

# KAMAMI

## KAmoD LTE CAT1-GNSS with A7670E-FASE module



# KAMAMI



Rev. 20260404130130

Źródło: [https://wiki.kamamilabs.com/index.php?title=KAmod\\_LTE\\_CAT1-GNSS\\_with\\_A7670E-FASE\\_module](https://wiki.kamamilabs.com/index.php?title=KAmod_LTE_CAT1-GNSS_with_A7670E-FASE_module)

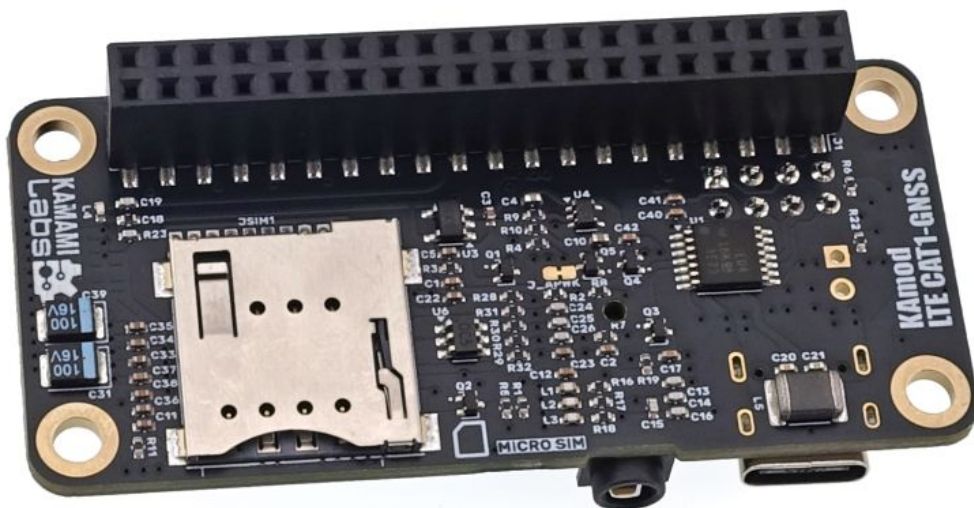
## Table of contents

Description .....	1
Basic Parameters .....	2
Standard Equipment .....	2
Block Diagram .....	3
Button Functions .....	4
Status Indicator Functions .....	5
Connection with Raspberry Pi Boards .....	6
USB Interface .....	8
SIM Card .....	9
Power Supply .....	9
Service Mode .....	10
Audio Connector - SPK&MIC .....	11
LTE/GSM Antenna .....	12
GNSS Antenna .....	12
Control via AT Commands .....	13
Basic Parameters .....	13
Voice Connection .....	14
Sending SMS Messages .....	15
Reading SMS Messages .....	16
GNSS Data Reading .....	17
Selected HTTP/HTTPS Functions .....	18
Selected SSL Secure Network Protocol Functions .....	19
Selected MQTT Functions .....	20
Running MQTT with HiveMQ Server .....	22
Links .....	26

## Description

### KAmoD LTE CAT1-GNSS - GSM/GPRS/GNSS HAT with A7670E-FASE module for Raspberry Pi

The SimCom A7670E/A7672E module is a GSM/LTE CAT-1 modem that provides wireless communication in LTE-FDD, GPRS/EDGE, and GSM standards. Additionally, it features an integrated GNSS receiver and supports numerous network functions, including TCP/IP, FTP/FTPS, HTTP/HTTPS, SSL, and MQTT. Controlling the modem's operation is very simple - it uses AT commands sent via the UART serial port. The KAmoD LTE CAT1-GNSS board contains all the components necessary for the modem to function and allows for easy connection with Raspberry Pi series computers and other similar devices.

