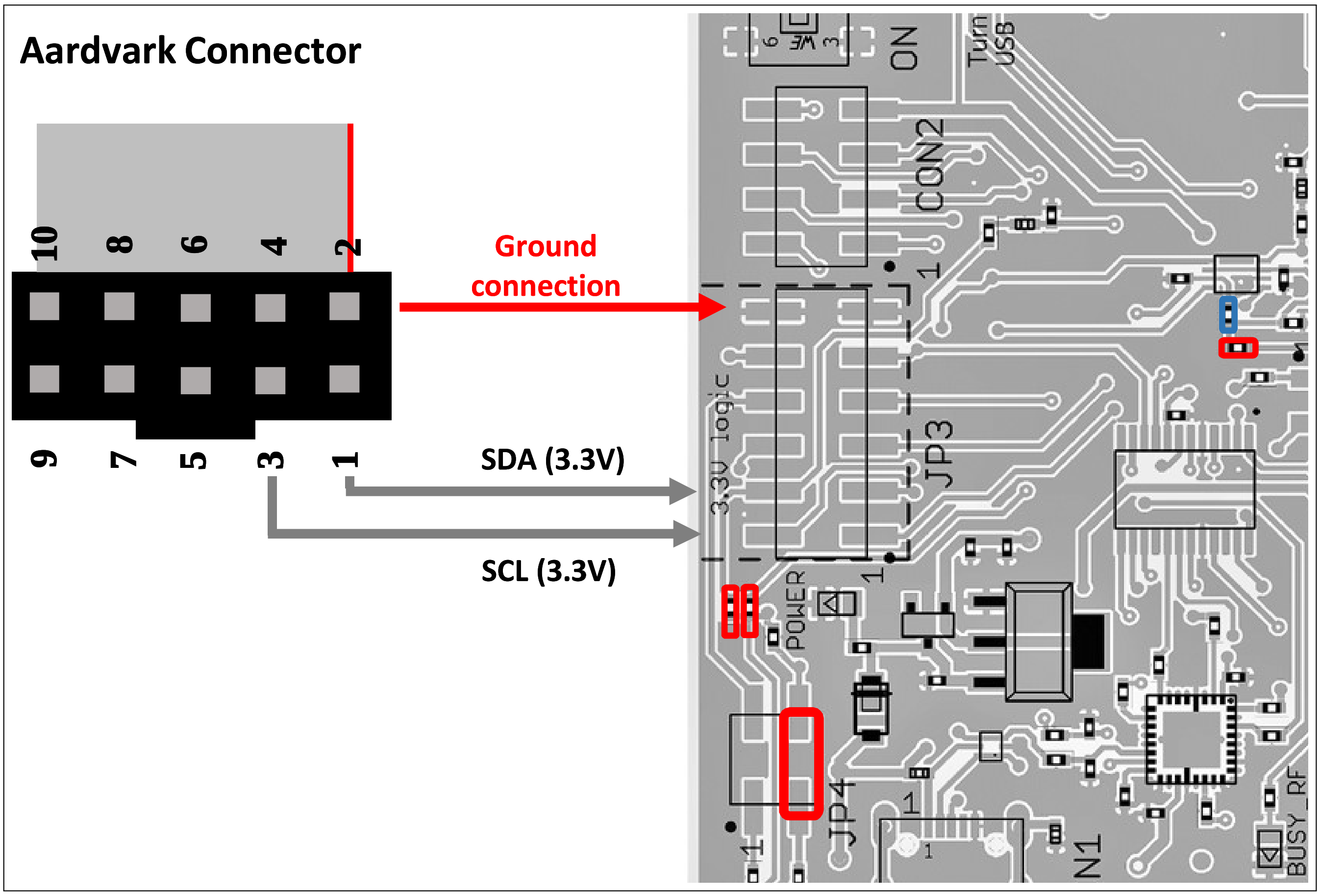

Figure 12: Aardvark to 3.3 V (Erinome-I) evaluation board

Connection	GNSS evaluation board JP3 pin	Aardvark pin
SCL 3.3 V to Aardvark	2	1
Ground connection	12	2
SDA 3.3 V to Aardvark	4	3

Table 8: Aardvark to 3.3 V (Erinome-I) evaluation board

To communicate with the GNSS module through Aardvark, prepare the hardware setup according to the GNSS module used. After that, follow the steps below to establish I^2C communication between the GNSS module and Aardvark.

