User Guide –
AT Command Specification

Aurawave AW100 Series

Version 1.0-r2

Revision History

Version	Date	Notes	Contributors	Approver
1.0	7 Aug 2025	Initial release.	Dave Drogowski	Josh Toole Jonathan Kaye
1.0-r2	5 Sep 2025	Add examples Fix error codes	Garrett LoVerde	Josh Toole Jonathan Kaye

Contents

1. Introduction

This document applies to the following Aurawave software versions:

- 1.0

1.1 Scope of the document

This document presents the AT Command Set for the Cloud2GND Aurawave AW100 series module used to control and configure the device.

1.2 Conventions and abbreviations

In this document, the Aurawave module is referred to as any of the following:

- ME (Mobile Equipment)
- TA (Terminal Adapter)
- DCE (Data Communication Equipment).

A host device controls the Aurawave Module by sending AT Command via its serial interface. The AT command 'host' is referred to as any of the following:

- TE (Terminal Equipment)
- DTE (Data Terminal Equipment)
- Plainly "the application", running on the host embedded system

2. AT Command syntax

The "AT" prefix must be set at the beginning of each Command line. To terminate a Command line, enter **<CR>**.

Words enclosed in ‹angle brackets› are references to syntactical elements.
Words enclosed in [square brackets] represent optional items which may be left out from the command line at the specified point.

The brackets are not used when the words appear in the command line.

A string type parameter input should be enclosed between quotation marks ("").

The "AT" prefix is followed by the name of a command which itself is often prefixed by a "+" which indicates an 'extended' command. Though "%" is commonly prescribed for use with vendor specific commands "+" is often used by vendors instead, as is the case here.

The command name is followed by a character or set of characters that indicate which 'sub-operation' of the command should be performed. These operations are outlined in the following subsections.

2.1 Set Operation ‹CMD› = [...]

Set Operations, are used to set values. A set operation implies that the command will also support a read operation. Example:

```
AT+CMD=1
```

Where:

- **AT** is the command line prefix
- **+** is the prefix for commands
- **CMD** is the body of a basic command
- **=** specifies the 'Set' operation
- **1** is a parameter (multiple parameters are separated by commas)

2.2 Read Operation <CMD>?

Read Operations check the current values of parameters.

Example:

```
AT+CMD?
```

Where:

- **AT** is the command line prefix
- **+** is the prefix for commands
- **CMD** is the body of a basic command
- **?** specifies the read operation

2.3 Test Operation <CMD>=?

Test Operations test the existence of the command and provide information about the type of its parameters.

Example:

```
AT+CMD=?
```

Where:

- **AT** is the command line prefix
- **+** is the prefix for commands
- **CMD** is the body of a basic command
- **=? Specifies** the test operation, often used for determining parameter values

2.4 Execute Operation <CMD>[=...]

Execute Operations require no operator character unless they support a parameter. Commands that support execute operations do not support read operations.

Examples:

```
AT+CMD
```

Where:

- **AT** is the command line prefix
- **+** is the prefix for commands
- **CMD** is the body of a basic command

2.5 Responses

AT responds to all commands with a final response.

The response is one of the following:

```
OK<CR><LF>
ERROR<CR><LF>
+CME ERROR: <cause value><CR><LF>
```

Some commands may also produce a varying number of information response lines before the final response. An information response can be received only when a command-specific response syntax is specified. An information response line usually starts with a prefix, which is the command entered:

```
+CMD: [...]<CR><LF>
```

2.6 Entering successive AT commands

When you need to enter a series of AT commands on separate lines, please note that you need to wait for the final response (for example OK, CME error) of the last AT Command you entered before you enter the next AT Command.

2.7 Supported character set

Only ASCII printable characters can be used for AT Command.

3. Use case Examples

3.1 Default/Factory Auracast broadcast, Stereo, USB audio input)

The default use case employs a factory default Aurawave module and thirdparty Auracast devices or a 2nd Aurawave module to demonstrate Auracast end to end.

