Manual

mioty® Gateway AVA

Created by

WEPTECH elektronik GmbH

Ostring 10/ 76829 Landau

Germany

1 Revision history

Order number:	**A001-0090-001**		
Revision	Date	Description	Author
V1.0	13.04.2021	Initial Release	SN2/ TB3

Contents

List of Figures

List of Tables

2 General information

2.1 Scope of delivery

- One (1) mioty® Gateway AVA
- One (1) Plugin USB C Raspberry PI power supply
- One (1) SMA antenna
- Two (2) mounting clips
- Four (4) screws and a screwdriver
- One (1) Quick Start Guide

2.2 Signal words and warning symbols used

Symbols used:

DANGER/ WARNING/ CAUTION

Note

Information

3 Notes on the documentation

3.1 Safety instructions

WARNING! In order to prevent danger to the operating personnel and damage to the device, the work described must be performed by qualified personnel.

Do not try to charge the battery! Do not disassemble the battery! Do not pierce, break, crush or cut the device or the battery! Do not short-circuit the contacts! Do not expose the battery to fire or moisture! Do not drop the device or the battery. Do not try to change or charge the battery in the device.

Do not expose the device or the battery to an open flame or extremely high temperatures! Exposing batteries to extremely high temperatures can cause an explosion or leakage of flammable liquids or gases. Do not expose the battery to temperatures below -20 or above 55 ° C. Do not expose the device or the battery to liquids or extremely low air pressure!

CAUTION: Failure to observe the safety instructions can result in fire, electric shock and other injuries or damage to the device or other property (The housing is made of plastic with sensitive electronic components and batteries inside).

At its core, the AVA Gateway is a Raspberry Pi computer with a regular Ubuntu operating system. It can be accessed via SSH with default usernames and passwords (see chapter 6.2SSH login) The default user has root access on the device. From a network security standpoint, all that is very unsafe. Talk to your network administrator about these issues before you install the device.

Note: Limitation of liability: The manufacturer assumes no liability for non-compliance with the operating instructions, improper use, modification or damage to the device.

3.2 Directives, laws and standards

The use of radio frequencies is limited by national regulations. This gateway was developed in accordance with the requirements of RED (Radio Equipment Directive) 2014/53 / EU.

3.3 Validity

These operating instructions apply exclusively to the product with the order number A001-0090-001.

3.4 Storage of documents

This manual and all other applicable documents should always be kept in a place where those working with the device have access to it.

4 Description of the device

4.1 Product information

AVA is a bidirectional, low cost gateway with the mioty® Profiles EU 0 EU 1 EU 2 US 0 for usage in Europe and the US.

AVA is easy to install. The gateway can be connected to your local network via Ethernet. It offers a web interface that gives overview and insights into all available mioty® traffic.

It also contains rudimentary development implementations for mioty® Application and Service Centers that allow you to configure your development sensors, decode their traffic, and forward their data via MQTT for easy evaluation. It is also possible to configure the gateway to connect to Application and Service Centers of your choice via BSSCI instead.

4.2 Features

Features	
Input frequency range (EU):	868 - 870 MHz
Input frequency range (US):	915 - 917 MHz
Tx power:	14 dBm
Sensitivity (EU1):	< -135 dBm
Input Voltage:	5V DC (USB C)
Typical Input Current (EU1):	approx. 900mA
Dimensions:	56 x 90 x 25 mm
Weight: approx.	0,16 kg
Temperature range:	0 - 35°C (indoor use only)
Humidity range:	20 - 80%

Table 1 Features

5 Device Usage

5.1 Device installation

Connect the AVA gateway to LAN with DHCP support, then attach the antenna to the gateway. Plugin the USB-C Raspberry Pi power supply and open your web browser. Type “ava” into the address bar. You will get to the dashboard of the gateway. If that fails, please contact your network administrator on how you can access the gateway in your network. In some networks the gateway is not accessible via name but via an IP address that your network administrator can provide to you.

