Fraunhofer
PORTUGAL

FRAUNHOFER CENTER FOR ASSISTIVE INFORMATION AND COMMUNICATION SOLUTIONS - AICOS

SENSOR IOT

Interfaces available on Kallisto

Porto, March 1, 2023

Revision History

Revision Number	Date	Comments
0.1	24/01/23	First release for approval
0.2	01/03/23	Added information on data stream (via USB)

Contents

1 Summary

This document provides to developers information on the communication interfaces available on Kallisto by Sensor IoT.

2 Introduction

TO DO

3 USB

Description

Sensor IoT USB protocol is implemented over USB Abstract Control Model behaving as a typical serial COM or UART communication. The protocol is implemented as a request-response transaction, with exception for data stream, and assumes one party is the master and the other is the slave. The later typically being the Kallisto with Sensor IoT firmware flashed.

As in any serial COM or UART communication, there are two communication channels: TX (Host Device → Kallisto) and RX (Kallisto → Host Device). Requests are handled by the first. Responses and streams of data by the later.

In both directions, the communication between parties is based on packets. These packets are described in detail from section 3.1 to section 3.12. And are composed of an identifier and payload. The identifier corresponds to the first byte of the packet and identifies the type of packet. Identifies are unintelligible between channels (TX & RX). Table 1 enumerates all identifiers used on packets in channel TX. Table 2 enumerates all identifiers used on packets in channel RX. The packet payload corresponds to the remaining bytes and its size is dependent on the type of packet and its parameters.

Streaming data doesn't comply with this protocol. And each packet corresponds to one measurement. All packets are encoded in Concise Binary Object Representation (CBOR) and must be decoded prior to any processing. The first field in each packet corresponds to the Sensor ID (see Annex A) of the sensor who captured the measurement. For more information on data streams, see section 3.13.

Table 1: List of packet identifiers (Host → Kallisto)

Packet ID	Description
0x01	SET Storage Request
0x02	GET Storage Request
0x03	SET Stream Request
0x04	SET Internet Connection Request
0x05	SET Sensor Request
0x06	SET Erase Request
0x07	SET RTC Request
0x08	GET RTC Request
0x09	GET Battery Request
0x0A	GET Storage List Request
0x0B	GET Status Request
0x0C	SET Calibration Request

Table 2: List of packet identifiers (Kallisto → Host)

Packet ID	Description
0x02	SET Storage Response
0x09	SET Stream Response
0x0F	SET Internet Connection Response
0x12	SET Sensor Response
0x15	SET Erase Response
0x18	SET RTC Response
0x1E	GET RTC Response
0x21	GET Storage List Response
0x24	GET Status Response
0x27	SET Calibration Response

3.1 SET Storage

Description

Enables and disables data storage in Kallisto. Each sensor is enabled separately. ***Sensor ID*** and ***Filename*** must be provided.

Request

	1 Byte	1 Byte	1Byte	Filename Length Bytes
0x01	Sensor ID	Enable/Disable	Filename Length	Filename

Sensor ID:

The values for ***Sensor ID*** are available on the Annex A.

Enable/Disable:

Code	Description
0x00	Disable
0x01	Enable

Filename Length:

Number of bytes in the Filename field.

Filename:

Name of the file in which samples from sensor with ***Sensor ID*** will be stored. The name must be encoded in ASCII.

Response

	1 Byte	
0x02	Status	0x0a

Status:

- 0, on success
- Otherwise, it failed

3.2 GET Storage

Description

Requests Kallisto to send data in the file with the provided ***Filename***.

Request

	1Byte	Filename Length Bytes
0x02	Filename Length	Filename

Filename Length:

Number of bytes in the Filename field.

Filename:

Name of the file in which samples from sensor with ***Sensor ID*** will be stored. The name must be encoded in ASCII.

Response

Binary data stored on the requested file.

3.3 SET Stream

Description

Requests Kallisto to continuosly send new sensor data. The ***Sensor ID*** of the sensor must be provided.

Request

	1 Byte	1 Byte
0x03	Sensor ID	Enable/Disable

Sensor ID:

The values for ***Sensor ID*** are available on the Annex A.

Enable/Disable:

Code	Description
0x00	Disable
0x01	Enable

Response

	1 Byte	
0x09	Status	0x0a

Status:

- 0, on success
- Otherwise, it failed

3.4 SET Internet Connection

Description

Configures connection between Kallisto and the internet or MQTT Broker.

