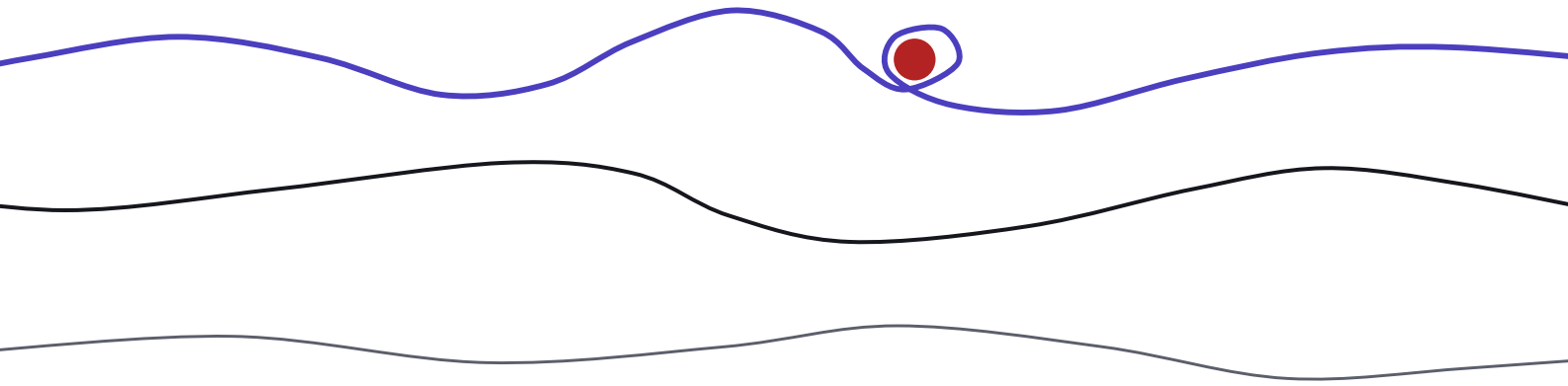
jupiter.au

LG WebOS TV (bscpylgtv)

Home Assistant integration for LG webOS TVs, built on the bscpylgtv library with one push websocket per TV.



VERSION

v2.1.1

DATE

25 November 2026

AUTHOR

belikh

SOURCE

github.com/belikh/ha-lg-webos-tv

Contents

00.1 What you get 2
00.2 Where to start 2

Start

01 Install 3
01.1 Install with HACS 3
01.2 Install by hand 3

02 Set up a TV 4
02.1 Add a TV 4
02.2 Re-pair after a lost key 4
02.3 Change the host 4
02.4 Setup messages 4

Guides

03 Turn the TV on 6
03.1 Check that turn on is offered 6
03.2 Enter the MAC address by hand 6
03.3 Wi-Fi connections 6
03.4 Wake timing 7
03.5 Turn on from an automation 7

04 Launch apps and tune channels 7
04.1 Launch an app 7
04.2 Tune a channel 7
04.3 Pick a source 8
04.4 Open a URL 8

05 Control soundbar volume 8
05.1 Switch the sound output 8
05.2 Get the volume slider back 9
05.3 What to expect 9
05.4 Tune the press interval 9

06 Take a screenshot 9
06.1 Save to a file 9
06.2 Get the image in the response 10

06.3 Errors 10

07 Troubleshoot 10

07.1 Pairing fails or loops 10
07.2 webOS 26 rejects pairing, or a working setup stopped after a firmware update 11
07.3 The TV shows unavailable 11
07.4 The connection went stale after a TV restart or network change 11
07.5 Picture mode shows unknown 11
07.6 Sharpness and color temperature show unknown 11
07.7 The channel select is empty 11
07.8 The channel select is very long 11
07.9 TPC or GSR show unknown 12
07.10 Service errors 12

08 Report a problem 12

08.1 Collect the debug log 12
08.2 Download diagnostics 13

09 Remove the integration 13

Reference

10 Entities 14

10.1 Media player behavior 15
10.2 Remote behavior 16
10.3 Notify behavior 16

11 Services 16

11.1 bscpylgtv.button 17
11.2 bscpylgtv.command 17
11.3 bscpylgtv.select_sound_output 17
11.4 bscpylgtv.launch_app 17
11.5 bscpylgtv.take_screenshot 17
11.6 bscpylgtv.set_settings 18
11.7 Standard services 18

12 Remote commands 18

12.1 Aliases 18
12.2 Common buttons 19
12.3 Pointer and text commands 19

13 Options 19

14 Sound outputs 20

15 Glossary 21

Concepts

16 How it works 22

16.1 Connection and subscriptions 22

16.2 Watchdog 22

16.3 Pairing 22

16.4 Power and wake 23

16.5 Entity availability 23

17 Why external outputs have no volume level 23

17.1 The experimental mode 23

18 Limitations 23

Develop

19 Build and test 25

19.1 Layout 25

20 Release 26

LG WebOS TV (bscpylgtv)

The integration connects Home Assistant to an LG TV that runs webOS. It uses the bscpylgtv library by chros73 and keeps one websocket open to the TV, so state changes arrive as they happen.

