INOGENI

INOGENI TOGGLE ROOMS

User guide

Version 0.1

January 18, 2024

| Version	Date	Description
0.1	January 17, 2024	Preliminary user guide for device launch.

CONTENTS

Here is a typical connection diagram used for the TOGGLE ROOMS device in a videoconferencing setup.

ROOM PC MODE WITH BYOD CONTENT SHARING

In this mode, only the Room PC USB and HDMI connections are routed to the main USB and HDMI peripherals.

The Room PC is the system that is currently selected to the main USB and HDMI peripherals. However, if the user would like to send HDMI content from the laptop's USB-C or HDMI connection to the Room PC, it is possible to do so with the HDMI SHARE output connection.
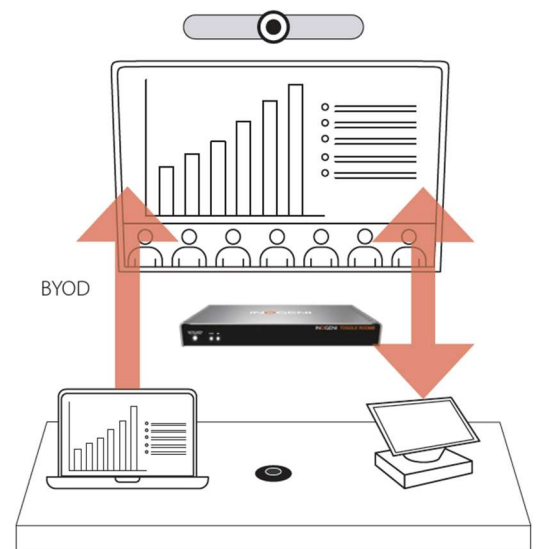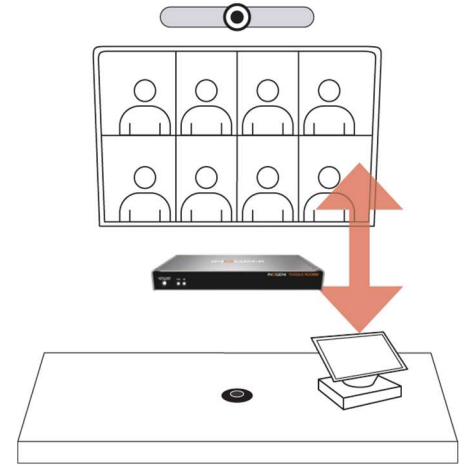
BYOM MODE

In this mode, the laptop is the system that is currently selected to the main USB and HDMI peripherals. The Room PC is completely disconnected from the setup.

Here is a simple block diagram to better understand the usage of the product.
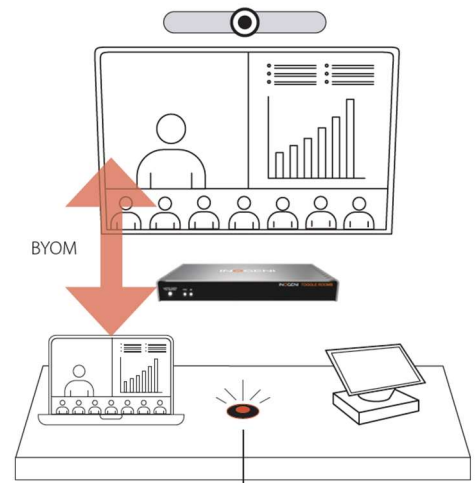
Figure 1: Basic block diagram