The default factory configuration for an AW100 module on bootup is as follows:

- 1 Public Broadcast Source instance.
- Broadcast Name: "Aurawave"
- 1 BIG with 2 BIS streams configured as Left and Right audio locations
- 48khz 'low-latency' audio
- USB Audio Input

Use case requires:

- USB Audio Host device and Source (Windows, iOS, Android) (Known issue with USB Audio on macOS see release notes)
- Aurawave module in factory default state
- 3rd Party Auracast Sink & Auracast Assistant or a 2nd Aurawave module + headphones with 3.5mm plug
    - Tested third party devices include
        - Samsung Galaxy Buds 3 Pro
        - Samsung S23
        - HomesSpot JM320 + iOS Device with QK Audio App

Configuration steps:

1. Plug a factory default Aurawave module into a USB Audio Host device and play audio. If your module is not in the factory default state it can be returned to factory defaults by applying power, holding the pushbutton for 5 seconds and releasing.

2. Use a thirdparty Auracast assistant and receiver to sync and listen to the Auracast broadcast named "Aurawave"

3. Alternatively sync to the Aurawave broadcast via a 2nd Aurawave module

A. Plug 2nd factory default Aurawave module into a usb Host device that supports USB CDC device class and has a serial terminal application installed
B. Plug headphones into the 3.5mm output
C. Connect to the serial port at 115200 baud
D. Execute 'AT+PBSNKDEMO' command, the module will sync to the "Aurawave" broadcast of the first module and play the left channel in both ears

3.2 Auracast Broadcast, analog input, custom broadcast name, persisted configuration

This use case configures an Aurawave module as an Auracast transmitter of audio input to the modules analog line level input. It uses a custom broadcast name and persists the configuration so that the broadcast will startup after a power cycle.

Use case requires:

- Analog Audio Source
- Aurawave module
- USB CDC Host device with serial terminal application (115200 baud)
- 3rd Party Auracast Sink & Auracast Assistant

Configuration steps:

1. Plug an analog audio source into Aurawave module
2. Plug Aurawave module into USB CDC host and connect to Aurawave module's serial port with terminal application at 115200 baud.
3. Select analog as the module's default audio input device

```
AT+AUDDEVSET=1
```

4. Delete any existing Public Broadcast Source (PBSource) instance. Command will return with OK if one existed, error 22 otherwise.

```
AT+PBSRCDEL=0
```

5. Create a new PBSource instance. The index of the instance will be returned.

```
AT+PBSRCCREATE
+PBSRCCREATE: 0
OK
```

6. Configure the Broadcast name of the new PBSource instance.

```
AT+PBSRCCONFIG=0,”NAME”,”Television-1”
+PBSRCCREATE: 0
OK
```

7. Start the PBSource instance. The LED should be solid blue, the broadcast is now active and can be received by an Auracast receiver.

```
AT+PBSRCSTART=0
OK
```

8. Save the Aurawave module's configuration to flash. Saving configuration while the broadcast is active saves the active state of the broadcast and will result in the broadcast restarting automatically after a power cycle.

```
AT+PBSRCSTART=0
OK
```

9. Identify the configured broadcast name and sync to the broadcast using your preferred Auracast assistant and receiver

4. General Commands

4.1 Attention command (AT)

Execute Operation	**AT**
Description	Can be used to confirm the interface is active
Parameters	None
Response	OK
Example	`AT` `OK`
Test Operation	**AT=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`AT=?` `OK`

4.2 Echo Mode enable (ATE1)

Execute Operation	**ATE1**
Description	Enable the interface echo mode. Characters sent to the interface will be echoed back.
Parameters	None
Response	OK
Example	`ATE1` `OK`
Test Operation	**ATE1=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`ATE1=?` `OK`

4.3 Echo Mode disable (ATE0)

Execute Operation	**ATE0**
Description	Disable the interface echo mode. Characters sent to the interface will NOT echo back.
Parameters	None
Response	OK
Example	`ATE0` `OK`
Test Operation	**ATE0=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`ATE0=?` `OK`

4.4 Reboot (AT+REBOOT)

Execute Operation	**AT+REBOOT**
Description	Soft reset of MCU, all current operations are dropped
Parameters	None
Response	None (System Reboot)
Example	`AT+REBOOT`
Test Operation	**AT+REBOOT=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`AT+REBOOT=?` `OK`