Its design follows the built-in Home Assistant `webostv` integration. It adds picture calibration sliders, picture mode, sound output and channel selects, power-state and channel sensors, a remote with pointer and text input, screen notifications and raw *SSAP* commands.

[image: Add to Home Assistant]

00.1

What you get

- Setup from the UI with SSDP discovery and a pairing prompt on the TV.
- A supervised connection that reconnects after TV restarts and network drops.
- Turn on through *Wake-on-LAN*, with the MAC address detected when the TV reports it.
- Reauthentication when the *Pairing key* is lost, and reconfigure when the host changes.
- Entities on 8 platforms: media player, remote, notify, button, number, select, sensor and switch.
- Six services, including raw SSAP commands and screenshots.
- A diagnostics download with the pairing key, host, MAC address and unique ID redacted.

00.2

Where to start

SEE ALSO

- Install
- Set up a TV
- Entities
- Services
- Troubleshoot

SECTION

Start

Install the integration and pair your TV.

01

Install

Add the integration to Home Assistant with HACS or by copying one folder.

Install with HACS, then restart Home Assistant.

01.1

Install with HACS

[image: Add to Home Assistant]

1. Select the badge above and choose **Download**. You can also search for **LG WebOS TV (bscpylgtv)** in HACS.
2. Restart Home Assistant.

If the search finds nothing, add the repository by hand:

1. In HACS, open the three-dot menu at the top right and choose **Custom repositories**.
2. Enter github.com/belikh/ha-lg-webos-tv with category **Integration**.
3. Install the integration and restart Home Assistant.

01.2

Install by hand

1. Copy the `custom_components/bscpylgtv` folder from the repository into the `custom_components` directory of your Home Assistant configuration.
2. Restart Home Assistant.

RESULTING LAYOUT

```
<config>/custom_components/bscpylgtv/manifest.json
```

SEE ALSO

- Set up a TV

02

Set up a TV

Add a TV to Home Assistant, accept the pairing prompt and recover from a lost key or a new IP address.

Add the TV from **Settings > Devices & Services**. The TV must be on and on the same network as Home Assistant.

02.1

Add a TV

[image: Add integration]

1. Select the badge above, or go to **Settings > Devices & Services > Add Integration** and search for **LG WebOS TV (bscpylgtv)**. Home Assistant often lists the TV as a discovered device already.
2. Enter the hostname or IP address of the TV and select **Submit**.
3. Accept the pairing prompt on the TV screen, then select **Submit** in Home Assistant.

The TV appears as a device with its entities. See Entities for the list and which entities start disabled.

PIN pairing is not supported. Pairing always uses the prompt on the TV.

02.2

Re-pair after a lost key

If the TV stops accepting the stored *Pairing key*, Home Assistant starts reauthentication. Open the notification, select **Submit** and accept the new prompt on the TV.

02.3

Change the host

If the IP address of the TV changes, discovery updates the entry. To change it by hand, open the config entry, choose the three-dot menu and select **Reconfigure**. Enter the new host. The flow aborts with `wrong_device` when the TV at that address is not the one that was configured.

02.4

Setup messages

MESSAGE	CAUSE	FIX
`Failed to connect, please turn on your TV and try again.`	The TV is off or unreachable.	Turn the TV on and check that Home Assistant can reach its address.
`Pairing failed, make sure to accept the pairing request on the TV and try again.`	The prompt was declined or timed out.	Submit again and accept the prompt on the TV in time.
`Device is already configured`	The TV has an entry already.	Open the existing entry instead.

SEE ALSO

- Turn the TV on
- Options
- Troubleshoot

SECTION

Guides

Common tasks, as numbered steps.

03

Turn the TV on

Wake a TV with Wake-on-LAN and set the MAC address when it was not detected.

`turn_on` sends a *Wake-on-LAN* magic packet (UDP, port 9) to the MAC address of the TV. The websocket is closed while the TV is off, so the packet is the wake method the integration uses.

03.1

Check that turn on is offered

The media player and the remote offer `turn_on` only when a MAC address is stored. The integration detects it when the TV reports it during pairing or on reconnect.

- With a MAC address, entities show `off` while the TV is off.
- Without one, entities show `unavailable` and there is no `turn_on`. This happens when the TV was off or unreachable during setup, or when it does not report its MAC. Some webOS 26 sets do not report it in *Compatibility mode*.

03.2

Enter the MAC address by hand

1. Find the address in the network settings of the TV or in your router. Use the format `aa:bb:cc:dd:ee:ff`.
2. Open **Settings > Devices & Services > LG WebOS TV (bscpylgtv)** and choose **Configure**.
3. Enter the address in **MAC address** and submit.

