

# User Guide – AT Command Specification

Aurawave AW100 Series

*Version 1.1*

---

## Revision History

Version	Date	Notes
1.0	7 Aug 2025	Initial release.
1.0-r2	5 Sep 2025	Add examples Fix error codes
1.0-r3	24 Oct 2025	Fix incorrect command in 3.2 step 8
1.0-r4	11 Feb 2026	Update AT+PBSNKDEMO Add Sink Use-case Add clarifications regarding the line endings on the serial interface
1.1-r1	23 Feb 2026	Update AT+PBSRCCONF Update AT+DFU Add AT+PBGRPCONF Add AT+PBSTRMCONF Add AT+USBID
1.1-r2	04 Mar 2026	Add small clarifications throughout the document Add the complete settings for the serial terminal Improve the instructions for running the Broadcast Sink Demo Improve the AT+PBSINKDEMO command documentation Add the I2S settings to be used for the 40-pin header ANLOG input Add notes about the AT+CONFDEF command resetting the board
1.1-r3	11 Mar 2026	Add small clarifications. Add notes about the Broadcast Sink Demo only working on boards with analog outputs

# Contents

1.	Introduction	5
1.1	Scope of the document	5
1.2	Conventions and abbreviations	5
2.	AT Command syntax	6
2.1	Set Operation <CMD>=[...]	6
2.2	Read Operation <CMD>?	7
2.3	Test Operation <CMD>=?	7
2.4	Execute Operation <CMD>[=...]	7
2.5	Responses	7
2.6	Entering successive AT commands	8
2.7	Supported character set	8
2.8	Supported line-endings	8
3.	Use case Examples	9
3.1	Default/Factory Auracast broadcast, Stereo, USB audio input)	9
3.2	Auracast Broadcast, analog input, custom broadcast name, persisted configuration	9
3.3	Broadcast source to sink demo	10
4.	General Commands	12
4.1	Attention command (AT)	12
4.2	Echo Mode enable (ATE1)	12
4.3	Echo Mode disable (ATE0)	13
4.4	Reboot (AT+REBOOT)	13
4.5	Version (AT+CGMR)	14
5.	Bluetooth Commands	15
5.1	Device name (+BTDEVNAME)	15
6.	LE Audio Commands	16
6.1	Public broadcast source create (AT+PBSRCCREATE)	16
6.2	Public broadcast source delete (AT+PBSRCDEL)	16
6.3	Public broadcast source configure (AT+PBSRCCONF)	16
6.4	Public broadcast sub-group configure (AT+PBGRPCONF)	18
6.5	Public broadcast stream configure (AT+PBSTRMCONF)	19
6.6	Public broadcast source start (AT+PBSRCSTART)	20
6.7	Public broadcast source state (AT+PBSRCSTATE)	20
6.8	Public broadcast source stop (AT+PBSRCSTOP)	21
6.9	Public broadcast sink demo (AT+PBSNKDEMO)	21
7.	Audio System Commands	22
7.1	Audio devices list (AT+AUDDEVLIST)	22

7.2	Audio device set (AT+AUDDEVSET)	22
8.	System configuration Commands	23
8.1	Configuration save (AT+CONFSAVE)	23
8.2	Configuration defaults (AT+CONFDEF)	23
9.	USB Configuration Commands	24
9.1	USB VID/PID (AT+USBID)	24
10.	Firmware update Commands	25
10.1	Enter DFU Mode (AT+DFU)	25
11.	Supported Result Codes	26
11.1	Summary of CME ERROR Codes	26
11.2	Summary of Unsolicited Result Codes	26
	Additional Information - Ezurio	27
	Additional Information - Cloud2GND	27

# 1. Introduction

This document applies to the following Aurawave software versions:

- 1.1

## 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.

## 2.8 Supported line-endings

Only CR or CRLF are supported.

