KAMAMI

KAmod LTE CAT1 with A7670E-LASE module

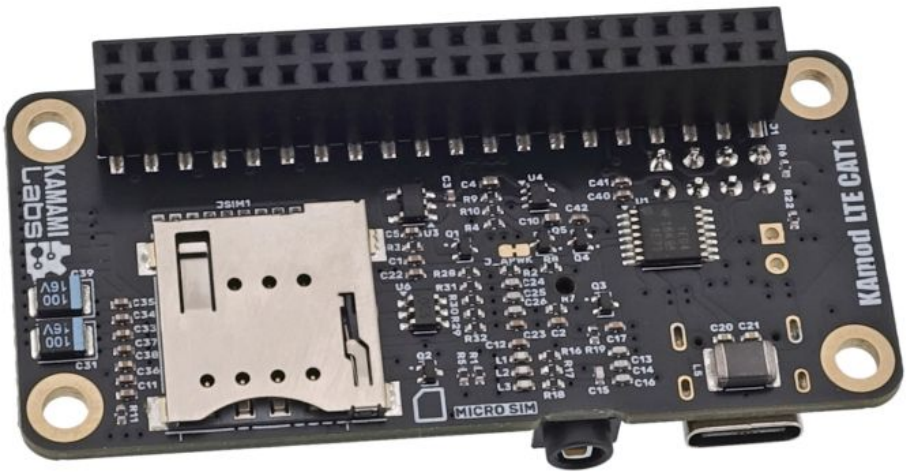
KAMAMI

Rev. 20260404130730
Źródło: https://wiki.kamamilabs.com/index.php?title=KAmod_LTE_CAT1_with_A7670E-LASE_module

Table of contents

Description

KAmod LTE CAT1 - GSM/GPRS HAT with A7670E-LASE 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 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 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.

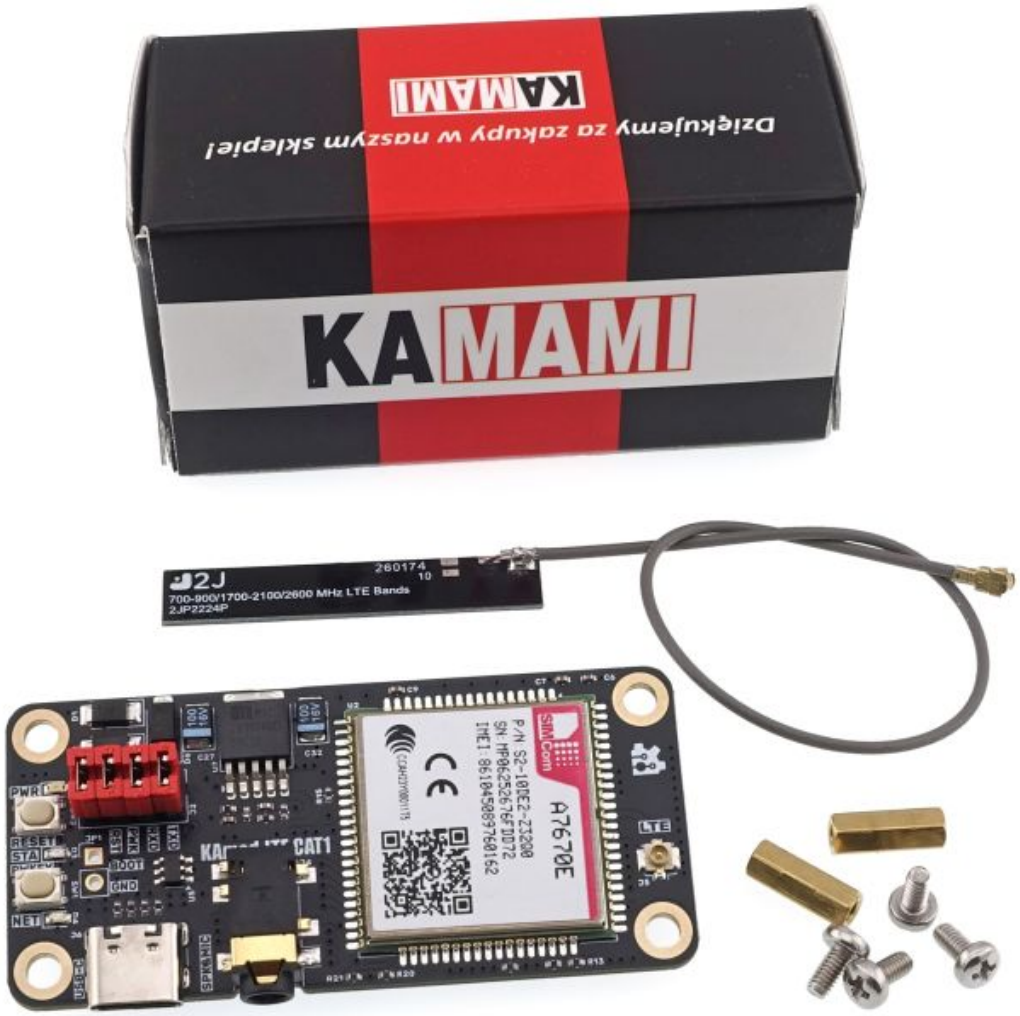


Basic Parameters

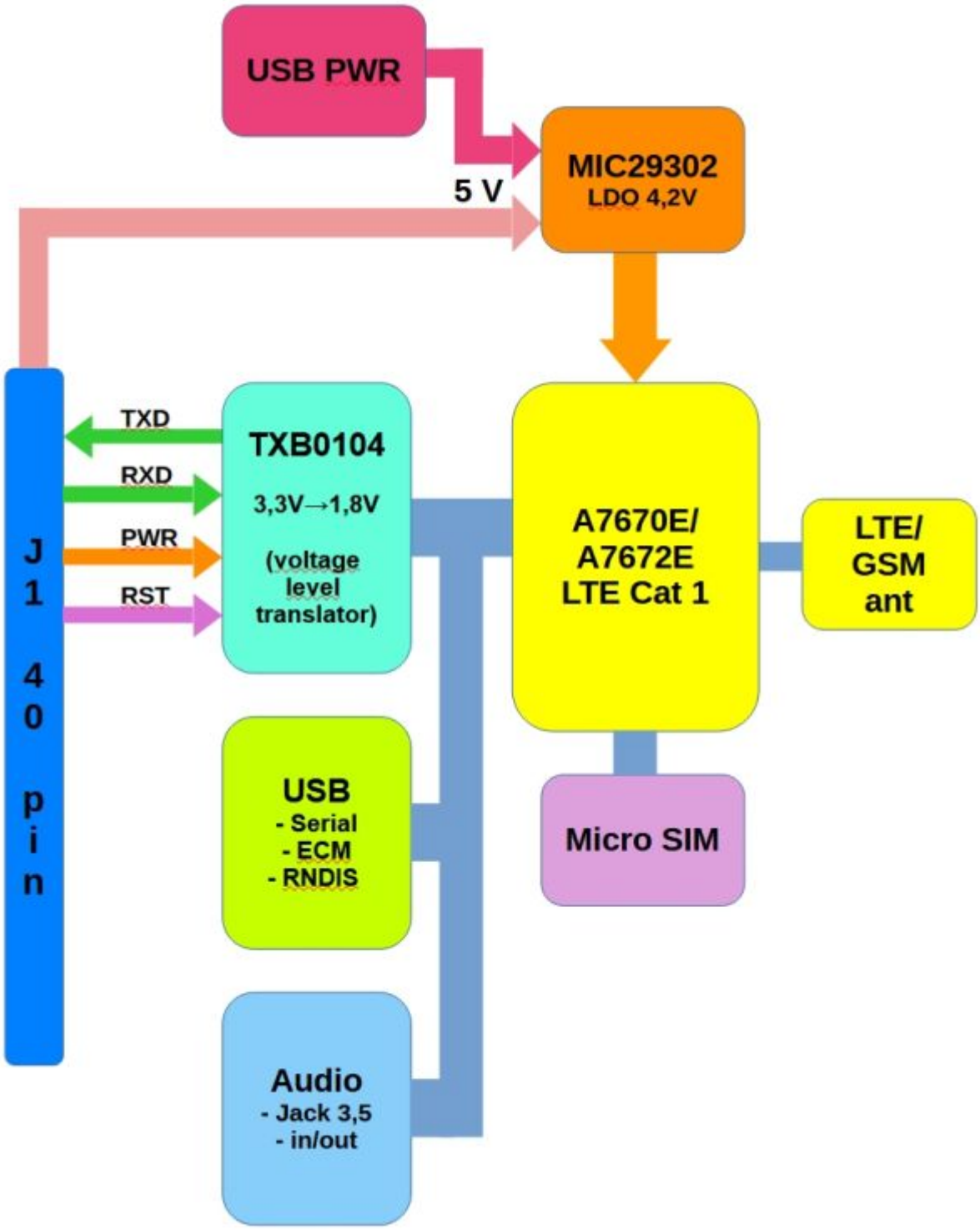
- LTE CAT-1 modem based on the SimCom A7670E or A7672E module
- LTE (4G) wireless communication bands: LTE-FDD B1/B3/B5/B7/B8/B20
- GSM (2G) wireless communication bands: 900/1800 MHz
- LTE data transfer speed: max 10 Mbps (DL); max 5 Mbps (UL)
- EDGE data transfer speed: max 236.8 kbps; GPRS: max 85.6 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
- 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
KAmod LTE CAT1	Assembled and tested module
GSM Antenna	GSM antenna with U.FL connector
Mounting Kit	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

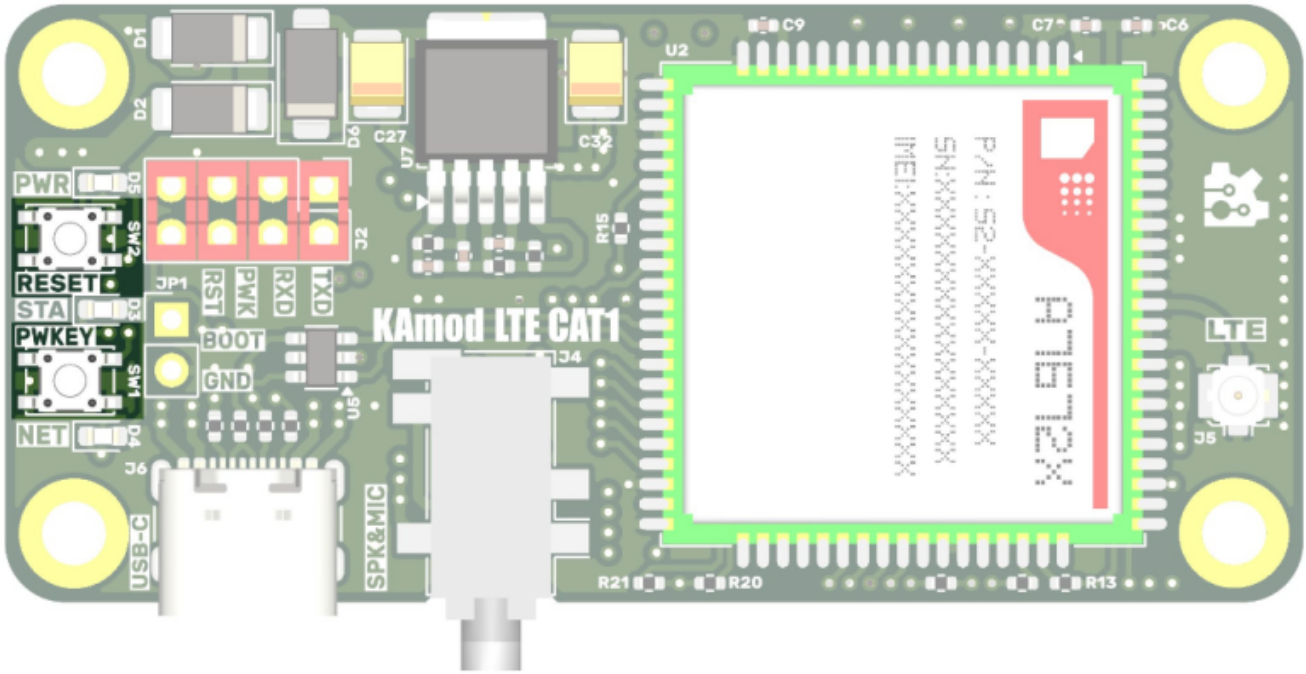
Button functions are described in the table below.

Button	Function

PWKEY (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).
RESET (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.

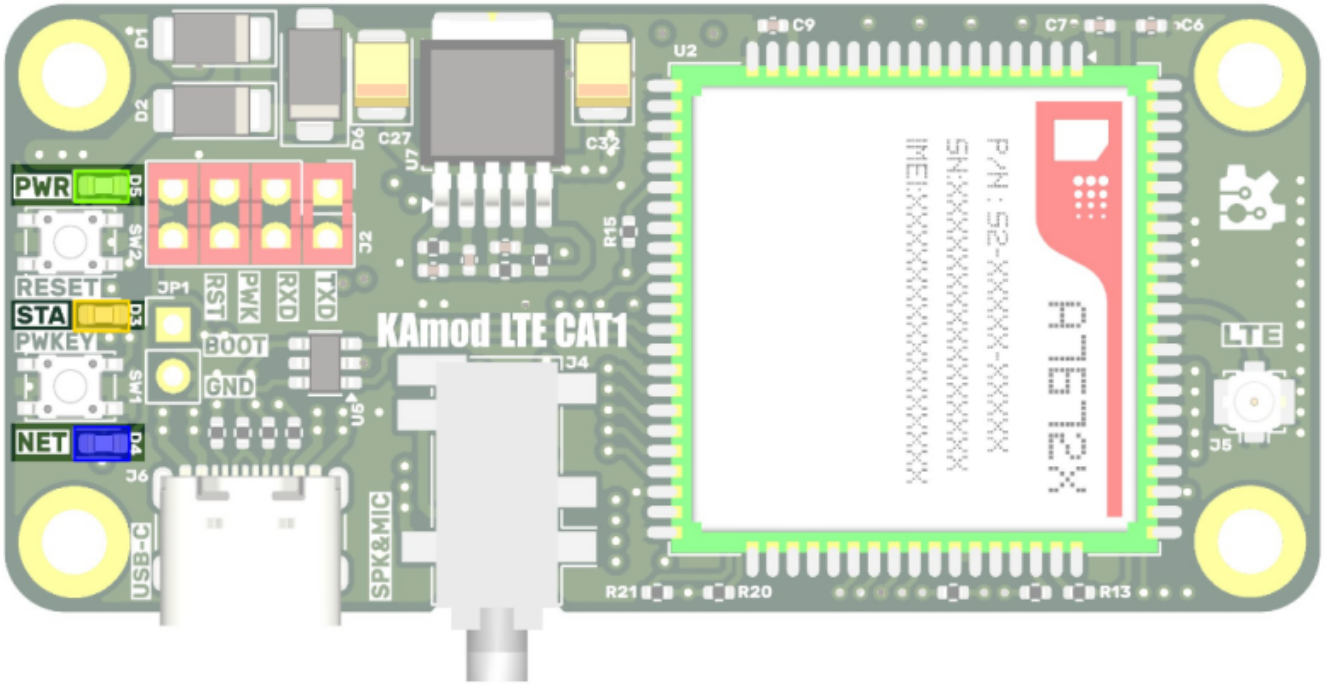
The KAmod LTE CAT1 module includes an automatic power-on circuit for the A7670E/A7672E module after the power supply is connected. This circuit generates a short pulse on the PWK line immediately after power-up. To deactivate this circuit, cut the jumper marked J_APWK located on the bottom of the board.



Status Indicator Functions

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

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



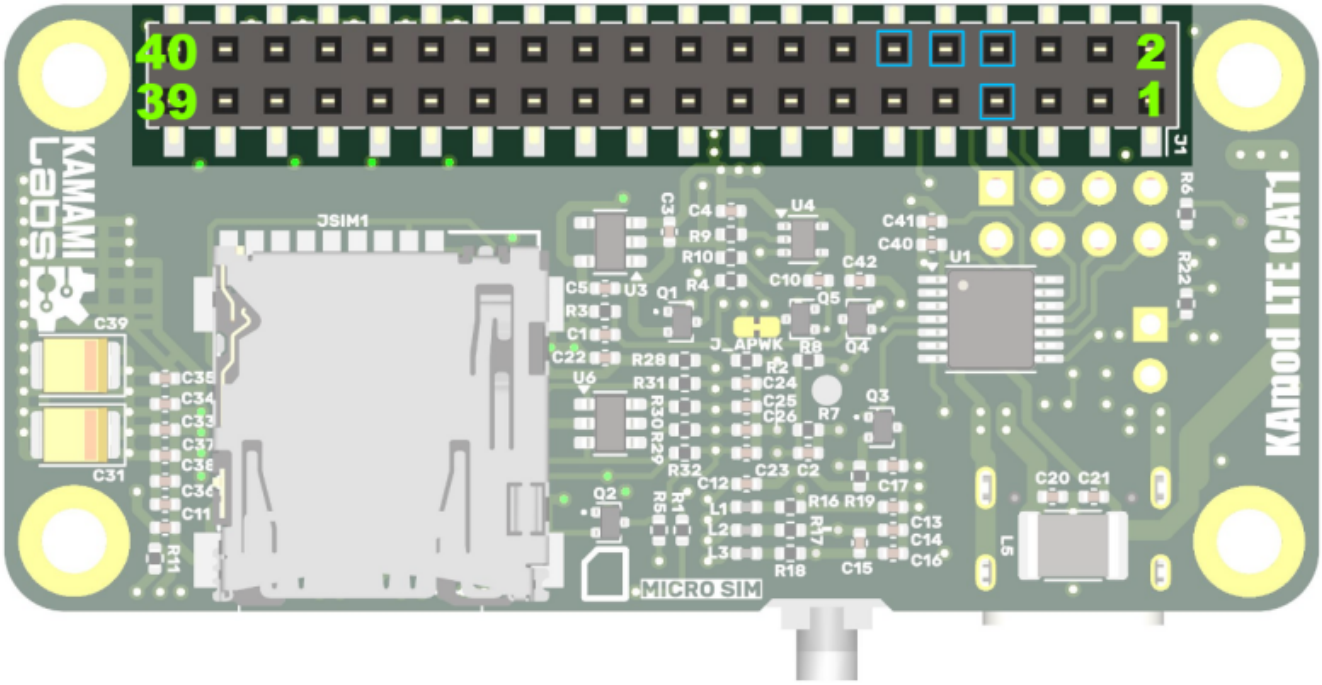
Connection with Raspberry Pi Boards

The KAmod LTE CAT1 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.

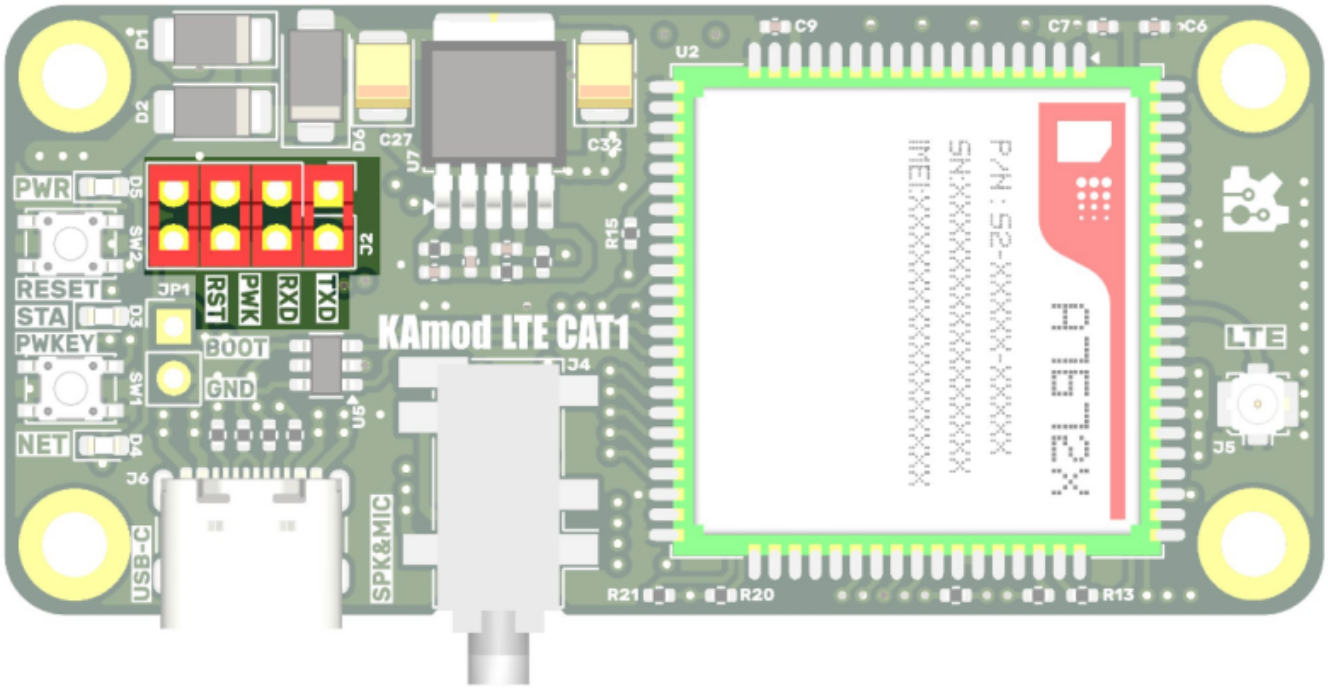
A7670E/A7672E module control signal	Control signal on Raspberry Pi family board (J1)	Function
UART TXD (out)	RXD/GPIO15 (pin 10)	Serial data output
UART RXD (in)	TXD/GPIO14 (pin 8)	Serial data input
RST - RESET (in)	GPIO18 (pin 12)	RESET signal input, active High
PWK - PWRKEY (in)	GPIO04 (pin 7)	Module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

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



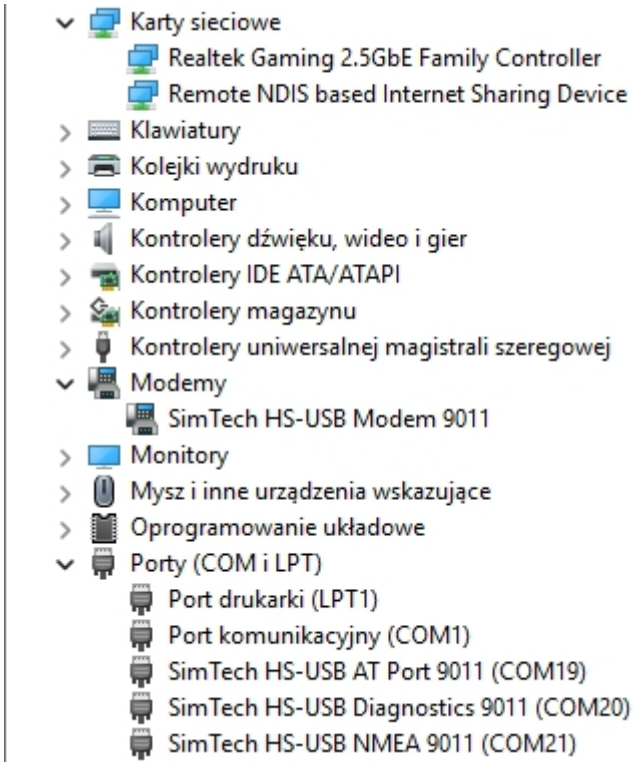
Not all control signals (TXD, RXD, PWK, and RST) must be connected to the Raspberry Pi. Signals will be connected if 4 jumpers are placed on the pins marked J2, as shown in the figure. Each jumper corresponds to a different signal. Removing a jumper disconnects the signal from the 40-pin Raspberry Pi connector. For example, you can omit the PWK and RST signals because the KAmod LTE CAT1 board provides the appropriate reset and startup for the A7670E/A7672E module.

The KAmod LTE CAT1 board should be attached to the 40-pin GPIO pin header available on Raspberry Pi series boards. This provides power and connects the UART serial port, enabling control via AT commands. To maintain better stability of this setup, it is recommended to use additional spacers and screws.

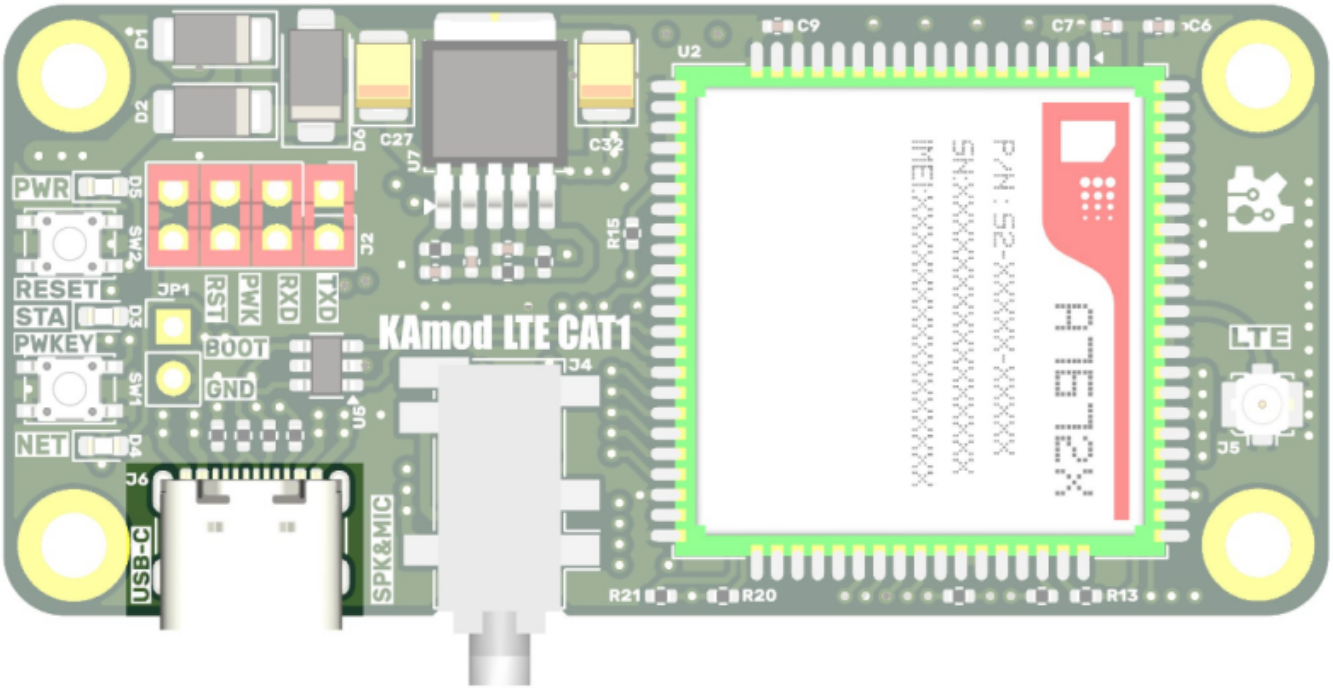


USB Interface

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”.

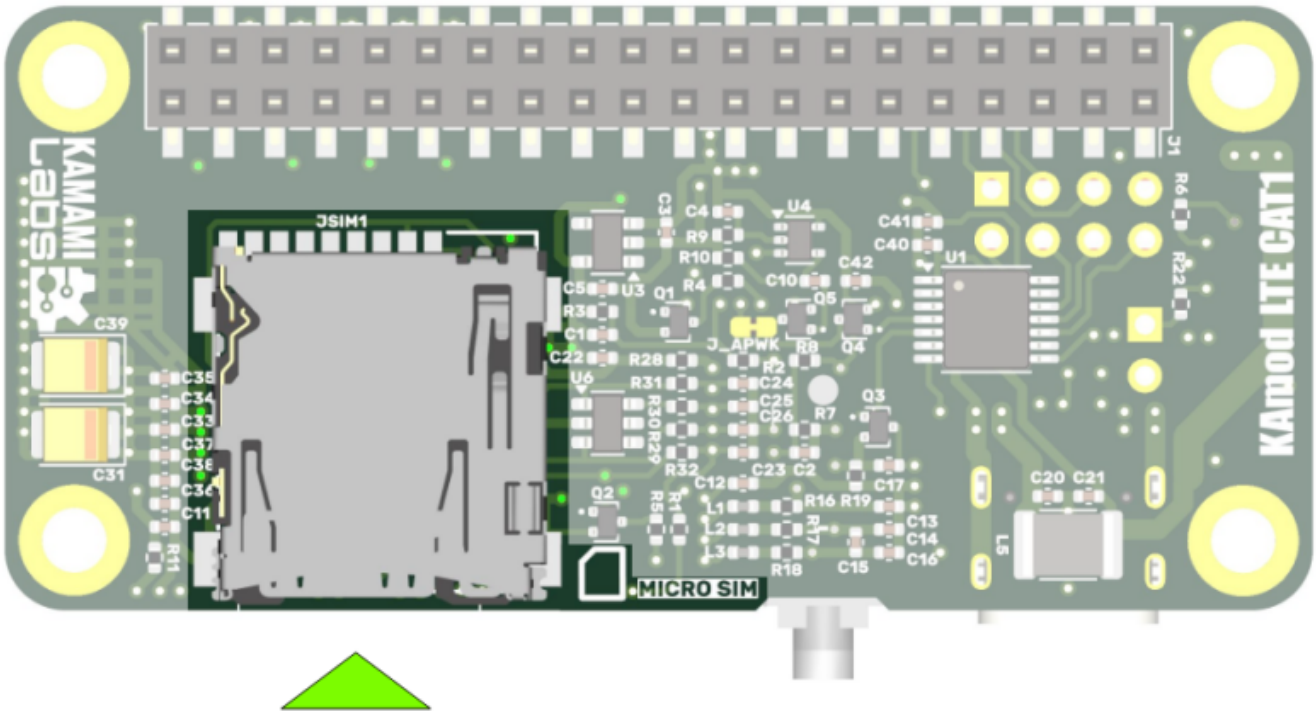


The instantaneous current consumption of the A7670E/A7672E module can exceed 2 A, so ensure that the USB port connected to the KAmod LTE CAT1 board has sufficient power. Use only high-quality USB cables with a maximum length of 0.5 m. In case of USB communication issues, it is worth checking the operation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



SIM Card

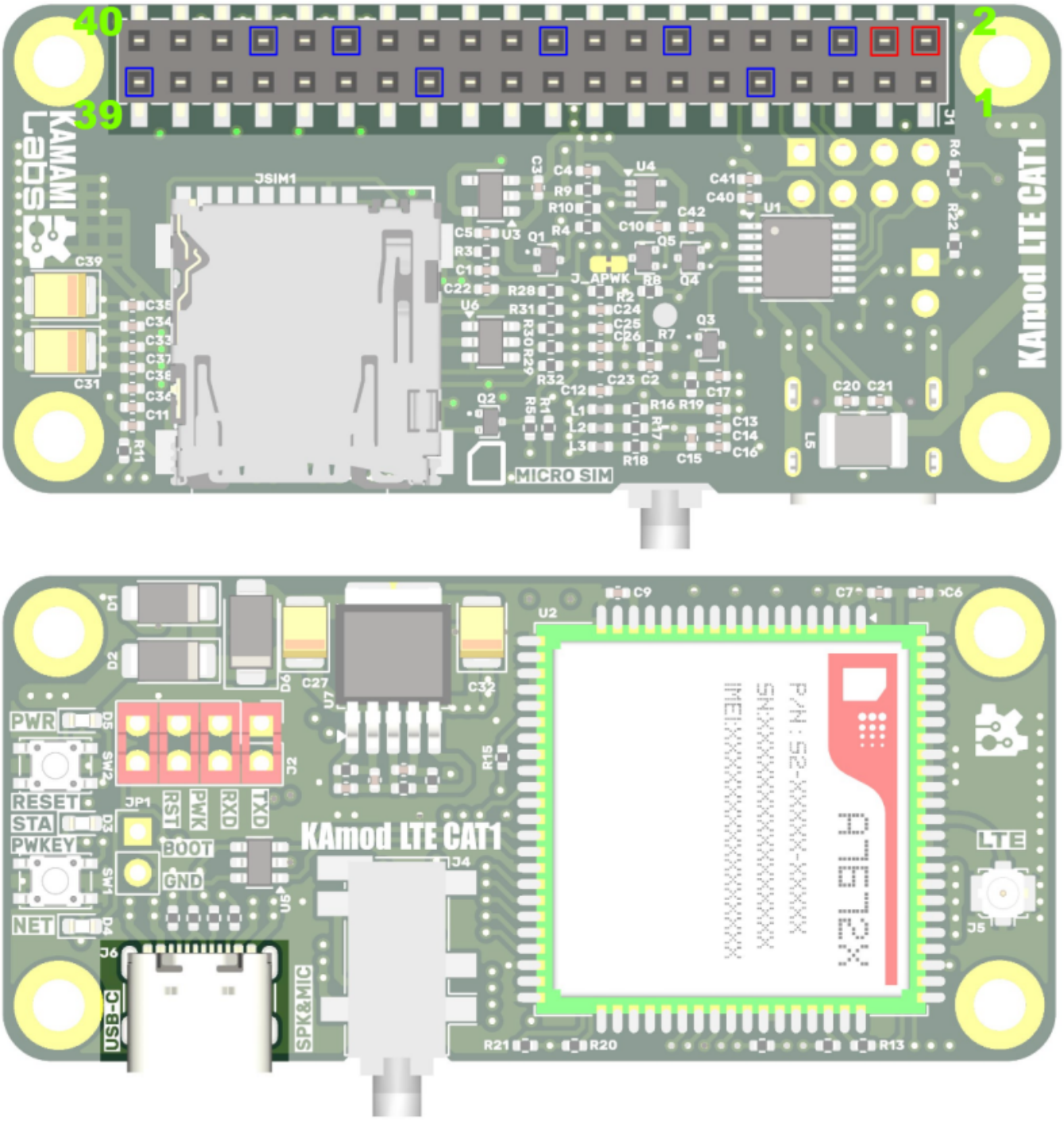
KAmod LTE CAT1 is equipped with a Micro SIM card slot (15x12 mm) and operates at 1.8/3.0 V. The card should be placed in the slot and gently pressed until it clicks. To remove the card, press it gently and release – the edge of the card will pop out, allowing it to be pulled out completely. All operations 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

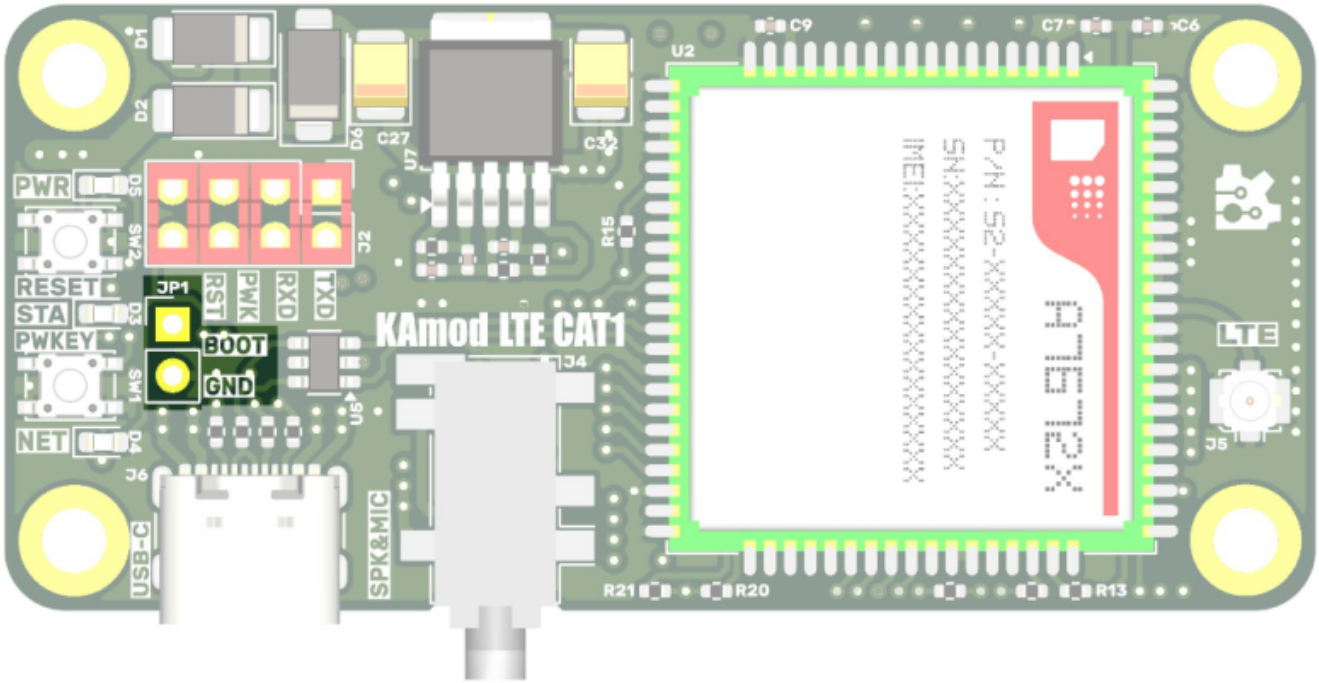
KAmod LTE CAT1 requires a 5 V power supply with a current capacity of at least 2 A. In the active state, when an LTE network connection is established but no tasks are being performed, the module consumes approximately 30 mA. However, during data transmission, instantaneous current consumption can exceed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

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



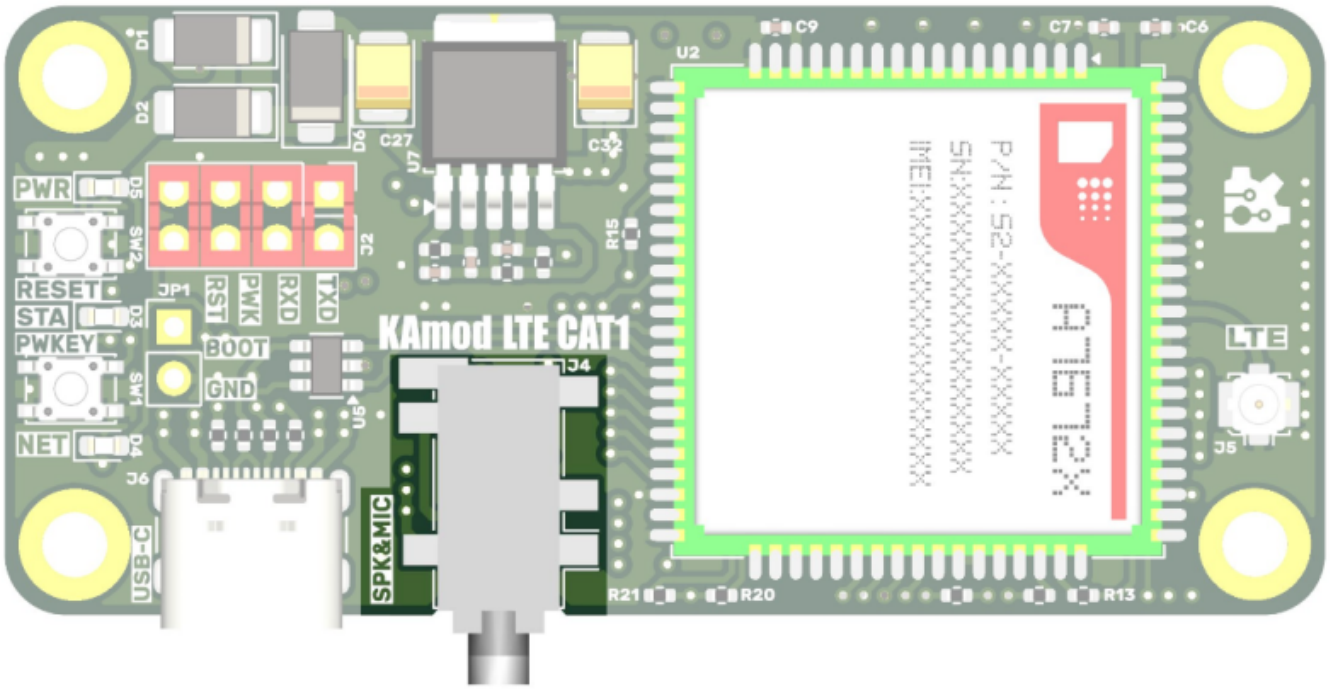
Service Mode

The manufacturer of A7670E/A7672E modules provides its components with the most optimized firmware and does not recommend updating it. The firmware version can be checked using the ATI command. However, JP1 contacts are provided on the KAmod LTE CAT1 board. Shorting them allows the module to start in a special service mode intended for firmware updates. During normal use, the JP1 contacts should not be shorted. More information about firmware updates can be found on the module manufacturer's website - SimCom - www.simcom.com.



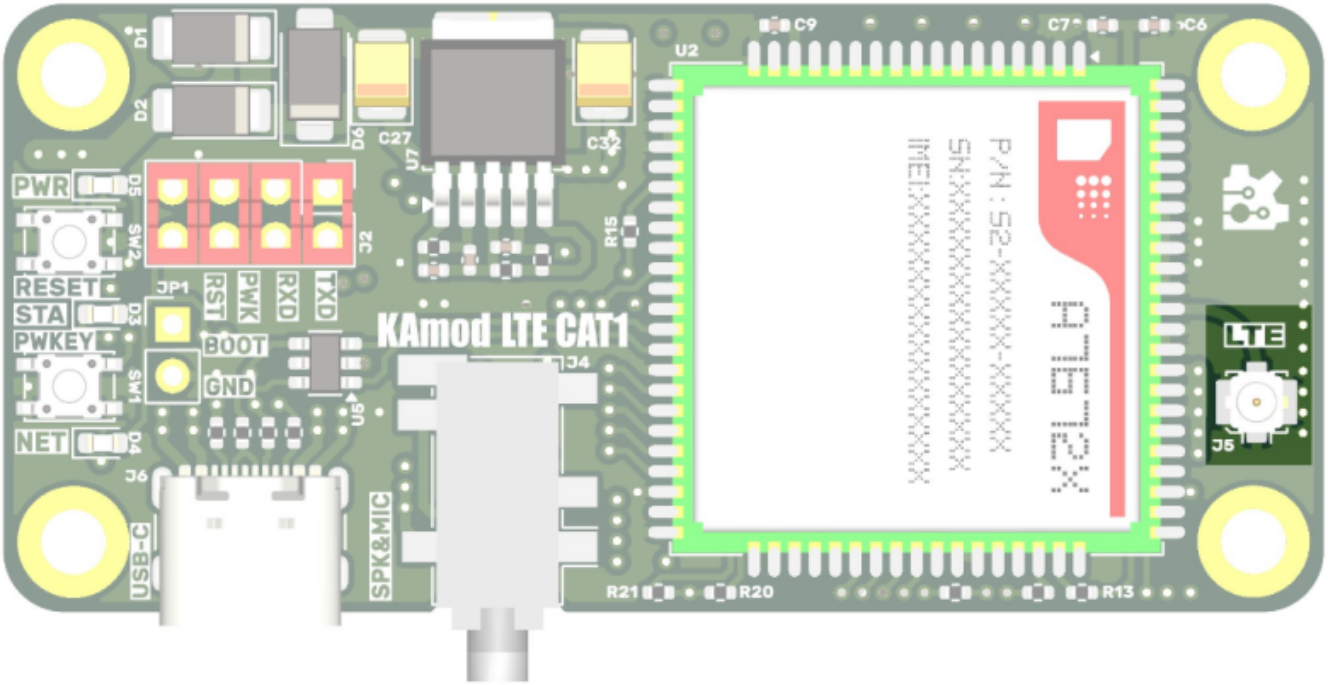
Audio Connector - SPK&MIC

KAmod LTE CAT1 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 allows listening and transmitting audio between call parties.



LTE/GSM Antenna

KAmod LTE CAT1 operates in GSM (2G) and LTE (4G) telephony standards. There is a U.FL (IPX) connector on the module board, marked LTE, to which an antenna designed for GSM/LTE operation must be connected.



AT Command Control

A7670E/A7672E modules are controlled using so-called AT commands sent via the UART serial port. Each 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, including names and phone numbers, start and end with a " character (double quote - hex 0x22). Commands can be entered using any terminal program, such as Putty or Minicom.

A full list of 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 antenna signal 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 defines the bit error rate, where:
0 = 0.01% or less; 7 = 8% or more; 99 = no information

• **AT+CREG?** - checks the GSM/LTE network connection status; a response similar to this will be sent:
`+CREG: 0,1`
The second parameter defines the connection status; significant values are:
0 = no connection; 1 = connection active; 2 = searching for operator; 3 = network connection failed

• **AT+COPS?** - reads the network operator's name; sends 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 Call

• **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). It must end with a semicolon (;). The responses sent will contain information about the call progress.

• **ATA** - allows answering an incoming voice call. An incoming call will be signaled by sending several commands, including:
`+CLIP: "+48123456789",145` - informs about the phone number the call is coming from.
`RING` - indicates the ringing signal.

• **ATH** - terminates the current voice call.

• **AT+COUTGAIN=7** - increases the earpiece volume to level 7 (7 = max, 0 = min).

• **AT+CMICGAIN=7** - increases microphone sensitivity to level 7 (7 = max, 0 = min).

Sending an SMS Message

• **AT+CSCA?** - checks the set message center number. For operators available in Poland, these may include:
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 with the command **AT+CSCA="<full_message_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 recipient's phone number. After confirming with the <CR> character (Enter key), a > character will be sent, signaling to type the message content. After typing the content, confirm the operation with the <1A> character (Ctrl + Z combination) or cancel with the <1B> character (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 number in the modem's memory.

Reading an SMS Message

• **AT+CMGR=4** - reads the message at the 4th position 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`
Meaning:
“REC UNREAD” - message was not previously read; after reading, its status changes to “REC READ”
"+48123456789" - 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 the 4th position in the message memory.

• **AT+CNMI=1,2,0,0,0** - after sending this command, new SMS messages will be automatically read – information similar to this will appear each time:
`+CMT: "+48123456789","","25/08/13,10:20:18+8"`
`Test Test`

Selected HTTP/HTTPS Functions

• **AT+HTTPINIT** - initializes the HTTP session.

• **AT+HTTPPARA="URL","https://www.example.org"** - sets HTTP parameters, in this case, the URL. Other available options include “CONTENT”, “ACCEPT”, or "USERDATA".

• **AT+HTTPACTION=0** - executes an HTTP/HTTPS request using a 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`
Meaning:
“200” - server response, 200 = OK
“1256” - amount of data to read (Content Length)

• **AT+HTTPHEAD** - reads the server response header (HTTP 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 (44 here) is the index to start reading from,
the second parameter is the amount of data to read.

• **AT+HTTPDATA=18,10** - used for transmitting data using the POST method. The first parameter defines the amount of data (from 1 to 153600 bytes), the second parameter defines the time in which 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 parameters for each context mean:
ssl version - SSL network protocol version,
auth mode - authentication mode,
ignore local time - certification time check,
negotiate time - maximum time for 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="sslversion",<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 plus client certificate and key);
3 = client-side authentication (client certificate and key).

• **AT+CSSLCFG="ignorelocaltime",<ssl_context>,1** - certification time check.
The first parameter is the context number: 0...9
the second parameter means:
0 = check certification time
1 = ignore certification time check

• **AT+CSSLCFG="cacert",<ssl_context>,"filename.pem"** - specifies the root CA certificate file for a given SSL context. The file must have a .pem or .der extension.

• **AT+CSSLCFG="enableSNI",<ssl_context>,1** - Server Name Indication function - enables support for multiple SSL certificates on a single IP address and port. 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 send - file length.
After sending the command, the "'
*" character will be displayed, indicating that the declared amount of data should be sent. If successful, the response "**OK**" will be sent.*

• **AT+CCERTLIST** - displays a 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.
First parameter: specifies the index assigned to the client name (0 or 1),
second parameter: client name (KAmod-LTE-CAT1),
third parameter: 0 - TCP connection; 1 - SSL/TLS connection.

• **AT+CMQTTACCQ?** - displays the 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`
First parameter is the client index: 0...1,
second parameter is the error code, e.g.: 0 = OK; 19 = client in use; 20 = client not yet set.

• **AT+CMQTTSSLCFG=<client>,<ssl_context>** - assigns the MQTT client index to an SSL context.
First parameter is the MQTT client index: 0...1,
second parameter is the SSL context: 0...9.

• **AT+CMQTTCONNECT=<client>,"<tcp://some.mqtt.broker.url:port>",<keepalive>,<clean_session>,"<user>","<password>"** - opens a connection with the MQTT server. First parameter - specifies the client index used to connect,
second parameter - MQTT server address (URL), must be preceded by “tcp://” and enclosed in " marks,
third parameter - keepalive time (e.g., 60),
fourth parameter - clean session flag - value 1 causes the server to delete all client information after disconnection,
parameters 5 and 6 - username and password for the MQTT server, each must be enclosed in " marks.

• **AT+CMQTTDISC=<client>,<timeout>** - closes the connection with the MQTT server.
First parameter - client index: 0...1,
second parameter - timeout time (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, a “>” 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, a “>” character will be displayed, indicating that the declared number of characters should be sent.

• **AT+CMQTTPUB=<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 = message delivered at most once
1 = message delivered at least once
2 = message delivered exactly once
third parameter - timeout time.

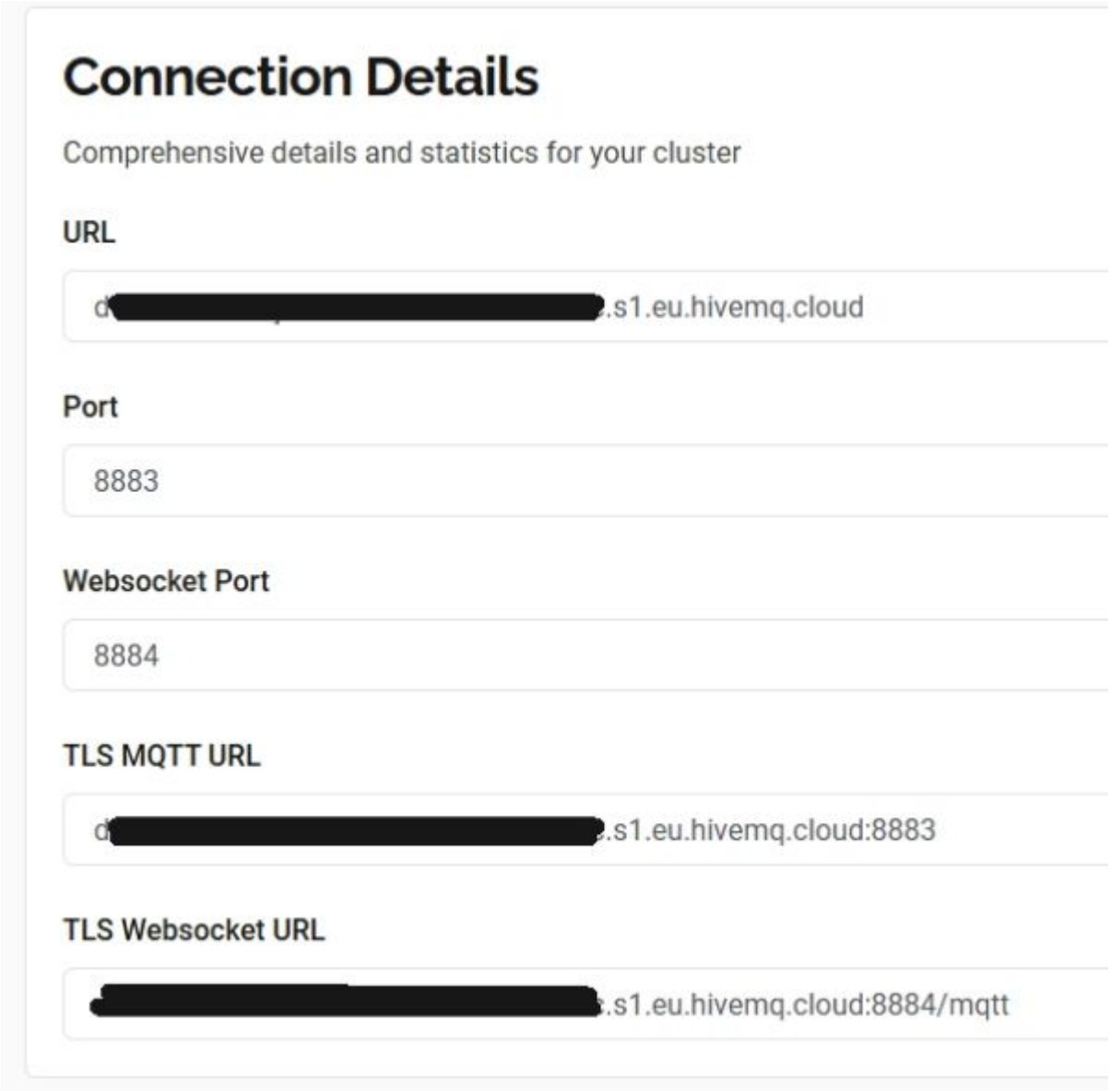
• **AT+CMQTTSUBTOPIC=<client>,<length>,<QoS>** - specifies the topic to subscribe to from the MQTT server.
First parameter - client index: 0...1,
second parameter - topic length,
third parameter - QoS (0, 1, or 2).
After sending the command, a “>” character will be displayed, indicating that the declared number of characters should be sent.

• **AT+CMQTTSUB=<client>** - sends a subscription request to the MQTT server for the previously specified topic.
First parameter - client index: 0...1.
When the MQTT server receives a message with the subscribed topic, the A7670/A7672 module will automatically display information like this:

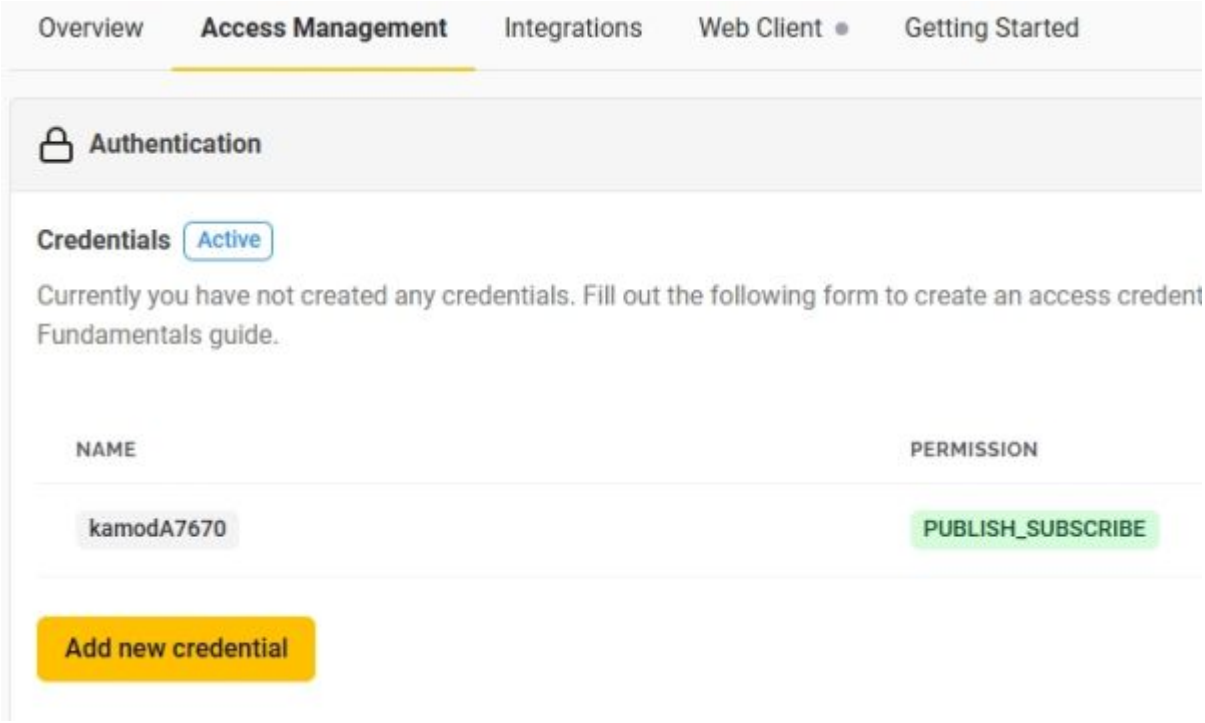
```
+CMQTTRXSTART: 0,7,20
+CMQTTRXTOPIC: 0,7
MyTopic
+CMQTTRXPAYLOAD: 0,20
KAmod LTE CAT1 Hello
+CMQTTRXEND: 0
```

Running MQTT with HiveMQ Server

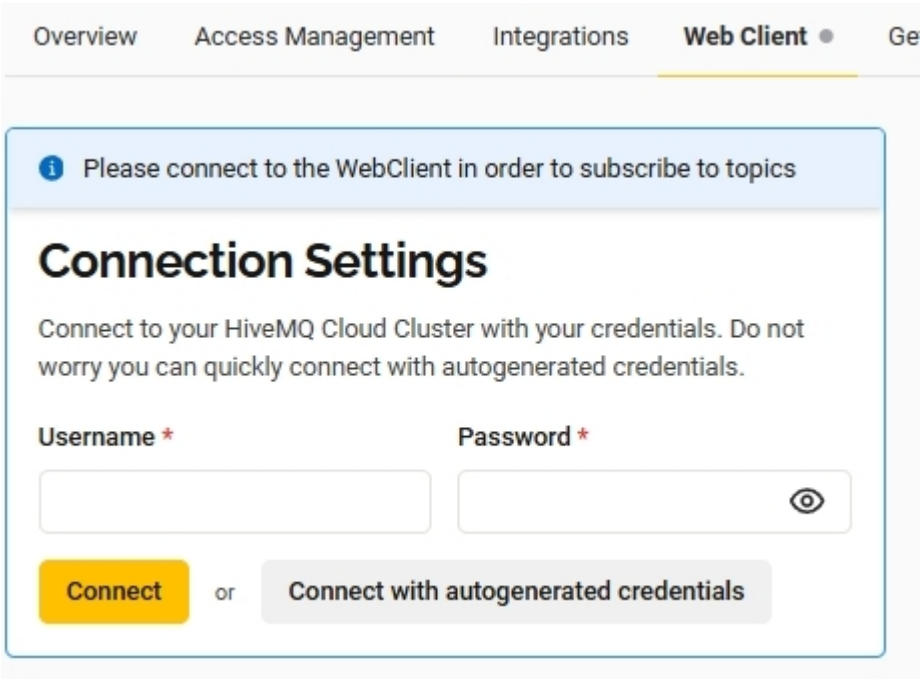
First, create an account at https://www.hivemq.com. In the dashboard under the *Overview* tab, the necessary addresses for the created MQTT broker will be provided. The most important one is the *TLS MQTT URL*.



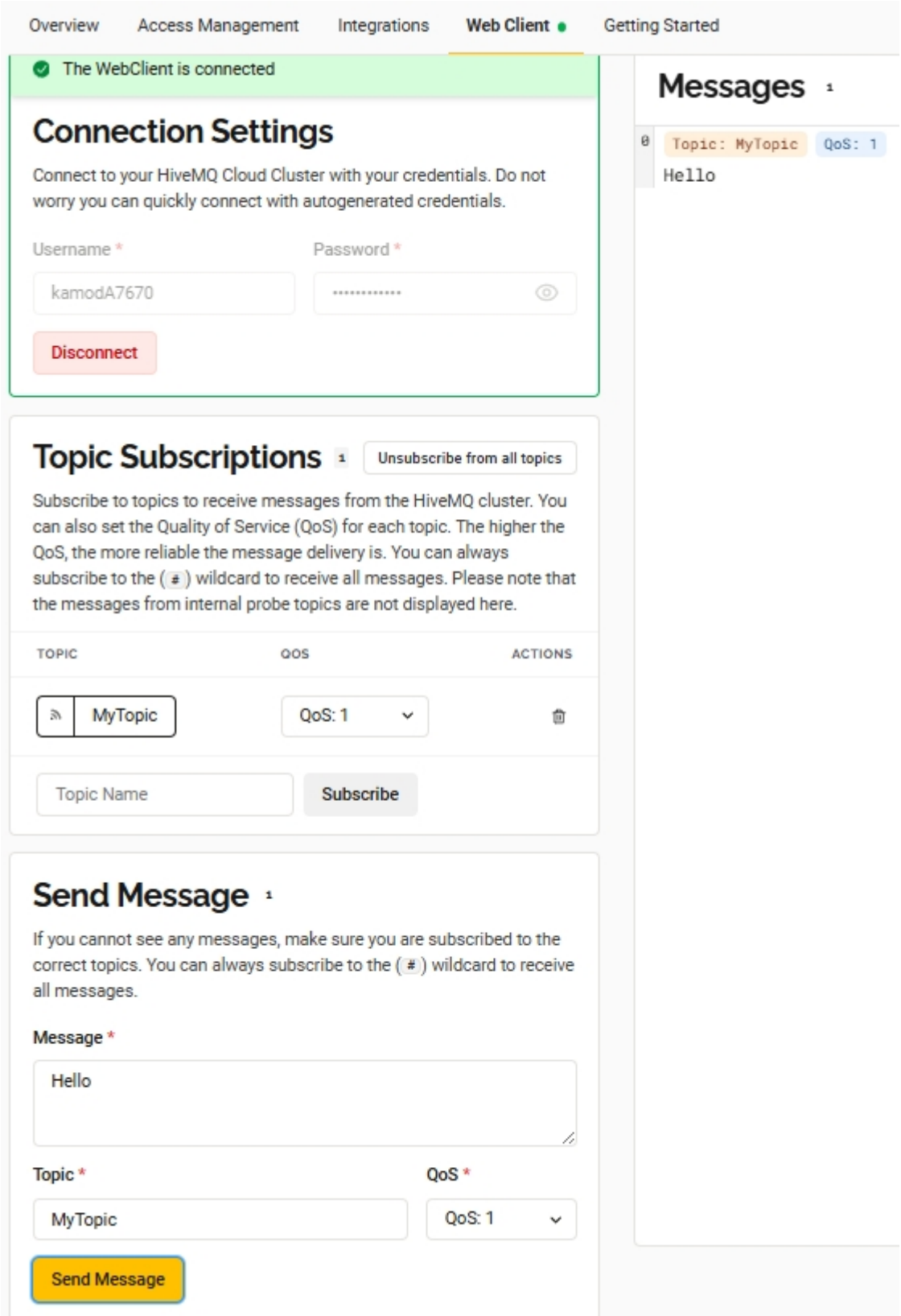
Next, go to the *Access Management* tab and add access credentials for MQTT clients – a username and password. Note these down as they will be needed shortly.



Go to the *Web Client* tab and connect using the username and password you noted. The *Topic Subscriptions* and *Send Message* windows will become active.



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.



Now, configure KAmod LTE CAT1 to work with the HiveMQ server. First, 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-
MIIFazCCA1OgAwIBAgIRAIIQz7DSQONZRGPgu2OCiwAwDQYJKoZIhvcNAQELBQAw
...
```

```
...
...
emyPxgcYxn/eR44/KJ4EBs+lVDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=
-END CERTIFICATE-
```

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

Send the command **AT+CCERTDOWN="isrgrootx1.pem",1938** to the module. A “>” character will be displayed. Use your terminal's "send file" feature to transmit the certificate content. If successful, the module will return **“OK”**.

Verify the installation by sending **AT+CCERTLIST**; you should receive:

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

Configure SSL by sending **AT+CSSLCFG="sslversion",0,4** – enabling all TLS and SSL versions.

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

Send **AT+CSSLCFG="cacert",0,"isrgrootx1.pem"** – set the root CA certificate for SSL context 0.

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

Verify settings with **AT+CSSLCFG?**; you should see:

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

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

Send **AT+CMQTTACCQ=0,"KAmod",1** – set client name to “KAmod” and use SSL/TLS.

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

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

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

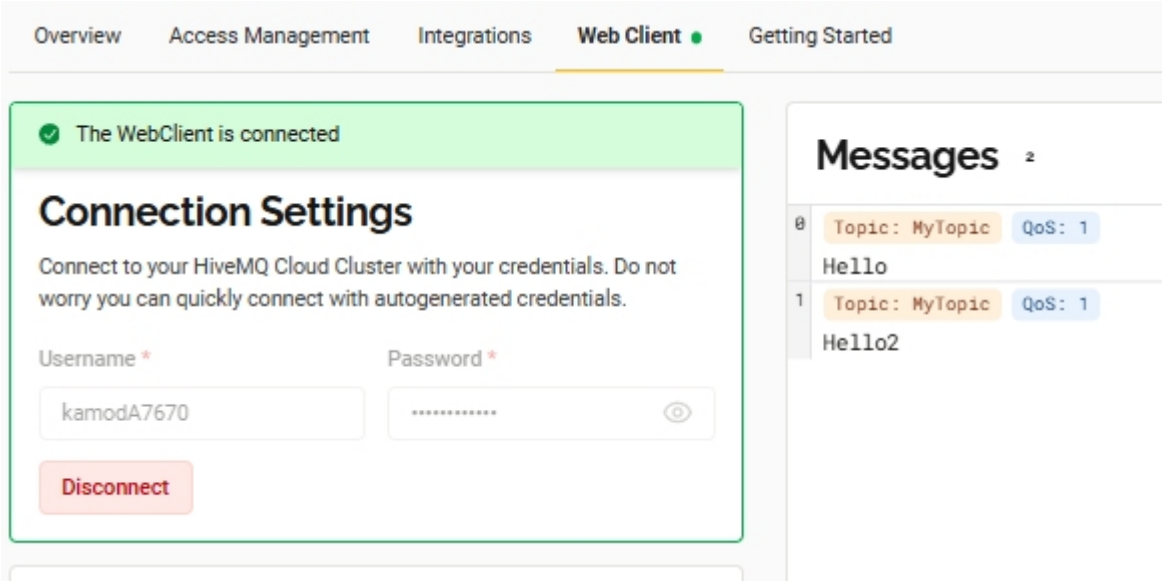
Set the topic: **AT+CMQTTTOPIC=0,7** then type: *MyTopic*

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

Publish the message: **AT+CMQTTPUB=0,1,60**. Response:

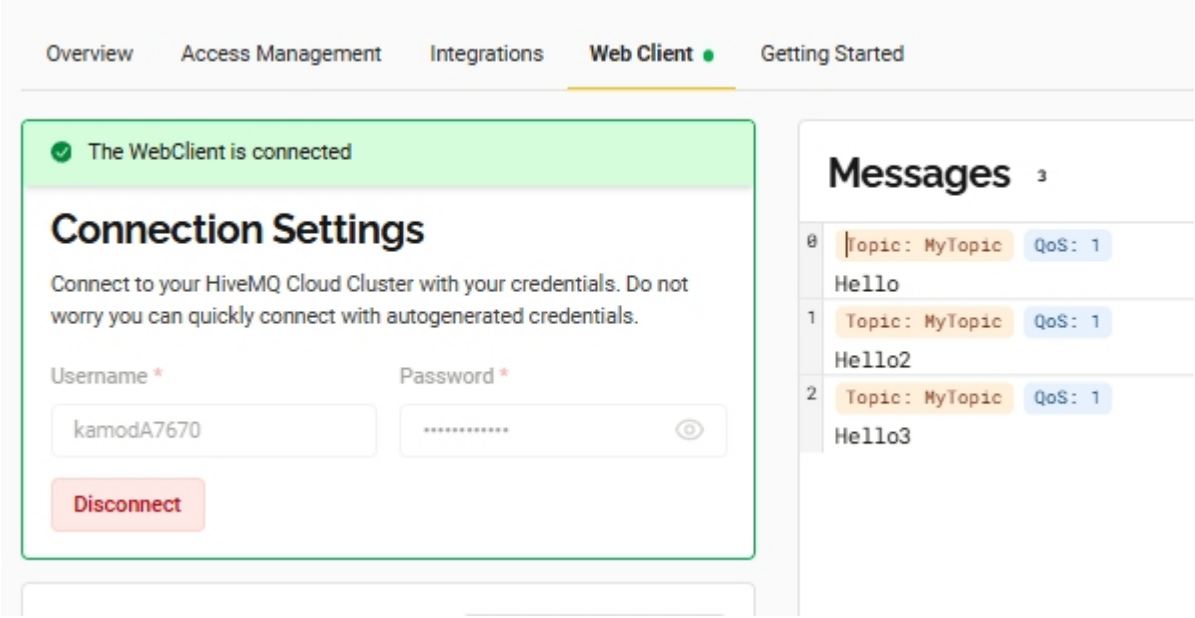
```
OK
+CMQTTPUB: 0,0
```

Check the HiveMQ Web Client; your message should appear there:



To subscribe to "MyTopic", enter: **AT+CMQTTSUBTOPIC=0,7,1** then type: *MyTopic*.

Send: **AT+CMQTTSUB=0**. Response:
OK
+CMQTTSUB: 0,0
If you now send *Hello3* from the HiveMQ panel:



The module will automatically display:
+CMQTTRXSTART: 0,7,6
+CMQTTRXTOPIC: 0,7
MyTopic
+CMQTTRXPAYLOAD: 0,6
Hello3
+CMQTTRXEND: 0

Links

- SimCom A7670/A7672 Modem Documentation
- AT Command Manual
- Code Examples
- SIMCOM Windows USB Drivers

BTC Korporacja
05-120 Legionowo
ul. Lwowska 5
tel.: (22) 767-36-20
faks: (22) 767-36-33
e-mail:
sprzedaz@kamami.pl
https://kamami.pl