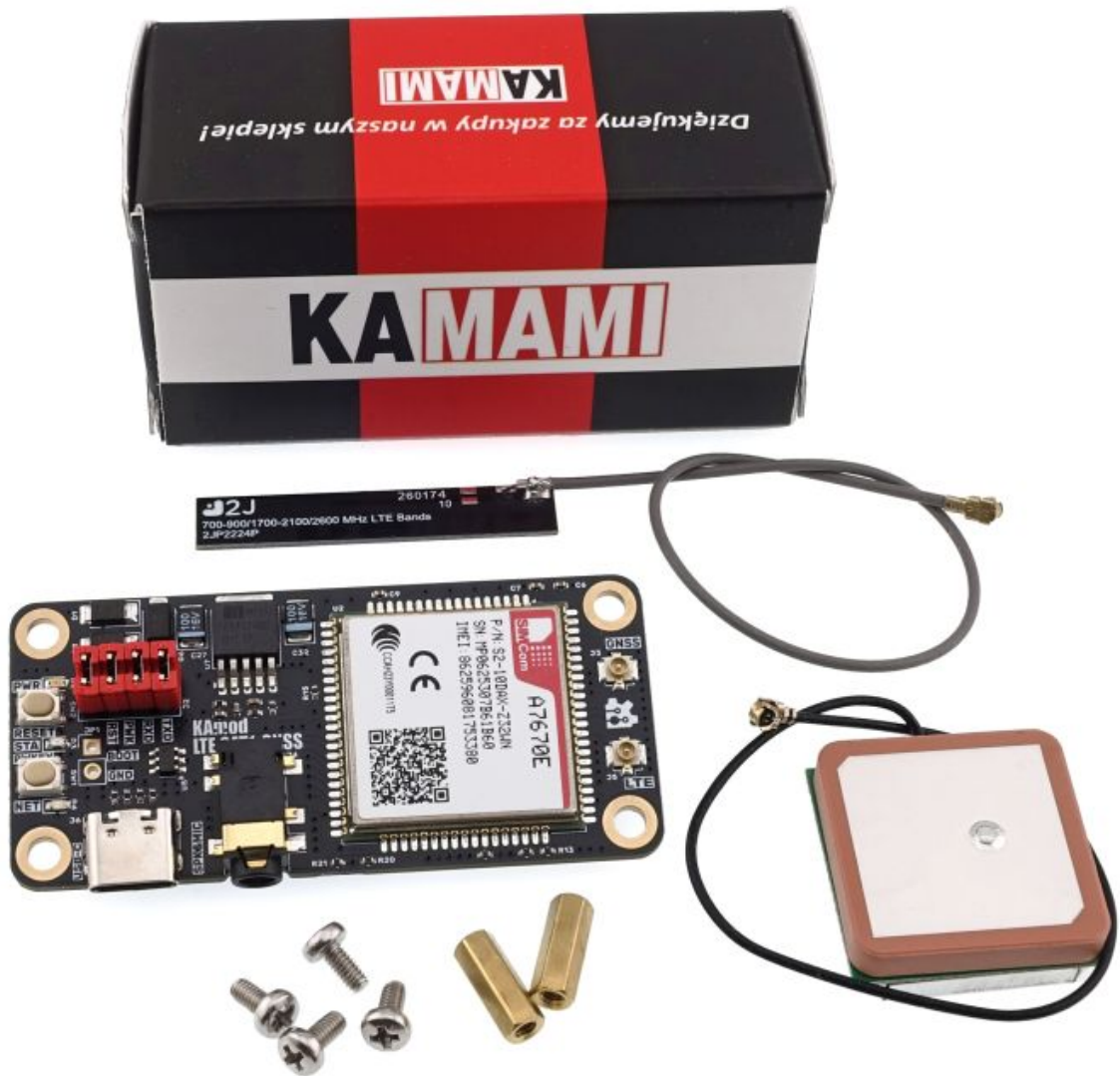


## Basic Parameters

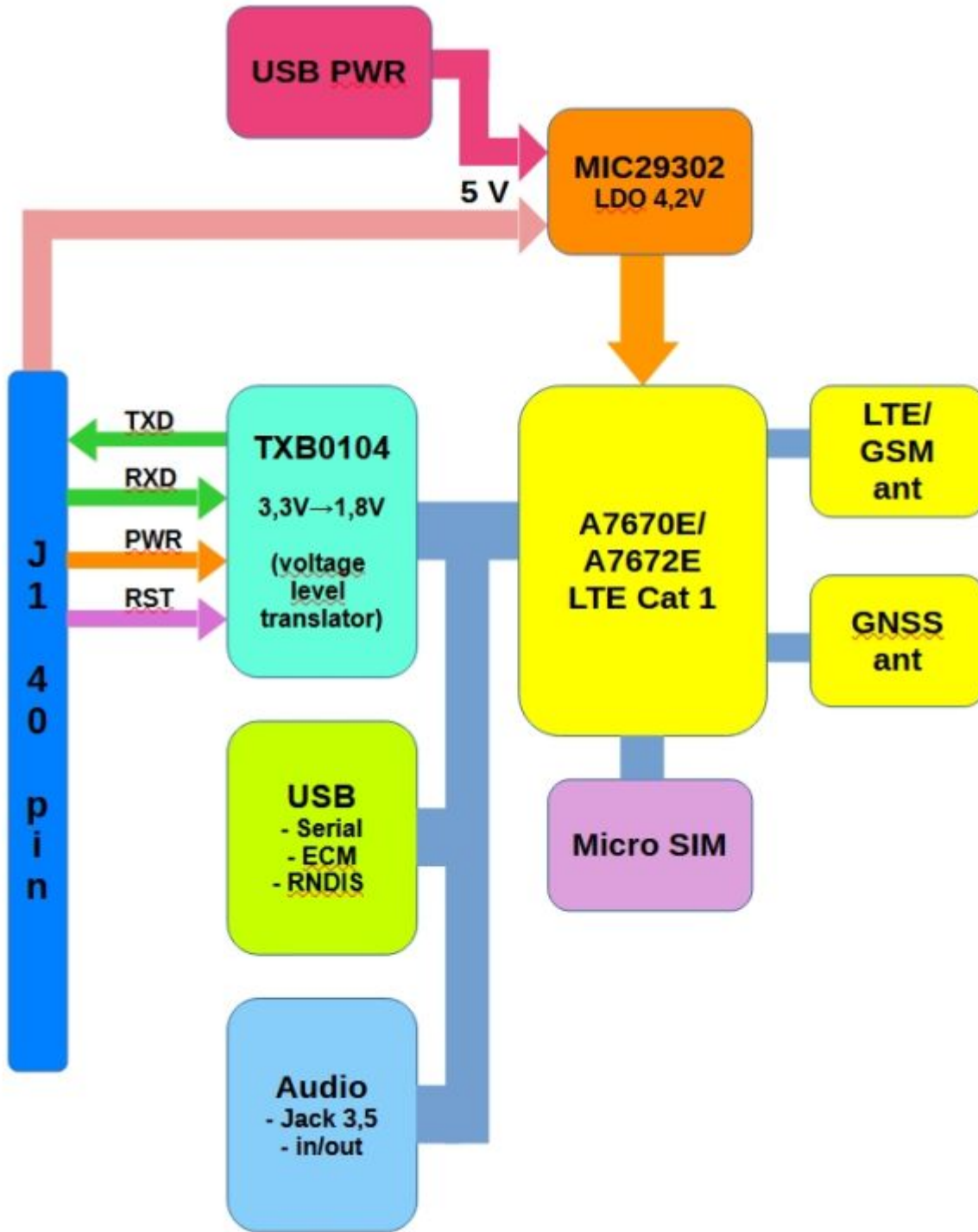
- LTE CAT-1 modem based on the SimCom A7670E or A7672E module
- LTE (4G) wireless communication bands: LTE-FDD B1/B3/B5/B7/B8/B20
- GSM (2G) wireless communication bands: 900/1800 MHz
- LTE data transfer speed: max 10 Mbps (DL); max 5 Mbps (UL)
- EDGE data transfer speed: max 236.8 kbps; GPRS: max 85.6 kbps
- Supported functions and protocols: TCP/IP, IPV4, IPV6, Multi-PDP, FTP/FTPS, HTTP/HTTPS, DNS, SSL, TLS, MQTT
- Supported SIM card: Micro SIM 1.8/3.0 V
- UART interface (3.3 V) for AT command control
- USB interface for PC connection (creates a serial port for AT commands and an ECM - Ethernet Control Model or RNDIS - Remote Network Driver Interface device for internet connectivity)
- 3.5 mm Jack audio in/out connector
- Integrated GNSS (GPS, GLONASS, BeiDou)
- 5 V / 2 A power supply via GPIO/USB-C
- Compatible with Raspberry Pi/Zero boards, includes a 40-pin GPIO connector

## Standard Equipment

Code	Description
<b>KAmoD LTE CAT1-GNSS</b>	Assembled and tested module
<b>GSM Antenna</b>	GSM antenna with U.FL connector
<b>GNSS Antenna</b>	Active GPS antenna with U.FL connector
<b>Mounting Kit</b>	Set of screws and spacers for attaching the HAT to the Raspberry Pi board



## Block Diagram



The electrical schematic is available here: [KAmoD\\_LTE\\_CAT1\\_sch](#)

## Button Functions

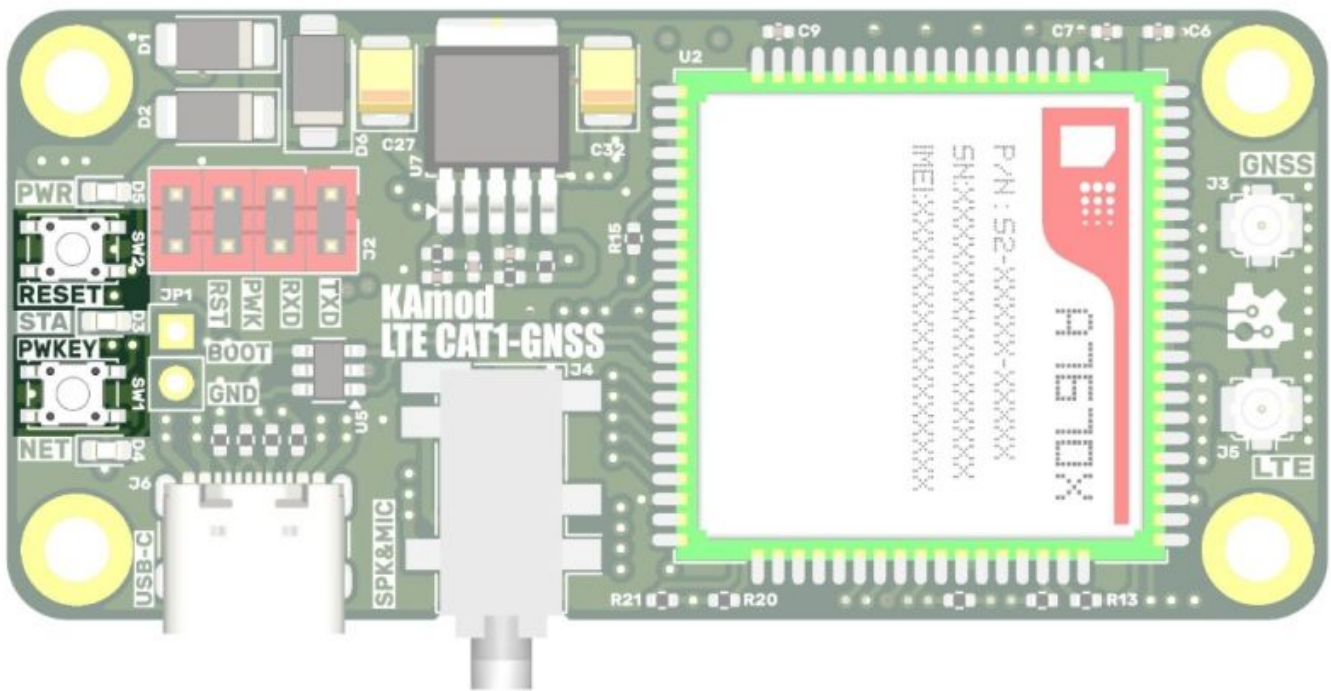
Button functions are described in the table below.

Button	Function
--------	----------

<b>PWKEY</b> (SW1)	Pressing the button for at least 50 ms turns on the A7670E/A7672E module (transition from power-off to normal operation). Pressing the button for at least 2.5 s turns off the module (transition from normal operation to Power OFF state).
<b>RESET</b> (SW2)	Pressing the button forces an active state on the RESET input of the A7670E/A7672E module and causes it to reset.

The buttons perform their functions in parallel with the PWK and RST signals from the Raspberry Pi connector. The manufacturer of the A7670E/A7672E module recommends not setting the PWK and RST signals active simultaneously.

