

Kappa4413USB

USB Isolated 4–20 mA Current-Loop Transmitter 3-Wire

Presentation

The **Kappa4413USB** is an isolated, **USB-controlled 4–20 mA current loop transmitter** (3-wire configuration) based on the **IS4413-M1** I2C 4–20 mA Current Loop Transmitter module. With it, you can generate a current loop signal from a PC **without writing any firmware**: just send simple text commands through a virtual COM port, from any serial terminal program or from your own software.

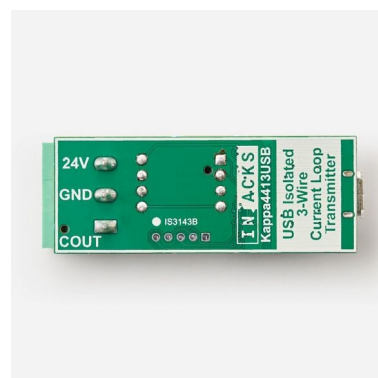
The board features the IS4413-M1 module, a **USB-to-UART bridge**, a **digital isolator** that galvanically isolates the PC from the current loop, a **microcontroller** that translates the text commands into IS4413-M1 I2C commands, and a **pluggable screw terminal** for the 24 V power supply and the current loop output.

The Kappa4413USB is designed to **evaluate the IS4413-M1** before integrating it into your own product, and to be used as a **ready-to-use PC-controlled current source**: to test and calibrate PLC analog inputs, to simulate sensors and transmitters, to drive current loop actuators, or to automate laboratory and production tests.

This manual describes the Kappa4413USB board, its firmware and how to use it. For the detailed specifications of the IS4413-M1 module, see the IS4413-M1 datasheet (ISDOC150).

IS4413-M1 Characteristics	
Current Loop Connection Type:	3-Wire
Output Current:	0 mA to 22 mA
Resolution:	12-bit (values 0 to 4095)
Safe Power-Up Current:	Configurable (stored in the module EEPROM)
Detailed Specifications:	IS4413-M1 datasheet (ISDOC150)

Kappa4413USB Characteristics	
PC Interface:	USB, Micro-B connector
PC Connection:	Virtual COM port (PL2303GL USB-to-UART bridge)
Serial Settings:	115200 baud, 8 data bits, no parity, 1 stop bit, no flow control
Command Protocol:	ASCII text lines, compatible with any serial terminal program
Available Commands:	Set Current Value, Set Default Value, Firmware Version
Isolation:	USB side galvanically isolated from the current loop side
Power Supply:	24 V through the screw terminal (the USB side is powered from USB)
Current Loop Connector:	3-way pluggable screw terminal: COUT, GND and 24V
Protections:	Reverse polarity protection on the 24 V input, reverse current protection on COUT, TVS diodes and EMI filters on the terminal lines



1 Ordering Information

Standard configuration

The standard configuration refers to the device variant that is mass-produced and offered at the most cost-efficient price.

Orderable Device	Status	Package	Connection	Description
IS4413-M1 www.inacks.com/is4413	ACTIVE	 Module 15.24×17.78 mm (Bread Board Compatible)	8 flat-castellated pads	I2C 4–20 mA current loop transmitter, 3-wire configuration. Solder it into your own design and control it from your microcontroller via I2C.
Kappa4413USB www.inacks.com/kappa4413usb	ACTIVE	 Module 49×19 mm	1x USB 1x Terminal Block	IS4413-M1 with isolated USB interface and screw terminal. Control the 4–20 mA output from a PC, from a terminal program or your own Python script.

Status Description:

ACTIVE: Product recommended for new designs.

SPECIAL: Available only for high-volume pre-orders. Contact sales.

NRND: Device is in production to support existing customers, but it is not recommended for use in new designs.

OBSOLETE: Production of the device has been discontinued.