An invalid value fails with `Invalid MAC address. Use the format aa:bb:cc:dd:ee:ff.` A stored or hand-entered address keeps working.

03.3

Wi-Fi connections

Wake-on-LAN reaches a TV on Ethernet without a setting change. On Wi-Fi, the TV needs one of these settings enabled, or it may not wake from a full power-off:

- **General > Quick Start+**
- **Turn On by Wi-Fi**

The menu names differ by model and firmware. A TV on Wi-Fi may also take longer to wake than one on Ethernet.

03.4 Wake timing

Timing varies by set. On an OLED48CXPTA one packet woke the TV after a full power-off, and the TV took a couple of minutes to accept connections. The integration keeps probing and reconnects when the TV is back, with no restart or reload.

If the TV never wakes, enable **General > Quick Start+** (or **Turn On by Wi-Fi**) in the settings of the TV.

03.5 Turn on from an automation

TURN THE TV ON

```
service: media_player.turn_on
target:
  entity_id: media_player.lg_webos_tv
```

Screen off is separate from power off. When the TV is on with the screen off, `turn_on` on the remote wakes the screen over the websocket. Use the **Turn screen on** button for the same effect.

SEE ALSO

- Options
- Troubleshoot

04 Launch apps and tune channels

Open an app, a channel or a URL on the TV from the media player, the selects or a service.

The media player starts apps and tunes channels with `media_player.play_media`.

04.1 Launch an app

LAUNCH NETFLIX

```
service: media_player.play_media
target:
  entity_id: media_player.lg_webos_tv
data:
  media_content_type: app
  media_content_id: netflix
```

The integration also launches an app when `media_content_id` is a full app id such as `com.webos.app.youtube`, whatever the content type. To pass parameters, use launch_app.

04.2 Tune a channel

TUNE A CHANNEL BY NUMBER AND NAME

```
service: media_player.play_media
target:
  entity_id: media_player.lg_webos_tv
data:
  media_content_type: channel
  media_content_id: "5 ARD"
```

The integration matches `media_content_id` in this order: exact channel number, exact channel name (case-insensitive), then a partial name. An unmatched value fails with `Channel 5 ARD not found in the channel list.`

04.3 Pick a source

`media_player.select_source` takes an app or input title from the source list. Choose which titles appear in Options. An unknown title fails with `Source <name> not found in the sources list for <tv>.`

04.4 Open a URL

OPEN A WEB PAGE

```
service: bscpylgtv.command
target:
  entity_id: media_player.lg_webos_tv
data:
  command: system.launcher/open
  payload:
    target: https://www.youtube.com
```

SEE ALSO

- Services
- Entities

05 Control soundbar volume

Switch sound outputs and turn on the experimental volume slider for a soundbar on ARC, optical or an external speaker.

On an external output the TV forwards volume keys to the soundbar and cannot read its level. The media player then shows no volume level and no slider. `volume_up`, `volume_down` and `volume_mute` still reach the soundbar.

05.1 Switch the sound output

Use the **Sound output** select, or call the service:

SEND AUDIO TO A SOUNDBAR ON HDMI ARC

```
service: bscpylgtv.select_sound_output
target:
  entity_id: media_player.lg_webos_tv
data:
  sound_output: external_arc
```

Sound outputs lists every value. The select adds the output the TV currently reports when it is not in the list.

05.2

Get the volume slider back

1. Open **Settings > Devices & Services > LG WebOS TV (bscpylgtv)** and choose **Configure**.
2. Turn on **Experimental: soundbar volume by key presses**.
3. Adjust the three timing options if your soundbar needs it. Defaults are in Options.

Setting a volume then presses volume-down until the soundbar is at its minimum, and presses volume-up once per step to the target. The soundbar does not report its level, so the integration shows its own estimate. While the mode is active the media player gains a `soundbar_steps` attribute.

05.3

What to expect

- The first set after a restart homes from the full step count, because the estimate starts empty.
- A newer volume set stops the one still running after its current press.
- Changes made with the remote of the soundbar or of the TV are not tracked. The next set re-homes with the extra presses.
- Setting 50 % on 40 steps takes about (estimate + 5 + 20) x 320 ms.

05.4

Tune the press interval

Presses sent faster than the soundbar accepts are lost. On an LG SN4 over ARC, 15 presses from 0 reached 13 at 220 ms and 240 ms, 15 at 267 ms and 370 ms, and 30 of 30 at 320 ms. Test your soundbar before you lower the interval.

SEE ALSO

- Why external outputs have no volume level
- Options

06

Take a screenshot

Capture the TV screen to a file in /media or as base64 in a service response.

`bscpylgtv.take_screenshot` captures the current screen as a JPEG.