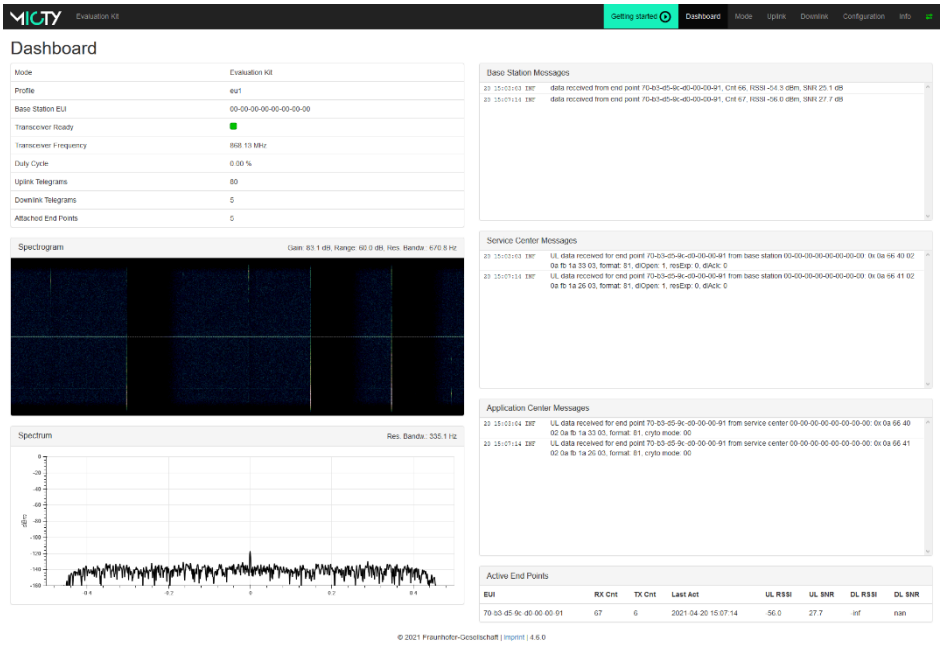

Figure 1 Dashboard

5.2 Sensor Registration

The AVA gateway can only decode data from sensors that have been registered on the gateway. To register a sensor, navigate to the configuration page on the gateways web interface and fill out the Endpoint Registration form as follows:

- EUI (extended unique identifier): Read this off the label of your sensor. It should be formatted as 8 hexadecimal numbers, separated by hyphens, e.g. 70-b3-d5-9c-d0-00-00-01.
- ShAddr (short address): Defaults to the 5th and 6th byte of the EUI, formatted as a single hexadecimal number (e.g. Sensor 70-b3-d5-9c-d0-00-00-01 would have the short address “d000”)
- BiDi (bidirectionality): Please check if your sensor is able to receive data as well as transmit it.
- PreAtt (preattachment): Please check if the sensor is configured to be pre-attached. Most sensors are. Can be left empty if the sensor supports over-the-air attachment.
- Network Key: The sensor’s network key, formatted as a single string of hexadecimal digits.
- Application Key (optional): The sensors application key, formatted as a single string of hexadecimal digits.
- Type EUI (optional): The EUI of the sensors type description.

When you have completed the form, click “Register” on the right. The sensor should then appear under “Registered Endpoints”.