2 Revision History

This manual and the IS4413-M1 datasheet may be revised without notice. Users are responsible for obtaining and reviewing the latest version of this document and the datasheet before purchasing the product or designing it into any application. The most recent revisions of this manual and of the datasheet, available on the company's website (www.inacks.com), shall be considered the only valid and authoritative source of product information.

Document Revision

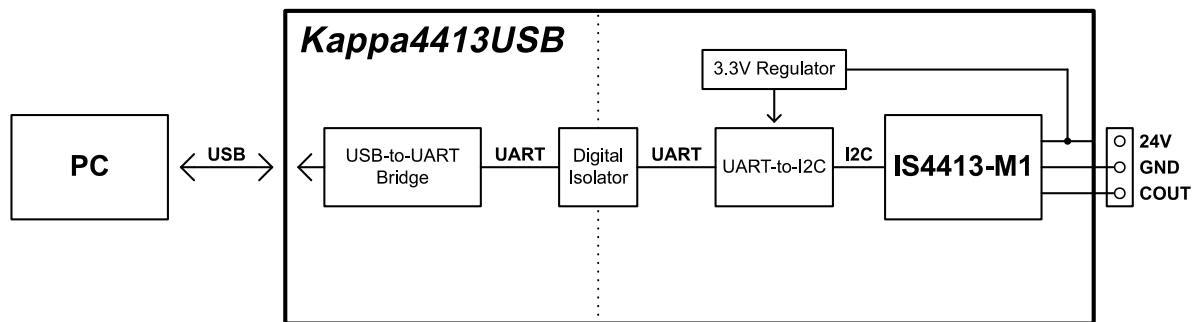
Date	Revision Code	Description
October 2026	ISDOC157A	- Initial Release

Module Revision

Date	Revision Code	Description
October 2026	ISB3143B	- Initial Release

3 Description

The Kappa4413USB converts the text commands sent by a PC into IS4413-M1 I2C commands:



SCH0104A

- **USB-to-UART bridge:** the PC sees the Kappa4413USB as a serial (COM) port. The bridge (PL2303GL) converts the USB data into a UART signal for the microcontroller. This side of the board is powered from the USB port.
- **Digital Isolator:** carries the UART signals across the isolation barrier. The PC ground and the current loop ground are not connected, which prevents ground loops and protects the PC from the current loop wiring.
- **UART-to-I2C:** receives each command, checks it, sends the corresponding I2C command to the IS4413-M1 and replies to the PC with OK, an error code or the requested data.
- **IS4413-M1:** generates the loop current on the COUT terminal.
- **3.3 V Regulator:** powers the UART-to-I2C from the 24 V power supply.

Because the UART-to-I2C is powered from the 24 V power supply, the Kappa4413USB only answers commands while the 24 V supply is connected. The COM port appears on the PC as soon as the USB cable is connected, even without the 24 V supply.

At power-up, the **IS4413-M1** generates its stored default current (see the Set Default Value command). The Kappa4413USB does not change the output current until it receives a command from the PC.

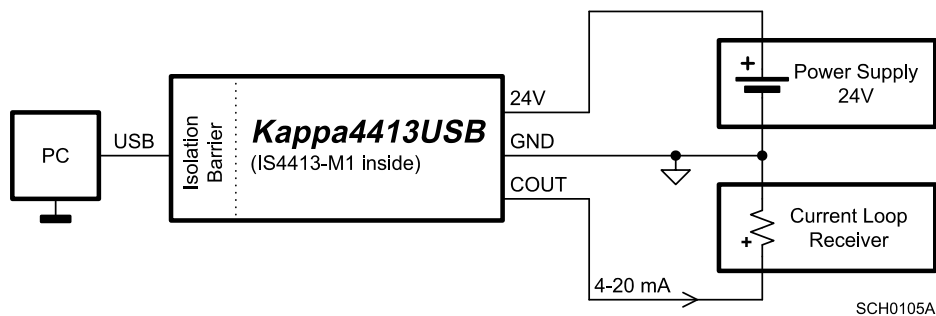
4 Pinout

The Kappa4413USB has one Micro-B USB port and one pluggable screw terminal block.