06.1

Save to a file

WRITE /MEDIA/TV_SCREEN.JPG

```
service: bscpylgtv.take_screenshot
target:
  entity_id: media_player.lg_webos_tv
data:
  filename: /media/tv_screen.jpg
```

Home Assistant checks the path with `hass.config.is_allowed_path` before it writes. The path must resolve inside an existing allowed directory. `/media` and `<config>/www` are allowed by default. Add other directories to `allowlist_external_dirs`.

> DO NOT WRITE TO THE WWW FOLDER
>
> Home Assistant serves `<config>/www` at `/local/` without authentication. Anyone who can reach your instance can open the picture. Write screenshots to `/media` or another private allowed directory.

06.2

Get the image in the response

Leave out `filename` and call the service with a response variable:

RESPONSE

```
image: "/9j/4AAQSkZJRgABAQ..."  # base64 JPEG
```

06.3

Errors

MESSAGE	CAUSE	FIX
`Screenshot path /x.jpg is not allowed.`	The path is outside every allowed directory.	Use `/media` or add the directory to `allowlist_external_dirs`.
`Could not write the screenshot file to /media/x.jpg.`	The directory is missing or not writable.	Create the directory and check permissions.

The integration has no Screenshot button. It was removed because it wrote into the public `www` folder. Call the service from an automation or script.

SEE ALSO

- Services

07

Troubleshoot

Causes and fixes for pairing failures, unavailable TVs and empty or unknown entities.

Find your symptom, then follow its fix. Collect logs with Report a problem when none applies.

07.1

Pairing fails or loops

Make sure the TV is on and accept the prompt before the timeout. If the TV keeps rejecting a stale key, start reauthentication from the integration page. It pairs again from scratch.

On webOS 25 sets (LG C3, C5 and similar), versions before 2.0.2 never showed the prompt, because the TV ignores the `hello` handshake of the library (issue #11). Update to 2.0.2 or later.

07.2

webOS 26 rejects pairing, or a working setup stopped after a firmware update

Some webOS 26 sets (firmware 43.x) reject the legacy signed manifest. From 2.0.4 the integration retries with an unsigned manifest after the known registration rejection. It remembers this *Compatibility mode* and skips the software information request, which that mode does not provide.

If the stored *Pairing key* is no longer registered, Home Assistant starts reauthentication. Use **Reconfigure** to check whether the normal signed manifest works again. This differs from the webOS 25 `hello` problem.

07.3

The TV shows unavailable

The TV is off or unreachable. With a known MAC address, entities show `off` and `turn_on` wakes the TV. See Turn the TV on. On a tested OLED48CXPTA the TV needed a couple of minutes after a full power-off before it accepted connections, and the integration recovered on its own.

07.4

The connection went stale after a TV restart or network change

The integration probes the connection every 10 s, drops dead sockets and rebuilds the connection. Recovery takes about 15 s and needs no restart of Home Assistant.

07.5

Picture mode shows unknown

Some models refuse every read of the current picture mode. Tested on an OLED48CXPTA: the settings service rejects the key and the config service does not carry it. Set a mode from Home Assistant once. The value is remembered across restarts.

The option list is a fixed fallback on such models. When the live read fails, the integration appends the unknown current mode so the state still displays.

07.6

Sharpness and color temperature show unknown

The same models reject reads of those keys, and the settings push of the TV does not carry them. They work after you set them once from Home Assistant, and the value is remembered across restarts. Backlight, contrast, brightness and color are pushed live and always show the real value.

07.7

The channel select is empty

The TV reports no tuner channels. Run a channel scan in **Settings > Channels** on the TV. The select fills on the next update. If you use only HDMI inputs, the channel entities do not apply.

07.8

The channel select is very long

A TV with 1000 or more channels produces a long dropdown, which is a Home Assistant UI limit. The list is rebuilt only when the lineup changes. Use `media_player.play_media` with a channel number instead (see Launch apps and tune channels).

07.9

TPC or GSR show unknown

The state reads of these switches are best-effort and return `unknown` on some models.

07.10

Service errors

MESSAGE	CAUSE	FIX
`Unable to connect to device <name>. Turn the TV on, or configure its MAC address in the integration options so it can be woken remotely.`	The TV is off and no MAC address is stored.	Turn the TV on, or set the MAC address in Options.
`Error calling <func> for device <name>: Device is off and cannot be controlled.`	The service needs a connected TV.	Turn the TV on first.
`Unknown button <name>.`	The button is not in the library list.	Use a name from Remote commands.
`Icon path <path> is not allowed.`	The notify icon is outside the allowed directories.	Use a path under `/media` or an entry of `allowlist_external_dirs`.
`Icon <path> not found when sending notification for device <name>`	The icon file does not exist.	Check the path.