4.5 Version (AT+CGMR)

Execute Operation	**AT+CGMR**
Description	Retrieves firmware version information
Parameters	None
Response	+CGMR: Application: "<nearest commit tag>-<number of commits ahead tag>-g<commit id>" [Controller: SoftDevice: Version <version as string> (<version as hex>), Revision <revision>] OK

Example	`AT+CGMR`
	`+CGMR: Application: "1.0.0-c479234d4109" Controller: SoftDevice: Version 6.0 (0x0e), Revision 8424`
	`OK`

Test Operation	**AT+CGMR=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`AT+CGMR=?` `OK`

5. Bluetooth Commands

5.1 Device name (+BTDEVNAME)

SET Operation	**AT+BTDEVNAME=<name>**
Description	Set Bluetooth device name
Parameters	**<name>** Name string value (1-31 bytes)
Response (OK)	OK
Example	AT+BTDEVNAME="Aurawave" OK
READ Operation	**AT+BTDEVNAME?**
Description	Read Bluetooth device name
Response	+BTDEVNAME: <name> OK
Example	AT+BTDEVNAME? +BTDEVNAME: "Aurawave" OK
TEST Operation	**AT+BTDEVNAME=?**
Description	Use to confirm command exists and possible parameter values
Response	+BTDEVNAME=<name>
Example	AT+BTDEVNAME=? +BTDEVNAME=<name> OK

6. LE Audio Commands

6.1 Public broadcast source create (AT+PBSRCCREATE)

Execute Operation	**AT+PBSRCCREATE**
Description	Create an instance of public broadcast source profile role
Parameters	None
Response	+PBSRCCREATE: <PBSourceID> OK
Example	AT+PBSRCCREATE +PBSRCCREATE: 0 OK
Test Operation	**AT+PBSRCCREATE=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	AT+PBSRCCREATE=? OK

6.2 Public broadcast source delete (AT+PBSRCDEL)

Execute Operation	**AT+PBSRCDEL=<PBSourceID>**
Description	Delete an instance of public broadcast source profile role. Stops broadcast if active.
Parameters	<PBSourceID> Public broadcast source context ID.
Response	OK
Example	AT+PBSRCDEL=0 OK
Test Operation	**AT+PBSRCDEL=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	AT+PBSRCDEL=? +PBSRCDEL: <(0)> OK

6.3 Public broadcast source configure (AT+PBSRCCONF)

Execute "Read" Operation	**AT+PBSRCCONF=<PBSourceID>,<ParamTag>**
Description	Reads the current value for a parameter of an existing public broadcast source instance.
Parameters	**<PBSourceID>** Public broadcast source context ID. **<ParamTag>** Tag string associated with parameter to read