Connector	Pin	Description
Terminal Block	24V	24 V power supply input (+). It powers the IS4413-M1 and the UART-to-I2C IC. Protected against reverse polarity.
	GND	0 V reference of the current loop side. Connect the negative (–) of the 24 V power supply and the return (–) of the current loop receiver. It is isolated from the USB ground.
	COUT	Current loop output. Connect it to the input (+) of the current loop receiver, typically a PLC analog input or a 250 Ω resistor.
USB	—	Micro-B USB connector. Connect it to the PC.

The following figure shows the typical connection. As the IS4413-M1 is a 3-wire transmitter, the return (–) of the current loop receiver must be connected to the 0 V of the 24 V power supply, together with the GND terminal.



5 Getting Started

5.1 USB Driver

The Kappa4413USB uses a Prolific PL2303GL USB-to-UART bridge, which the PC sees as a standard serial port:

- **Windows:** the driver is usually installed automatically the first time the board is connected. The port appears in the Device Manager, under *Ports (COM & LPT)*. If it does not appear, install the PL2303GL driver from the Prolific website.
- **Linux:** recent kernels include the PL2303GL driver. The port appears as `/dev/ttyUSB0` (or the next free number). Your user needs permission to open it, typically by being a member of the `dialout` group.
- **macOS:** if the port does not appear, install the PL2303GL driver from the Prolific website. The port appears as a `/dev/cu.*` device.

5.2 Quick Start

1. Connect the 24V terminal to the positive (+) of the 24 V power supply, the GND terminal to its negative (–) and to the return (–) of the current loop receiver, and the COUT terminal to the input (+) of the receiver. Then connect the Kappa4413USB to the PC with a USB cable.
2. Open the COM port at 115200 baud, 8 data bits, no parity, 1 stop bit and no flow control, with a serial terminal program, Python script or with your own software.
3. Type `ver` and press Enter. The Kappa4413USB answers 1, its firmware version. If there is no answer, see section Troubleshooting.
4. Type `725` and press Enter. The Kappa4413USB answers OK and the output current changes to approximately 4 mA.

5.3 Output Value and Current

The output current is set with a value from 0 to 4095 (12-bit). The value 0 generates the minimum output current (0 mA) and 4095 the maximum (approximately 22 mA). Between these limits, the output current is proportional to the value. The following table shows approximate values for the standard current loop range:

Output Current	Value
4 mA	725
8 mA	1454
12 mA	2182
16 mA	2911
20 mA	3640

These values are approximate: the exact current depends on each unit and on the temperature. For maximum accuracy, measure the output current at two values (for example, near 4 mA and near 20 mA) and calculate the values from your own measurements.

6 Communication Protocol

6.1 Serial Port Settings

Parameter	Value
Baud rate	115200
Data bits	8
Parity	None
Stop bits	1
Flow control	None
Character set	ASCII
Line ending of the commands	CR, LF or CR+LF
Line ending of the replies	CR+LF
Echo	No. The Kappa4413USB does not return the characters it receives.

6.2 Command Format

- Each command is one line of text ending with Enter (CR, LF or CR+LF). The Kappa4413USB executes the command when it receives the line ending.
- Values are decimal numbers from 0 to 4095, without sign, decimal point or spaces. Leading zeros are accepted (0725 is the same as 725).
- Letters can be upper or lower case: ver and VER, d and D are equivalent.
- Empty lines are ignored.
- The Kappa4413USB answers every command with one reply line. Wait for the reply before sending the next command.