## 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 third-party Auracast devices or a 2<sup>nd</sup> 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
- 3<sup>rd</sup> Party Auracast Sink & Auracast Assistant or a 2<sup>nd</sup> Aurawave module + headphones with 3.5mm plug
  - Tested third party devices include
    - Samsung Galaxy Buds 3 Pro
    - Samsung S23
    - HomeSpot 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 third-party Auracast assistant and receiver to sync and listen to the Auracast broadcast named "Aurawave"
3. Alternatively sync to the Aurawave broadcast via a 2<sup>nd</sup> Aurawave module
  - A. Plug 2<sup>nd</sup> 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)
- 3<sup>rd</sup> 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+AUDEVSET=1
```

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

```
AT+PBRCDL=0
```

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

```
AT+PBRCCREATE
+PBRCCREATE: 0
OK
```

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

```
AT+PBRCCONFIG=0,"NAME","Television-1"
+PBRCCREATE: 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+PBRCCSTART=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+CONFSAVE
OK
```

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

### 3.3 Broadcast source to sink demo

This use case configures TWO Aurawave modules, first as a public broadcast source (transmitter), and second as public broadcast sink (receiver) of audio. The sink Aurawave will look for a broadcast source named "Aurawave" (the default stream name) to sync with and pass the received audio to the analog output.

The source configuration for an AW100 module 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 (analog input also supported)
- *This is the factory default configuration.*

The sink configuration for an AW100 module is as follows:

- 1 Public Broadcast Sink instance.
- 48khz 'low-latency' audio
- Analog Audio Output

**Use case requires:**

- TWO Aurawave modules.
- USB CDC Host device with serial terminal application
- USB Audio Source (analog input also supported)
- Analog Audio Output.

**Terminal application configuration:**

- 115200 bps, 8, N, 1

**Configuration steps:**

- For the TWO Aurawave modules (if they are not already in default factory state):
  1. Plug the Aurawave module into USB CDC host and connect to Aurawave module's serial port with terminal application.

2. Restore default configuration using the AT+CONFDEF command. This command will reset the board, and you will need to manually reconnect to the COM port if your terminal application is not configured to automatically reconnect.

```
AT+CONFDEF
OK
```

3. Save configuration

```
AT+CONFSAVE
OK
```

- Broadcast source 1<sup>st</sup> Aurawave:
  1. Plug the USB audio source into the module's USB port.
  2. Start the USB audio source.

- Broadcast sink 2<sup>nd</sup> Aurawave:

1. Start sink demo

```
AT+PBSNKDEMO
OK
```

2. **(Optional)** Save configuration to persist the broadcast sink demo during reboots