Here is a simple connectivity diagram showing

TOGGLE ROOMS
BYOD/BYOM switcher

Push button for BYOM

USB speakerphone

Touch panel

Microsoft Teams

ZOOM

Google Meet

Room PC

USB-C 100W

BYOD BYOM

TOGGLE ROOMS

Under the table installation
Mounting hardware included

USB-C (100W) or USB & HDMI

LEGEND

USB-C	
USB	
HDMI	
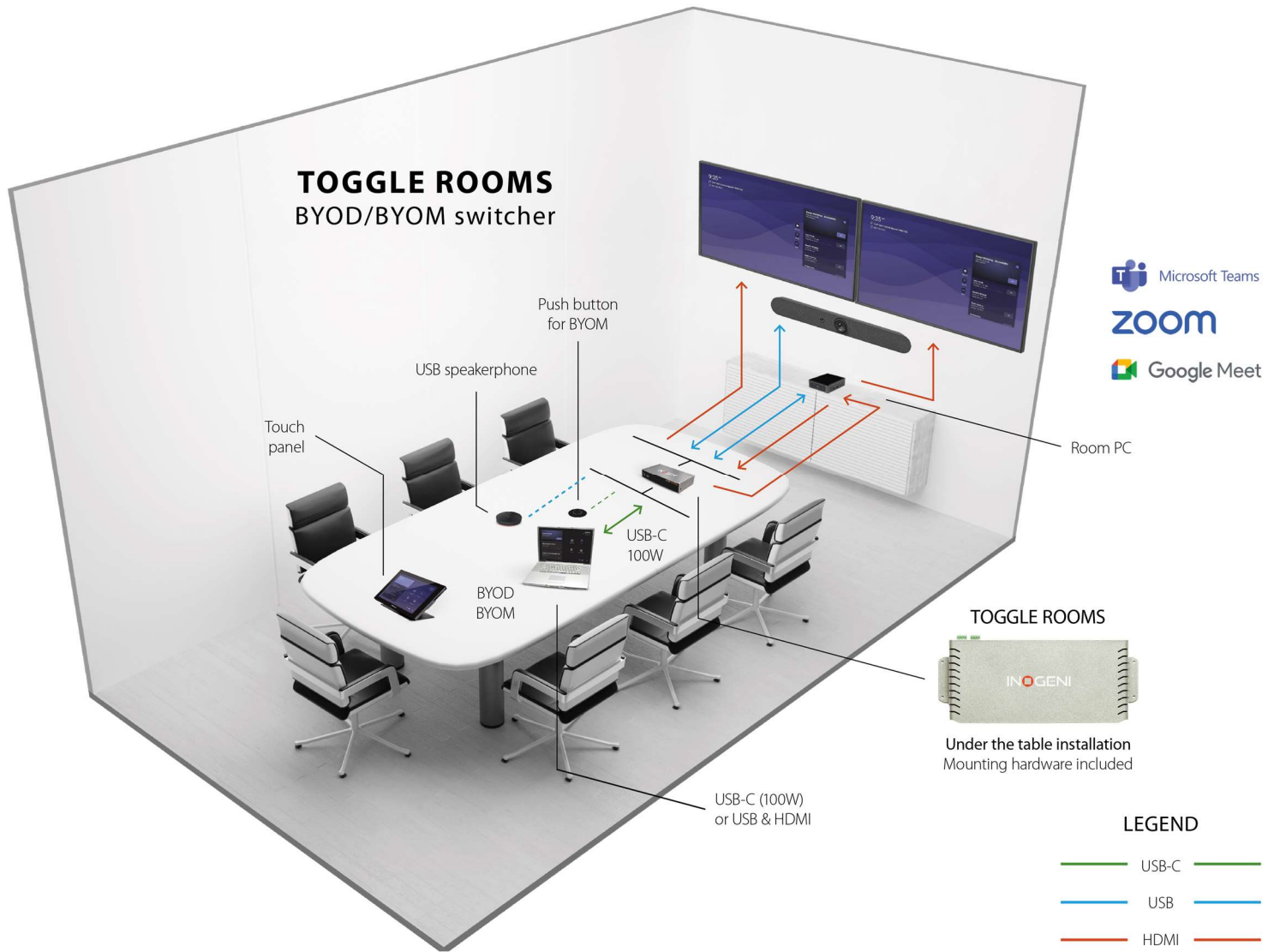
DEVICE INTERFACES

Here are the devices interfaces.
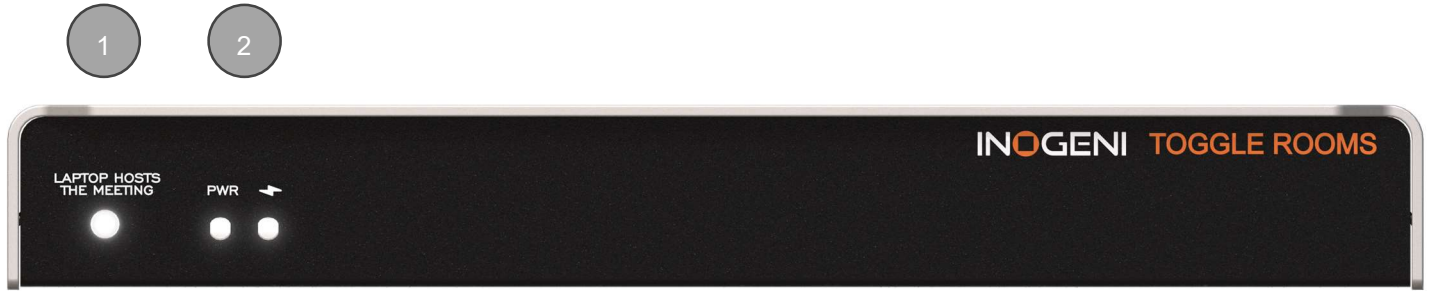
Figure 2: Front side connections

Figure 3: Back side connections

	Items
1	LAPTOP HOST THE MEETING button. This button will connect HDMI and USB peripherals to the laptop connection for BYOM.
2	PWR and charging status leds.
3	24VDC power input.
4	USB-C laptop connection.
5	USB-B laptop connection.
6	HDMI laptop connection.
7	USB-B Room PC connection.
8	HDMI Room PC connection.
9	USB devices.
10	HDMI share output from laptop.
11	HDMI display output.
12	LAN interface.
13	RS232 and remote interface.
14	GPI/button interface.

LEDS BEHAVIOR

Here are the LEDs behavior:

LAPTOP HOSTS THE MEETING	
OFF	Laptop not selected.
SOLID	Laptop selected.
BLINK	Error condition. 1. When the user tries to switch to laptop if this one is not present or if USB or HDMI connections are missing. 2. When the user tries to switch host if button is locked through our API.
PWR	
OFF	Device not powered.
SOLID	Device powered.
Charging �	
OFF	Laptop is not charging.
SOLID	Laptop is charging.

OPERATING MODES

There are the operating modes supported by the device. They will be explained here.

AUTOMATIC

This is the default mode. This mode will switch automatically to the last source (USB or HDMI) connected. If the current source is disconnected, the device will switch back to the other source if it is detected. Push-button action and remote control are also supported.

MANUAL

The manual mode will enable you to force a specific source selection. Push-button action and remote control are also supported.

MANUAL WITH FALLBACK

The manual mode with fallback supports the same features as the manual mode. It will only add the possibility to switch to the other detected source connection automatically if the selected source is disconnected.

SPECIFICATIONS

Here is the complete specification.

Physical details	
Dimensions (W x L x H)	25.11 cm x 10.97 cm x 3.26 cm 9.89" x 4.32" x 1.28"
Weight	770g
Power supply	160W (85-264VAC 50/60Hz to 24V/6.67A DC)
Power supply dimensions (W x L x H)	175 mm x 72 mm x 35 mm 6.89" x 2.83" x 1.38"
Package contents	1 x Toggle Rooms 1 x USB-C to USB-C cable – 6ft 1 x USB3.0 cable (USB-A to USB-B) – 3ft 2 x terminal block 4-pos 2 x mounting brackets 4 x M2.5 mounting screws for brackets on product 4 x screws for Toggle Rooms table/wall mount 1 x 24V/160W PSU 1 x AC power cord 1 x country-specific power plug (USA/CA or EU/UK/AU/BIS) 1 x PSU mounting brackets 4 x screws for PSU table/wall mount 1 x quickstart guide 4 x rubber feet
Operating temperature	0° to 45° C (32° to 113° F)
Storage temperature	-40° to 105° C (-40° to 221° F)
Relative humidity	0% to 90% non-condensing
Mounting options	Ability to mount under the table or on a wall.
UPC code	051497418694
Origin	Canada
Warranty	2 years

HOST - LAPTOP	
1x USB-C connector	Supports USB-C DisplayPort Alternate Mode - DisplayPort up to 3840x2160p60 / 4096x2160p60 - USB3.0 (USB 3.1 Gen 1 / 5 Gbps) - USB2.0 (480 Mbps) - Charging up to 100W - USB-C cable locking option
1x USB connector	USB 3.0 Type-B
1x HDMI connector	Up to 3840x2160p60 / 4096x2160p60 Cable locking option.

HOST - ROOMPC	
1x USB connector	USB 3.0 Type-B
1x HDMI connector	Up to 3840x2160p60 / 4096x2160p60. Cable locking option.

HDMI DISPLAY output	
Resolution	Up to 3840x2160p60 / 4096x2160p60.
Connector	HDMI with cable locking option.

HDMI SHARE output	
Resolution	Up to 3840x2160p60 / 4096x2160p60.
Connector	HDMI with cable locking option.

USB devices	
Connectors	3 x USB3.0 Type-A ports.
Power	1.8A shared between downstream ports.