6.3 Commands

Command	Format	Example	Description	Reply
Set Current Value	<value> (0 to 4095)	3640	Sets the output current. The Kappa4413USB sends the IS4413-M1 Set Current Value command. This is the normal operation.	OK
Set Default Value	<value>d	0d	Sets the output current and stores the value in the IS4413-M1 EEPROM as the safe power-up current. The Kappa4413USB sends the IS4413-M1 Set Default Value command and waits 50 ms for the EEPROM write to finish before replying.	OK
Firmware Version	ver	ver	Returns the firmware version number, digits only.	1

Note: every Set Default Value command writes the IS4413-M1 EEPROM, which supports a limited number of write cycles (see the IS4413-M1 datasheet). Use it only to change the power-up current, not to update the output current continuously. For the recommended safe power-up current, see the IS4413-M1 datasheet.

6.4 Replies and Error Codes

The Kappa4413USB answers every command with one line ending with CR+LF:

- OK: the command was executed.
- A number: the answer to the ver command.
- ERRn: followed by a description: the command was not executed. The number n identifies the error, as shown in the following table.

Code	Reply	Cause	Solution
ERR1	ERR1: missing value, send a number from 0 to 4095	The command has no value, for example d.	Send the value before the d.
ERR2	ERR2: value out of range, must be 0 to 4095	The value is greater than 4095.	Send a value from 0 to 4095.
ERR3	ERR3: unexpected 'x' after 'd'	There are characters after the d, for example 100d5.	The d must be the last character.
ERR4	ERR4: invalid character 'x', use digits and optional 'd'	The command contains a character that is not allowed: a space, a sign, a decimal point or a letter other than d.	Send digits only, optionally followed by d.
ERR5	ERR5: IS4413-M1 is not responding, is the 24V power present?	The IS4413-M1 did not answer the I2C command.	Check the 24 V power supply and its connection.
ERR6	ERR6: I2C bus error, check for noise or a shorted SDA/SCL line	Error on the internal I2C bus.	Repeat the command. If the error persists, power-cycle the board.
ERR7	ERR7: I2C timeout, SDA or SCL may be held low	The internal I2C bus is blocked.	Power-cycle the board.
ERR8	ERR8: I2C communication failed	Other internal I2C error.	Repeat the command. If the error persists, power-cycle the board.
ERR9	ERR9: busy, command ignored, wait for OK before sending the next one	A new command arrived before the previous one was finished. The new command was discarded.	Wait for the reply before sending the next command.

- In ERR3 and ERR4, 'x' is the character that caused the error. Characters that cannot be printed are shown as '?'.
• If a command contains several errors, only the first one is reported.
- ERR9 is sent once, after the reply to the command in progress, even if several commands were discarded.
- To power-cycle the board, disconnect the 24 V power supply and connect it again.

7 Examples

7.1 Python Example

This example (ISXMPL4413ex2) controls the Kappa4413USB from a Python 3 script with the pyserial library. It defines a small Kappa4413USB class that you can reuse in your own scripts:

- `send()` sends one command, reads the reply and raises an exception if the Kappa4413USB answers with an error or does not answer.
- `version()`, `set_value()` and `set_default()` implement the three commands.
- `set_ma()` converts a current in mA into a value, using the two points `VALUE_AT_4MA` and `VALUE_AT_20MA`. Replace them with your own measurements for better accuracy.

Before running it, set `PORT` to the COM port of your Kappa4413USB. The script reads the firmware version, stores 0 mA as the power-up current, generates 4, 8, 12, 16 and 20 mA for two seconds each and finally shows how an error is reported.