```
AT+CONFSAVE
OK
```

**NOTE:** Switching back to the Broadcast Source role requires returning to factory defaults by long pressing the button on the board or by sending the AT+CONFDEF command (which will reset the board)

**NOTE:** The Broadcast Sink Demo only works on AW100 variants which have an analog output (through an onboard codec chip)

3. Plug the analog audio output into Aurawave module analog OUT.
4. Now you should be hearing audio being broadcasted from the broadcast source 1<sup>st</sup> Aurawave coming out from the 2<sup>nd</sup> Aurawave analog audio output.

## 4. General Commands

### 4.1 Attention command (AT)

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

### 4.2 Echo Mode enable (ATE1)

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

### 4.3 Echo Mode disable (ATE0)

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

### 4.4 Reboot (AT+REBOOT)

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

## 4.5 Version (AT+CGMR)

<b>Execute Operation</b>	<b>AT+CGMR</b>
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	<pre>AT+CGMR +CGMR: Application: "1.0.0-c479234d4109" Controller: SoftDevice: Version 6.0 (0x0e), Revision 8424 OK</pre>
<b>Test Operation</b>	<b>AT+CGMR=?</b>
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	<pre>AT+CGMR=? OK</pre>

## 5. Bluetooth Commands

### 5.1 Device name (+BTDEVNAME)

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

## 6. LE Audio Commands

### 6.1 Public broadcast source create (AT+PBSRCCREATE)

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

### 6.2 Public broadcast source delete (AT+PBSRCDEL)

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

### 6.3 Public broadcast source configure (AT+PBSRCCONF)

<b>Execute "Read" Operation</b>	<b>AT+PBSRCCONF= &lt;PBSourceID&gt;, &lt;ParamTag&gt;</b>				
Description	Reads the current value for a parameter of an existing public broadcast source instance.				
Parameters	<p>&lt;PBSourceID&gt; Public broadcast source context ID.</p> <p>&lt;ParamTag&gt; Tag string associated with parameter to read</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">ParamTag String</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">"NAME"</td> <td>Broadcast Name</td> </tr> </tbody> </table>	ParamTag String	Description	"NAME"	Broadcast Name
ParamTag String	Description				
"NAME"	Broadcast Name				

	<table border="1"> <tr> <td>"PRESET"</td> <td>LC3 Codec &amp; QOS Preset</td> </tr> <tr> <td>"CODE"</td> <td>Broadcast Code: Set to enable encrypted broadcast. <b>KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</b></td> </tr> <tr> <td>"SUBGROUPS"</td> <td>Number of associated subgroups</td> </tr> </table>	"PRESET"	LC3 Codec & QOS Preset	"CODE"	Broadcast Code: Set to enable encrypted broadcast. <b>KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</b>	"SUBGROUPS"	Number of associated subgroups														
"PRESET"	LC3 Codec & QOS Preset																				
"CODE"	Broadcast Code: Set to enable encrypted broadcast. <b>KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</b>																				
"SUBGROUPS"	Number of associated subgroups																				
Response	+PBSRCCONF: <PBSourceId>, <ParamTag>, <ParamValue> OK																				
Example	<pre>AT+PBSRCCONF=0, "NAME" +PBSRCCONF: 0, "NAME", "Aurawave" OK</pre>																				
<b>Execute "Set" Operation</b>	<b>AT+PBSRCCONF=&lt;PBSourceID&gt;, &lt;ParamTag&gt;, &lt;ParamValue&gt;</b>																				
Description	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.																				
Parameters	<p>&lt;PBSourceID&gt; Public broadcast source context ID.</p> <p>&lt;ParamTag&gt; Tag string associated with parameter to set</p> <table border="1"> <thead> <tr> <th>ParamTag String</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>"NAME"</td> <td>Broadcast Name</td> </tr> <tr> <td>"PRESET"</td> <td>LC3 Codec &amp; QOS Preset</td> </tr> <tr> <td>"CODE"</td> <td>Broadcast Code: Set to enable encrypted broadcast. <b>KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</b></td> </tr> <tr> <td>"SUBGROUPS"</td> <td>Count of associated subgroups</td> </tr> </tbody> </table> <p>&lt;ParamValue&gt; Desired value of parameter indicated by &lt;ParamTag&gt; argument. Type differs based &lt;ParamTag&gt;</p> <table border="1"> <thead> <tr> <th>ParamTag String Used</th> <th>Acceptable