Request

	1 Byte	1 Byte	...
0x04	Connection Type	Enable/Disable	Connection Parameters

Connection Type:

Code	Description
0x00	WiFi

Enable/Disable:

Code	Description
0x00	Disable
0x01	Enable

Connection Parameters:

- WiFi

1 Byte	...
Mode	Mode Parameters

  Mode:

Code	Description
0x00	Station Mode
0x01	Access Point ***(Not implemented)***
0x02	Disconnect
0x03	MQTT Connect
0x04	MQTT Disconnect

Mode Parameters:

- Station

1 Byte	SSID Length Bytes	1 Byte	...
SSID Length	SSID	Security Type	Security Parameters

SSID Length:

Number of bytes in the SSID field.

SSID:

SSID of the WiFi Network. The SSID must be encoded in ASCII.

Security Type:

Code	Description
0x00	Open
0x01	WEP
0x02	WPA PSK
0x03	***Reserved***
0x04	802.1x (MS-CHAP v2)
0x05	802.1x (TLS)

Security Parameters:

* Open
* WEP/WPA

1 Byte	KPL Bytes
KPL	Key/Password

KPL - Key/Password Length:

Number of bytes in the Key/Password field.

Key/Password:

Security Key of the WEP Network. Or the Password of the WPA. Both Security Key and Password must be encoded in ASCII.

* 802.1X (MSCHAPv2)

1 Byte	DL Bytes	1 Byte	UL Bytes	1 Byte	PL Bytes
DL	Domain	UL	Username	PL	Password

DL - Domain Length:

Number of bytes in the Domain field.

Domain:

MSCHAPv2 authentication server address. The field must be enconded in ASCII.

UL - Username Length:

Number of bytes in the Username field.

Username:

Username for MSCHAPv2 authentication. The field must be encoded in ASCII.

PL - Password Length:

Number of bytes in the Password field.

Password:

Password for MSCHAPv2 authentication. The field must be enconded in ASCII.

* 802.1X (TLS)

1 Byte	DL Bytes	1 Byte	UL Bytes	1 Byte	ML Bytes	1 Byte	EL Bytes	1 Byte	CE Bytes
DL	D	UL	U	ML	M	EL	E	CE	C

DL - Domain Length:

Number of bytes in the Domain field.

D - Domain:

TLS authentication server address. The field must be enconded in ASCII.

UL - Username Length:

Number of bytes in the Username field.

U - Username:

Username for TLS authentication. The field must be encoded in ASCII.

ML - Modulus Length:

Number of bytes in the Modulus field.

M - Modulus:

Modulus n in the TLS authentication key. The field must be encoded in ASCII.

EL - Exponent Length:

Number of bytes in the Exponent field.

E - Exponent:

Exponent in the TLS authentication key. The field must be encoded in ASCII.

CL - Certificate Length:

Number of bytes in the Certificate field.

C - Certificate:

TLS Certificate. The field must be encoded in ASCII.

- Disconnect
- MQTT Connect

1 Byte	AL Bytes	1 Byte	1 Byte	CIL Bytes	1 Byte	UL Bytes	1 Byte	PL Bytes	1 Byte
AL	Address	Port	CIDL	Client ID	UL	Username	PL	Password	Will Flag

1 Byte	1 Byte	1 Byte	...
MQTT Version	Keep Alive Interval	Clean Session	Extended Params

AL - Address Length:

Number of bytes in the Address field.

Address:

MQTT broker address. The field must be enconded in ASCII.

Port:

Port in which MQTT broker is running.

CIDL - Client ID Length:

Number of Bytes in the Address field.

Client ID:

MQTT Client ID. The field must be encoded in ASCII.

UL - Username Length:

Number of bytes in the Username field.

Username:

Username for MQQT broker authentication. The field must be encoded in ASCII.

PL - Password Length:

Password for MQTT broker authentication. The field must be encoded in ASCII.

Will Flag:

Code	Description
0x00	Disable
0x01	Enable

MQTT Version:

Code	Version
0x03	3.1
0x04	3.1.1

Keep Alive Interval:

Maximum time in seconds allowed to elapse between MQTT protocol packets sent by the client.

Clean Session:

Code	Description
0x00	Disable
0x01	Enable

Extended Parameters:

* Will Flag Enabled

1 Byte	1 Byte	1 Byte	ML Bytes	1 Byte	1 Byte
Topic Length	Topic Name	ML	Message	Retained	QoS

Topic Length:

Number of bytes in the Topic Name field.