Find the complete example and buy the Kappa4413USB at: www.inacks.com/kappa4413usb

```
# -----
# Kappa4413USB - Python example
# Controls the IS4413-M1 output current from a PC through the Kappa4413USB.
# Requirements: Python 3 and pyserial (install with: pip install pyserial)
# -----

import time
import serial

PORT = "COM3" # Windows: "COM3", Linux: "/dev/ttyUSB0"

# Approximate conversion from mA to output value (0..4095).
# For the best accuracy, calibrate your unit and replace these two points.
VALUE_AT_4MA = 725
VALUE_AT_20MA = 3640

class Kappa4413USB:
    def __init__(self, port):
        # 115200 baud, 8 data bits, no parity, 1 stop bit (8N1)
        self.ser = serial.Serial(port, 115200, timeout=1)
        self.ser.reset_input_buffer()

    def close(self):
        self.ser.close()

    def send(self, command):
        """Sends one command and returns the reply without the line ending."""
        self.ser.write((command + "\n").encode("ascii"))
        reply = self.ser.readline().decode("ascii").strip()
        if reply == "":
            raise TimeoutError("No reply: check the COM port and the 24 V supply")
        if reply.startswith("ERR"):
            raise RuntimeError(reply)
        return reply

    def version(self):
        """Returns the firmware version number."""
        return int(self.send("ver"))

    def set_value(self, value):
        """Set Current Value: updates the output current (value 0..4095)."""
        self.send(str(value))

    def set_default(self, value):
        """Set Default Value: stores the power-up current in the IS4413-M1."""
        self.send(str(value) + "d")

    def set_ma(self, ma):
        """Sets the output current in mA using the conversion points above."""
```

```
        value = VALUE_AT_4MA + (ma - 4) * (VALUE_AT_20MA - VALUE_AT_4MA) / 16
        self.set_value(max(0, min(4095, round(value))))

kappa = Kappa4413USB(PORT)
try:
    print("Firmware version:", kappa.version())

    # Optional, only needed once: output 0 mA at power-up (safe power-up)
    kappa.set_default(0)

    # Generate 4, 8, 12, 16 and 20 mA, two seconds each
    for ma in (4, 8, 12, 16, 20):
        kappa.set_ma(ma)
        print("Output set to", ma, "mA")
        time.sleep(2)

    # Errors reported by the Kappa4413USB raise an exception
    try:
        kappa.set_value(5000)
    except RuntimeError as error:
        print("Kappa4413USB answered:", error)
finally:
    kappa.close()
```

7.2 Serial Terminal Example

Any serial terminal program can control the Kappa4413USB: PuTTY, Tera Term, CoolTerm, minicom, the Arduino IDE Serial Monitor, etc. Configure the COM port as described and make sure that the program sends a line ending (CR, LF or both) when you press Enter.

The following example uses PuTTY, a free terminal program for Windows and Linux:

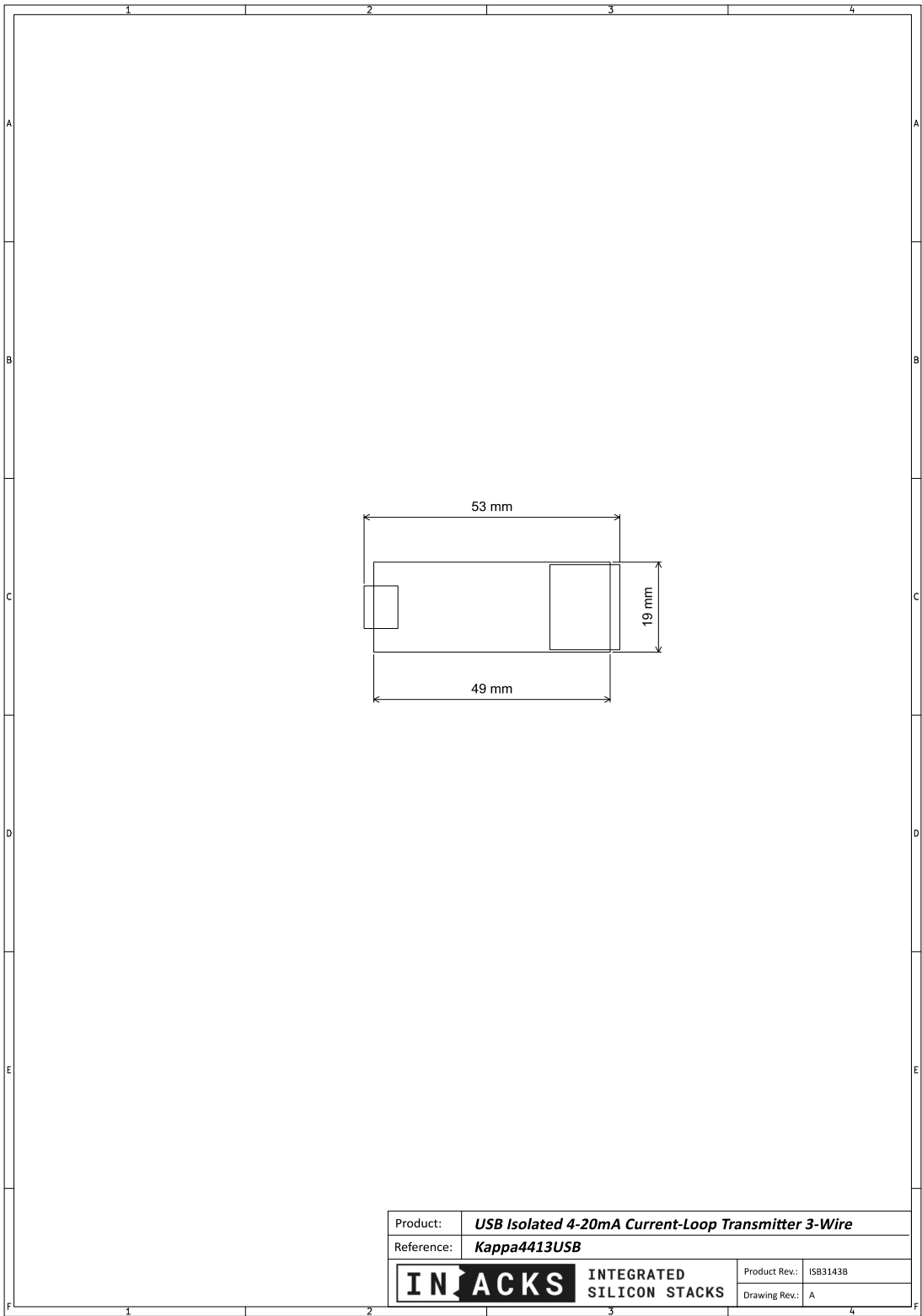
1. In the *Session* category, select *Serial* as the connection type, enter the COM port in *Serial line* (for example, COM3) and 115200 in *Speed*.
2. In the *Connection > Serial* category, select 8 data bits, 1 stop bit, no parity and no flow control.
3. In the *Terminal* category, set *Local echo* and *Local line editing* to *Force on*. As the Kappa4413USB does not echo the received characters, local echo shows what you type, and local line editing lets you correct a command before pressing Enter.
4. Click *Open*, type the commands and press Enter after each one.

Example session (the commands typed by the user are shown in bold):

```
ver
1
725
OK
3640
OK
0d
OK
5000
ERR2: value out of range, must be 0 to 4095
12.5
ERR4: invalid character '.', use digits and optional 'd'
```

In this session, `ver` returns the firmware version; `725` and `3640` set the output current to approximately 4 mA and 20 mA; `0d` sets 0 mA and stores it as the power-up current, so the IS4413-M1 outputs 0 mA every time the 24 V supply is switched on, until the first command arrives; `5000` and `12.5` are rejected with the error codes ERR2 and ERR4, and the output current does not change.

8 Mechanical



9 Troubleshooting

Symptom	Possible Cause	Solution
No COM port appears when the USB cable is connected.	The USB driver is not installed, or the USB cable only carries power.	Install the PL2303GL driver. Try another USB cable.
The COM port opens, but the Kappa4413USB never answers.	The 24 V power supply is not connected (it powers the microcontroller), the COM port or the serial settings are wrong, or no line ending is sent.	Connect the 24 V power supply. Check the COM port and the settings (115200 baud, 8N1, no flow control). Make sure your program sends CR or LF.
The typed characters are not shown.	The Kappa4413USB does not echo the received characters.	Enable local echo in your terminal program.
The reply is ERR5.	The IS4413-M1 does not answer, for example because the 24 V supply voltage is too low.	Check the voltage between the 24V and GND terminals.
The reply is OK, but there is no current in the loop.	The loop is open: the receiver is not connected between COUT and the 0 V of the power supply.	Check the wiring (section Pinout).
The current is slightly different from the expected value.	Unit-to-unit variation and temperature.	Calculate the values from your own measurements.
The reply is ERR9.	Commands are sent faster than the Kappa4413USB executes them.	Wait for each reply before sending the next command.