Control	
Control options	Front button – for laptop selection RS232 GPI LAN USB
IP interface	10/100Mbps Supports DHCP or static addressing. IP control available through RESTAPI and telnet connections.
RS232 interface	4-pos terminal block connector Baud rates: 9600 [default], 19200, 38400 and 115200 Data bits: 8 Stop bits: 1 Parity: None Flow control: None
GPI interface	4-pos terminal block connector 2x Contact-closure control. GPI: - Controlled by open-drain IO (short to ground) or driven IO. - Supported voltage range: 0 to 12V max. - Voltage threshold is 2.3V. VOUT: - Able to power up the led on the button of our INO-Button accessory. - Logic-low level: 0 @ 0.5V - Logic-high level: 4.5 @ 5V

HDMI video	
HDCP compliance	Compliant with HDCP2.3, HDCP2.2 and HDCP1.4
HDMI compliance	Compliant with HDMI2.0b, HDMI1.4 and DVI1.0
Sampling frequency	600MHz
Video scaling	Crosspoint switch supports video downscaling from 4K to 1080p.
Chroma subsampling	YUV/RGB 4:4:4, 4:2:2
CEC	Ability to send CEC commands to connected HDMI display sink.

HDMI audio	
Audio	Audio passthrough from input to output
Formats	LPCM, Dolby Digital, DTS up to 192kHz

Certifications	
Device	FCC, CE, UKCA, RoHS, IEC62368, RCM, SoV
Power supply	FCC, CE, UKCA, RoHS, IEC62368, RCM, CCC, CB, EAC, VI, UL
TAA-compliance	Yes

Compatibility	
Operating system	NO driver installation necessary Windows 7 and above (32/64-bit) macOS 10.10 and above Linux (kernel v2.6.38 and above)

Here is the complete list of commands provided through the serial connection. As written on the back of the device, here is the pinout of the terminal block.

Pin 1: Receive
Pin 2: GND
Pin 3: Transmit
Pin 4: 5V supply (for INOGENI Remote)
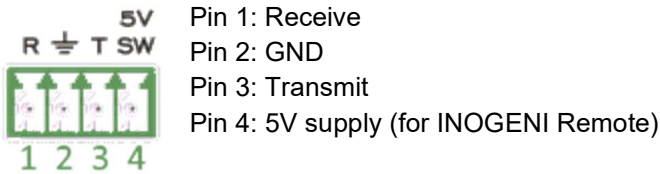
> ⚠️ **NOTE:** The user needs to put a **space character between the command name and argument**.

ARG	Lists all the available options for the arguments to be used with the command.
RX	When command does not have any argument or only first argument is provided, it will return information from the device.
TX	When command have all arguments, it will apply the configuration to the device.

You need to add a carriage return **<CR>** character and a line feed **<LF>** character at the end of the command string.

Typically, commands will return "ACK**<CR><LF>**" in case of success and "NACK**<CR><LF>**" in case of failure.

Baud rate: 9600 [default] // **Data bits**: 8 // **Stop bits:** 1 // **Parity:** None // **Flow control:** None

Command	REQ/ ARG	Arguments	Return
HELP Return commands list with description.	RX	N/A	List of all the supported commands.
RSTR Restore default settings (including password and REST API token).	RX	N/A	ACK**<CR><LF>**
REBOOT Reboot the device.	RX	N/A	ACK**<CR><LF>**
VERSION Return firmware version.	RX	N/A	MAJOR=<Integer>**<CR><LF>** MINOR=<Integer>**<CR><LF>** ACK**<CR><LF>**
STATUS Return laptop and RoomPC information, display and share output timings.	RX	N/A	List of all the status of the device.
USBHOST Get/Set USB host to use.	ARG	**<host> options:** 0 => RoomPC 1 => Laptop USB-C 2 => Laptop USB-B 3 => OFF	
	TX	<host>	ACK**<CR><LF>**
	RX	N/A	USBHOST=<host>**<CR><LF>** ACK**<CR><LF>**
DISPLAYSRC Get/Set which HDMI source to be routed to display output.	ARG	**<src> options:** 0 => RoomPC 1 => Laptop USB-C 2 => Laptop HDMI 3 => OFF	
	TX	<src>	ACK**<CR><LF>**

Command	REQ/ARG	Arguments	Return
	RX	N/A	DISPLAYSRC=\<src\>**\<CR\>\<LF\>** ACK**\<CR\>\<LF\>**
SHARESRC Get/Set which HDMI source to be routed to share output.	ARG	**\<src\> options:** 0 => RoomPC [Not supported in automatic mode] 1 => Laptop USB-C 2 => Laptop USB-B 3 => OFF	
	TX	\<src\>	ACK**\<CR\>\<LF\>**
	RX	N/A	SHARESRC=\<src\>**\<CR\>\<LF\>** ACK**\<CR\>\<LF\>**
OPMODE Get/Set operation mode. By default, the device will operate in **RoomPC / BYOD mode** – RoomPC USB and HDMI peripherals selected, and laptop sends video content only to SHARE output. The user will need to trigger our API or use the GPI interface to enter BYOM mode. When **BYOM mode** is set, the device will automatically switch all HDMI and USB peripherals to the laptop as soon as it is detected. When **Custom mode** is set, the user can set the USB, display and share source switching modes independently.	ARG	**\<opMode\> options:** 0 => RoomPC with BYOD/content sharing [default] 1 => BYOM 2 => Custom	
	TX	\<opMode\>	ACK**\<CR\>\<LF\>**
	RX	N/A	OPMODE=\<opMode\>**\<CR\>\<LF\>** ACK**\<CR\>\<LF\>**
USBHOSTSWMODE Get/Set USB host switching mode. The operation mode must be set to "Custom" to use this.	ARG	**\<swMode\> options:** 0 => Automatic mode [default] 1 => Manual mode 2 => Manual mode with fallback	
	TX	\<swMode\>	ACK**\<CR\>\<LF\>**
	RX	N/A	USBHOSTSWMODE=\<swMode\>**\<CR\>\<LF\>** ACK**\<CR\>\<LF\>**
DISPLAYSWMODE Get/Set HDMI display source switching mode. The operation mode must be set to "Custom" to use this.	ARG	**\<swMode\> options:** 0 => Automatic mode [default] 1 => Manual mode 2 => Manual mode with fallback	
	TX	\<swMode\>	ACK**\<CR\>\<LF\>**
	RX	N/A	DISPLAYSWMODE=\<swMode\>**\<CR\>\<LF\>** ACK**\<CR\>\<LF\>**
SHARESWMODE Get/Set HDMI share source switching mode. The operation mode must be set to "Custom" to use this.	ARG	**\<swMode\> options:** 0 => Automatic mode [default] 1 => Manual mode 2 => Manual mode with fallback	
	TX	\<swMode\>	ACK**\<CR\>\<LF\>**
	RX	N/A	SHARESWMODE=\<swMode\>**\<CR\>\<LF\>** ACK**\<CR\>\<LF\>**