- Connect the GNSS evaluation board to PC through a micro USB cable.
- Make sure that the module is in hibernate mode. Please refer to module evaluation board specific user manual.
- Module power supply (VCC) is stable and able to reliably supply the module's static and peak current consumption as specified by the module manual.
- Connect the Aardvark USB cable to PC.
- Open control center serial software in PC.

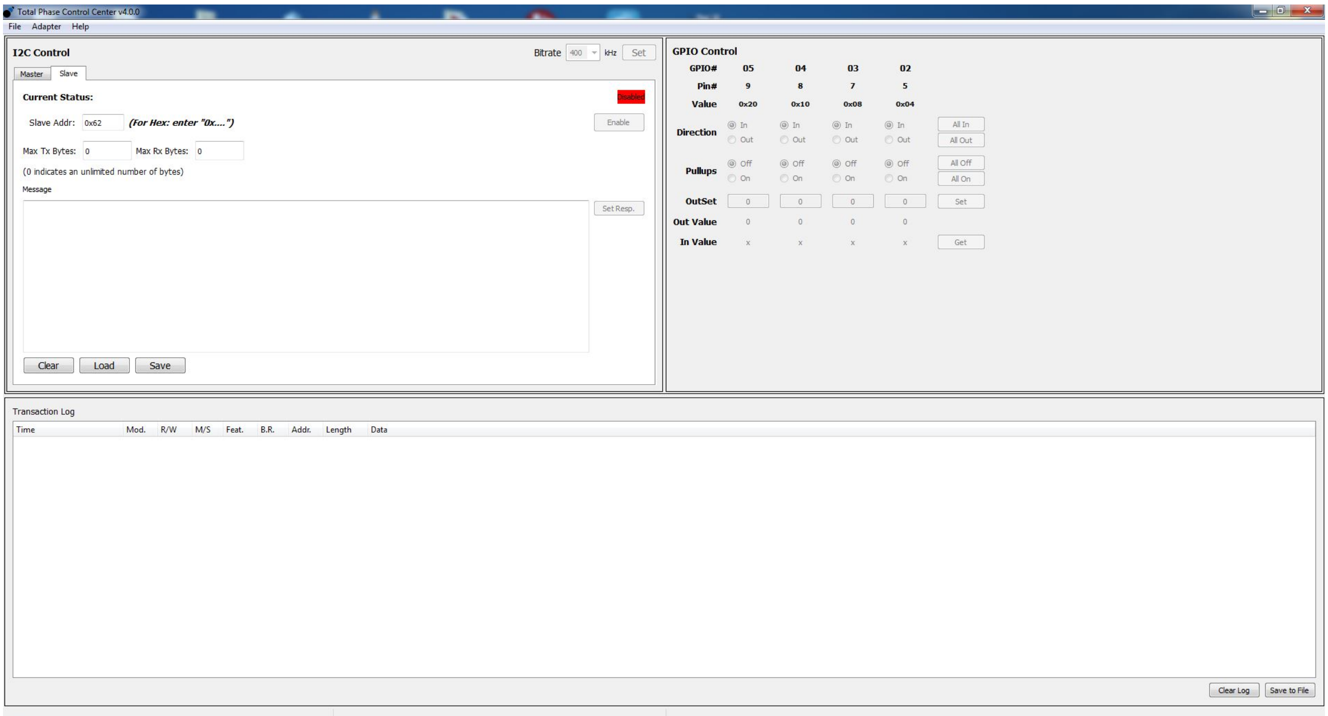

Figure 13: Control center serial software open window

- Select 'Adapter-> Connect'.
- Select 'I2C - GPIO'.
- Click 'OK'.

