Kappa4420USB

USB Isolated 4–20 mA Current-Loop Receiver

Presentation

The **Kappa4420USB** is a **USB Isolated 4–20 mA Current-Loop Receiver** based on the **IS4420-M1** I2C/UART 4–20 mA Current Loop Receiver module. Connect it to a PC and read the loop current **without writing any firmware**: the readings arrive as lines of text on a virtual COM port, ready for the free IS4420WIN software, a serial terminal or your own program.

The board carries the **IS4420-M1** module, a **USB-to-UART bridge** with an isolation barrier between the PC and the current loop, and a **pluggable screw terminal** for the loop. The loop input is **protected** against reversed wires, a 24 V supply connected straight to it, and voltage spikes.

Use the **Kappa4420USB** to **evaluate the IS4420-M1**, or as a **ready-to-use current loop meter** on a PC: to read 4–20 mA temperature, pressure and flow transmitters, to test and calibrate sensors, and to log the readings with IS4420WIN.

This manual describes the Kappa4420USB board and how to use it.

Kappa4420USB Characteristics	
Measuring Range:	0 mA to 21.5 mA
Resolution:	0.01 mA, and 0.1 % of the 4–20 mA span
Calibration:	Factory calibrated
Protection:	Isolated USB; loop input protected against reversed wires, a direct 24 V supply and voltage spikes
Software:	IS4420WIN, free Windows software that displays and logs the readings
Python:	Python compatible: write your own programs to read the loop current
For the detailed specifications of the module, refer to the IS4420-M1 datasheet (ISDOC146)	







1 Ordering Information

Orderable Device	Status	Package	Connection	Description
IS4420-M1 www.inacks.com/is4420	ACTIVE	 Module 12.7×10.16 mm (Breadboard Compatible)	8 flat-castellated pads	I2C/UART 4–20 mA current loop receiver, 2-wire, 3-wire and 4-wire configurations. Solder it into your own design and read it from your microcontroller via I2C or UART.
Kappa4420USB www.inacks.com/ kappa4420usb	ACTIVE	 Board 50×19 mm	1x Micro-B USB 1x Terminal Block	IS4420-M1 with isolated USB interface and screw terminal. Read the 4–20 mA loop current on a PC, with IS4420WIN, 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 IS4420-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

The revision history of this manual.

Date	Revision Code	Description
November 2026	ISDOC158**A**	- Initial Release

Hardware Revision

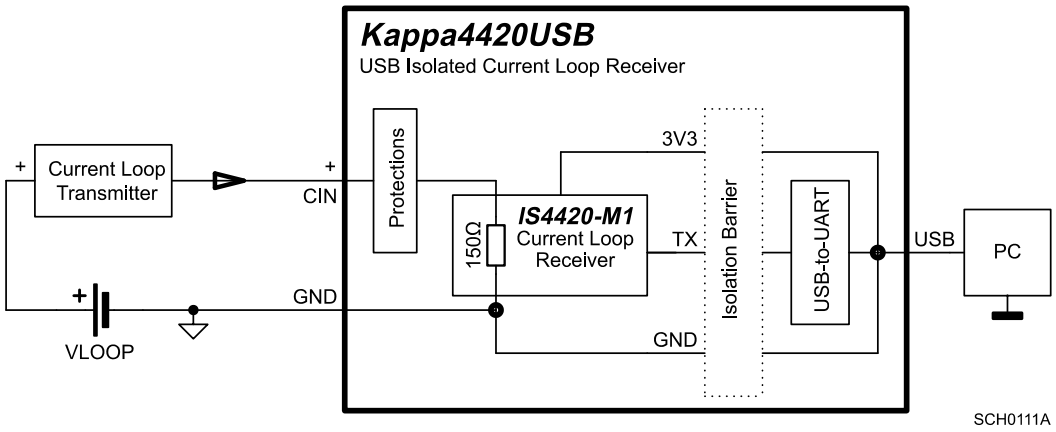
The revision history of the Kappa4420USB hardware.

Date	Revision Code	Description
November 2026	ISB3158**B**	- Silkscreen Improvement
November 2026	ISB3158**A**	- Initial Release

3 Description

The Kappa4420USB makes current loop readings easy. It connects the TX pad of the IS4420-M1 to a PC through an isolation barrier and a USB-to-UART bridge (PL2303GL), so the readings show up on a virtual COM port and the PC stays isolated from the loop. Read them with the free IS4420WIN software, or log them with your own Python scripts.