ParamTag String	**Description**
"NAME"	Broadcast Name

<table>
<tr><td></td><td>
<table>
<tr><td>“PRESET”</td><td>LC3 Codec & QOS Preset</td></tr>
<tr><td>“CODE”</td><td>Broadcast Code: Set to enable encrypted broadcast.
KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</td></tr>
</table>
</td></tr>
<tr><td>Response</td><td>+ PBSRCCONF: <PBSourceId>,<ParamTag>,<ParamValue>
OK</td></tr>
<tr><td>Example</td><td><pre>AT+PBSRCCONF=0,"NAME"
+PBSRCCONF: 0,"NAME","Aurawave"
OK</pre></td></tr>
<tr><td>Execute “Set” Operation</td><td>AT+PBSRCCONF=<PBSourceID>,<ParamTag>,<ParamValue></td></tr>
<tr><td>Description</td><td>Set a parameter of an existing public broadcast source instance.
NOTE: Broadcasts that have been started can no longer be re-configured, even after stopping them. They must be deleted (AT+PBSRCDEL=) and a new instance created.</td></tr>
<tr><td>Parameters</td><td><PBSourceID> Public broadcast source context ID.
<ParamTag> Tag string associated with parameter to set
<table>
<tr><th>ParamTag String</th><th>Description</th></tr>
<tr><td>“NAME”</td><td>Broadcast Name</td></tr>
<tr><td>“PRESET”</td><td>LC3 Codec & QOS Preset</td></tr>
<tr><td>“CODE”</td><td>Broadcast Code: Set to enable encrypted broadcast.
KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</td></tr>
</table>
<ParamValue> Desired value of parameter indicated by <ParamTag> argument. Type differs based <ParamTag>
<table>
<tr><th>ParamTag String Used</th><th>Acceptable Input</th></tr>
<tr><td>“NAME”</td><td>ASCII String (4-32 bytes)</td></tr>
<tr><td>“PRESET”</td><td>0 - 16kHz Low Latency
1 - 16kHz High Reliability
2 - 48kHz Low Latency
3 – 48kHz High Reliability</td></tr>
<tr><td>“CODE”</td><td>ASCII String (4-16 bytes)</td></tr>
</table>
</td></tr>
<tr><td>Response</td><td>OK</td></tr>
<tr><td>Example</td><td><pre>AT+PBSRCCONF=0,"NAME","TV 1"
OK</pre></td></tr>
<tr><td>Test Operation</td><td>AT+PBSRCCONF=?</td></tr>
<tr><td>Description</td><td>Use to confirm command exists and possible parameter values</td></tr>
<tr><td>Response</td><td>+PBSRCCONF: <(0)>,<ParamTag>,<ParamValue></td></tr>
<tr><td>Example</td><td><pre>AT+PBSRCCONF=?
+PBSRCCONF: <(0)>,<ParamTag>,<ParamValue>
OK</pre></td></tr>
</table>

6.4 Public broadcast source start (AT+PBSRCSTART)

Execute Operation	**AT+PBSRCSTART=<PBSourceID>**
Description	Start LE audio broadcast for a public broadcast source instance
Parameters	<PBSourceID> Public broadcast source context ID.
Response	OK
Example	AT+PBSRCSTART=0 OK
Test Operation	**AT+PBSRCSTART=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	AT+PBSRCSTART=? +PBSRCSTART: <(0)> OK

6.5 Public broadcast source state (AT+PBSRCSTATE)

Execute Operation	**AT+PBSRCSTATE=<PBSourceID>**
Description	Retrieve state of specified instance of public broadcast source.
Parameters	<PBSourceID> Public broadcast source context ID.
Response	OK
Example	AT+PBSRCSTATE=0 +PBSRCSTATE: 0,0,"NOT_STARTED" OK
Test Operation	**AT+PBSRCSTATE=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	AT+PBSRCSTATE=? +PBSRCSTATE: <(0)> OK

6.6 Public broadcast source stop (AT+PBSRCSTOP)

Execute Operation	**AT+PBSRCSTOP=<PBSourceID>**
Description	Stop LE audio broadcast for a public broadcast source instance.
Parameters	<PBSourceID> Public broadcast source context ID.
Response	OK
Example	AT+PBSRCSTOP=0 OK
Test Operation	**AT+PBSRCSTOP=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	AT+PBSRCSTOP=? +PBSRCSTOP: <(0)> OK

6.7 Public broadcast sink demo (AT+PBSNKDEMO)

Execute Operation	**AT+PBSNKDEMO[=<ChannelLocation>]**
Description	Create, configure, and enable public broadcast sink demo. The AW100 module will look for Auracasts with Broadcast Names set to “Aurawave” and attempt to sync. If no argument is provided, default to LEFT channel. Supports only 48khz Codec presets and Analog audio out. Note: a system reboot must be executed to return to normal operation after executing this command.
Parameters	**<ChannelLocation>** Channel of remote Aurawave Auracast stream to render locally.