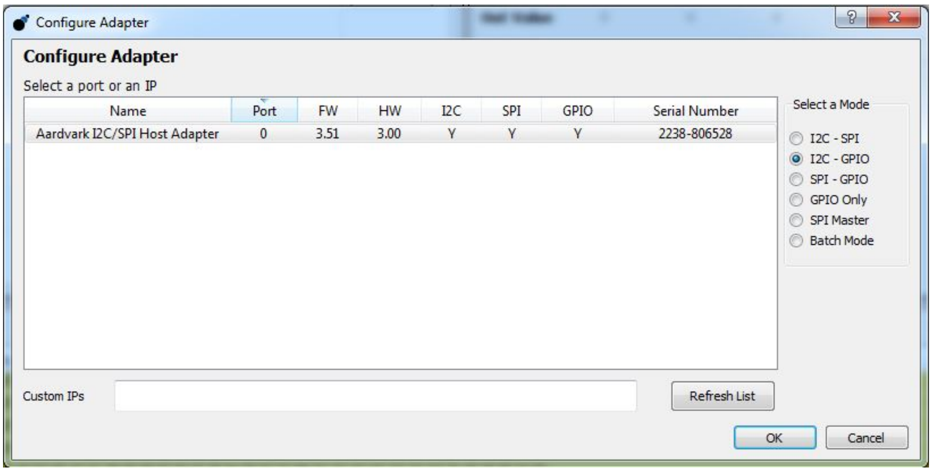

Figure 14: Connect I^2C

- Select Slave tab.
- Select the bitrate to 400kHz from the dropdown list.
- Enter '0x62' Slave Addr.
- Click 'Set'.