The Kappa4420USB passes the readings straight to the PC. There are no commands and nothing to configure: open the COM port and the readings arrive.

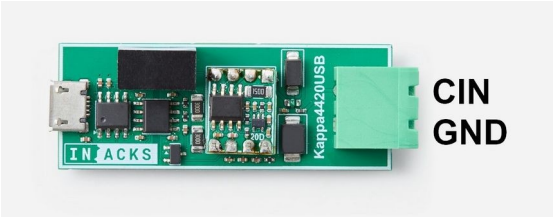


VLOOP is the current loop power supply, from which the transmitter generates the 4–20 mA current. It is an external supply: the Kappa4420USB does not provide VLOOP.

- **IS4420-M1:** the main part of the board. It measures the loop current and has I2C and UART interfaces. It sends the readings to the PC through the UART.
- **USB-to-UART bridge:** the PL2303GL makes the Kappa4420USB appear on the PC as a serial (COM) port.
- **Isolation barrier:** isolates both the USB power and the UART data, so the PC and the current loop do not share a ground.
- **Protections:** the current loop input is protected against:
  - **Reversed wires:** a diode blocks the current, and the reading stays at `00.00mA` until the wires are connected the right way.
  - **Direct 24 V connection:** if a 24 V supply is connected straight between CIN and GND, without a transmitter, a resettable fuse limits the current. The fuse resets by itself once the supply is removed. The fuse depends on temperature: a high ambient temperature increases its resistance. This protection works for a direct 24 V supply, but not for a lower voltage connected the same way.
  - **Voltage spikes:** a TVS diode absorbs the spikes picked up by the loop wiring.

4 Pinout

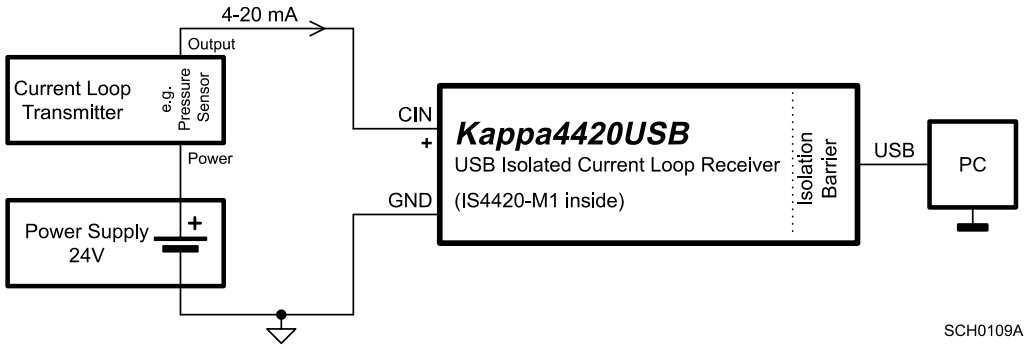
The Kappa4420USB has one Micro-B USB port and one pluggable screw terminal block.



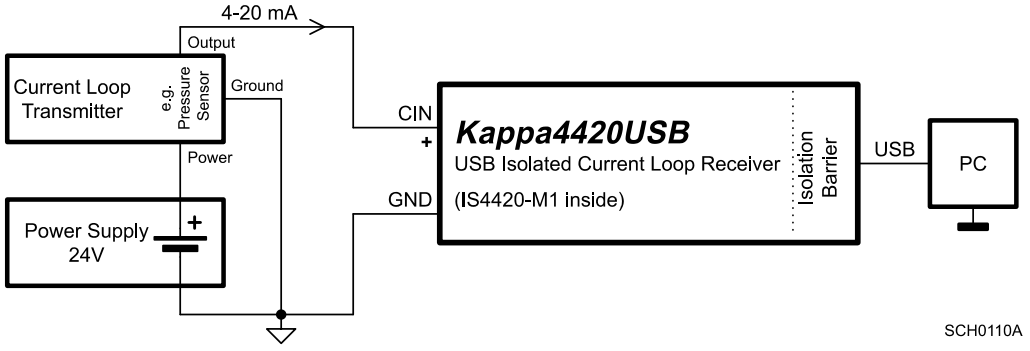
Connector	Pin	Description
Terminal Block	CIN	Current loop input. The loop current enters through this terminal: connect it to the current output of the transmitter.
	GND	Current loop return. In 2-wire and 3-wire loops, connect it to the negative terminal of the loop power supply. It is isolated from the USB ground.
USB	—	Micro-B USB connector. Connect it to the PC. It also powers the board.