Input</th> </tr> </thead> <tbody> <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> <tr> <td>"SUBGROUPS"</td> <td>Integer value starting from 1 or greater</td> </tr> </tbody> </table>	ParamTag String	Description	"NAME"	Broadcast Name	"PRESET"	LC3 Codec & QOS Preset	"CODE"	Broadcast Code: Set to enable encrypted broadcast. <b>KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</b>	"SUBGROUPS"	Count of associated subgroups	ParamTag String Used	Acceptable Input	"NAME"	ASCII String (4-32 bytes)	"PRESET"	0 - 16kHz Low Latency 1 - 16kHz High Reliability 2 - 48kHz Low Latency 3 - 48kHz High Reliability	"CODE"	ASCII String (4-16 bytes)	"SUBGROUPS"	Integer value starting from 1 or greater
ParamTag String	Description																				
"NAME"	Broadcast Name																				
"PRESET"	LC3 Codec & QOS Preset																				
"CODE"	Broadcast Code: Set to enable encrypted broadcast. <b>KNOWN ISSUE: Setting CODE to less than 4 characters factory defaults the system</b>																				
"SUBGROUPS"	Count of associated subgroups																				
ParamTag String Used	Acceptable Input																				
"NAME"	ASCII String (4-32 bytes)																				
"PRESET"	0 - 16kHz Low Latency 1 - 16kHz High Reliability 2 - 48kHz Low Latency 3 - 48kHz High Reliability																				
"CODE"	ASCII String (4-16 bytes)																				
"SUBGROUPS"	Integer value starting from 1 or greater																				
Response	OK																				
Example	<pre>AT+PBSRCCONF=0, "NAME", "TV 1" OK</pre>																				
<b>Test Operation</b>	<b>AT+PBSRCCONF=?</b>																				
Description	Use to confirm command exists and possible parameter values																				
Response	+PBSRCCONF: <(0)>, <ParamTag>, <ParamValue>																				
Example	<pre>AT+PBSRCCONF=? +PBSRCCONF: &lt;(0)&gt;, &lt;ParamTag&gt;, &lt;ParamValue&gt;</pre>																				

	OK
--	----

## 6.4 Public broadcast sub-group configure (AT+PBGRPCONF)

<b>Execute "Read" Operation</b>	<b>AT+PBGRPCONF= &lt;PBSourceID&gt;, &lt;PBSubGroupID&gt;, &lt;ParamTag&gt;</b>						
Description	Reads the current value for a parameter of an existing public broadcast source's subgroup instance.						
Parameters	<PBSourceID> Public broadcast source context ID. <PBSubGroupID> Public broadcast source subgroup context ID. <ParamTag> Tag string associated with parameter to read						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">ParamTag String</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>"LANG"</td> <td>Subgroup associated language</td> </tr> <tr> <td>"STREAMS"</td> <td>Subgroup associated number of streams</td> </tr> </tbody> </table>	ParamTag String	Description	"LANG"	Subgroup associated language	"STREAMS"	Subgroup associated number of streams
	ParamTag String	Description					
	"LANG"	Subgroup associated language					
"STREAMS"	Subgroup associated number of streams						
Response	+PBGRPCONF: <PBSourceID>, <PBSubGroupID>, <ParamTag>, <ParamValue> OK						
Example	<pre>AT+PBGRPCONF=0,0,"LANG" +PBGRPCONF: 0,0,"LANG","eng" OK</pre>						
<b>Execute "Set" Operation</b>	<b>AT+PBGRPCONF= &lt;PBSourceID&gt;, &lt;PBSubGroupID&gt;, &lt;ParamTag&gt;, &lt;ParamValue&gt;</b>						
Description	Set a parameter of an existing public broadcast source's subgroup 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.						
Parameters	<PBSourceID> Public broadcast source context ID. <PBSubGroupID> Public broadcast source subgroup context ID. <ParamTag> Tag string associated with parameter to set						
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">ParamTag String</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>"LANG"</td> <td>Subgroup associated language</td> </tr> <tr> <td>"STREAMS"</td> <td>Subgroup associated number of streams</td> </tr> </tbody> </table>	ParamTag String	Description	"LANG"	Subgroup associated language	"STREAMS"	Subgroup associated number of streams
	ParamTag String	Description					
	"LANG"	Subgroup associated language					
	"STREAMS"	Subgroup associated number of streams					
Response	OK						
Example	<pre>AT+PBGRPCONF=0,0,"LANG","eng" OK</pre>						
<b>Test Operation</b>	<b>AT+PBGRPCONF=?</b>						
Description	Use to confirm command exists and possible parameter values						
Response	+PBGRPCONF: <(0)>, <(0)>, <ParamTag>, <ParamValue>						