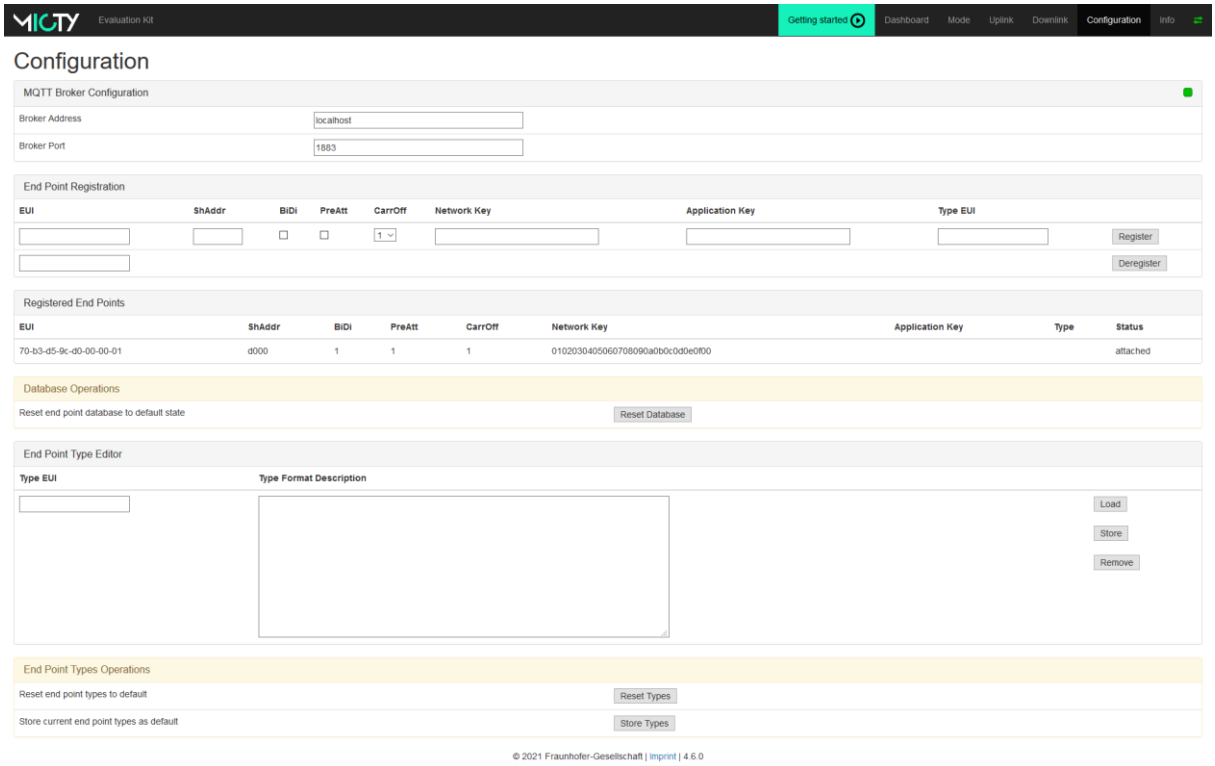

Figure 2 Configuration

5.3 Data reception

Any time a sensor, that has been correctly registered on the gateway, transmits data, the following information will appear on the dashboard (see Figure 1):

- An entry in the “Base Station Messages” box that indicates the EUI of the sensor, the packet counter of the received transmission, and the RSSI and SNR of the transmission.
- An entry in the “Service Center Messages” box that shows the EUI of the sensor, the raw data payload of the message, the payload format indicator (if different from 0), as well as the values of the dlOpen (Downlink Window Open), resExp (Respoinse Expected) and dlAck (Downlink Acknowledgement) flags
- An entry in the “Application Center Messages” box that shows the EUI of the sensor, the raw data payload of the messages, the payload format indicator (if different from 0), and the application layer crypto mode.

The last received packet of each registered sensor is also displayed on the Uplink page.

5.4 Data transmission

You can enqueue downlink transmissions to bidirectional sensors on the Downlink page. Fill out the form as follows:

- EUI (extended unique identifier): Read this off the label of your sensor. It should be formatted as 8 hexadecimal numbers, separated by hyphens, e.g. 70-b3-d5-9c-d0-00-00-01.

- Data: Formatted as a JSON array of numbers, e.g. “[1, 14, 128]”.
- Preschedule: Check to preschedule the data in the Service Center; If left unchecked, the data is held in the Application Center. On the Evaluation Gateway, this difference is almost irrelevant. Leave this unchecked
- Counter: If ”Preschedule” is checked, enter the packet counter to which the data should be transmitted in response. Otherwise, enter 0.

When you have completed the form, click “Queue” on the right. The data should then appear under “Locally Queued Downlink Data”. It will then be transmitted on the next or prescheduled downlink window.

5.5 MQTT

The gateway comes with a preinstalled and preconfigured MQTT broker. The mioty® Application Center uses this broker to publish MQTT messages with the received mioty® uplink messages as well as subscribe to MQTT messages with mioty® downlink messages to be transmitted.

You can use an MQTT client like mosquito_client or MQTT Explorer to send and receive those MQTT messages.

Also take a look at the provided Python MQTT Client Example which you can download on the Info page of the device.