The figures below show how to connect a 2-wire and a 3-wire transmitter. The loop current enters through CIN and leaves through GND.

2-Wire Transmitter Connections



3-Wire Transmitter Connections



You can also read 4-wire transmitters, but take special care: some of them have galvanic isolation between their power supply and their current output. Check how their grounds are connected and watch out for ground loops.

The loop power supply must have enough voltage for the transmitter, the cable and the Kappa4420USB. A 24 V supply is the usual choice.

5 Getting Started

5.1 USB Driver

The Kappa4420USB 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 transmitter and the loop power supply to the CIN and GND terminals, as shown in section Pinout. Then connect the Kappa4420USB to the PC with a USB cable, and switch on the loop power supply.
2. Start IS4420WIN, which finds the COM port by itself. Or open the COM port at 115200 baud (8N1, no flow control) with a serial terminal, a Python script or your own software.
3. A new line arrives about every 103 ms, for example `12.34mA_+052.1%_+025deg_IS4420M1_ver0`. The first five characters are the loop current in mA. If no line arrives, see section Troubleshooting.
4. Check the reading: with 4 mA in the loop, the line starts with `04.00mA_+000.0%`, and with 20 mA, with `20.00mA_+100.0%`.

5.3 Current and Percentage

Each line starts with the loop current in mA, followed by its percentage of the 4–20 mA span: 4 mA is 0 % and 20 mA is 100 %. Below 4 mA the percentage is negative. Some examples:

Loop Current	Beginning of the Line
0.00 mA	`00.00mA_-025.0%`
3.00 mA	`03.00mA_-006.3%`
4.00 mA	`04.00mA_+000.0%`
12.00 mA	`12.00mA_+050.0%`
20.00 mA	`20.00mA_+100.0%`
20.50 mA	`20.50mA_+103.1%`
Above 21.50 mA	`EOVERmA_+EOVER%`

The IS4420-M1 is calibrated at the factory, so the readings need no conversion or calibration.

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
Direction	From the Kappa4420USB to the PC only. There are no commands.
Line ending	CR+LF
Line length	39 bytes: 37 characters plus CR+LF
Line rate	One line about every 103 ms (about 9.7 lines per second)

6.2 Line Format

- The IS4420-M1 sends one line of text per measurement, about every 103 ms. The Kappa4420USB passes each line to the PC unchanged.
- Every line has the same length, and every field is always at the same position. An underscore separates the fields.
- Keep only the lines with exactly 37 characters before CR+LF. This skips a line caught halfway when the COM port is opened, and a stray byte that may arrive at power-up.

Example lines:

```
12.34mA_+052.1%_+025deg_IS4420M1_ver0
03.00mA_-006.3%_+025deg_IS4420M1_ver0
EOVERmA_+EOVER%_+025deg_IS4420M1_ver0
ERRORmA_+ERROR%_+ERRdeg_IS4420M1_ver0
```

6.3 Fields

Each line has the format `xx.xxmA_yzzz.z%_stttdeg_IS4420M1_verv`:

Field	Characters	Example	Description	Unit
Current	1 to 5	**`12.34`**	Loop current, with two decimals. `EOVER` above the measuring range and `ERROR` without a measurement.	mA
Percentage	9 to 14	**`+052.1`**	Percentage of the 4–20 mA span, with one decimal. Negative below 4 mA.	%
Temperature	17 to 20	**`+025`**	Temperature inside the IS4420-M1, in whole degrees. `ERR` without a measurement.	°C
Model	25 to 32	**`IS4420M1`**	Model of the module, always `IS4420M1`.	—
Revision	37	**`0`**	Revision of the IS4420-M1. It may change during the product life cycle.	—