Command	REQ/ARG	Arguments	Return
NETWORK Get/Set network settings.	ARG	**`<mode>` options:** `static` => addressing is static `dhcp` => use DHCP addressing If mode is static, ip and netmask are required while gateway is optional. **`<ip>` option:** String defined IP address. Example: `192.168.0.20` **`<netmask>` option:** String defined netmask address. Example: `255.255.0.0` **`<gateway>` option:** String defined gateway address. Example: `192.168.0.1`	
	TX	`<mode> <ip> <netmask>` `<gateway>`	ACK**`<CR><LF>`**
	RX	N/A	MODE=`<mode>`**`<CR><LF>`** IP=`<ip>`**`<CR><LF>`** NETMASK=`<netmask>`**`<CR><LF>`** GATEWAY=`<gateway>`**`<CR><LF>`**
CECPOWER Power ON/OFF the display.	ARG	**`<ctrl>` options:** `0` => power off `1` => power on	
	TX	`<ctrl>`	ACK**`<CR><LF>`**
CECTOGGLEMUTE Toggle mute control.	TX		ACK**`<CR><LF>`**
CECVOLUP Increase display volume.	TX		ACK**`<CR><LF>`**
CECVOLDOWN Decrease display volume.	TX		ACK**`<CR><LF>`**
EDID Set specific EDID modes to be reported to video source.	ARG	**`<src>` options:** `0` => RoomPC `1` => Laptop USB-C `2` => Laptop HDMI **`<edid>` options:** `0` => Passthrough `1` => 4K60 `2` => 1080p60 `3` => 720p60 `4` => User EDID	
	TX	`<src> <edid>`	ACK**`<CR><LF>`**
EDIDUSR Set user EDID to be sent to specified source. Must have set the according video source EDID in user EDID mode.	ARG	**`<src>` options** `0` => RoomPC `1` => Laptop USB-C `2` => Laptop HDMI **`<edid>` => formatted 256 bytes array**	
	TX	`<src> <256 bytes array>`	ACK**`<CR><LF>`**
USBC4K60EN Get/Set the USB-C working mode. __NOTE:__ Enabling DisplayPort signal to support 4K60 will disable USB3.0	ARG	**`<mode>` options:** `0` => Disable 4K60 [default] `1` => Enable 4K60	
	TX	`<mode>`	ACK**`<CR><LF>`**

Command	REQ/ARG	Arguments	Return
connectivity on USB-C port. USB2.0 will remain active. Disabling this option will allow user to support USB3.0 and 4K30 video.	RX	N/A	USBC4K60EN=<mode>**<CR><LF>** ACK**<CR><LF>**
HDCPCTL Get/Set the HDCP setting.	ARG	**<src> options:** 0 => RoomPC 1 => Laptop USB-C 2 => Laptop HDMI **<mode> options:** 0 => Disabled 1 => HDCP v1.4 2 => HDCP v2.2 3 => Auto	
	TX	<src> <mode>	ACK**<CR><LF>**
	RX	<src>	SRC=<src>**<CR><LF>** MODE=<mode>**<CR><LF>** ACK**<CR><LF>**

GPICFG
 	
Get/Set the GPI configuration.
 	
<u>NOTE:</u>
A short to GND on this pin will trigger the function.
The function will be executed on GPI rising/falling edge.
	ARG	**<gpi> options:**
		0 => GPI1
		1 => GPI2
		
		<mode> options:
		0 => Disabled.
		1 => BYOM mode control [default GPI1]
		    SHORT = BYOM
		    OPEN = ROOMPC
		2 => USB host control [default GPI2]
		    SHORT = LAPTOP
		    OPEN = ROOMPC
		3 => Display video source control
		    SHORT = LAPTOP USB-C/HDMI
		    OPEN = ROOMPC
		4 => Laptop video source control
		    SHORT = LAPTOP USB-C
		    OPEN = LAPTOP HDMI	

	TX	<gpi> <mode>	ACK**<CR><LF>**
	RX	<gpi>	GPI=<gpi>**<CR><LF>** MODE=<mode>**<CR><LF>** ACK**<CR><LF>**
VOUT Get/Set the VOUT level. <u>NOTE:</u> By default, the firmware will drive this output to power up a led when enabling the BYOM mode. If the user wants to override this behavior, it is possible by changing the <vout> option.	ARG	**<vout> options:** 0 => Controlled by firmware. 1 => Logic-low. 2 => Logic-high.	
	TX	<vout>	ACK**<CR><LF>**
	RX	N/A	VOUT=<vout>**<CR><LF>** ACK**<CR><LF>**
BAUDRATE Set RS232 baud rate.	ARG	**<baudrate> options** 0 => 9600 1 => 19200 2 => 38400 3 => 115200	
	TX	<baudrate>	ACK**<CR><LF>**
BTNLOCK Get/Set the button lock status.	ARG	**<lockState> options:** 0 => Not locked 1 => Locked	
	TX	<lockState>	ACK**<CR><LF>**
	RX	N/A	BTNLOCK=<lockState>**<CR><LF>** ACK**<CR><LF>**

Command	REQ/ ARG	Arguments	Return
SCALER Get/Set the scaler options over the HDMI video outputs.	ARG	**`<output>` options:** `0 =>` Display output `1 =>` Share output **`<enable>` options:** `0 =>` OFF `1 =>` ON	
	TX	`<output> <enable>`	`ACK`**`<CR><LF>`**
	RX	`<output>`	`ENABLE=<enable>`**`<CR><LF>`** `ACK`**`<CR><LF>`**

You can enable a bearer authentication in the HTTP header (Authorization: Bearer <token>) through our configuration page to increase security on the API.

There will be a return code to each call with the following commands:

```
200 => success
400 => error
401 => authorization error
```

ARG	Lists all the available options for the arguments to be used with the command.
RX	When command does not have any body arguments or only first argument is provided, it will return information from the device.
TX	When command have all body arguments, it will apply the configuration to the device.

The return body will usually be JSON formatted with a `"message"` field containing a JSON string explaining the cause of the error or "success" in case of success. Note that we are using self-signed certificates.