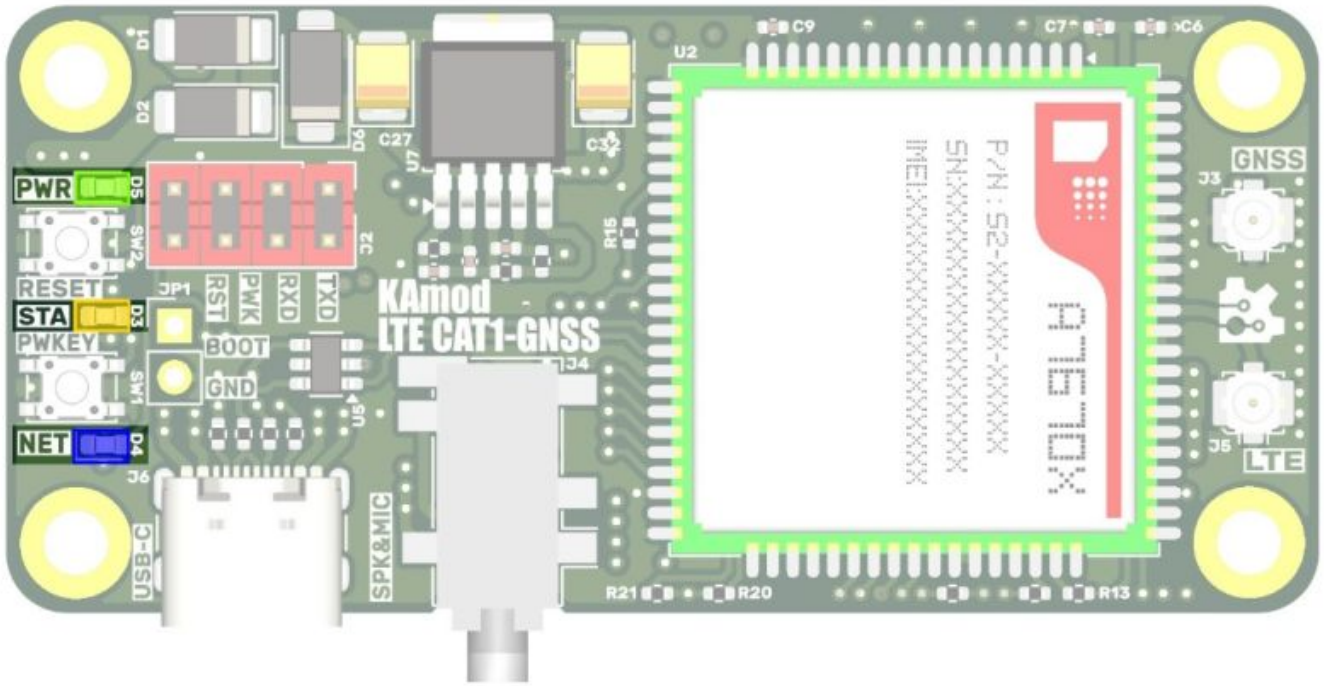
The KAmoD LTE CAT1-GNSS module includes an automatic power-on circuit for the A7670E/A7672E module after the power supply is connected. This circuit generates a short pulse on the PWK line immediately after power-up. To deactivate this circuit, cut the jumper marked J\_APWK located on the bottom of the board.



## Status Indicator Functions

The meaning of the LED indicators is described in the table, and their arrangement is shown in the figure.

Indicator	Function
<b>PWR</b> (D5)	The LED indicates the presence of power supply voltage for the A7670E/A7672E module
<b>STA</b> (D3)	The LED indicates that the A7670E/A7672E module is in an active state
<b>NET</b> (D4)	Steady light indicates GSM/LTE network searching status Flashing LED signals a connection to the GSM/LTE network



## Connection with Raspberry Pi Boards

The KAmoD LTE CAT1-GNSS module is designed to connect with the Raspberry Pi family and other similar devices featuring a 40-pin GPIO goldpin connector compatible with Raspberry Pi. Power and necessary control signals, described in the table below, are provided through this connector.

Control signals are adapted to 3.3 V voltage. The default communication speed for the UART serial interface is set to 115200 bps.

A7670E/A7672E module control signal	Control signal on Raspberry Pi family board (J1)	Function
<b>UART TXD</b> (out)	RXD/GPIO15 (pin 10)	Serial data output
<b>UART RXD</b> (in)	TXD/GPIO14 (pin 8)	Serial data input
<b>RST - RESET</b> (in)	GPIO18 (pin 12)	RESET signal input, active High
<b>PWK - PWRKEY</b> (in)	GPIO04 (pin 7)	Module activation signal input, active High

Power Line	Function
<b>5 V Power</b> (in) (pins 2, 4)	5 V power input from the Raspberry Pi board
<b>GND</b> (in) (pins 6, 9, 14, 20, 25, 30, 34, 39)	Power ground (GND)

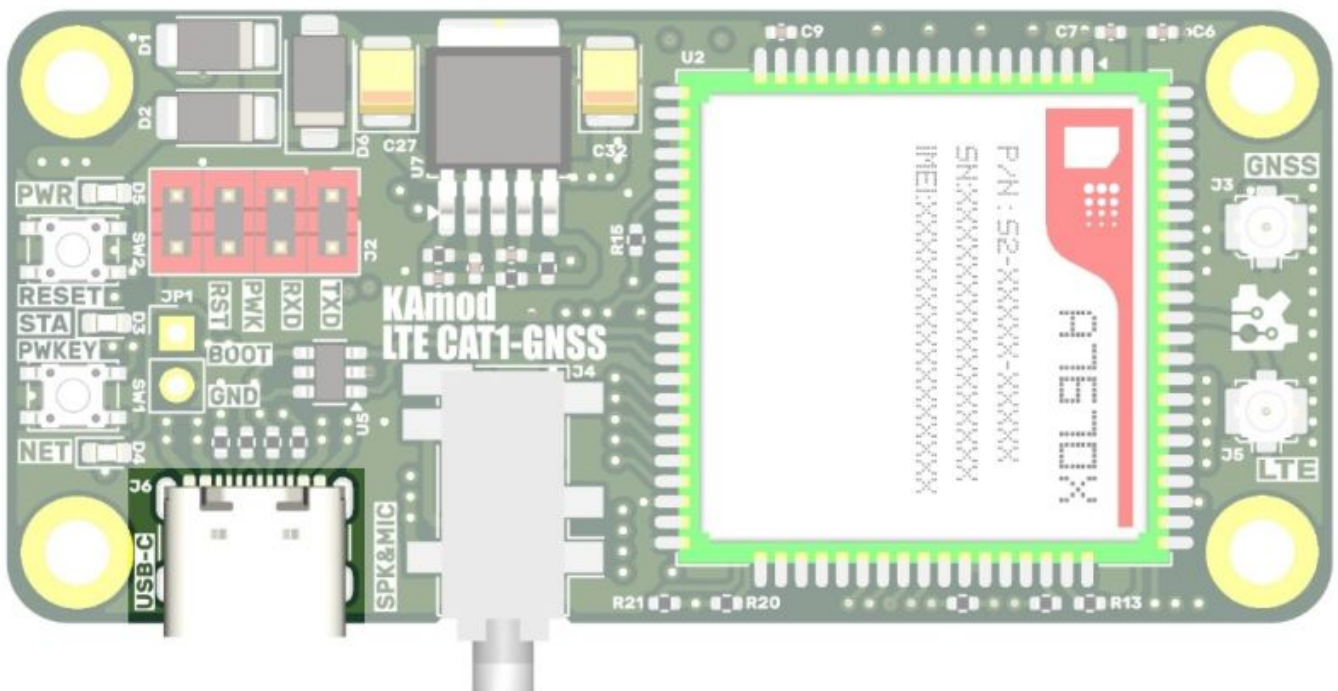


## USB Interface

The A7670E/A7672E module can operate as an ECM (Ethernet Control Model) or RNDIS (Remote Network Driver Interface Specification), which easily creates an internet connection for Windows-based devices. To achieve this functionality, the KAmoD LTE CAT1 board must be connected to a PC via the USB-C connector. After installing the drivers, several new devices will appear in the system. AT commands can be sent to the module via the "AT Port".