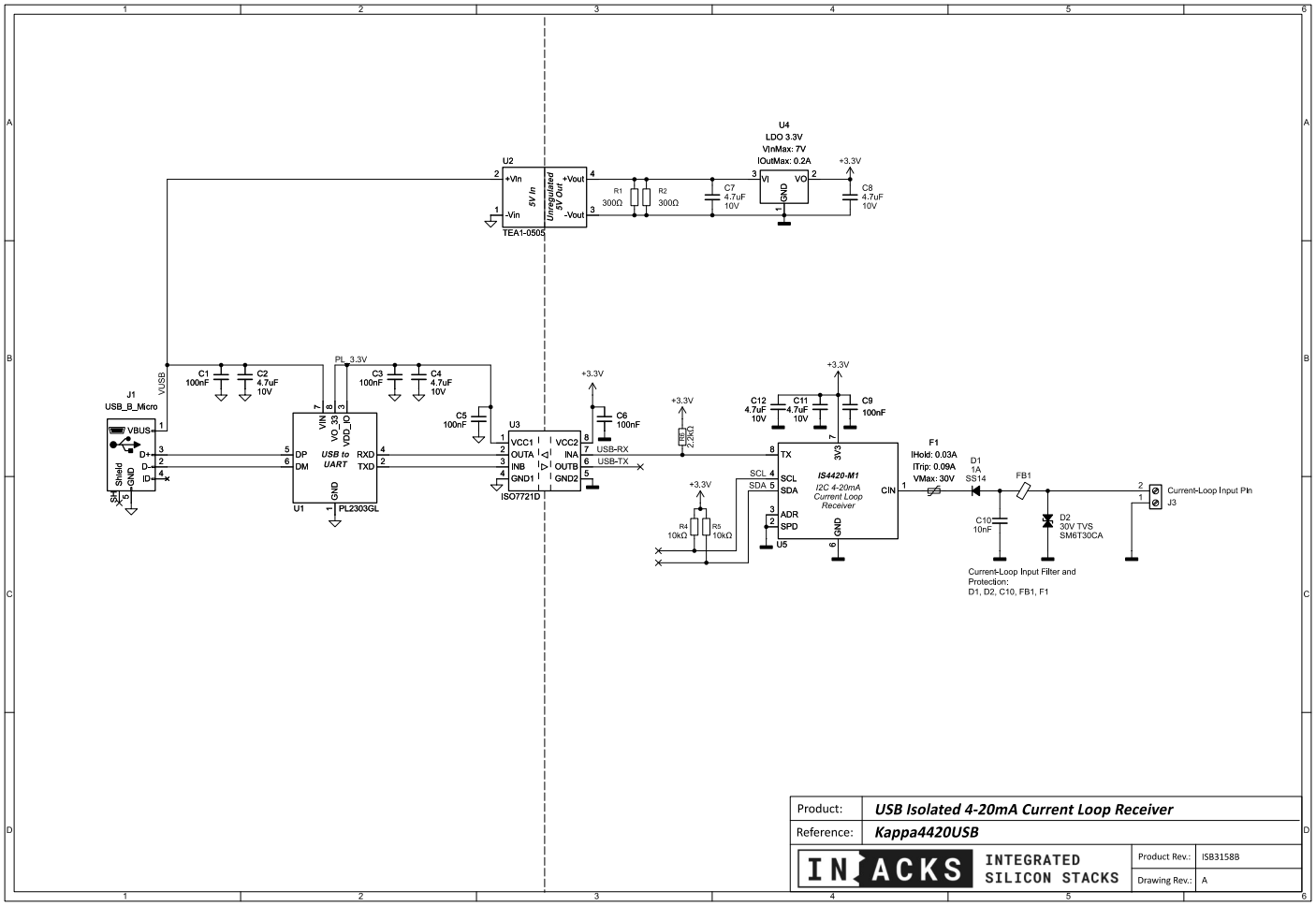
Note: the temperature is measured inside the IS4420-M1 and reads a little above the ambient temperature. Use it to monitor the module, not as a thermometer.

6.4 Error Reporting

When the IS4420-M1 cannot give a valid reading, the current and percentage fields show a code instead of a number:

Code	Line	Cause	Solution
EOVER	`EOVERmA_+EOVER%_+025deg_IS4420M1_ver0`	The loop current is above the measuring range, 21.50 mA.	Check the transmitter: many transmitters drive more than 21 mA to signal a fault.
ERROR	`ERRORmA_+ERROR%_+ERRdeg_IS4420M1_ver0`	The IS4420-M1 could not complete a measurement.	Normal for a moment after power-up. If it persists, disconnect the USB cable and connect it again.