Here is the complete list of commands supported through the REST API (excluding password change, firmware update, bearer token get/set):

Command URL / Description	REQ/ ARG	Body arguments	Return body
HTTP GET `https://<IP>/api/v1/help` Return commands list with description.	RX	N/A	JSON object with multiple fields

HTTP GET/POST
 `https://<IP>/api/v1/rstr`
 	
 Restore default settings (including password and REST API token).
	RX	N/A	`{`
			     `"message": <String>`
			`}`

HTTP GET/POST
 `https://<IP>/api/v1/reboot`
 	
 Reboot the device.
	RX	N/A	`{`
			     `"message": <String>`
			`}`

HTTP GET
 `https://<IP>/api/v1/version`
 	
 Return firmware version.
	RX	N/A	`{`
			     `"major": <Integer>,`
			     `"minor": <Integer>`
			`}`

HTTP GET `https://<IP>/api/v1/status` Return laptop and RoomPC information, display and share output timings.	RX	N/A	JSON object with multiple fields
HTTP GET/POST `https://<IP>/api/v1/usbHost` Get/Set USB host to use.	ARG	`<usbHost>` **options:** `0 =>` RoomPC `1 =>` Laptop USB-C `2 =>` Laptop USB-B `3 =>` OFF	

	TX	usbHost=<host>	`{`
			     `"message": <String>`
			`}`

	RX	N/A	`{`
			     `"usbHost": <host>,`
			     `"message": <String>`
			`}`

HTTP GET/POST `https://<IP>/api/v1/ displaySrc`	ARG	`<displaySrc>` **options:** `0 =>` RoomPC `1 =>` Laptop USB-C `2 =>` Laptop HDMI `3 =>` OFF	

Command URL / Description	REQ/ARG	Body arguments	Return body

Get/Set which HDMI source to be routed to display output.	TX	displaySrc=<src>	{
			   "message": <String>
			}

	RX	N/A	{
			   "displaySrc": <src>,
			   "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/** **shareSrc** Get/Set which HDMI source to be routed to share output.	ARG	**<shareSrc> options:** 0 => RoomPC [Not supported in automatic mode] 1 => Laptop USB-C 2 => Laptop HDMI 3 => OFF	

	TX	shareSrc=<src>	{
			   "message": <String>
			}

	RX	N/A	{
			   "shareSrc": <src>,
			   "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/** **opMode** Get/Set operation mode. By default, the device will operate in **RoomPC / BYOD mode** – RoomPC USB and HDMI peripherals selected, and laptop sends video content only to SHARE output. The user will need to trigger our API or use the GPI interface to enter BYOM mode. When **BYOM mode** is set, the device will automatically switch all HDMI and USB peripherals to the laptop as soon as it is detected. When **Custom mode** is set, the user can set the USB, display and share source switching modes independently.	ARG	**<opMode> options:** 0 => RoomPC with BYOD/content sharing [default] 1 => BYOM 2 => Custom	

	TX	opMode=<opMode>	{
			   "message": <String>
			}

	RX	N/A	{
			   "opMode": <opMode>,
			   "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/** **usbHostSwMode** Get/Set USB host switching mode. The operation mode must be set to "Custom" to use this.	ARG	**<usbHostSwMode> options:** 0 => Automatic mode [default] 1 => Manual mode 2 => Manual mode with fallback	

	TX	usbHostSwMode=<swMode>	{
			   "message": <String>
			}

	RX	N/A	{
			   "usbHostSwMode": <swMode>,
			   "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/** **displaySwMode** Get/Set HDMI display source switching mode. The operation mode must be set to "Custom" to use this.	ARG	**<displaySwMode> options:** 0 => Automatic mode [default] 1 => Manual mode 2 => Manual mode with fallback	

	TX	displaySwMode=<swMode>	{
			   "message": <String>
			}

	RX	N/A	{
			   "displaySwMode": <swMode>,
			   "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/** **shareSwMode**	ARG	**<shareSwMode> options:** 0 => Automatic mode [default] 1 => Manual mode 2 => Manual mode with fallback	

Command URL / Description	REQ/ARG	Body arguments	Return body

Get/Set HDMI share source switching mode. The operation mode must be set to "Custom" to use this.	TX	shareSwMode=<swMode>	{
			    "message": <String>
			}

	RX	N/A	{
			    "shareSwMode": <swMode>,
			    "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/network** Get/Set network settings.	ARG	**<mode> options:** static => addressing is static dhcp => use DHCP addressing If mode is static, ip and netmask are required while gateway is optional. **<ip> option:** String defined IP address. Example: 192.168.0.20 **<netmask> option:** String defined netmask address. Example: 255.255.0.0 **<gateway> option:** String defined gateway address. Example: 192.168.0.1	

	TX	mode=<mode>
		ip=<ip>
		netmask=<netmask>
		gateway=<gateway>
			{
			    "message": <String>
			}

	RX	N/A	{
			    "mode": <static,dhcp>,
			    "ip": <ip>,
			    "netmask": <netmask>,
			    "gateway": <gateway>,
			    "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/hostname** Get/Set the hostname of the device.	ARG	**<hostname> option:** String defined hostname to be shown on the network. This string must not have space characters.	

	TX	hostname=<hostname>	{
			    "message": <String>
			}

	RX	N/A	{
			    "hostname": <hostname>,
			    "message": <String>
			}

HTTP GET/POST **https://<IP>/api/v1/cecPower** Power ON/OFF the display.	ARG	**<ctrl> options:** 0 => power off 1 => power on	

	TX	ctrl=<ctrl>	{
			    "message": <String>
			}

HTTP GET/POST
https://<IP>/api/v1/cecToggleMute
 	
Toggle mute control.
	TX	N/A	{
			    "message": <String>
			}

HTTP GET/POST
https://<IP>/api/v1/cecVolUp
 	
Increase display volume.
	TX	N/A	{
			    "message": <String>
			}

HTTP GET/POST
https://<IP>/api/v1/cecVolDown
 	
Decrease display volume.
	TX	N/A	{
			    "message": <String>
			}

Command URL / Description	REQ/ARG	Body arguments	Return body
HTTP GET/POST **https://<IP>/api/v1/** **edid** Set specific EDID modes to be reported to video source.	ARG	**<src> options:** `0` => RoomPC `1` => Laptop USB-C `2` => Laptop HDMI **<edid> options:** `0` => Passthrough `1` => 4K60 `2` => 1080p60 `3` => 720p60 `4` => User EDID	

	TX	`src=<src>`
		`edid=<edid>`
			`{`
			`    "message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **edidUsr** Set user EDID to be sent to specified source. Must have set the according video source EDID in user EDID mode.	ARG	**<src> options** `0` => RoomPC `1` => Laptop USB-C `2` => Laptop HDMI **<edid> => Filetype formatted 256 bytes array**	

	TX	`src=<src>`
		`edid=<256 bytes array>`
			`{`
			`    "message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **usbc4K60En** Get/Set the USB-C working mode. **NOTE:** Enabling DisplayPort signal to support 4K60 will disable USB3.0 connectivity on USB-C port. USB2.0 will remain active. Disabling this option will allow user to support USB3.0 and 4K30 video.	ARG	**<mode> options:** `0` => Disable 4K60 [default] `1` => Enable 4K60	

	TX	`mode=<mode>`	`{`
			`    "message": <String>`
			`}`

	RX	N/A	`{`
			`    "mode": <mode>,`
			`    "message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **hdcpCtl** Get/Set the HDCP setting.	ARG	**<src> options:** `0` => RoomPC `1` => Laptop USB-C `2` => Laptop HDMI **<mode> options:** `0` => Disabled `1` => HDCP v1.4 `2` => HDCP v2.2 `3` => Auto	

	TX	`src=<src>`
		`mode=<mode>`
			`{`
			`    "message": <String>`
			`}`

	RX	`src=<src>`	`{`
			`    "src": <src>,`
			`    "mode": <mode>,`
			`    "message": <String>`
			`}`

Command URL / Description	REQ/ARG	Body arguments	Return body

HTTP GET/POST
https://<IP>/api/v1/
gpiCfg
 	
Get/Set the GPI configuration.
 	