SEE ALSO

- Limitations
- Report a problem

08

Report a problem

Collect a debug log and a diagnostics file for a bug report.

Open an issue and fill in the bug report form. It asks for the TV model, the webOS version and two files.

08.1

Collect the debug log

1. Open **Settings > Devices & services > LG WebOS TV (bscpylgtv)**.
2. Choose **Enable debug logging** from the three-dot menu.
3. Repeat the problem.
4. Choose **Disable debug logging**. The browser downloads `home-assistant_bscpylgtv_<date>.log`.

The log covers only the time between the two clicks. The integration does not log the pairing key.

Connection errors from the library can include the local IP address of the TV, so read the file before you post it.

To keep debug logging on across restarts, add this to `configuration.yaml`:

CONFIGURATION.YAML

```
logger:
  logs:
    custom_components.bscpylgtv: debug
    bscpylgtv: debug
```

08.2

Download diagnostics

1. Open the device page of the TV.
2. Choose **Download diagnostics**.

The file lists the integration, library and Home Assistant versions, the model and power state of the TV, and the current app and channel. It redacts the pairing key, host, MAC address and unique ID.

SEE ALSO

- Troubleshoot

09

Remove the integration

Delete the config entry, uninstall the code and clean up old v1 files.

1. Open **Settings > Devices & Services** and open the **LG WebOS TV (bscpylgtv)** entry.
2. Choose the three-dot menu, then **Delete**. This removes the config entry, its device and its entities.
3. To remove the code, uninstall through HACS or delete `custom_components/bscpylgtv`, then restart Home Assistant.
4. Optional: delete the pairing files that v1 left in `.storage/bscpylgtv_*.sqlite`.

SECTION

Reference

Entities, attributes and configuration options.

10

Entities

Every entity the integration creates, with its platform, category, default state and range.

Each TV becomes one device. Entities marked disabled are created disabled and you enable them on the entity settings page.

PLATFORM	ENTITY	CATEGORY	DEFAULT	DESCRIPTION
`media_player`	LG WebOS TV (named after the device)	none	enabled	Power, volume, mute, source (apps and inputs), play, pause, stop, next, previous and `play_media` for apps and channels.
`remote`	Remote	none	enabled	Library buttons plus pointer (`MOVE`, `CLICK`, `SCROLL`) and text (`TEXT`) commands.
`notify`	Notifications	none	enabled	Toast message on the TV screen.
`button`	Turn screen off	none	enabled	Switches the panel off and keeps the TV running.
`button`	Turn screen on	none	enabled	Switches the panel back on.
`button`	Reboot	config	disabled	Reboots the TV.
`number`	Backlight	config	disabled	Backlight or OLED light, 0 to 100.

PLATFORM	ENTITY	CATEGORY	DEFAULT	DESCRIPTION
`number`	Contrast	config	disabled	0 to 100.
`number`	Brightness	config	disabled	0 to 100.
`number`	Color	config	disabled	Color saturation, 0 to 100.
`number`	Sharpness	config	disabled	0 to 50.
`number`	Color temperature	config	disabled	-50 to +50.
`select`	Picture mode	none	enabled	Live-read mode list, with a fixed fallback list if the TV refuses the read.
`select`	Sound output	none	enabled	Output list in Sound outputs.
`select`	Channel	none	enabled	Channel list. Appears after you open Live TV once.
`sensor`	Current app	none	enabled	Name of the running app.
`sensor`	Volume	none	enabled	Volume in %.
`sensor`	Power state	none	enabled	`Active`, `Active Standby`, `Screen Off`, `Suspend` and other TV states.
`sensor`	Current channel	diagnostic	enabled	Channel number and name while Live TV runs.
`switch`	TPC	config	disabled	Temporal Peak Luminance Control, OLED burn-in protection.
`switch`	GSR	config	disabled	Global Sticky Reduction, OLED burn-in protection.

BURN-IN PROTECTION

TPC and GSR are OLED panel-protection settings. Turning them off can raise the risk of burn-in.

10.1

Media player behavior

- Supported features: turn off, pause, play, stop, next, previous, select source and play media. Turn on appears when a MAC address is stored.
- Volume features depend on the sound output. `tv_speaker` and `headphone` support absolute volume. External outputs support step and mute only. See Why external outputs have no volume level.

- The `sound_output` attribute carries the current output.
- `turn_off` does nothing when the TV is already off or disconnected, so automations do not fail.
- The state restores after a restart while the TV is off.

10.2 Remote behavior

The remote reports on while the TV is in `Screen Off`, because the websocket stays open and the remote can wake the screen. Its current activity is the title of the current input or app. Commands are in Remote commands.

10.3 Notify behavior

A webOS toast is one line, so a title is prefixed to the message as `<title>: <message>`.

The notify entity can attach an `icon` file path. Home Assistant currently passes only `message` and `title` to notify entities, so the icon is reachable only from direct entity calls. The path must be inside an allowed directory.