Example	<pre>AT+PBGRPCONF=? +PBGRPCONF: &lt;(0)&gt;,&lt;(0)&gt;,&lt;ParamTag&gt;,&lt;ParamValue&gt; OK</pre>
---------	--

## 6.5 Public broadcast stream configure (AT+PBSTRMCONF)

<b>Execute "Read" Operation</b>	<b>AT+PBSTRMCONF= &lt;PBSourceID&gt; , &lt;PBSubGroupID&gt; , &lt;PBStreamID&gt; , &lt;ParamTag&gt;</b>	
Description	Reads the current value for a parameter of an existing public broadcast source's stream instance.	
Parameters	<PBSourceID> Public broadcast source context ID. <PBSubGroupID> Public broadcast source subgroup context ID. <PBStreamID> Public broadcast source stream context ID. <ParamTag> Tag string associated with parameter to read	
	<b>ParamTag String</b>	<b>Description</b>
	"LOC"	Stream associated channel allocation
Response	+PBSTRMCONF: <PBSourceID> , <PBSubGroupID> , <PBStreamID> , <ParamTag> , <ParamValue> OK	
Example	<pre>AT+PBSTRMCONF=0,0,0,"LOC" +PBSTRMCONF: 0,0,0,"LOC",0 OK</pre>	
<b>Execute "Set" Operation</b>	<b>AT+PBSTRMCONF= &lt;PBSourceID&gt; , &lt;PBSubGroupID&gt; , &lt;PBStreamID&gt; , &lt;ParamTag&gt; , &lt;ParamValue&gt;</b>	
Description	Set a parameter of an existing public broadcast source's stream 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.	
Parameters	<PBSourceID> Public broadcast source context ID. <PBSubGroupID> Public broadcast source subgroup context ID. <PBStreamID> Public broadcast source stream context ID. <ParamTag> Tag string associated with parameter to set	
	<b>ParamTag String</b>	<b>Description</b>
	"LOC"	Stream associated channel allocation
Parameters	<ParamValue> Desired value of parameter indicated by <ParamTag> argument. Type differs based <ParamTag>	
	<b>ParamTag String Used</b>	<b>Acceptable Input</b>
	"LOC"	Bit field value <i>as mentioned in BLE Assigned Numbers: 6.12.1 Audio Location Definitions</i>
Response	OK	
Example	<pre>AT+PBSTRMCONF=0,0,0,"LOC",0 OK</pre>	
<b>Test Operation</b>	<b>AT+PBSTRMCONF=?</b>	
Description	Use to confirm command exists and possible parameter values	
Response	+PBSTRMCONF: <(0)> , <(0)> , <(0-1)> , <ParamTag> , <ParamValue>	
Example	<pre>AT+PBSTRMCONF=? +PBSTRMCONF: &lt;(0)&gt; , &lt;(0)&gt; , &lt;(0-1)&gt; , &lt;ParamTag&gt; , &lt;ParamValue&gt;</pre>	

	OK
--	----

## 6.6 Public broadcast source start (AT+PBSRCSTART)

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

## 6.7 Public broadcast source state (AT+PBSRCSTATE)

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

## 6.8 Public broadcast source stop (AT+PBSRCSTOP)

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

## 6.9 Public broadcast sink demo (AT+PBSNKDEMO)

<b>Execute Operation</b>	<b>AT+PBSNKDEMO[= &lt;ChannelLocation&gt;]</b>						
Description	<p>Create, configure, and enable public broadcast sink demo.</p> <p>The AW100 module will look for Auracasts with Broadcast Names set to "Aurawave" and attempt to sync.</p> <p>If no argument is provided, default to LEFT channel.</p> <p>Supports only 48khz Codec presets and Analog audio out.</p> <p>Example: <a href="#">See section: 3.3 Broadcast source to sink demo</a></p> <p><b>NOTE: Switching back to default (broadcast source) operation requires restoring factory defaults.</b></p> <p><b>NOTE: This command only works on AW100 variants which have an analog output (through an onboard codec chip)</b></p>						
Parameters	<p>&lt;ChannelLocation&gt; Channel of remote Aurawave Auracast stream to render locally.</p> <table border="1"> <thead> <tr> <th>Location String</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>"LEFT"</td> <td>Left channel location</td> </tr> <tr> <td>"RIGHT"</td> <td>Right channel location</td> </tr> </tbody> </table>	Location String	Description	"LEFT"	Left channel location	"RIGHT"	Right channel location
Location String	Description						
"LEFT"	Left channel location						
"RIGHT"	Right channel location						
Response	OK						
Example	<pre>AT+PBSNKDEMO="RIGHT" OK</pre>						
<b>Test Operation</b>	<b>AT+PBSNKDEMO=?</b>						
Description	Use to confirm command exists and possible parameter values						
Response	+PBSNKDEMO: ["LEFT" "RIGHT"]						
Example	<pre>AT+PBSNKDEMO=? +PBSNKDEMO: ["LEFT" "RIGHT"] OK</pre>						

## 7. Audio System Commands

### 7.1 Audio devices list (AT+AUDDEVLIST)

<b>Execute Operation</b>	<b>AT+AUDDEVLIST</b>