<u>NOTE:</u>
A short to GND on this pin will trigger the function.
The function will be executed on GPI rising/falling edge.
	ARG	`<gpi>` **options:**
		`0 =>` GPI1
		`1 =>` GPI2
		
		`<mode>` **options:**
		`0 =>` Disabled.
		`1 =>` BYOM mode control [default GPI1]
		    `SHORT` = BYOM
		    `OPEN` = ROOMPC
		`2 =>` USB host control [default GPI2]
		    `SHORT` = LAPTOP
		    `OPEN` = ROOMPC
		`3 =>` Display video source control
		    `SHORT` = LAPTOP USB-C/HDMI
		    `OPEN` = ROOMPC
		`4 =>` Laptop video source control
		    `SHORT` = LAPTOP USB-C
		    `OPEN` = LAPTOP HDMI	

	TX	`gpi=<gpi>`
		`mode=<mode>`
			`{`
			  `"message": <String>`
			`}`

	RX	`gpi=<gpi>`	`{`
			  `"gpi": <gpi>,`
			  `"mode": <mode>,`
			  `"message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **vout** Get/Set the VOUT level. <u>NOTE:</u> By default, the firmware will drive this output to power up a led when enabling the BYOM mode. If the user wants to override this behavior, it is possible by changing the `<vout>` option.	ARG	`<vout>` **options:** `0 =>` Controlled by firmware. `1 =>` Logic-low. `2 =>` Logic-high.	

	TX	`vout=<vout>`	`{`
			  `"message": <String>`
			`}`

	RX	N/A	`{`
			  `"vout": <vout>,`
			  `"message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **baudRate** Set RS232 baud rate.	ARG	`<baudrate>` **options** `0 =>` 9600 `1 =>` 19200 `2 =>` 38400 `3 =>` 115200	

	TX	`baudrate=<baudrate>`	`{`
			  `"message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **btnLock** Get/Set the button lock status.	ARG	`<lockState>` **options:** `0 =>` Not locked `1 =>` Locked	

	TX	`btnLock=<lockState>`	`{`
			  `"message": <String>`
			`}`

	RX	N/A	`{`
			  `"btnLock": <lockState>,`
			  `"message": <String>`
			`}`

HTTP GET/POST **https://<IP>/api/v1/** **scaler** Get/Set the scaler options over the HDMI video outputs.	ARG	`<output>` **options:** `0 =>` Display output `1 =>` Share output `<enable>` **options:** `0 =>` OFF `1 =>` ON	

	TX	`output=<output>`
		`enable=<enable>`
			`{`
			  `"message": <String>`
			`}`

	RX	`output=<output>`	`{`
			  `"enable": <enable>,`
			  `"message": <String>`
			`}`

It is also possible to embed arguments to an API call inside the URL to ease configuration with some control systems with the following topology:

`GET` `https://<IP>/api/v1/`**`<COMMAND>`**`?`**`<ARG1>=value`**`&`**`<ARG2>=value`**

where **`<COMMAND>`**, **`<ARG1>`** and **`<ARG2>`** are command and associated arguments.

For example, using the **`usbHost`** command, you can issue the following request:

`GET` `https://<IP>/api/v1/`**`usbHost`**`?`**`host=1`**

This request will set the USB host to laptop USB-C port.

The following commands allow to perform password management, bearer token management and firmware update. The authentication used is basic auth, and we use the same user and password as the webpage (default user=admin password=admin<LAST-4-digits-serial-number>).

Command URL / Description	Body arguments	Return body

HTTP POST
 `https://<IP>/api/v1/`
 `changePassword?`
 `password=<newPassword>`
 	
 Change the webpage password to <newPassword>.
		`{`
		     `"message": <String>`
		 `}`

HTTP GET
 `https://<IP>/api/v1/`
 `getAccessToken`
 	
 Return the bearer token.
		`{`
		     `"token": <String>`
		 `}`
		
		 If no bearer token is set, the "token" field will be null.

HTTP POST
 `https://<IP>/api/v1/`
 `generateAccessToken`
 	
 Generate random access token.
		`{`
		     `"message": <String>`
		 `}`

HTTP POST
 `https://<IP>/api/v1/update`
 	
 Sends update file to device.
	Must use formdata body.
	 The key must be myFile, and the value is of type file. We expect a .wic file that should be present in our official update packages
		`{`
		     `"message": <String>`
		 `}`

TELNET

You can use any telnet application in order to communicate with the device using TCP. Make sure to use the right IP address and **port 50000**. You can configure the telnet connection with a username/password to increase security.

Use the serial communication protocol to configure the device.

You can use our INOGENI Maestro application to monitor firmware information and upgrade your unit.

> **NOTE:** You need to use the USB-B to USB-A cable provided with the box for the Maestro application to detect the unit.