Topic Name:

Name of topic to publish Will Message. The field must be enconded in ASCII.

ML - Message Length:

Number of bytes in the Message field.

Message:

The field must be enconded in ASCII.

Retained:

Code	Description
0x00	Disable
0x01	Enable

QoS - Quality of Service:

Quality of Service level for the Will Message.

– MQTT Disconnect

Response

1 Byte		
0x0F	Status	0x0a

Status:

- 0, on success
- Otherwise, it failed

3.5 SET Sensor

Description

Requests Kallisto to enable/disable a sensor with the provided ***Sensor ID*** and ***Sampling period***.

Request

	1 Byte	1 Byte	4 Bytes
0x05	Sensor ID	Enable/Disable	Sampling period

Sensor ID:

The values for ***Sensor ID*** are available on the Annex A.

Enable/Disable:

Code	Description
0x00	Disable
0x01	Enable

Sampling period:

The sampling period units change according to the sensor. For more information on the units used for each sensor, please read annex B. The field must be encoded in ***uint32***.

Response

	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
0x12	Acc. Status	Mag. Status	Gyro. Status	Temp. Status	Press. Status	Hum. Status	IAQ Status

1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
bVOC Status	eCO2 Status	Light Status	Noise Status	Light Status	0x0a

Acc. Status - Accelerometer Status:

- 0, on success
- Otherwise, it failed

Mag. Status - Magnetometer Status:

- 0, on success
- Otherwise, it failed

Gyro. Status - Gyroscope Status:

- 0, on success
- Otherwise, it failed

Temp. Status - Temperature Status:

- 0, on success
- Otherwise, it failed

Press. Status - Pressure Status:

- 0, on success
- Otherwise, it failed

Hum. Status - Humidity Status:

- 0, on success
- Otherwise, it failed

IAQ Status:

- 0, on success
- Otherwise, it failed

bVOC Status:

- 0, on success
- Otherwise, it failed

eCO2 Status:

- 0, on success
- Otherwise, it failed

Light Status:

- 0, on success
- Otherwise, it failed

Noise Status:

- 0, on success
- Otherwise, it failed

Light Status: **????????**

3.6 SET Erase

Description

Requests Kallisto to erase file from its default storage method.

Request

	1 Byte	1 Byte	PL Bytes
0x06	Sensor ID	PL	Path

Sensor ID:

The values for ***Sensor ID*** are available on the Annex A.

PL - Path Length:

Number of bytes in the Path field.

Path:

Path of the file to be erased. The field must be encoded in ASCII.

Response

	1 Byte	
0x15	Status	0x0a

Status:

- 0, on success
- Otherwise, it failed

3.7 SET RTC

Description

Requests Kallisto to set the current time value of the Real Time Clock.

Request

	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
0x07	Month	Day	Year	Weekday	Hour	Minutes	Seconds	Centiseconds

Month:

January to December (0x01 - 0x0c);

Day:

1 to 31 (0x01 - 0x1F);

Year:

0 to 99 (0x00 - 0x63)

Weekday:

Sunday to Saturday (0x00 - 0x06)

Hour:

0 to 23 (0x00 - 0x17)

Minutes:

0 to 59 (0x00 - 0x3B)

Seconds:

0 to 59 (0x00 - 0x3B)

Centiseconds:

0 to 99 (0x00 - 0x63)

Response

	1 Byte	
0x18	Status	0x0a

Status:

- 0, on success
- Otherwise, it failed

3.8 GET RTC

Description

Requests Kallisto to provide the current Real Time Clock value.

Request

0x08	0x01

Response

- OK

	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	
0x1B	Month	Day	Year	Weekday	Hour	Minutes	Seconds	Centiseconds	0x0a

NOTE: Information on each field is available on section 3.7.

3.9 GET Battery

Description

Requests Kallisto to provide the current Battery state of charge value.

Request

0x09	0x01

Response

	1 Byte	1 Byte	
0x1E	SoC	Status	0x0a

SoC - State of Charge:

Level of charge of a battery relative to its capacity (in %).

Charging Status:

- 0, when not charging
- 1, when charging

3.10 GET Storage List

Description

Requests Kallisto to provide list of files in the root directory of the default storage method.

Request

0x0a	0x01

Response

	1 Byte	2 Byte	SLL Bytes	
0x21	Number of Files	SLL	Storage List	0x0a

Number of files:

Number of file detected in the default storage method root directory.