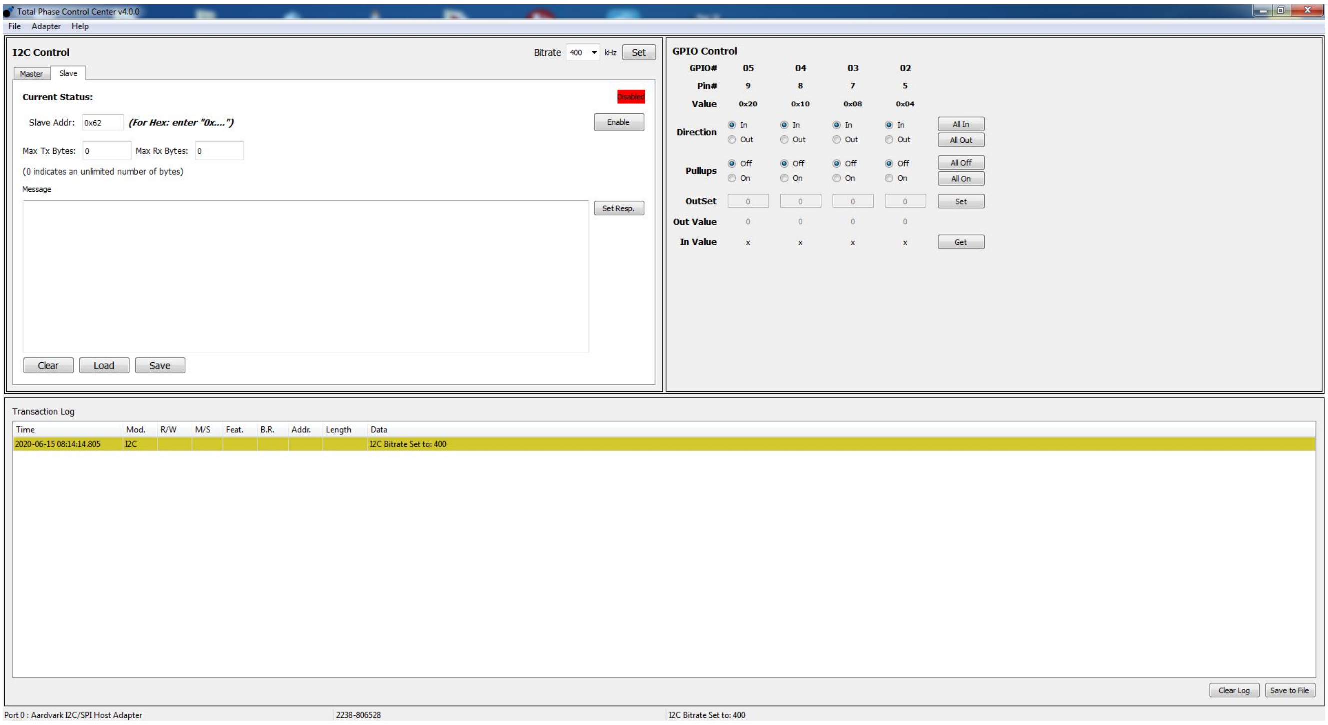

Figure 15: I^2C Bitrate set

- Make sure 'Adapter -> I2C Pullups' is deactivated.
- Click 'Enable' in the Slave tab.
- Switch GNSS module into full power mode. Please refer to module evaluation board specific user manual.

After the steps are properly followed the I^2C interface communication between module and Aardvark will be started. GNSS messages from the module can be viewed in the transaction log of the control center serial software in hexadecimal form. These messages in the transaction log have to be manually converted into ASCII form.

For information about the NMEA messages supported by the GNSS module, please refer to the module specific user manual.