Figure 4: INOGENI Maestro application preview

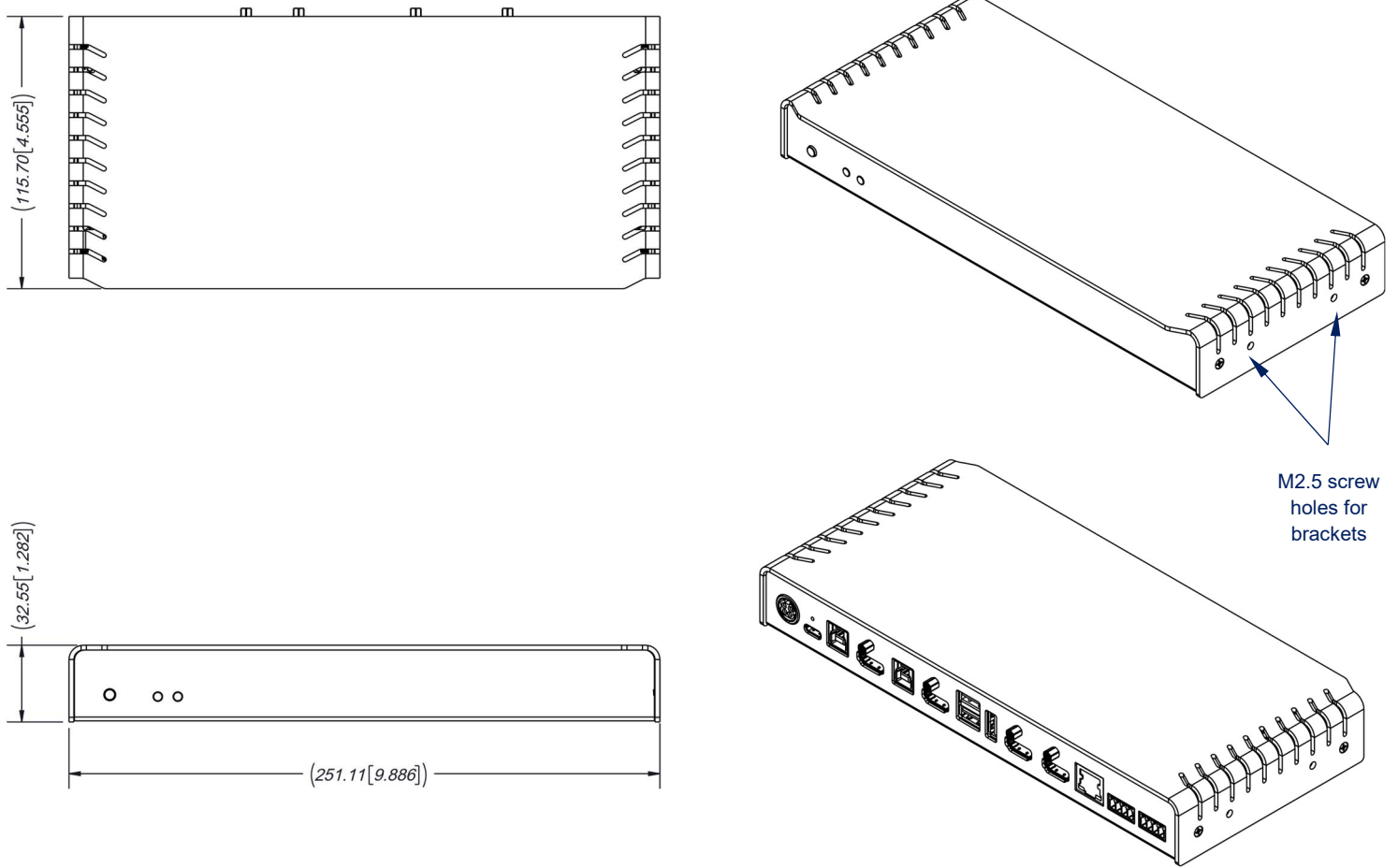You can find the mechanical specification of the device. All dimensions are in **mm [in]**.
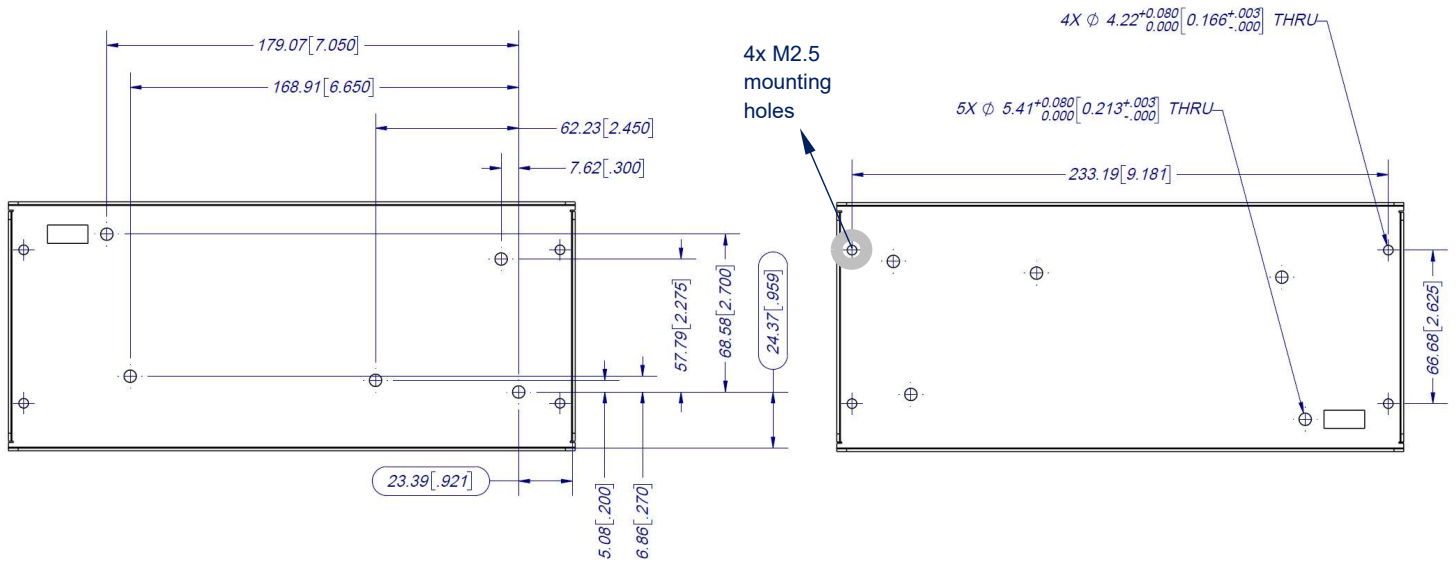