Location String	**Description**
“LEFT”	Left channel location
“RIGHT”	Right channel location

Response	OK
Example	AT+PBSNKDEMO="RIGHT" OK
Test Operation	**AT+PBSNKDEMO=?**
Description	Use to confirm command exists and possible parameter values
Response	+PBSNKDEMO: ["LEFT"|"RIGHT"]
Example	AT+PBSNKDEMO=? +PBSNKDEMO: ["LEFT"|"RIGHT"] OK

7. Audio System Commands

7.1 Audio devices list (AT+AUDDEVLIST)

Execute Operation	**AT+AUDDEVLIST**
Description	List audio devices supported by AW100 series module
Parameters	None
Response	+AUDDEVLIST: <ListLength> 0,<Device0> 1,<Device1> ... OK
Example	AT+AUDDEVLIST +AUDDEVLIST: 2 0,USB 1,ANALOG OK
Test Operation	**AT+AUDDEVLIST=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	AT+AUDDEVLIST=? OK

7.2 Audio device set (AT+AUDDEVSET)

Execute Operation	**AT+AUDDEVSET=<AudioDeviceId>**
Description	Set default input audio device used by Public Broadcast Source instances. For retrieving a list of supported audio devices refer to AUDDEVLIST.
Parameters	**<AudioDeviceId>** ID of device to set as default audio device for broadcasts. Supported values can be found via AT+AUDDEVLIST command.
Response	OK
Example	AT+AUDDEVSET=0 OK
Test Operation	**AT+AUDDEVSET=?**
Description	Use to confirm command exists and possible parameter values
Response	+AUDDEVSET: <(0-1)>
Example	AT+AUDDEVSET=? +AUDDEVSET: <(0-1)> OK

8. System configuration Commands

8.1 Configuration save (AT+CONFSAVE)

Execute Operation	**AT+CONFSAVE**
Description	Save current system configuration, including: • Bluetooth Device Name • Public Broadcast Source Instances or lack thereof and their configuration • Broadcast state • Default Audio Input Note: In this release, if a public broadcast source is active when AT+CONFSAVE is executed then the broadcast will automatically start on system startup.
Parameters	None
Response	OK
Example	`AT+CONFSAVE` `OK`
Test Operation	**AT+CONFSAVE=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`AT+CONFSAVE=?` `OK`

8.2 Configuration defaults (AT+CONFDEF)

Execute Operation	**AT+CONFDEF**
Description	Reset system configuration to defaults. Executing AT+CONFSAVE is still required to persist defaults.
Parameters	None
Response	OK
Example	`AT+CONFDEF` `OK`
Test Operation	**AT+CONFDEF=?**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`AT+CONFDEF=?` `OK`

9. Firmware update Commands

9.1 Enter DFU Mode (AT+DFU)

Execute Operation	**AT+DFU**
Description	Enter firmware DFU mode. Refer to AW100 DFU Guide for firmware upgrade procedure.
Parameters	None
Response	None (Reboot into DFU mode)
Example	`AT+DFU` `OK`
Test Operation	**AT+DFU**
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	`AT+DFU=?` `OK`

10. Supported Result Codes

9.1 Summary of CME ERROR Codes

A result code of "+CME ERROR: <err>" indicates an error.
The following table lists the possible error codes returned by Aurawave AT commands.

Value	Error Code	Context	AT Commands
12	ENOMEM	• Not enough memory to complete action • No more contexts supported	AT+PBSRCCREATE AT+BTDEVNAME
22	EINVAL	Invalid argument/parameter validation failed	AT+PBSRCCONF AT+BTDEVNAME AT+PBSRCDEL
37	ECHRNG	Parameter out of range/index validation	AT+AUDDEVSET AT+PBSRCDEL AT+PBSRC*
120	EALREADY	Operation already in progress/state conflict	AT+AUDDEVSET AT+PBSRCSTART AT+PBSRCSTOP

9.2 Summary of Unsolicited Result Codes

As of firmware version 1.0, no unsolicited result codes are indicated

Additional Information - Ezurio

Please contact your local sales representative or our support team for further assistance:

Headquarters	Ezurio 50 S. Main St. Suite 1100 Akron, OH 44308 USA
Website	http://www.ezurio.com
Technical Support	http://www.ezurio.com/resources/support
Sales Contact	http://www.ezurio.com/contact

Note: Information contained in this document is subject to change.

Additional Information – Cloud2GND

Headquarters	Cloud2GND 4500 Bowling Blvd #100 Louisville, KY 40207
Website	http://www.cloud2gnd.com/
Technical Support	https://cloud2gnd.com/contact/

Ezurio's products are subject to standard Terms & Conditions.