SLL - Storage List Length:

Number of bytes in the ***Storage List*** field.

Storage List:

List of files detected in the default storage method root directory.

3.11 GET Status

Description

Requests Kallisto to provide the current status of each sensor. Including: if the sensor is enabled, if the sensor data is being stored, if the sensor data is being streamed and the current sampling period.

Request

0x0b	0x01

Response

7 Bytes[1]		
0x24	Sensor Status	0x0a

Sensor Status:

1 Byte	1 Byte	1 Byte	4 Bytes
Storage	Stream	Sensor	Frequency

Storage:

Code	Description
0x00	Disable
0x01	Enable

Stream:

Code	Description
0x00	Disable
0x01	Enable

[1]For each sensor in the following order: Accelerometer, Gyroscope, Magnetometer, Temperature, Pressure, Humidity, eCO2, TVOC, Light, bVOC, IAQ, Noise and Microphone.

Sensor:

Code	Description
0x00	Disable
0x01	Enable

Sampling period:

The sampling period units change according to the sensor. For more information on the units used, read annex B. The field must be encoded in ***uint32***.

3.12 SET Calibration

Request

1 Byte		
0x0c	Sensor ID	0x01

Sensor ID:

The values for ***Sensor ID*** are available on the Annex A.

Response

	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
0x27	Acc. Status	Mag. Status	Gyro. Status	Temp. Status	Press. Status	Hum. Status	IAQ Status

1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
bVOC Status	eCO2 Status	Light Status	Noise Status	Light Status	0x0a

Acc. Status - Accelerometer Status:

- 0, on success
- Otherwise, it failed

Mag. Status - Magnetometer Status:

- 0, on success
- Otherwise, it failed

Gyro. Status - Gyroscope Status:

- 0, on success
- Otherwise, it failed

Temp. Status - Temperature Status:

- 0, on success
- Otherwise, it failed

Press. Status - Pressure Status:

- 0, on success
- Otherwise, it failed

Hum. Status - Humidity Status:

- 0, on success
- Otherwise, it failed

IAQ Status:

- 0, on success
- Otherwise, it failed

bVOC Status:

- 0, on success
- Otherwise, it failed

eCO2 Status:

- 0, on success
- Otherwise, it failed

Light Status:

- 0, on success
- Otherwise, it failed

Noise Status:

- 0, on success
- Otherwise, it failed

Light Status: **????????**

3.13 Data Stream

Description

Data streaming packets are encoded in CBOR. Each packet corresponds to a single measurement and composes of multiple fields. In CBOR, field sizes are not fixed and are handled by the decoder. To extract information from a packet using a CBOR decoder, it is only required to know the order of the fields. The first two fields in each measurement packet correspond to the Sensor ID (see Annex A) and the timestamp of the measurement. All other fields are sensor dependent and are described in the following sections:

Accelerometer

Sensor ID	Timestamp (μs)	X	Y	Z

Gyroscope

Sensor ID	Timestamp (μs)	X	Y	Z

Magnetometer

Sensor ID	Timestamp (μs)	X	Y	Z

Temperature

Sensor ID	Timestamp (μs)	Temperature

Pressure

Sensor ID	Timestamp (μs)	Pressure

Humidity

Sensor ID	Timestamp (μs)	Humidity

eCO2

Sensor ID	Timestamp (μs)	eCO2

tVOC

Sensor ID	Timestamp (μs)	tVOC

Light

Sensor ID	Timestamp (μs)	Light Intensity

bVOC

Sensor ID	Timestamp (μs)	bVOC

IAQ

Sensor ID	Timestamp (μs)	IAQ

Noise

Sensor ID	Timestamp (μs)	Noise (dB)

Microphone

Sensor ID	Timestamp (μs)	PCM	Length

A List of USB Sensor IDs

ID	Sensor
0x01	Accelerometer
0x02	Gyroscope
0x03	Magnetometer
0x04	Temperature
0x05	Pressure
0x06	Humidity
0x07	eCO2
0x08	TVOC
0x09	Light
0x0a	bVOC
0x0b	IAQ
0x0c	Noise
0x0d	Microphone

B List of sensors' frequency units

Sensor	miliseconds	microseconds
Accelerometer	-	x
Gyroscope	-	x
Magnetometer	-	x
Temperature	x	-
Pressure	x	-
Humidity	x	-
eco2	x	-
TVOC	x	-
Light	x	-
BVOC	x	-
IAQ	x	-
Microphone	x	-
Noise	x	-