7 Hardware


U4
LDO 3.3V
VinMax: 7V
IOutMax: 0.2A
U2
+Vin
-Vin
+Vout
-Vout
5V In
Unregulated 5V Out
TEA1-0505
R1
300Ω
R2
300Ω
C7
4.7uF
10V
VI
VO
GND
C8
4.7uF
10V
+3.3V
J1
USB_B_Micro
VBUS
D+
D-
ID
GND
Shield
VUSB
C1
100nF
C2
4.7uF
10V
PL_3.3V
C3
100nF
C4
4.7uF
10V
U1
PL2303GL
VIN
VO_33
VDD_IO
DP
DM
USB to UART
RXD
TXD
GND
C5
100nF
U3
ISO7721D
VCC1
OUTA
INB
GND1
VCC2
INA
OUTB
GND2
C6
100nF
USB-RX
USB-TX
TXB
2.2kΩ
C12
4.7uF
10V
C11
4.7uF
10V
C9
100nF
3V3
TX
SCL
SDA
ADR
SPD
IS4420-M1
I2C 4-20mA Current Loop Receiver
CIN
U5
R4
10kΩ
R5
10kΩ
F1
IHold: 0.03A
ITrip: 0.09A
VMax: 30V
D1
1A
SS14
FB1
C10
10nF
D2
30V TVS
SM6T30CA
J3
Current-Loop Input Pin
Current-Loop Input Filter and Protection:
D1, D2, C10, FB1, F1
Product: USB Isolated 4-20mA Current Loop Receiver
Reference: Kappa4420USB
INTEGRATED SILICON STACKS
Product Rev.: ISB3158B
Drawing Rev.: A

8 Software Examples

8.1 Python Example

This example (ISXMPL4420ex5) reads the Kappa4420USB from a Python 3 script with the pyserial library. It defines a small `Kappa4420USB` class that you can reuse in your own scripts:

- `read_line()` returns the next complete line. It raises an exception if no data arrives within one second.
- `read()` splits the line and returns the current in mA, the percentage and the temperature. The current and the percentage are `None` above the measuring range (`EOVER`), and the three values are `None` without a measurement (`ERROR`).

Set `PORT` to the COM port of your Kappa4420USB before running it. The script prints the readings, about ten per second, until you press Ctrl+C.

Find more information and buy the Kappa4420USB at: www.inacks.com/kappa4420usb

```
# ------------------------------------------------------------------------------
# Kappa4420USB - Python example
# Reads the IS4420-M1 loop current from a PC through the Kappa4420USB.
# Requirements: Python 3 and pyserial (install with: pip install pyserial)
# ------------------------------------------------------------------------------
import serial

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


class Kappa4420USB:
    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 read_line(self):
        """Waits for the next complete line and returns it without CR LF."""
        while True:
            raw = self.ser.readline()
            if not raw:
                raise TimeoutError("No data: check the port and the cable")
            # A complete line has 37 characters plus CR LF. Skip anything
            # else, such as a line caught halfway when the port opens.
            if len(raw) == 39 and raw.endswith(b"\r\n"):
                line = raw[:37].decode("ascii", "replace")
                if line.count("_") == 4:
                    return line

    def read(self):
        """Returns the current (mA), the percentage and the temperature
        (degC). The current and the percentage are None above the
        measuring range (EOVER), and all three are None without a
        measurement (ERROR)."""
        current, percent, temp, model, version = self.read_line().split("_")
        if current == "ERRORmA":                    # no measurement
            return None, None, None
        degc = int(temp[:-3])                       # "+025deg" -> 25
        if current == "EOVERmA":                    # above the measuring range
            return None, None, degc
        # "12.34mA" -> 12.34 and "+052.1%" -> 52.1
        return float(current[:-2]), float(percent[:-1]), degc


kappa = Kappa4420USB(PORT)
try:
    print("Line received:", kappa.read_line())
```

```
    # Show every reading, about ten per second, until Ctrl+C is pressed
    while True:
        ma, pct, degc = kappa.read()
        if ma is not None:
            print(f"{ma:5.2f} mA  {pct:+6.1f} %  {degc:+d} degC")
        elif degc is not None:
            print("Over-range: the current is above the measuring range")
        else:
            print("No measurement")
except KeyboardInterrupt:
    pass
finally:
    kappa.close()
```

8.2 Serial Terminal Example

Any serial terminal program can show the readings of the Kappa4420USB: PuTTY, Tera Term, CoolTerm, minicom, the Arduino IDE Serial Monitor, etc. Configure the COM port as described and open it: there is nothing to send.