Description	List audio devices supported by AW100 series module Note: The configuration of the ANALOG audio device, I2S input on the 40 pin header, is: stereo, 48 KHz sampling rate, 32-bit bit-depth, master mode.
Parameters	None
Response	+AUDDEVLIST: <ListLength> 0,<Device0> 1,<Device1> ... OK
Example	<pre>AT+AUDDEVLIST +AUDDEVLIST: 2 0,USB 1,ANALOG OK</pre>
<b>Test Operation</b>	<b>AT+AUDDEVLIST=?</b>
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	<pre>AT+AUDDEVLIST=? OK</pre>

### 7.2 Audio device set (AT+AUDDEVSET)

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

## 8. System configuration Commands

### 8.1 Configuration save (AT+CONFSAVE)

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

### 8.2 Configuration defaults (AT+CONFDEF)

Execute Operation	AT+CONFDEF
Description	<p>Reset system configuration to defaults. Executing AT+CONFSAVE is still required to persist defaults.</p> <p><b>Note:</b> The AT+CONFDEF command will reset the board, and you will need to manually reconnect to the COM port if your terminal application is not configured to automatically reconnect.</p>
Parameters	None
Response	OK
Example	<pre>AT+CONFDEF OK</pre>
Test Operation	AT+CONFDEF=?
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	<pre>AT+CONFDEF=? OK</pre>

## 9. USB Configuration Commands

### 9.1 USB VID/PID (AT+USBID)

<b>SET Operation</b>	<b>AT+USBID=&lt;VendorID&gt;,&lt;ProductID&gt;</b>
Description	Set USB Vendor and Product IDs
Parameters	<VendorID> USB Vendor ID in HEX (2-bytes) <ProductID> USB Product ID in HEX (2-bytes)
Response (OK)	OK
Example	<pre>AT+USBID=1915,530A OK</pre>
<b>READ Operation</b>	<b>AT+USBID?</b>
Description	Read USB Vendor and Product IDs
Response	+USBID: <VID>,<PID> OK
Example	<pre>AT+USBID? +USBID: "0x1915","0x530A" OK</pre>
<b>TEST Operation</b>	<b>AT+USBID=?</b>
Description	Use to confirm command exists and possible parameter values
Response	+ USBID: <VID>,<PID>
Example	<pre>AT+USBID=? +USBID= &lt;Vendor-ID&gt;,&lt;Product-ID&gt; OK</pre>

## 10. Firmware update Commands

### 10.1 Enter DFU Mode (AT+DFU)

<b>Execute Operation</b>	<b>AT+DFU</b>
Description	<p>Enter firmware DFU mode.</p> <p>Refer to AW100 DFU Guide for firmware upgrade procedure.</p> <p><i>Note: This command is not supported by boards shipped with Release 1.0 as it needs the bootloader to be updated.</i></p>
Parameters	None
Response	None (Reboot into DFU mode)
Example	<pre>AT+DFU OK</pre>
<b>Test Operation</b>	<b>AT+DFU</b>
Description	Use to confirm command exists and possible parameter values
Response	OK
Example	<pre>AT+DFU=? OK</pre>

## 11. Supported Result Codes

### 11.1 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	<ul style="list-style-type: none"> <li>Not enough memory to complete action</li> <li>No more contexts supported</li> </ul>	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

### 11.2 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:

---

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

---

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

---

## Additional Information – Cloud2GND

---

<b>Headquarters</b>	Cloud2GND 4500 Bowling Blvd #100 Louisville, KY 40207
<b>Website</b>	<a href="http://www.cloud2gnd.com/">http://www.cloud2gnd.com/</a>
<b>Technical Support</b>	<a href="https://cloud2gnd.com/contact/">https://cloud2gnd.com/contact/</a>

---

Ezurio's products are subject to standard [Terms & Conditions](#).

---

© Copyright 2025 Ezurio All Rights Reserved. Any information furnished by Ezurio and its agents is believed to be accurate but cannot be guaranteed. All specifications are subject to change without notice. Responsibility for the use and application of Ezurio materials or products rests with the end user since Ezurio and its agents cannot be aware of all potential uses. Ezurio makes no warranties as to non-infringement nor as to the fitness, merchantability, or sustainability of any Ezurio materials or products for any specific or general uses. Ezurio or any of its affiliates or agents shall not be liable for incidental or consequential damages of any kind. All Ezurio products are sold pursuant to the Ezurio Terms and Conditions of Sale in effect from time to time, a copy of which will be furnished upon request. Nothing herein provides a license under any Ezurio or any third-party intellectual property right. Ezurio and its associated logos are trademarks owned by Ezurio and/or its affiliates.

---