SHOW A TOAST

```
service: notify.send_message
target:
  entity_id: notify.lg_webos_tv_notifications
data:
  message: "Dinner is ready!"
```

SEE ALSO

- Services
- Options

11 Services

The six bscpylgtv services with their fields, plus the standard services the TV supports.

All services target the `media_player` entity of the TV. Developer tools in Home Assistant show the same names and descriptions.

SERVICE	FIELDS	PURPOSE
`bscpylgtv.button`	`button` (required)	Press a remote button.
`bscpylgtv.command`	`command` (required), `payload`	Send a raw *SSAP* endpoint.
`bscpylgtv.select_sound_output`	`sound_output` (required)	Change the sound output.
`bscpylgtv.launch_app`	`app_id` (required), `params`	Launch an app.
`bscpylgtv.take_screenshot`	`filename`	Capture the screen.
`bscpylgtv.set_settings`	`category` (required), `settings` (required)	Write TV settings.

11.1 bscpylgtv.button

PRESS INFO

```
service: bscpylgtv.button
target:
  entity_id: media_player.lg_webos_tv
data:
  button: INFO
```

Button names are in Remote commands. An unknown name fails with `Unknown button <name>`.

11.2 bscpylgtv.command

Sends a raw SSAP endpoint. The response comes back when you call the service with a response variable. This is for expert use.

OPEN A URL

```
service: bscpylgtv.command
target:
  entity_id: media_player.lg_webos_tv
data:
  command: system.launcher/open
  payload:
    target: https://www.youtube.com
```

The service cannot read return values of Luna-only endpoints (library limitation).

11.3 bscpylgtv.select_sound_output

SELECT HDMI ARC

```
service: bscpylgtv.select_sound_output
target:
  entity_id: media_player.lg_webos_tv
data:
  sound_output: external_arc
```

Values are in Sound outputs.

11.4 bscpylgtv.launch_app

OPEN A YOUTUBE VIDEO

```
service: bscpylgtv.launch_app
target:
  entity_id: media_player.lg_webos_tv
data:
  app_id: com.webos.app.youtube
  params:
    contentTarget: https://www.youtube.com/watch?v=dQw4w9WgXcQ
```

11.5 bscpylgtv.take_screenshot

Without `filename` the JPEG returns as base64 in the response. With `filename` the integration writes the file. Path rules and the warning about www are in Take a screenshot.

WRITE A FILE

```
service: bscpylgtv.take_screenshot
target:
  entity_id: media_player.lg_webos_tv
data:
  filename: /media/tv_screen.jpg
```

11.6 bscpylgtv.set_settings

Writes settings through the settings API of the TV.

SET BACKLIGHT AND CONTRAST

```
service: bscpylgtv.set_settings
target:
  entity_id: media_player.lg_webos_tv
data:
  category: picture
  settings:
    backlight: 55
    contrast: 80
```

11.7 Standard services

SERVICE	USE
`media_player.play_media` with `app` or `channel`	See Launch apps and tune channels.
`media_player.select_source`	Pick an app or input from the source list.
`media_player.turn_on`, `turn_off`, `volume_*`, `media_*`	The usual media player controls.
`notify.send_message`	Toast on the TV. See Entities.
`remote.send_command`	See Remote commands.

SEE ALSO

- Entities
- Take a screenshot

12 Remote commands

Button names, aliases and the pointer and text commands that remote.send_command accepts.

`remote.send_command` and `bscpylgtv.button` take button names from the bscpylgtv library. The library has 77 names. Names are case-insensitive.

12.1 Aliases

NAME	SENT AS
SELECT	ENTER
VOLUME_UP	VOLUMEUP
VOLUME_DOWN	VOLUMEDOWN

12.2 Common buttons

LEFT, RIGHT, UP, DOWN, HOME, BACK, ENTER, MENU, EXIT, INFO, DASH, ASTERISK, CC, MUTE, VOLUMEUP, VOLUMEDOWN, CHANNELUP, CHANNELDOWN, PLAY, PAUSE, STOP, RED, GREEN, BLUE, YELLOW, POWER and 0 to 9.

12.3 Pointer and text commands

COMMAND	FORMAT	EFFECT
CLICK	CLICK	Click at the pointer.
MOVE	MOVE:dx,dy or MOVE:dx,dy,down	Move the pointer by dx, dy. down is 0 or 1 and defaults to 0.
SCROLL	SCROLL:dx,dy	Scroll by dx, dy.
TEXT	TEXT:some text	Type text into the focused field. Case is kept.

A malformed command fails with Unknown button <command>.

MOVE THE POINTER, CLICK, TYPE

```
service: remote.send_command
target:
  entity_id: remote.lg_webos_tv_remote
data:
  command:
    - "MOVE:100,50"
    - CLICK
    - "TEXT:Dark"
```

The remote accepts a list and sends the commands in order.