5.5.1 Topics

For each sensor, the Gateway will use two MQTT topics on the MQTT broker:

`mioty/00-00-00-00-00-00-00-00/70-b3-d5-9c-d0-00-00-01/uplink`

`mioty/00-00-00-00-00-00-00-00/70-b3-d5-9c-d0-00-00-01/downlink`

In these examples, “70-b3-d5-9c-d0-00-00-01” is the EUI of the sensor.

MQTT also allows wildcards “+” for a single level, and “#” for several levels. E.g. subscribe to “`mioty/00-00-00-00-00-00-00-00/+/uplink`” to receive uplink messages of any sensor or „`mioty/#`“ to receive all messages that are transmitted on the broker.

5.5.2 Uplink messages

Here is an example JSON message that the gateway will publish in the uplink topic for every uplink message that it receives:

```
{
      "baseStations":[
      {     "bsEui":0,
            "rssi":20.0,
            "rxTime":1613134205634647808,
            "snr":10.0  },
      {     "bsEui":0,
            … }
      ],
      "cnt":30,
      "components":null,
      "data":[1,2,3,4],
      "format":0,
      "meta":null,
      "typeEui":0
}
```

Key	Type	Description
baseStations	Array of Objects	Reception information
bsEui	String	EUI of the base station
rssi	Number	RSSI in dBm
rxTime	Number	Unix time stamp in ns
snr	Number	SNR in db
cnt	Number	Packet counter
components	Object	Interpreted payload components if blueprint is available
data	Array of Numbers	User payload as byte array
format	Number	MPF field
meta	Object	End Point meta information if blueprint is available
typeEui	Number	typeEui of associated blue print

5.5.3 Downlink messages

Here is an example JSON message that the gateway can receive in the downlink topic to enqueue a downlink transmission:

```
{
      "data":[1,2,3,4],
      "format":0
}
```

Key	Type	Description
data	Array of Numbers	User payload as byte array
format	Number	MPF field

6 Advanced Topics

6.1 Manipulating Debug parameters

For select advanced topics it is necessary to manipulate Debug parameters.

Please change these parameters only if you have been specifically advised to do so, and/or know what you are doing. Manipulating these parameters can make the device inoperable. Weptech will provide no support or replacement in that case.

Navigate to “ava/debug.html” (or IP address + “/debug.html” if your network does not provide access via hostname). The web interface will now display an additional Debug page in the menu at the top. On that page, you will see a list of parameters, sorted in to modules.

You can edit the values of the parameters. These changes will persist until you reboot the device.

If you want to persist these values beyond that, change the “storeSystemConfig” key at the very top to “true.” It will be immediately reset to “false” but the current configuration will be persisted to the SD card.

You can also manipulate these parameters by logging into the device via SSH (see chapter 6.2) and editing the file /data/evalkit4_config.xml directly. That XML can optionally contain values for any key in any module. The file is read on startup. For any missing keys, the default values are used. This file is overwritten by the “storeSystemConfig” key described above.

6.2 SSH login

For select advanced topics it is necessary to log into the underlying operating system.

At its core, the AVA gateway is a Raspberry Pi computer with an Ubuntu operating system. It runs an SSH server on port 22. The default user and password are both “ubuntu”. You can log in with any SSH client, like the “ssh” command that comes preinstalled on Linux or Windows 10, or a dedicated terminal program like PuTTY.

When you log in for the first time you are required to change the default password. Please follow the steps on the command line. Once the password has been changed successfully, you will be disconnected automatically. Just reconnect with your new password.

6.3 Changing the mioty® Profile

To change the chosen mioty® Profile, log into the device via SSH (see chapter 6.2) and edit the file `/etc/default/mioty-evalkit`. That file defines an environment variable for the mioty_evalkit service, EVAL_OPT, which contains command line parameters which are passed to the service on startup.

Add or change the command line parameter "`--profile <value>`", where <value> can be one of "eu0", "eu1", "eu2" or "us0" (without the quotes). "eu1" is the default.

Restart the device for the change to take effect.

6.4 Disabling packet deduplication

Mioty® uses a packet counter. It is used for the deduplication of packets (which may be received by several base stations at once), but also as a countermeasure against replay attacks.