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. Click *Open*. A new line appears about every 103 ms.

Example session, with some of the lines received while the loop current changes:

```
04.00mA_+000.0%_+025deg_IS4420M1_ver0
04.00mA_+000.0%_+025deg_IS4420M1_ver0
12.34mA_+052.1%_+025deg_IS4420M1_ver0
20.00mA_+100.0%_+026deg_IS4420M1_ver0
EOVERmA_+EOVER%_+026deg_IS4420M1_ver0
03.00mA_-006.3%_+026deg_IS4420M1_ver0
00.00mA_-025.0%_+026deg_IS4420M1_ver0
```

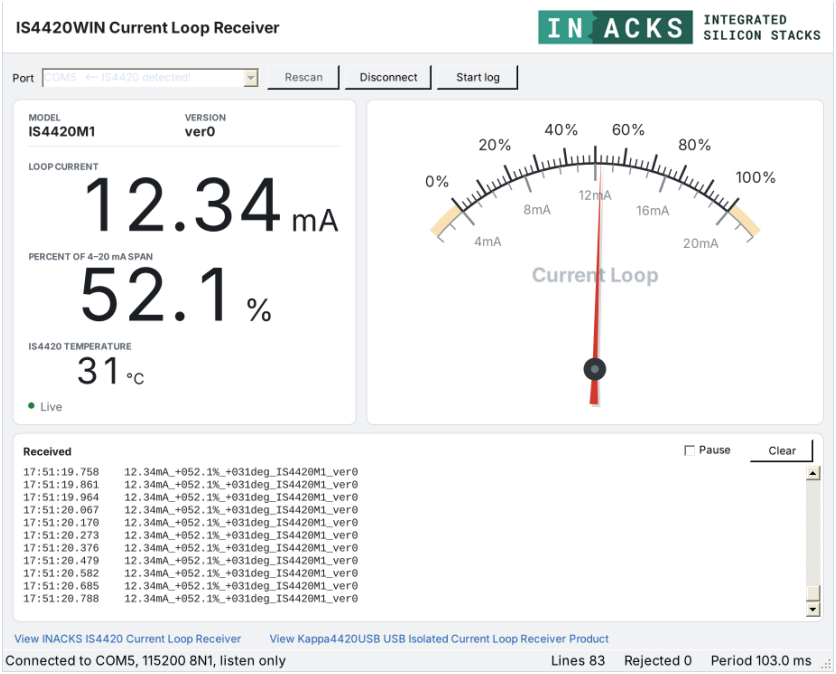
In this session, the current rises from 4 mA to 20 mA and above the measuring range (`EOVER`), then drops below 4 mA and down to 0 mA, for example because the loop was opened.