Content

Presentation.....	1
1 Ordering Information.....	2
2 Revision History.....	3
3 Description.....	4
4 Pinout.....	5
5 Getting Started.....	6
5.1 USB Driver.....	6
5.2 Quick Start.....	6
5.3 Output Value and Current.....	6
6 Communication Protocol.....	7
6.1 Serial Port Settings.....	7
6.2 Command Format.....	7
6.3 Commands.....	7
6.4 Replies and Error Codes.....	8
7 Examples.....	9
7.1 Python Example.....	9
7.2 Serial Terminal Example.....	11
8 Mechanical.....	12
9 Troubleshooting.....	13
Content.....	14
Appendix.....	15
Documentation Feedback.....	15
Sales Contact.....	15
Customization.....	15
Independence and Trademarks Notice.....	15
Disclaimer.....	16

Appendix

Documentation Feedback

Feedback and error reporting on this document are very much appreciated.

feedback@inacks.com

Sales Contact

For special order requirements, large volume orders, or scheduled orders, please contact our sales department at:

sales@inacks.com

Customization

INACKS can develop new products or customize existing ones to meet specific client needs. Please contact our engineering department at:

engineering@inacks.com

Independence and Trademarks Notice

This company and the products provided herein are developed independently and are not affiliated with, endorsed by, or associated with any official protocol or standardization entity.

All trademarks, names, and references to specific protocols remain the property of their respective owners.

Disclaimer

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, INACKS does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. INACKS takes no responsibility for the content in this document if provided by an information source outside of INACKS.

In no event shall INACKS be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, INACKS's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of INACKS.

Right to make changes — INACKS reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — INACKS products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an INACKS product can reasonably be expected to result in personal injury, death or severe property or environmental damage. INACKS and its suppliers accept no liability for inclusion and/or use of INACKS products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. INACKS makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using INACKS products, and INACKS accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the INACKS product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

INACKS does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using INACKS products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). INACKS does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — INACKS products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.inacks.com/comercialsaleterms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. INACKS hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of INACKS products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — This INACKS product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. INACKS accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

Protocol Guidance Disclaimer: The information provided herein regarding the protocol is intended for guidance purposes only. While INACKS strive to provide accurate and up-to-date information, this content should not be considered a substitute for official protocol documentation. It is the responsibility of the client to consult and adhere to the official protocol documentation when designing or implementing systems based on this protocol.

INACKS make no representations or warranties, either expressed or implied, as to the accuracy, completeness, or reliability of the information contained in this document. INACKS shall not be held liable for any errors, omissions, or inaccuracies in the information or for any user's reliance on the information.

The client is solely responsible for verifying the suitability and compliance of the provided information with the official protocol standards and for ensuring that their implementation or usage of the protocol meets all required specifications and regulations. Any reliance on the information provided is strictly at the user's own risk.

Certification and Compliance Disclaimer: Please be advised that the product described herein has not been certified by any competent authority or organization responsible for protocol standards. INACKS do not guarantee that the chip meets any specific protocol compliance or certification standards.

It is the responsibility of the client to ensure that the final product incorporating this product is tested and certified according to the relevant protocol standards before use or commercialization. The certification process may result in the product passing or failing to meet these standards, and the outcome of such certification tests is beyond our control.

INACKS disclaim any liability for non-compliance with protocol standards and certification failures. The client acknowledges and agrees that they bear sole responsibility for any legal, compliance, or technical issues that arise due to the use of this product in their products, including but not limited to the acquisition of necessary protocol certifications.