During development, your end device may not yet comply with the requirement that the packet counter be persisted on the device and increase with every packet, even when the device is rebooted. When working with such device you may want to turn off the Gateway's deduplication feature. It is controlled via the "duplicateDiscard" key in the "sc" module of the debug parameters (see chapter 6.1).

6.5 Time synchronization

It is a known issue that if the device has not been configured with a proper date and time, the device may be unable to decode incoming mioty® traffic. The device will try to automatically configure its date and time via NTP. If that fails, please contact your network administrator so they can provide the gateway with access to the NTP service on port 123.

6.6 External Service Centers through HTTP proxy servers

Many corporate networks use HTTP proxies for internet access. While evaluating other service and application center implementations, like Behrtech MyCloud, you may experience problems connecting the gateway to those external servers through the HTTP proxy.

Please note that the following section provides only rudimentary guidance for a use case that is not officially supported. If you have any trouble understanding and following the proposed steps, please do not attempt it.

To route TCP traffic through the HTTP proxy to an external server, install simpleproxy:

```
$ sudo apt install simpleproxy
```

Next, create a service file called `/etc/system/system/simpleproxy.service` to automatically start the simpleproxy service at boot:

```
[Service]

Type=simple

ExecStartPre=iptables -t nat -I OUTPUT -p tcp -d <external server> --dport <external
port> -j REDIRECT --to-port 9009

ExecStart=/usr/bin/simpleproxy -v -L 9009 -R <external server>:<external port> -S
<proxy server>:<proxy port>

ExecStopPost=-iptables -t nat -D OUTPUT -p tcp -d <external server> --dport
<external port> -j REDIRECT --to-port 9009

Restart=always

RestartSec=20s


[Install]

WantedBy=mioty-evalkit.service


[Unit]

Description=Simple Proxy for Mioty Evalkit

After=systemd-resolved.service

Wants=systemd-resolved.service
```

The ExecStartPre statement installs an iptables rule at start which will reroute all traffic that is destined to the remote server back to port 9009 on the local machine. The ExecStart statement will start the simpleproxy program that will open the local port 9009 and route all traffic from this port through the HTTP proxy to the actual remote server. The ExecStop Post statement will in turn remove the rerouting rule from iptables after the service has shut down.

After you have created the file, let system reload all service files and enable the service:

```
$ sudo systemctl daemon-reload && sudo systemctl enable simpleproxy
```

Afterwards, reboot the gateway.

7 Disposal

All old electric and electronic appliances have to be exposed of separate from the domestic waste over locations provided for this by the state. In accordance with EU Directive 2012/19 // EC on old electrical and electronic devices and their implementation in national law, old electrical devices must be collected separately and recycled in an environmentally friendly manner.

8 Copyright protection

9 Warranty information

Great care was taken in preparing this manual. No guarantee is given for the correctness of the contents of this manual since errors, in spite of all efforts, can never be completely avoided.

10 Disclaimer

The author reserves the right to not be responsible for the topicality, correctness, completeness, or quality of the information provided. Liability claims regarding damage caused by the use of any information provided, including any kind of information which is incomplete or incorrect, will therefore be rejected. Parts of the pages or the complete publication including all offers and information may be extended, changed or partly or completely deleted by the author without separate announcement.

This disclaimer is to be regarded as part of the internet publication which you were referred from. If sections or individual terms of this statement are not legal or correct, the content or validity of the other parts remain uninfluenced by this fact.

11 Operating limitations

WEPTECH elektronik products are not authorized for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. WEPTECH elektronik GmbH customers using or selling these products for use in such applications, do so at their own risk and agree to fully indemnify WEPTECH elektronik GmbH for any damages resulting from any improper use or sale.
Use of WEPTECH elektronik GmbH products commits the user to the terms and conditions set out herein.

WEPTECH elektronik GmbH

Ostring 10

D-76829 Landau

http://www.WEPTECH.de

info@WEPTECH.de

Tel: +49 6341 9255-0

Fax: +49 6341 9255-100

Note:

The information, instructions and descriptions in this manual refer to the actual operating and service conditions herein.

For technical questions, safety notes or technical failure, please contact WEPTECH elektronik GmbH.

We reserve the right to technical modifications.