SEE ALSO

- Services

13 Options

The Configure flow, with the source list, MAC address and soundbar volume settings and their defaults.

Open **Settings > Devices & Services > LG WebOS TV (bscpylgtv)** and choose **Configure**.

OPTION	TYPE	DEFAULT	MEANING
Sources list	multi-select	all sources	Apps and inputs shown in the media player source list.
MAC address	text	detected	Address for *Wake-on-LAN*, format `aa:bb:cc:dd:ee:ff`. Set it only when detection failed.
Experimental: soundbar volume by key presses	boolean	off	Restores a volume slider on external sound outputs.
Soundbar press interval (ms)	number	320	Time between key presses. Faster presses are lost.
Extra volume-down presses when homing	number	5	Added to the estimated level when pressing down, so the soundbar reaches its minimum if the estimate is too high. Each one adds one interval to every volume change.
Soundbar volume steps (0 to max)	number	40	Steps from minimum to maximum. The slider maps 0 to 100 % onto 0 to this value. LG soundbars use 40.

An invalid MAC address fails with `Invalid MAC address. Use the format aa:bb:cc:dd:ee:ff.`

Option keys in the config entry are `sources`, `soundbar_volume`, `soundbar_step_interval_ms`, `soundbar_extra_down_presses` and `soundbar_max_steps`. The entry data holds `host`, `mac`, `client_key` and `unsigned_manifest`.

SEE ALSO

- Control soundbar volume
- Turn the TV on

14

Sound outputs

The sound output values the select and the select_sound_output service accept.

The select lists these values and adds the output the TV currently reports when it is missing.

VALUE	MEANING	VOLUME CONTROL
`tv_speaker`	TV speakers	Absolute
`headphone`	Headphone jack	Absolute

VALUE	MEANING	VOLUME CONTROL
`external_arc`	Soundbar on HDMI ARC	Step and mute
`external_optical`	Optical output	Step and mute
`external_speaker`	External speaker	Step and mute
`tv_external_speaker`	TV speakers plus an external speaker	Step and mute
`tv_speaker_external_arc`	TV speakers plus a soundbar on ARC (WOW Orchestra, from 2.1.1)	Step and mute
`lineout`	Line out	None
`bt_soundbar`	Bluetooth soundbar	Step and mute
`lgSoundSync`	LG Sound Sync	Step and mute
`wisa_speaker`	WiSA speaker	Step and mute
`usb_speaker`	USB speaker	Step and mute

Absolute volume means the slider works. Step and mute means `volume_up`, `volume_down` and `volume_mute` work and the slider is hidden. The experimental slider in Control soundbar volume applies to the five outputs from `external_arc` to `tv_speaker_external_arc`, not to the Bluetooth, Sound Sync, WiSA and USB outputs. When the TV reports no output, the media player offers absolute volume.

SEE ALSO

- Why external outputs have no volume level
- Control soundbar volume

15

Glossary

Terms used in these pages.

ARC
Audio Return Channel, the HDMI path that carries TV audio to a soundbar.

Compatibility mode
Registration with an unsigned manifest. The integration switches to it when a webOS 26 TV rejects the signed manifest.

Pairing key
The client key a TV issues after you accept the pairing prompt. Home Assistant stores it in the config entry and sends it on every connection.

SSAP
Second Screen Application Protocol, the websocket API a webOS TV exposes to companion apps. The integration sends every command through it.

Wake-on-LAN
A UDP magic packet sent to the TV's MAC address. It is the only way to turn a webOS TV on from Home Assistant.

SECTION

Concepts

How the integration works and why.

16

How it works

The connection, pairing, reconnect and wake behavior of the integration.

The integration keeps one websocket open per TV and subscribes to state pushes. It does not poll for state.

16.1

Connection and subscriptions

A coordinator owns the bscpylgtv client. It subscribes to `system_info`, `software_info`, `power`, `current_app`, `muted`, `volume`, `apps`, `inputs`, `sound_output` and `picture_settings`. The channel list and current channel subscribe lazily once Live TV is in the foreground, with 15 attempts spaced 2 s apart. In *Compatibility mode* the integration drops `software_info`.

The integration is `local_push`, with `config_flow` and SSDP discovery on `urn:lge-com:service:webos-second-screen:1`.

16.2

Watchdog

The coordinator interval of 10 s drives a watchdog that never fetches state. Each tick probes a healthy connection (5 s budget) and rebuilds a dead one. Commands time out after 8 s and a reconnect after 15 s. After a TV restart or network change, things recover in about 15 s.