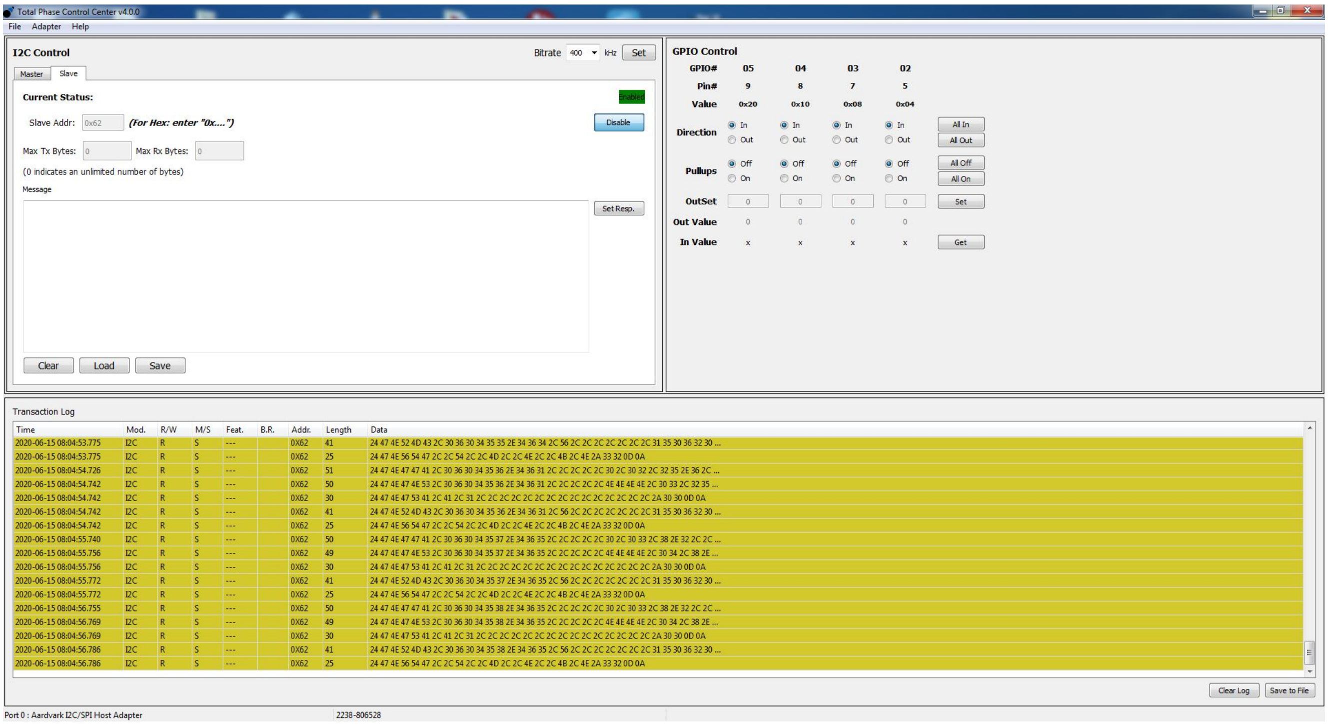

Figure 16: I^{2}C communication

4.2 Generic Microcontroller

The following flowchart shows the principle that needs to be implemented into the µC firmware to allow a communication with the GNSS module over the I^2C interface and the protocol used in the GNSS module.

The protocol used by the GNSS module is typically different to common sensor protocols. Therewith many existing drivers or hardware abstraction layers are not matching the requirements of the GNSS protocol. It may be necessary to modify existing drivers or write new drivers.

A µC host driver for use with the GNSS module is required to support at least 400kHz clock, slave mode, 7-bit addressing with the own address 0x62 and a variable I^2C transfer length.

The application in the host is also required to support concatenating and parsing NMEA messages that will be sent by the GNSS module. For information regarding NMEA messages supported and used by the GNSS module refer to the module specific user manual.