Figure 5: Top plate dimensions
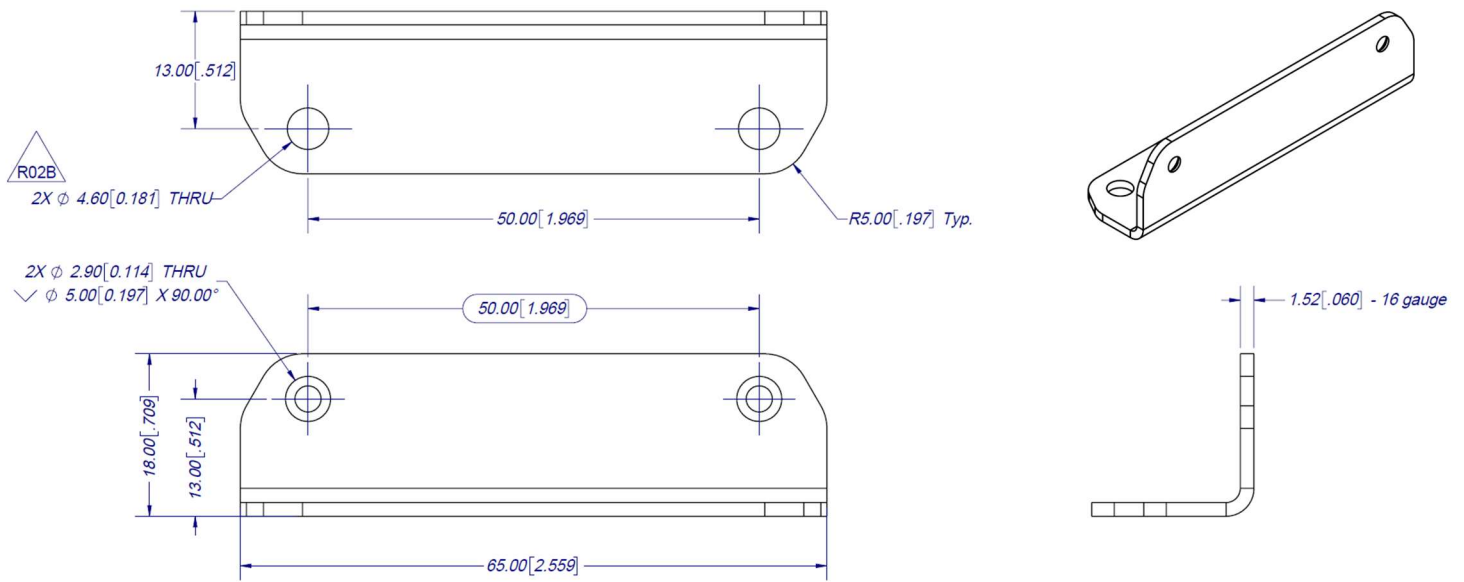

Figure 6: Bottom plate dimensions and holes positions

Figure 7: Bracket dimensions
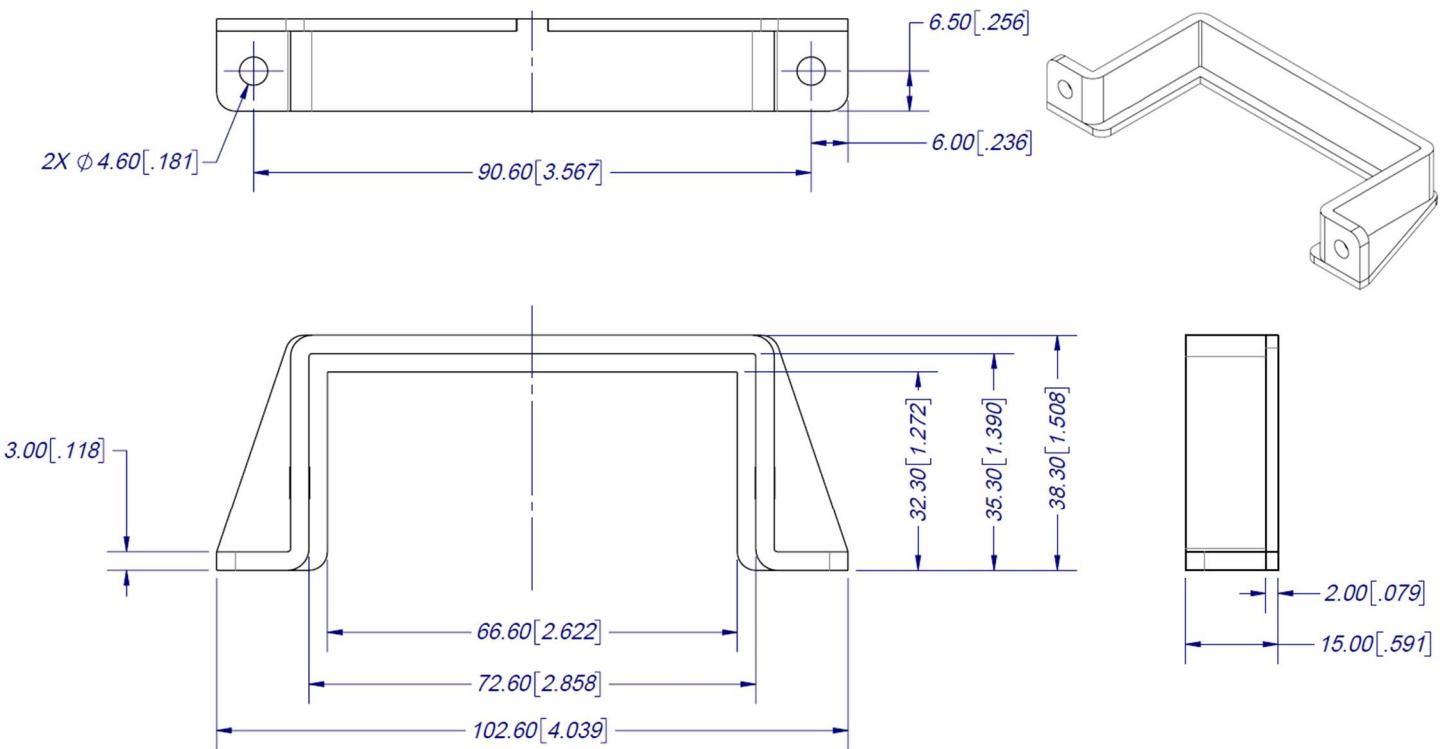
Figure 8: Power supply bracket dimensions

DIP SWITCHES

Here you can find the behavior of the DIP switches located at the back of the unit.

Switch	Position	Description
SW1	OFF / ON	For future use.
SW2	OFF / ON	For future use.
SW3	OFF / ON	For future use.
SW4	OFF / ON	For future use.
SW5	OFF / ON	Reserved.
SW6	OFF	Disable 5V on terminal block
	ON	Enable 5V on terminal block. This switch must be set to power up the connected remote.

TROUBLESHOOTING SECTION

Here is the troubleshooting section for the device.

Problem	Resolution
My laptop is not charging using my USB-C cable.	Check if the cable is rated to support USB-C power delivery. Also check if the cable used is among the ones that we already support. Visit https://inogeni.com/product/toggle-rooms/ for the complete list.

CERTIFICATIONS

FCC Radio Frequency Interference Statement Warning
This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions:
(1) this device may not cause harmful interference, and
(2) this device must accept any interference received including interference that may cause undesired operation.

IC Statement
This Class A digital apparatus complies with Canadian CAN ICES-3(A)/NMB-3(A).

CE Statement
We, INOGENI Inc., declare under our sole responsibility that the Toggle Rooms, to which this declaration relates, is in conformity with European Standards EN 55032, EN 55035, and RoHS Directive 2011/65/EU + 2015/863/EU.

UKCA Statement
This device is compliant with the Electromagnetic Compatibility Regulations 2016 No. 1091 as part of the requirements leading to the UKCA marking.

WEEE Statement
The European Union has established regulations for the collection and recycling of all waste electrical and electronic equipment (WEEE). Implementation of WEEE regulations may vary slightly by individual EU member states. Please check with your local and state government guidelines for safe disposal and recycling or contact your national WEEE recycling agency for more information.

RCM Statement
This device is compliant with Regulator Compliance Mark (RCM) certification.