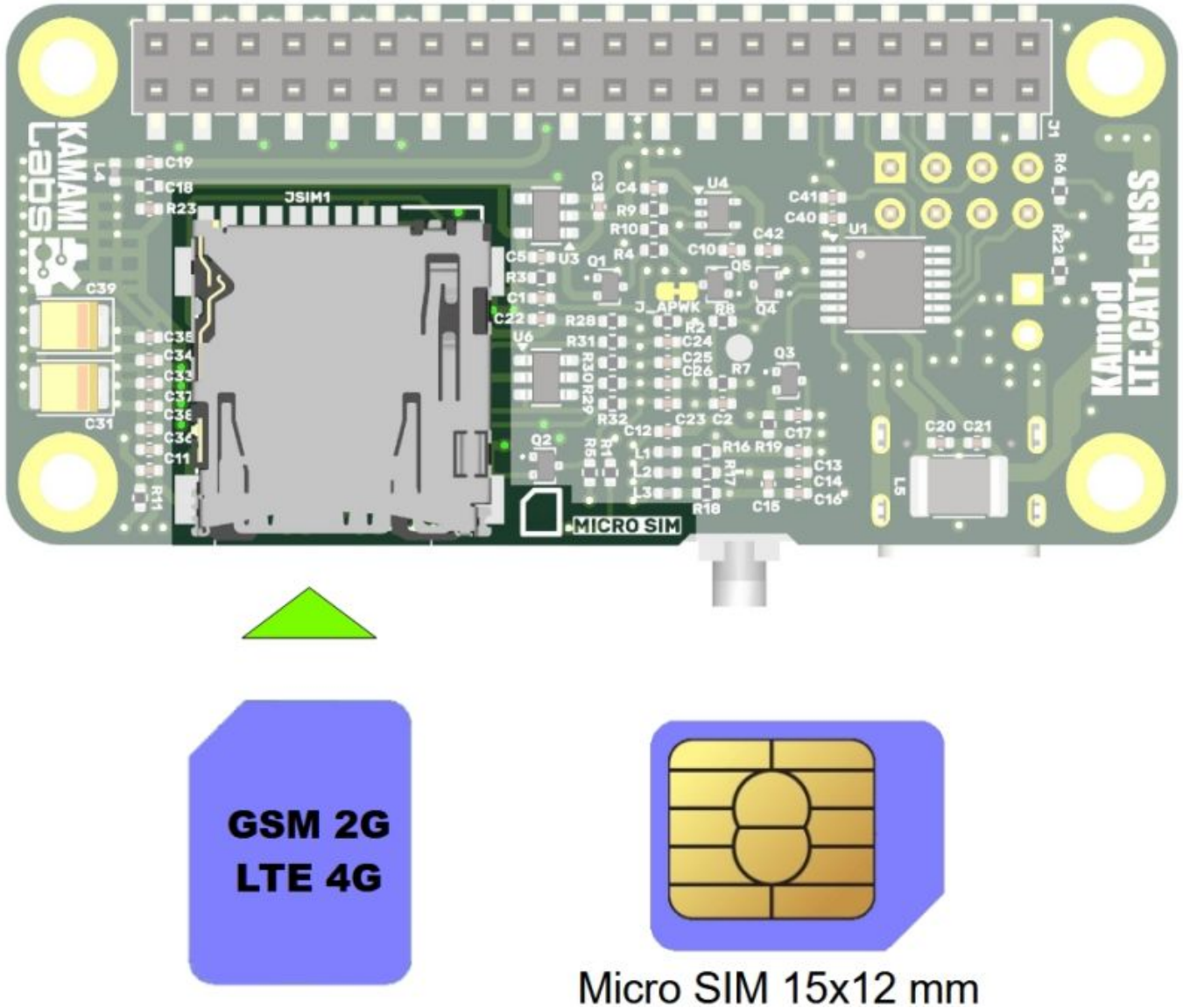
- ▼ Karty sieciowe
      - Realtek Gaming 2.5GbE Family Controller
      - Remote NDIS based Internet Sharing Device
    - > Klawiatury
    - > Kolejki wydruku
    - > Komputer
    - > Kontrolery dźwięku, wideo i gier
    - > Kontrolery IDE ATA/ATAPI
    - > Kontrolery magazynu
    - > Kontrolery uniwersalnej magistrali szeregowej
    - ▼ Modemy
      - SimTech HS-USB Modem 9011
    - > Monitory
    - > Mysz i inne urządzenia wskazujące
    - > Oprogramowanie układowe
    - ▼ Porty (COM i LPT)
      - Port drukarki (LPT1)
      - Port komunikacyjny (COM1)
      - SimTech HS-USB AT Port 9011 (COM19)
      - SimTech HS-USB Diagnostics 9011 (COM20)
      - SimTech HS-USB NMEA 9011 (COM21)

The instantaneous current consumption of the A7670E/A7672E module can exceed 2 A, so ensure that the USB port connected to the KAmoD LTE CAT1-GNSS board has sufficient power. Use only high-quality USB cables with a maximum length of 0.5 m. In case of USB communication issues, it is worth checking the operation with the LTE antenna disconnected and/or without a SIM card.



## SIM Card

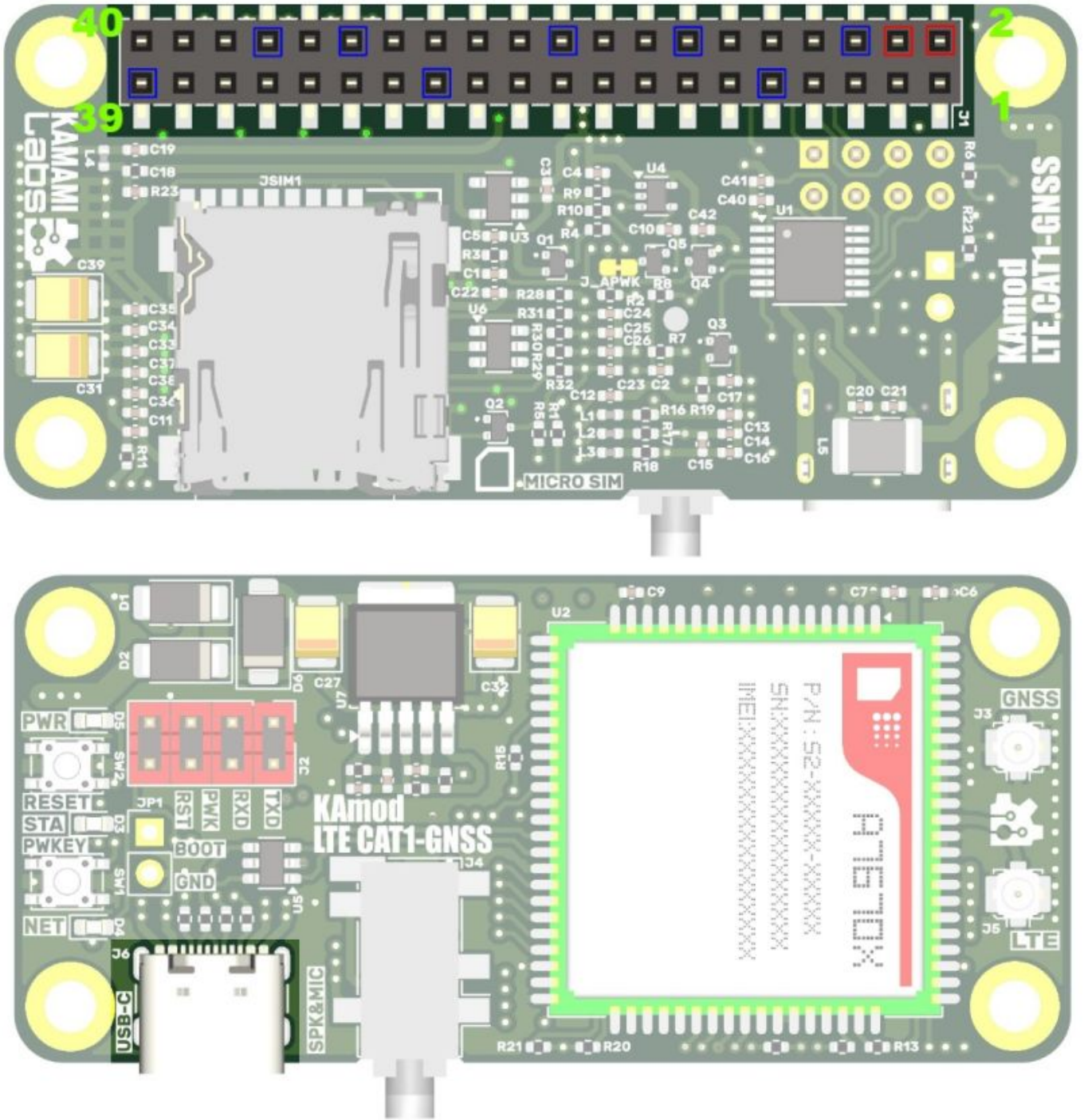
KAmoD LTE CAT1 is equipped with a Micro SIM card slot (15x12 mm) and operates at 1.8/3.0 V. The card should be placed in the slot and gently pressed until it clicks. To remove the card, press it gently and release - the edge of the card will pop out, allowing it to be pulled out completely. All operations with the SIM card should be performed with the power disconnected from the KAmoD LTE CAT1 board (and thus from the RPi computer). The clipped corner of the card should point towards the outside of the board, as shown in the figure and marked on the PCB.



## Power Supply

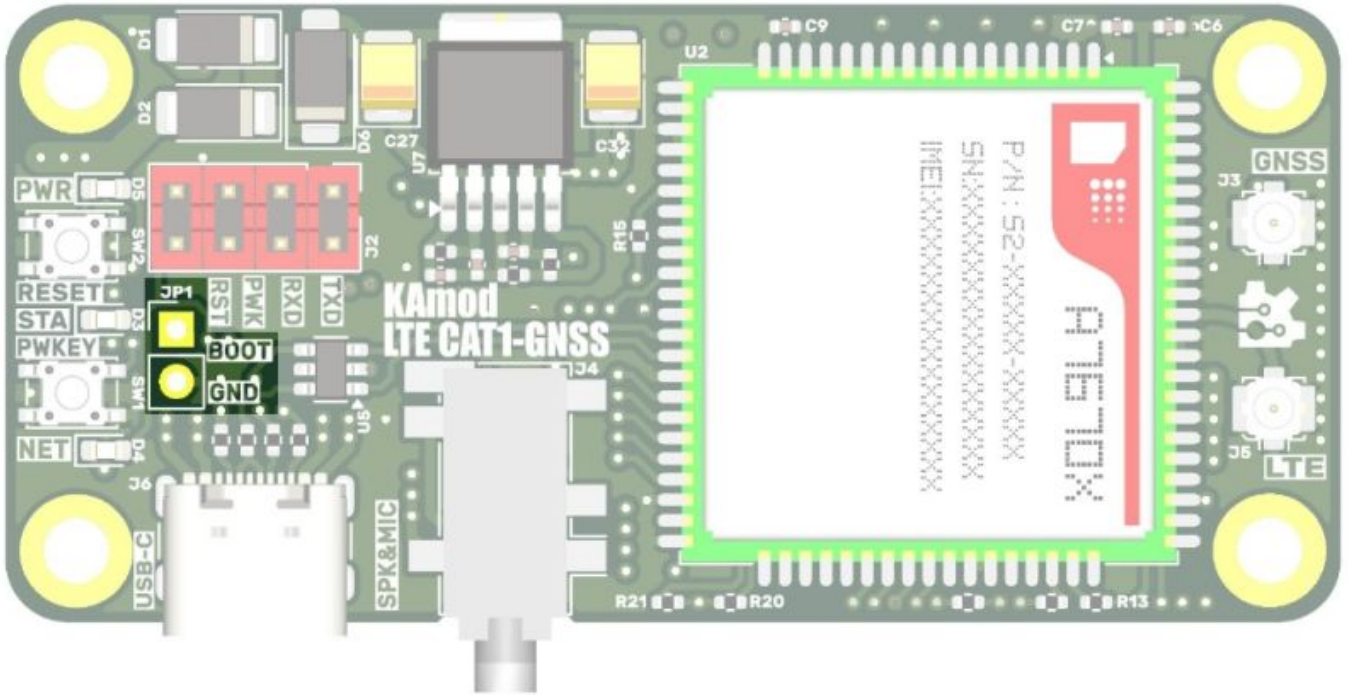
KAmoD LTE CAT1-GNSS requires a 5 V power supply with a minimum current capacity of 2 A. In the active state, when an LTE network connection is established but no tasks are being performed, the module consumes approximately 30 mA. However, during data transmission, instantaneous current consumption can exceed 2 A. Therefore, for the proper operation of the KAmoD LTE CAT1-GNSS module, a power supply with an instantaneous capacity of at least 2 A is required.

Power can be supplied via the 40-pin GPIO goldpin connector (J1), compatible with Raspberry Pi, and/or simultaneously via the USB connector.



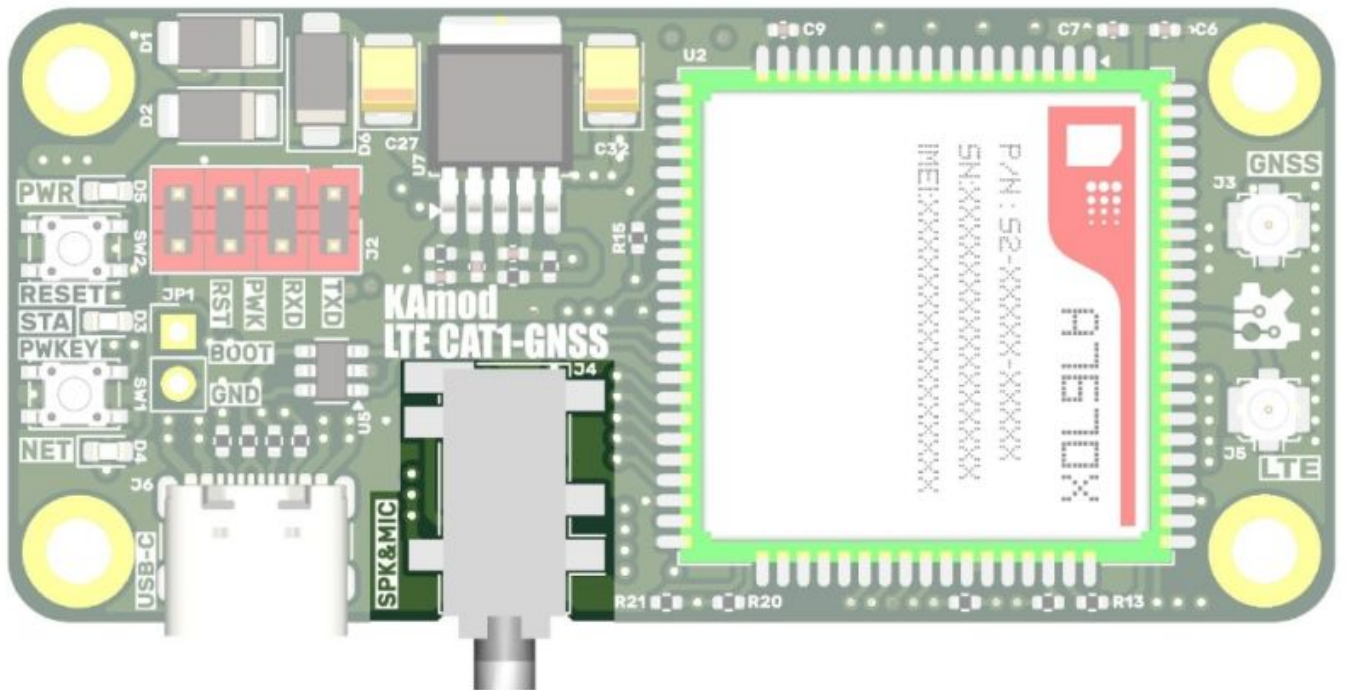
## Service Mode

The manufacturer of the A7670E/A7672E modules delivers components with the most optimized firmware and does not recommend updating it. The firmware version can be checked using the ATI command. However, the Kamod LTE CAT1-GNSS board features JP1 goldpins. Shorting them allows the module to start in a special service mode intended for firmware updates. During normal use, the JP1 pins should not be shorted. For more information regarding firmware updates, please visit the SimCom module manufacturer's website - [www.simcom.com](http://www.simcom.com).



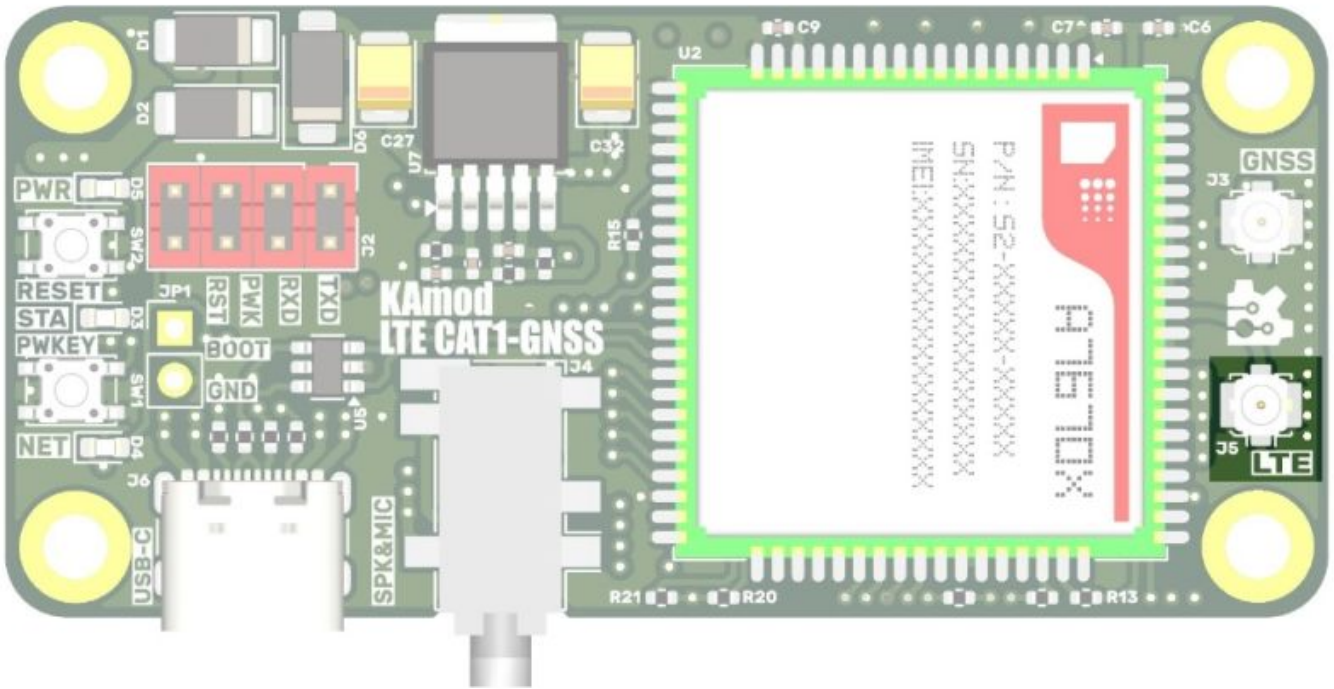
## Audio Connector - SPK&MIC

KAmoD LTE CAT1-GNSS allows for standard voice calls, provided the installed SIM card supports this feature. A 3.5 mm Jack headset (headphones and microphone) should be connected to the J4 connector – a standard headset that enables listening and transmitting sound between parties in a phone call.



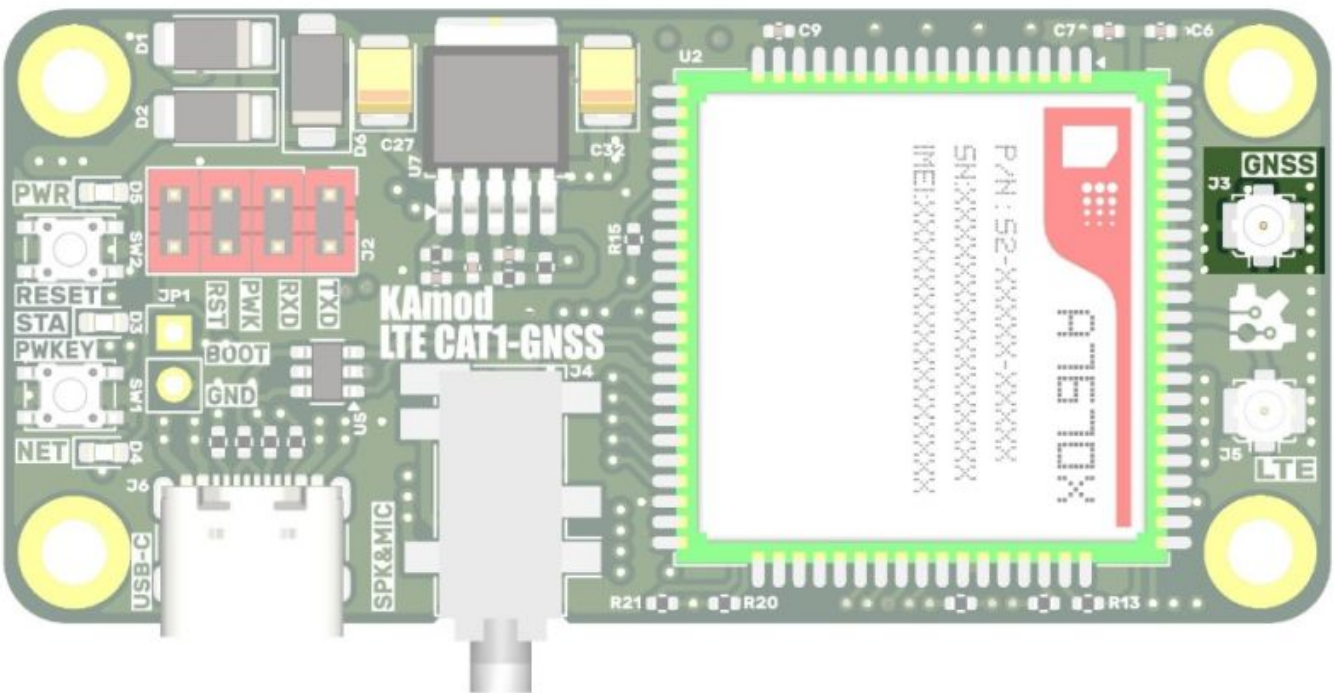
## LTE/GSM Antenna

KAmoD LTE CAT1-GNSS operates in GSM (2G) and LTE (4G) telephony standards. The module board features a U.FL (IPX) connector marked LTE, to which an antenna designed for GSM/LTE operation must be attached.



## GNSS Antenna

The module also features a U.FL (IPX) connector for the satellite positioning system. An active GNSS antenna should be connected to the connector marked GNSS.



## Control via AT Commands

A7670E/A7672E modules are controlled using so-called AT commands sent via the UART serial port. Every command starts with the characters "AT" and ends with a <CR> character (hex 0x0D), which corresponds to the ENTER key on a computer keyboard. Some parameters, such as names and phone numbers, start and end with a " (double quote - hex 0x22) character. Commands can be entered using any terminal program, such as Putty or Minicom.

A list of all AT commands with descriptions can be found here: <A76XX\_Series\_AT\_Command\_Manual\_V1.09.pdf> The following part of the description presents some AT commands for performing basic tasks and selected functions of the A7670E/A7672E modem.

### Basic Parameters

- **AT** - test command; if the module is working correctly, the following response will be sent:  
OK

- **ATI** - read basic information; a response similar to this will be sent:

```
Manufacturer: INCORPORATED
Model: A7670E-FASE
Revision: A7670M7_V1.11.1
IMEI: 863957078398663
+GCAP: +CGSM,+FCLASS,+DS
```

- **AT+CPIN?** - checks the PIN code status for the SIM card; if a PIN is not required, the following response will be sent:

```
+CPIN: READY
```

If a PIN is required, use the command **AT+CPIN=<sim\_card\_pin>**

- **AT+CSQ** - allows reading the signal strength level (network coverage); a response similar to this will be sent:

```
+CSQ: 23,99
```

The first parameter (23) is the signal level (RSSI), where:

0 = -113 dBm or less; 31 = -51 dBm or more.

The second parameter specifies the bit error rate, where:

0 = 0.01% or less; 7 = 8% or more; 99 = no information

- **AT+CREG?** - checks the GSM/LTE network registration status; a response similar to this will be sent:

```
+CREG: 0,1
```

The second parameter specifies the connection status; significant values are:

0 = not registered; 1 = registered, home network; 2 = searching for operator; 3 = registration denied

- **AT+COPS?** - reads the network operator name; returns a response similar to this:

```
+COPS: 0,0,"Orange",7
```

If a number appears instead of the operator's name, enter the command **AT+COPS=3,0** and then **AT+COPS?** again.

## Voice Connection

- **ATD<full\_phone\_number>;** - starts a voice call to the selected phone number. The number must be entered with a prefix (e.g., +48 for Poland). There must be a semicolon (;) at the end. The responses sent will contain information about the call progress.
- **ATA** - allows answering an incoming voice call. An incoming call will be signaled by several commands, including:  
+CLIP: "+48123456789", 145 - informs about the phone number the call is coming from.  
RING - indicates the ringing signal.
- **ATH** - hangs up the current voice call.
- **AT+COUGAIN=7** - increases the earpiece volume to level 7 (7 = max, 0 = min).
- **AT+CMICGAIN=7** - increases the microphone sensitivity to level 7 (7 = max, 0 = min).

## Sending SMS Messages

- **AT+CSCA?** - checks the set service center number. For operators available in Poland, these may be the following numbers:

Orange: +48602951111

Play: +48602295000

Plus: +48601000310

T-Mobile: +48602951111 (same as Orange)

This information should be confirmed with the SIM card operator. The number can be set using the command

**AT+CSCA="<full\_service\_center\_number>"**

- **AT+CMGF=1** - enables text mode, allowing the message content to be saved and read as plain text.
- **AT+CSCS="GSM"** - sets the character set.
- **AT+CMGS="+48123456789"** - sets the message recipient's number. After confirming with <CR> (Enter key), the > character will be sent, signaling to type the message content. After typing the content, confirm the operation with <1A> (corresponds to Ctrl + Z) or cancel with <1B> (ESC key). If the process is successful, the SMS will be sent, and a response similar to this will be returned in the terminal:

+CMGS: 15

Parameter 15 is the message index in the modem's memory.

## Reading SMS Messages

- **AT+CMGR=4** - reads the message at index 4 in the message memory. A response similar to this will be sent:

+CMGR: "REC UNREAD", "+48123456789", "", "25/03/23,23:51:43+4"

Test

OK

Which means: "REC UNREAD" - the message was not previously read; after reading, its status changes to "REC READ"

"+48123456789" - the sender's phone number

"25/03/23,23:51:43+4" - date and time the message was received

Test - message content

- **AT+CMGD=4** - deletes the message at index 4 in the message memory.

- **AT+CNMI=1,2,0,0,0** - after sending this command, new SMS messages will be automatically displayed - information similar to this will appear each time:

+CMT: "+48123456789", "", "25/08/13,10:20:18+8"

Test Test

## GNSS Data Reading

- **ATI** - first, it is worth checking if the A7670E/A7672E module has an integrated GNSS receiver. The ATI command displays essential information; if the module is described as **A7670E-FASE** or **A7672E-FASE**, it is the GNSS version. If the model ends with the letters "LASE", it does not include GNSS.

- **AT+CGNSSPWR=1** - activates the GNSS satellite positioning system module. A response similar to this will be sent:  
OK  
+CGNSSPWR: READY!

- **AT+CGNSSTST=1** - activates the GNSS data output.

- **AT+CGNSSPORTSWITCH=0,1** - switches the GNSS data output to the UART port. From this moment, a full GNSS data package in NMEA standard will be sent via the UART port every 1 second. The data will have a structure similar to this:

```
$GNRMC,071922.00,V,,,,,,,,180825,,,N,V*10
$GNGGA,071922.00,,,,,0,00,99.99,,,,,*77
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,1*33
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,3*31
$GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99,2*30
$GPGSV,3,1,12,08,43,196,26,27,18,170,21,32,50,087,22,01,45,288,,0*61
$GPGSV,3,2,12,02,79,288,,24,05,019,,03,,,19,33,21,223,,0*59
$GPGSV,3,3,12,37,30,179,,39,30,174,,40,22,139,,38,21,223,,0*61
$GAGSV,1,1,00,0*74
$GLGSV,1,1,03,,,,,28,,,,,28,,,,,29,0*71
```

The meaning of individual lines is described, for example, here: <https://gpsd.gitlab.io/gpsd/NMEA.html>

To deactivate sending GNSS data via UART, send the command: AT+CGNSSPORTSWITCH=0,1

- **AT+CGPSINFO** - reads basic GNSS data. A response similar to this will be sent:  
+CGPSINFO: 5224.16696,N,02054.71634,E,180825,072826.00,195.0,,

These are, in order: latitude, N/S, longitude, E/W, date, UTC time, altitude, speed [expressed in knots; multiply the value in knots by 1.852 to get speed in km/h], direction [expressed in degrees].

## Selected HTTP/HTTPS Functions

- **AT+HTTPIPINIT** - initializes the HTTP session.
- **AT+HTTTPARA="URL","https://www.example.org"** - sets HTTP parameters, in this case, the URL. Other available options include "CONTENT", "ACCEPT", or "USERDATA".
- **AT+HTTPACTION=0** - performs an HTTP/HTTPS request using the specified method; available options are: 0 = GET, 1 = POST, 2 = HEAD, 3 = DELETE, 4 = PUT.  
A response similar to this will be sent:  
+HTTPACTION: 0,200,1256  
Which means:  
"200" - server response, 200 = OK  
"1256" - data length to read (Content Length)
- **AT+HTTPHEAD** - reads the server response header information. A response similar to this will be sent:  
+HTTPHEAD: 44  
HTTP/1.1 200 OK  
Content-Type: text/html  
Content-Length: 1256  
Connection: keep-alive
- **AT+HTTPREAD=44,1000** - reads the entire content of the server response.  
The first parameter (here 44) is the index from which to start reading,  
the second parameter is the amount of data to read.
- **AT+HTTPDATA=18,10** - used for sending data using the POST method. The first parameter specifies the amount of data (from 1 to 153600 bytes), the second parameter specifies the time in which the data must be entered (from 10 to 65535 seconds).
- **AT+HTTPTERM** - terminates the HTTP session.

## Selected SSL Secure Network Protocol Functions

- **AT+CSSLCFG?** - displays the configuration of all 9 SSL contexts. The parameter group for each context means:

ssl version - SSL network protocol version,

auth mode - authentication mode,

ignore local time - certificate time check,

negotiate time - maximum time to perform authentication (default 300 s)

ca file - CA (Certificate Authority) certificate file name

client cert file - client CA certificate file name

client key file - client key file name

password file - password file name, required to read the client key file

enable SNI flag - Server Name Indication function - enables support for multiple SSL certificates on a single IP address and port

- **AT+CSSLCFG="sslverson",<ssl\_context>,4** - sets the SSL network protocol version.

The first parameter is the context number: 0...9

the second parameter means: 0 = SSL3.0; 1 = TLS1.0; 2 = TLS1.1; 3 = TLS1.2, 4 = all (0...3)

- **AT+CSSLCFG="authmode",<ssl\_context>,1** - specifies the authentication mode.

The first parameter is the context number: 0...9

the second parameter means:

0 = no authentication;

1 = server-side authentication (requires root CA certificate)

2 = server and client-side authentication (requires root CA certificate and client certificate and key)

3 = client-side authentication (client certificate and key)

- **AT+CSSLCFG="ignorelocaltime",<ssl\_context>,1** - certificate time check.

The first parameter is the context number: 0...9

the second parameter means:

0 = check certificate time

1 = ignore certificate time check

- **AT+CSSLCFG="cacert",<ssl\_context>,"filename.pem"** - specifies the root CA certificate file for the given SSL context.

The file must have a .pem or .der extension.

- **AT+CSSLCFG="enableSNI",<ssl\_context>,1** - Server Name Indication function.

The first parameter is the context number: 0...9

the second parameter means:

0 = SNI disabled

1 = SNI active

- **AT+CCERTDOWN="<name.pem>,<length>** - allows installing a CA certificate.

The first parameter is the certificate name (must have .pem or .der extension),

the second parameter means the amount of data to be sent - file length.

After sending the command, the ">" character will be displayed, indicating that the declared amount of data should be sent.

If successful, an "OK" response will be sent.

- **AT+CCERTLIST** - displays the list of installed certificates.

## Selected MQTT Functions

- **AT+CMQTTSTART** - activates the MQTT function.
- **AT+CMQTTSTOP** - deactivates the MQTT function.
- **AT+CMQTTACCQ=<client>,"KAmoD-LTE-CAT1",1** - MQTT client initialization.  
The first parameter: specifies the index to which we assign the client name (0 or 1),  
the second parameter: client name (KAmoD-LTE-CAT1),  
the third parameter: 0 - TCP connection; 1 - SSL/TLS connection.
- **AT+CMQTTACCQ?** - displays set clients.
- **AT+CMQTTREL=<client>** - removes set clients; the parameter specifies the client index. A response similar to this will be sent:  
`+CMQTTREL: 0,0`  
The first parameter is the client index: 0...1,  
the second parameter is the error code (0 = OK; 19 = client in use; 20 = client not set yet).
- **AT+CMQTTSSLCFG=<client>,<ssl\_context>** - assigns the MQTT client index to the SSL context.  
The first parameter is the MQTT client index: 0...1,  
the second parameter is the SSL context: 0...9.
- **AT+CMQTTCONNECT=<client>,"<tcp://some.mqtt.brocker.url:port>",<keepalive>,<clean\_session>,<user>",<password>** - opens a connection to the MQTT server.  
First parameter - MQTT client index,  
Second parameter - MQTT server address (URL), must be preceded by "tcp://" and enclosed in "  
Third parameter - keepalive time (e.g., 60),  
Fourth parameter - clean session flag - value 1 causes the server to remove all client info after disconnection,  
Parameters 5 and 6 - username and password for the MQTT server, each enclosed in "
- **AT+CMQTTDISC=<client>,<timeout>** - closes the connection to the MQTT server.  
First parameter - client index: 0...1,  
Second parameter - timeout value (e.g., 120).
- **AT+CMQTTTOPIC=<client>,<length>** - specifies the topic for the MQTT message.  
First parameter - client index: 0...1,  
Second parameter - topic length (number of characters).  
After sending the command, the ">" character will be displayed, indicating that the declared number of characters should be sent.
- **AT+CMQTTPAYLOAD=<client>,<length>** - specifies the payload (content) of the MQTT message.  
First parameter - client index: 0...1,  
Second parameter - message length (number of characters).  
After sending the command, the ">" character will be displayed, indicating that the declared number of characters should be sent.
- **AT+CMQTT PUB=<client>,<QoS>,<timeout>** - sends the message to the MQTT server with the previously specified topic and content.

First parameter - client index: 0...1,

Second parameter - QoS:

0 = at most once

1 = at least once

2 = exactly once

Third parameter - timeout value.

• **AT+CMQTTSUBTOPIC=<client>,<length>,<QoS>** - specifies the topic to subscribe to from the MQTT server.

First parameter - client index: 0...1,

Second parameter - topic length,

Third parameter - QoS.

After sending the command, the ">" character will be displayed.

• **AT+CMQTTSUB=<client>** - sends the subscription message to the MQTT server.

First parameter - client index: 0...1.

When the server receives a message with the subscribed topic, the module will automatically display content similar to this:

```
+CMQTTRXSTART: 0,7,20
```

```
+CMQTTRXTOPIC: 0,7
```

```
MyTopic
```

```
+CMQTTRXPAYLOAD: 0,20
```

```
KAmod LTE CAT1 Hello
```

```
+CMQTTRXEND: 0
```

## Running MQTT with HiveMQ Server

First, you must create an account at <https://www.hivemq.com>. In the information panel, under the *Overview* tab, the necessary addresses for the created MQTT broker will be provided. The most important is the *TLS MQTT URL*.

### Connection Details

Comprehensive details and statistics for your cluster

**URL**

  
**Port**

**Websocket Port**


**TLS MQTT URL**

**TLS Websocket URL**

Next, go to the *Access Management* tab and add access credentials for MQTT clients - a username and password. Note these down as they will be needed shortly.

Overview **Access Management** Integrations Web Client • Getting Started

---

 Authentication

**Credentials** Active

Currently you have not created any credentials. Fill out the following form to create an access credential  
Fundamentals guide.


NAME	PERMISSION
kamodA7670	PUBLISH_SUBSCRIBE

**Add new credential**

Go to the *Web Client* tab; there, in the *Connection Settings* window, connect using the username and password you noted. The *Topic Subscriptions* and *Send Message* windows will be unlocked.

Overview Access Management Integrations **Web Client** • Ge


---

 Please connect to the WebClient in order to subscribe to topics

## Connection Settings

Connect to your HiveMQ Cloud Cluster with your credentials. Do not worry you can quickly connect with autogenerated credentials.

**Username \***

**Password \***  

**Connect** or **Connect with autogenerated credentials**

Subscribe to an MQTT topic - enter e.g., *MyTopic* in the *TOPIC* window and set QoS to 1. Then you can send a message with the same topic *MyTopic*, QoS 1, and content e.g., Hello - it should appear in the *Messages* window on the right.

Overview Access Management Integrations **Web Client** Getting Started

✓ The WebClient is connected

## Connection Settings

Connect to your HiveMQ Cloud Cluster with your credentials. Do not worry you can quickly connect with autogenerated credentials.

Username \* Password \*

kamodA7670 .....

**Disconnect**

## Topic Subscriptions

**Unsubscribe from all topics**

Subscribe to topics to receive messages from the HiveMQ cluster. You can also set the Quality of Service (QoS) for each topic. The higher the QoS, the more reliable the message delivery is. You can always subscribe to the (#) wildcard to receive all messages. Please note that the messages from internal probe topics are not displayed here.

TOPIC	QoS	ACTIONS
<input type="text" value="MyTopic"/>	QoS: 1	<input type="button" value="🗑"/>

## Send Message

If you cannot see any messages, make sure you are subscribed to the correct topics. You can always subscribe to the (#) wildcard to receive all messages.

**Message \***

**Topic \*** **QoS \***

QoS: 1

## Messages

0 **Topic: MyTopic** **QoS: 1**

Hello

Now we will configure KAmoD LTE CAT1 to work with the created HiveMQ MQTT server. First, we must install the CA certificate. The correct file can be downloaded from <https://letsencrypt.org/certs/isrgrootx1.pem>. Its content is 1938 characters long and looks like this:

-BEGIN CERTIFICATE-

```
MIIIFazCCA10gAwIBAgIRAIQz7DSQONZRGpGu20CiwAwDQYJKoZIhvcNAQELBQAw
```

...  
-END CERTIFICATE-

Connect KAmoD LTE CAT1 to the computer using a high-quality, short USB cable. Connect to the "SimTech AT Port" COM port using any terminal program. Set parameters to 115200, 8, N, 1. Remember to end each command with <CR> (hex 0x0D).

Send the command **AT+CCERTDOWN="isrgrootx1.pem",1938** to the module. The ">" character will be displayed. Use your terminal's "send file" function to transfer the content. If successful, an **"OK"** response will be sent.

Check if the certificate was installed – send **AT+CCERTLIST**; you should receive a response similar to:

```
+CCERTLIST: "isrgrootx1.pem"  
OK
```

Now configure SSL: send **AT+CSSLFCFG="sslversion",0,4** – this enables support for all TLS and SSL versions.

Send **AT+CSSLFCFG="authmode",0,1** – for server-side authentication.

Send **AT+CSSLFCFG="cacert",0,"isrgrootx1.pem"** – set the root CA certificate file.

Send **AT+CSSLFCFG="enableSNI",0,1** – enable Server Name Indication.

Check settings with **AT+CSSLFCFG?**; you should see:

```
+CSSLFCFG: 0,4,1,1,300,"isrgrootx1.pem","","",1  
OK
```

Start the MQTT session. Send **AT+CMQTTSTART**.

Send **AT+CMQTTACCQ=0,"KAmoD",1** – set the client name to "KAmoD" and set the connection to SSL/TLS.

Send **AT+CMQTTSSLFCFG=0,0** – assign the MQTT client to the SSL context.

Send **AT+CMQTTCONNECT=0,"tcp://xxxxxxxxxx.s1.eu.hivemq.cloud:8883",30,1,"<user>","<password>"** – use your TLS MQTT URL preceded by "tcp://" and your credentials. You should receive:

```
OK  
+CMQTTCONNECT: 0,0
```

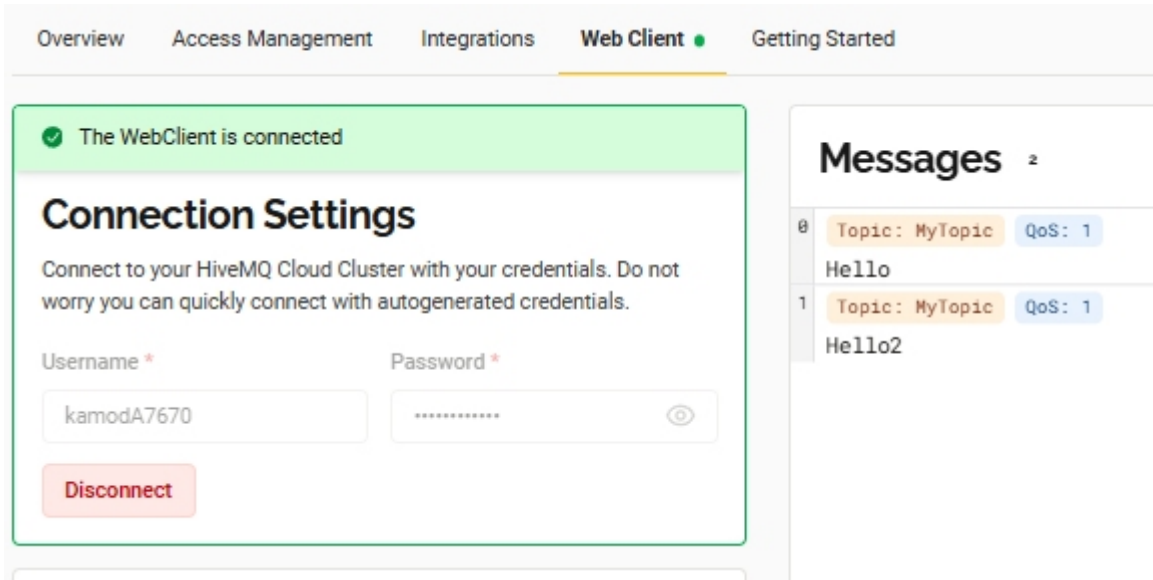
Set the topic: **AT+CMQTTTOPIC=0,7** then type: *MyTopic*

Set the payload: **AT+CMQTTPAYLOAD=0,6** then type: *Hello2*

Publish: **AT+CMQTT PUB=0,1,60**

```
OK  
+CMQTT PUB: 0,0
```

Check the HiveMQ Web Client panel to see the message.



To subscribe to MyTopic: **AT+CMQTTSUBTOPIC=0,7,1** then type: *MyTopic*, followed by **AT+CMQTTSUB=0**.

If you send *Hello3* from the HiveMQ panel, the module will display:

```
+CMQTTRXSTART: 0,7,6
+CMQTTRXTOPIC: 0,7
MyTopic
+CMQTTRXPAYLOAD: 0,6
Hello3
+CMQTTRXEND: 0
```

## Links

- [SimCom A7670/A7672 Modem Documentation](#)
- [AT Command Description](#)
- [Examples](#)
- [SIMCOM Windows USB Drivers](#)



Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.

Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.

BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.

BTC Korporacja nie ponosi odpowiedzialności za jakiegokolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.

BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.