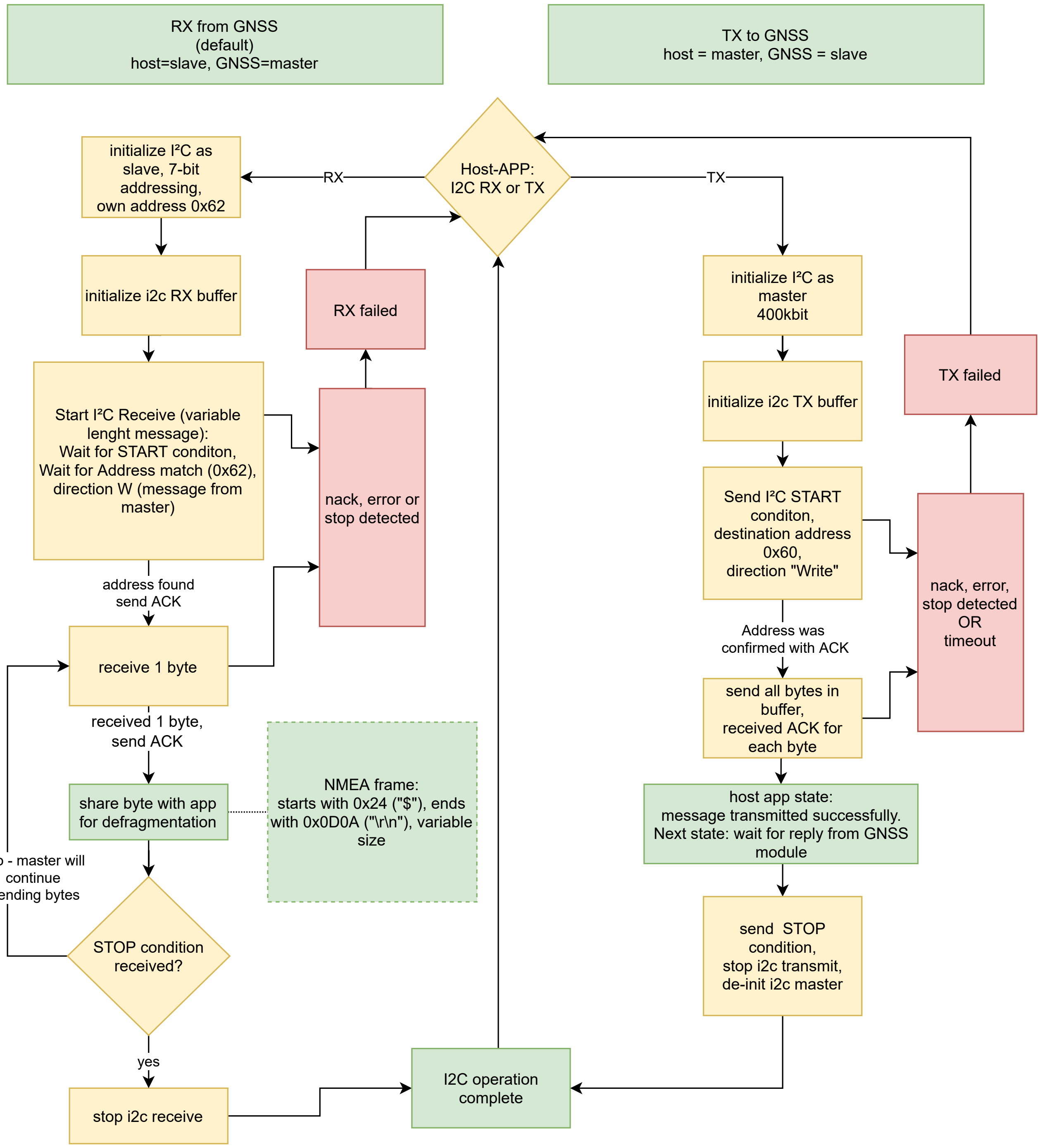

Figure 17: Flowchart for host implementation

5 Troubleshooting

If the communication between host and GNSS modules cannot be established, please verify the following points.

- Check the logic level compatibility between GNSS module and host.
- Check hardware setup is properly connected according to the relevant GNSS module used.
- Check if recommended hardware modifications on the relevant GNSS evaluation board are made.
  - For Elara-I, Elara-II and Elara-II evaluation board hardware modification refer to chapter 3.2
  - For Erinome-I evaluation board hardware modification refer to chapter 3.4
- Check if the level shifter supports the required logic level translation between GNSS module and Host.
- Check whether the level shifter uses open drain circuit and is I^{2}C compatible.
- Check whether the level shifter supports the required bitrate of 400 kbits/s.
- Check power up sequence- 'Module Power up-> Reset -> Hibernate mode -> Full power mode'. Refer to the module specific manual for further information.
- Check if pull-down on */RTS* line is active during power up.
- Check if I^{2}C operation mode 'Master or Slave' on GNSS module and Host.
- Check the bitrate on both Host and GNSS module.
- Check I^{2}C bus signals using a logic analyser.
- Check if pull-up resistors are needed in I^{2}C bus on the host controller.
- Check if the length of I^{2}C bus is kept as short as possible.

6 References

[1] Würth Elektronik. Application note 18 - GNSS I2C communication. http://www.we-online.com/ANR018.

7 Important notes

The Application Note and its containing information ("Information") is based on Würth Elektronik eiSos GmbH & Co. KG and its subsidiaries and affiliates ("WE eiSos") knowledge and experience of typical requirements concerning these areas. It serves as general guidance and shall not be construed as a commitment for the suitability for customer applications by WE eiSos. While WE eiSos has used reasonable efforts to ensure the accuracy of the Information, WE eiSos does not guarantee that the Information is error-free, nor makes any other representation, warranty or guarantee that the Information is completely accurate or up-to-date. The Information is subject to change without notice. To the extent permitted by law, the Information shall not be reproduced or copied without WE eiSos' prior written permission. In any case, the Information, in full or in parts, may not be altered, falsified or distorted nor be used for any unauthorized purpose.