8.3 IS4420WIN Software

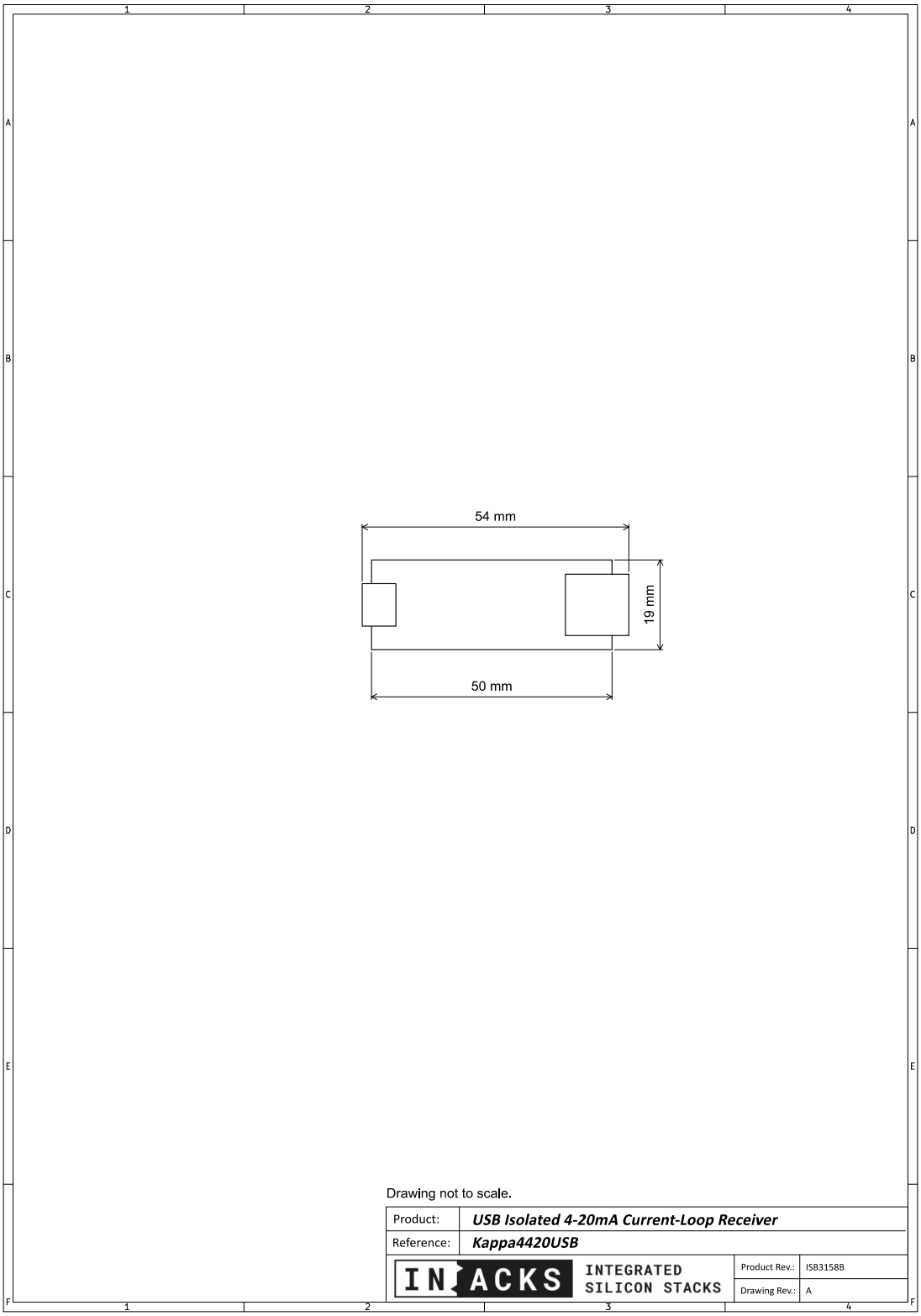
IS4420WIN is a free INACKS program for Windows 10 and 11 that shows the loop current live on the PC. It needs no configuration: it finds the Kappa4420USB by itself.

It shows the current in mA and in % with large digits and a needle gauge, shows the module temperature, and can save the readings to a CSV file. It only listens: it never sends anything to the board.

Download IS4420WIN at: github.com/inacks/IS4420WIN



9 Mechanical



10 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 no lines arrive, or unreadable characters arrive.	The COM port or the serial settings are wrong.	Check the COM port and the settings (115200 baud, 8N1, no flow control).
The first line is incomplete or has an extra character.	The COM port was opened in the middle of a line, or a stray byte arrived at power-up.	Keep only the lines with exactly 37 characters before CR+LF.
The lines show `00.00mA`.	No current flows: the loop is open, the wires are swapped, the loop power supply is off, or the transmitter is not working.	Check the wiring (section Pinout) and the loop power supply.
The lines show `EOVER`.	The loop current is above 21.50 mA. Many transmitters do this to signal a fault.	Check the transmitter and the loop.
The lines show `ERROR`.	The IS4420-M1 could not complete a measurement.	If it persists, disconnect the USB cable and connect it again.
The reading is right at low currents, but too low near 20 mA.	The loop power supply voltage is too low.	Use a loop power supply with more voltage, for example 24 V.
The reading is always a little lower than expected.	Part of the loop current returns through another path, for example a second ground connection.	Make sure that the loop current only returns through the GND terminal.

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 Current and Percentage 6
6 Communication Protocol 7
6.1 Serial Port Settings 7
6.2 Line Format 7
6.3 Fields 7
6.4 Error Reporting 8
7 Hardware 9
8 Software Examples 10
8.1 Python Example 10
8.2 Serial Terminal Example 12
8.3 IS4420WIN Software 13
9 Mechanical 14
10 Troubleshooting 15
Content 16
Appendix 17
Documentation Feedback 17
Sales Contact 17
Customization 17
Independence and Trademarks Notice 17
Disclaimer 18

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.

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 strives 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 makes 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 does not guarantee that the product 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 disclaims 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.