16.3

Pairing

The TV shows a prompt and returns a *Pairing key* on accept. The integration stores it as `client_key` in the config entry. Two firmware behaviors shape the flow:

- webOS 25 ignores the `hello` handshake of the library, so a connect that requests it hangs until the caller's timeout. The integration bounds the probe to 5 s (fixed in 2.0.2).
- webOS 26 firmware 43.x can reject the signed manifest. The integration retries with an unsigned manifest, stores `unsigned_manifest`, and keeps using it (from 2.0.4).

A key the TV no longer accepts raises reauthentication.

16.4

Power and wake

The websocket is closed while the TV is off, so `turn_on` cannot use it. `turn_on` sends *Wake-on-LAN* to UDP port 9 and asks the coordinator to reconnect as soon as the TV boots. The watchdog keeps retrying every 10 s in any case. In art standby (`Screen Off`) the socket stays open, so a command can wake the screen.

16.5

Entity availability

With a stored MAC address, entities show `off` while the TV is off. Without one they show `unavailable`. The media player restores its supported features after a restart while the TV is off.

SEE ALSO

- Why external outputs have no volume level
- Limitations

17

Why external outputs have no volume level

The TV forwards volume keys to a soundbar and cannot read its level, so the integration estimates it.

On `external_arc`, `external_optical`, `external_speaker`, `tv_external_speaker` and `tv_speaker_external_arc`, the TV only counts key presses and sends them to the soundbar. It has no reading of the soundbar level and no way to set it.

Because of that, the media player hides the volume slider and the volume level on these outputs. Up, down and mute still work.

17.1

The experimental mode

The option **Experimental: soundbar volume by key presses** builds a slider from presses. A volume set presses volume-down until the soundbar bottoms out, then volume-up once per step to the target. The integration remembers the level it believes the soundbar has. That level is an estimate, and any change made outside Home Assistant makes it wrong until the next set re-homes the soundbar.

The mode is off by default because it sends dozens of key presses for one change and presses can be lost on a slow soundbar. See Control soundbar volume for timing.

SEE ALSO

- Sound outputs
- Options

18

Limitations

What the integration cannot do because of webOS or the library.

- **Firmware updates.** webOS has no firmware-update API for websocket clients. Update from the settings of the TV.
- **Turn on.** The integration turns the TV on with *Wake-on-LAN* and does not use the websocket power-on call. Wake-on-LAN over Wi-Fi needs a setting on the TV. See Turn the TV on.
- **PIN pairing.** Pairing always uses the prompt on the TV.
- **Luna-only endpoints.** The `command` service cannot read their return values (library limitation).
- **TPC and GSR state.** Reads are best-effort and show `unknown` on some models.
- **Picture settings reads.** Some models refuse reads of picture mode, sharpness and color temperature until you set them once. See Troubleshoot.
- **Soundbar volume.** The level is an estimate. See Why external outputs have no volume level.

SECTION

Develop

Build, test and release the integration.

19

Build and test

Run the tests, lint and type checks against the pinned Home Assistant version.

Tests run against a pinned Home Assistant version with uv.

CHECK THE PROJECT

```
uv sync                                          # create .venv from uv.lock
uv run pytest
uv run ruff check custom_components tests
uv run mypy
```

`pyproject.toml` holds the dev dependency group: pytest-homeassistant-custom-component, bscpylgtv, ruff and mypy. `uv.lock` pins the full tree.

After you change the `bscpylgtv` pin in `custom_components/bscpylgtv/manifest.json`, update the dev group and run `uv lock` in the same change. The pin points at the `belikh/bscpylgtv` fork (0.5.5.post1), which carries a standby fix until chros73/bscpylgtv#13 merges upstream.

19.1

Layout

FILE	PURPOSE
`__init__.py`	Entry setup and unload.
`coordinator.py`	Client ownership, subscriptions, watchdog.
`config_flow.py`	Setup, pairing, reauth, reconfigure, options.
`media_player.py`, `remote.py`, `notify.py`	Main platforms and the service methods.
`button.py`, `number.py`, `select.py`, `sensor.py`, `switch.py`	Other platforms.
`services.py`, `services.yaml`	Service registration.
`diagnostics.py`	Diagnostics with redaction.

FILE	PURPOSE
`const.py`	Constants, sound output list, picture mode fallback.
`translations/en.json`	All user-facing strings.

SEE ALSO

- Release

20

Release

Tag a version and let the release workflow publish the notes from CHANGELOG.md.

1. Add a `## <version>` section to `CHANGELOG.md` and set `version` in `custom_components/bscpylgtv/manifest.json` to the same value.
2. Push a tag named `<version>` or `v<version>`.
3. The workflow `.github/workflows/release.yaml` copies the matching `CHANGELOG.md` section into the release body and attaches `manifest.json`. Without a matching section it generates notes. A tag with a hyphen, such as `2.1.1-beta.1`, publishes a pre-release.

The HACS and hassfest workflows validate the repository on push.

SEE ALSO

- Build and test