WE eiSos is not liable for application assistance of any kind. Customer may use WE eiSos' assistance and product recommendations for customer's applications and design. No oral or written Information given by WE eiSos or its distributors, agents or employees will operate to create any warranty or guarantee or vary any official documentation of the product e.g. data sheets and user manuals towards customer and customer shall not rely on any provided Information. THE INFORMATION IS PROVIDED "AS IS". CUSTOMER ACKNOWLEDGES THAT WE EISOS MAKES NO REPRESENTATIONS AND WARRANTIES OF ANY KIND RELATED TO, BUT NOT LIMITED TO THE NON-INFRINGEMENT OF THIRD PARTIES' INTELLECTUAL PROPERTY RIGHTS OR THE MERCHANTABILITY OR FITNESS FOR A PURPOSE OR USAGE. WE EISOS DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT RELATING TO ANY COMBINATION, MACHINE, OR PROCESS IN WHICH WE EISOS INFORMATION IS USED. INFORMATION PUBLISHED BY WE EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WE eiSos TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

The responsibility for the applicability and use of WE eiSos' components in a particular customer design is always solely within the authority of the customer. Due to this fact it is up to the customer to evaluate and investigate, where appropriate, and decide whether the device with the specific characteristics described in the specification is valid and suitable for the respective customer application or not. The technical specifications are stated in the current data sheet and user manual of the component. Therefore the customers shall use the data sheets and user manuals and are cautioned to verify that they are current. The data sheets and user manuals can be downloaded at *www.we-online.com*. Customers shall strictly observe any product-specific notes, cautions and warnings. WE eiSos reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time without notice.

WE eiSos will in no case be liable for customer's use, or the results of the use, of the components or any accompanying written materials. IT IS CUSTOMER'S RESPONSIBILITY TO VERIFY THE RESULTS OF THE USE OF THIS INFORMATION IN IT'S OWN PARTICULAR ENGINEERING AND PRODUCT ENVIRONMENT AND CUSTOMER ASSUMES THE ENTIRE RISK OF DOING SO OR FAILING TO DO SO. IN NO CASE WILL WE EISOS BE LIABLE FOR

CUSTOMER'S USE, OR THE RESULTS OF IT'S USE OF THE COMPONENTS OR ANY ACCOMPANYING WRITTEN MATERIAL IF CUSTOMER TRANSLATES, ALTERS, ARRANGES, TRANSFORMS, OR OTHERWISE MODIFIES THE INFORMATION IN ANY WAY, SHAPE OR FORM.

If customer determines that the components are valid and suitable for a particular design and wants to order the corresponding components, customer acknowledges to minimize the risk of loss and harm to individuals and bears the risk for failure leading to personal injury or death due to customers usage of the components. The components have been designed and developed for usage in general electronic equipment only. The components are not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the components is reasonably expected to cause severe personal injury or death, unless WE eiSos and customer have executed an agreement specifically governing such use. Moreover WE eiSos components are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation, transportation signal, disaster prevention, medical, public information network etc. WE eiSos must be informed about the intent of such usage before the design-in stage. In addition, sufficient reliability evaluation checks for safety must be performed on every component which is used in electrical circuits that require high safety and reliability functions or performance. COSTUMER SHALL INDEMNIFY WE EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF THE COMPONENTS IN SUCH SAFETY-CRITICAL APPLICATIONS.

List of Figures

List of Tables

Contact
Würth Elektronik eiSos GmbH & Co. KG
Division Wireless Connectivity & Sensors

Max-Eyth-Straße 1
74638 Waldenburg
Germany

Tel.: +49 651 99355-0
Fax.: +49 651 99355-69
www.we-online.com/wireless-connectivity

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT