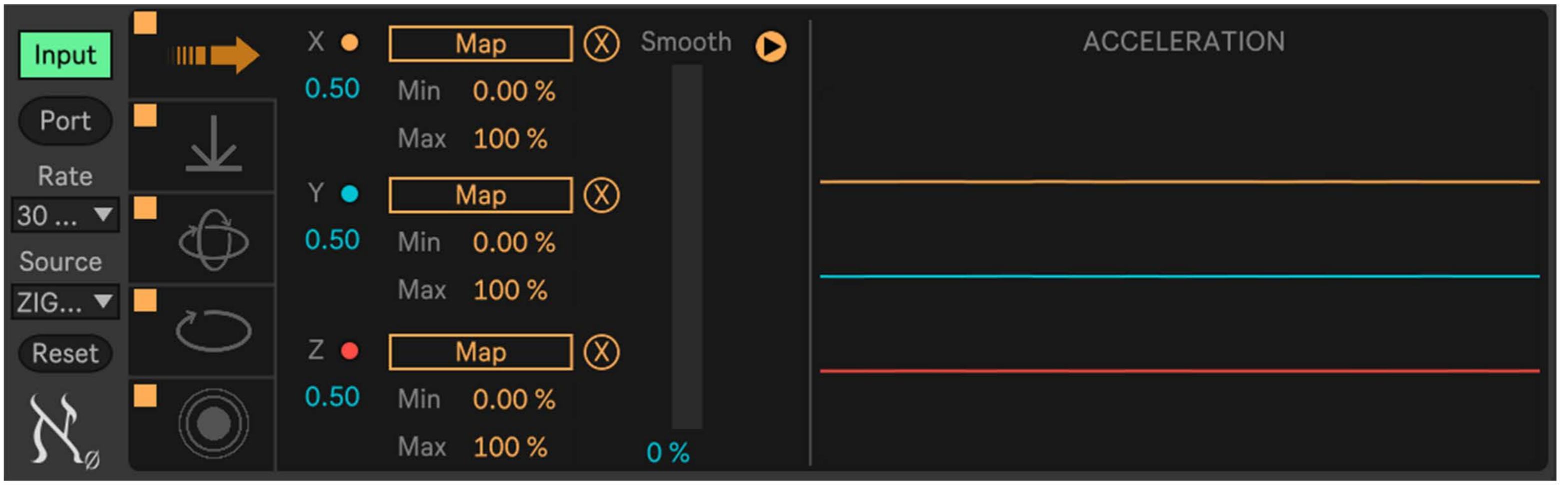


Rate 30 ...

Source ZIG...

X 0.50 Min 0.00 % Max 100 %

Y 0.50 Min 0.00 % Max 100 %

Z 0.50 Min 0.00 % Max 100 %

Smooth 0 %

ALEPH NULL

A0.Gest

OSC Motion Control Device

User Manual

v1.2

Table of Contents

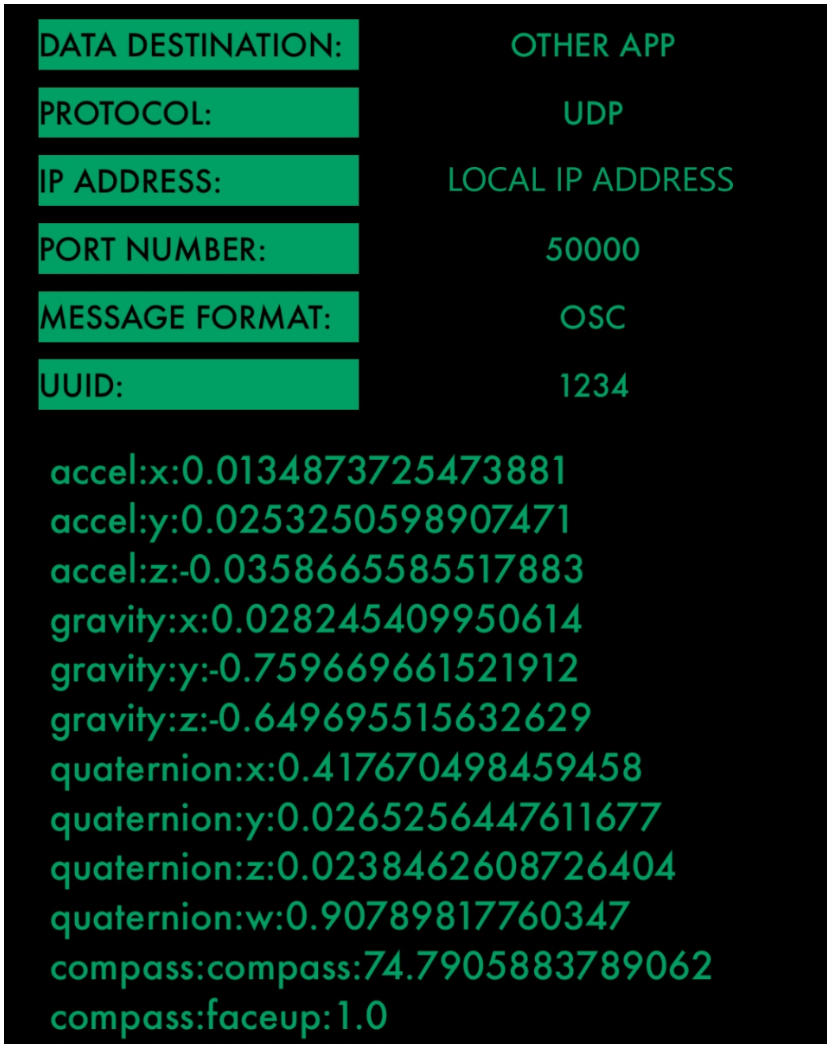
Initial Setup

1. Download the **ZIG SIM** app from the Apple App Store or Google Play store to the device that will be used for motion capture.
2. Find the local IP address of the computer that will be running A0.gest.
    a. **Windows:**
        i. Open command prompt.
        ii. Type “**ipconfig**” and press enter.
        iii. The address next to **IPv4 Address** is your local IP address.
    b. **OSX:**
        i. In the Spotlight search, type “**Terminal”** and press enter.
        ii. Type “**ipconfig getifaddr en0”** and press enter.
        iii. Your local IP address will be displayed in Terminal.
3. For first time setup of ZIG SIM, go to the Settings tab and configure the settings as follows:

    Data Destination → **Other App**
    Protocol → **UDP**
    IP Address → **local IP address**
    Port Number → **50000**
    Message Format → **OSC**
    Message Rate (per sec) → **30**
    Device UUID → **1234**

    - After this initial setup, you should not need to enter these settings again for the same network.
4. Navigate to the **Sensor** tab in ZIG SIM and enable the following sensors:
    1. ACCEL
    2. GRAVITY
    3. QUATERNION
    4. COMPASS
    5. 2D TOUCH
5. Place a single instance of A0.gest on any audio or MIDI track in Ableton Live.

6. In ZIG SIM, open the **Start** tab to begin sending your device's motion data to A0.gest.
7. If configuration is successful, the settings in the top half of ZIG SIM should be colored green with data from each sensor flowing below it. The axis indicators in A0.gest should also be lit up and receiving data, and the visualizations should be colored and moving.



DATA DESTINATION:	OTHER APP
PROTOCOL:	UDP
IP ADDRESS:	LOCAL IP ADDRESS
PORT NUMBER:	50000
MESSAGE FORMAT:	OSC
UUID:	1234

```
accel:x:0.0134873725473881
accel:y:0.0253250598907471
accel:z:-0.0358665585517883
gravity:x:0.028245409950614
gravity:y:-0.759669661521912
gravity:z:-0.649695515632629
quaternion:x:0.417670498459458
quaternion:y:0.0265256447611677
quaternion:z:0.0238462608726404
quaternion:w:0.90789817760347
compass:compass:74.7905883789062
compass:faceup:1.0
```

Fig. 1 ZIG SIM running successfully. Note that the touch sensor's input will not appear in the list of sensor inputs unless you are actively touching the screen.

How to Use

After the initial setup, running A0.gest is simple:

1. Open ZIG SIM on your smartphone.
2. Place A0.gest on any MIDI or audio track in Ableton Live.
3. Open the **Start** tab in ZIG SIM to begin sending your device's motion sensor data to A0.gest.

Once A0.gest is successfully receiving motion data, proceed to mapping.

Mapping

A0.gest has an individual map button assigned to each axis of every sensor (for more information on each sensor's functions, see page 7).

To map a sensor's axis to a parameter in Ableton:

1. Click the desired **Map** button so it begins flashing, indicating it is ready to map to a parameter.
2. Click the desired parameter to map.

The map button should now display the parameter's name, indicating that it is successfully mapped.

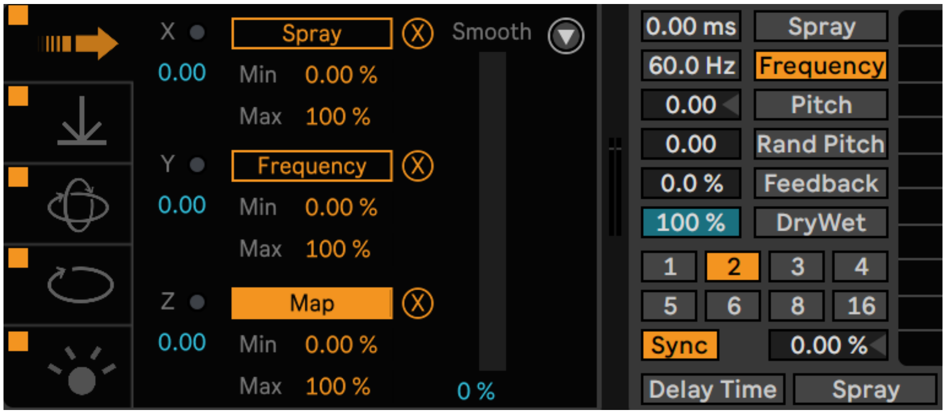


Fig. 2 The x and y-axes are mapped to Spray and Frequency, while the z-axis is ready to map.

To clear the mapped parameter, click the **X** button to the right of the respective map button. The map button now displays **Map** again, showing it is ready for a new parameter.

Configuring Parameters

Once a parameter is mapped, it can be configured in three ways:

1. The minimum value can be set with the **Min** parameter underneath its respective map button.
   - This is a percentage of the maximum value of the parameter.
2. The maximum value can be set with the respective **Max** parameter.
   - Configuring the Min and Max values allows for a range to be defined for each mapped parameter.
   - To **invert a signal** (e.g: cause the x-axis of gravity to decrease a parameter when rotated to the right rather than increase it), set Min to a higher value than Max (e.g: Min 100%, Max 0%).
3. The **Smooth** slider is used to smooth out the incoming signal for each respective sensor. Use this to create slower, more gradual changes to the mapped parameter.

The following diagram explains the primary functions of A0.gest:

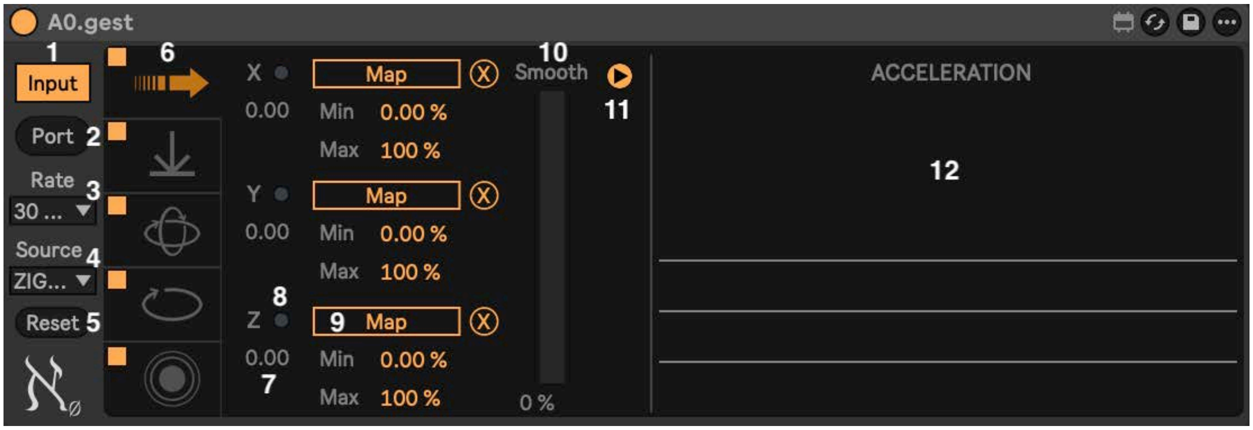


Fig. 3 The primary UI of A0.gest. Each label is named and described below.

Label	Name	Function
1	Input button	Turn on/off receiving input from all sensors.
2	Port button	Set the port number to match the sending device.
3	Rate	Set the rate A0.gest will allow sensor messages through.
4	Source	Change the source of the motion sensor data.
5	Reset	Reset all sensor values to 0.
6	Tab selection	Select a sensor to view. Turn on/off with the buttons.
7	Axis value	Displays the live numeric value of each sensor's axes.
8	Axis indicator	Lights up when actively receiving sensor input.
9	Map	Click to map, clear the mapping, and set min/max values.
10	Smooth	Apply smoothing to the incoming signal for each sensor.
11	Show display	Show or hide the visualizer for each sensor.
12	Visualizer	Display the live reading of each sensor.

How A0.gest Works

OSC

"OpenSoundControl (OSC) is a data transport specification (an encoding) for realtime message communication among applications and hardware" developed at CNMAT.[1] In other words, OSC is a way of formatting and packaging a collection of data to be received and unpackaged elsewhere in realtime.

OSC is an exceptionally powerful, lightweight, and flexible way of sending data between devices. It can be thought of as a more robust alternative to MIDI and supports a wide range of data types.

In A0.gest, OSC is used to send motion sensor data from a device, as well as to receive and format the incoming data into Ableton Live.

Altogether, it works as follows:

1. ZIG SIM packages the live reading of your smartphone's motion sensors, then sends it all as a single bundle 30 times a second to A0.gest.
2. A0.gest receives, unbundles, and sends each sensor's data to its respective location in the Max for Live device.
3. The realtime readings of each sensors' axes by A0.gest can then be used to modulate whichever parameter(s) within Ableton Live you choose to map to.

Sensors

Acceleration

The accelerometer sensor measures the force of acceleration on the device. Because gravity is always pulling on the device, a high-pass filter is automatically applied to the sensor to subtract the force of gravity (**g** = 9.81 m/s^{2}) and isolate the real acceleration of the device.[2]

This sensor is highly flexible when using the smoothing function, allowing both the most rapid and jumpy parameter changes of any sensor with smoothing off, as well as fluid sweeps when smoothing is increased.

Acceleration is best used when wanting to emphasize motion and is less ideal when precise or sustained values are required.

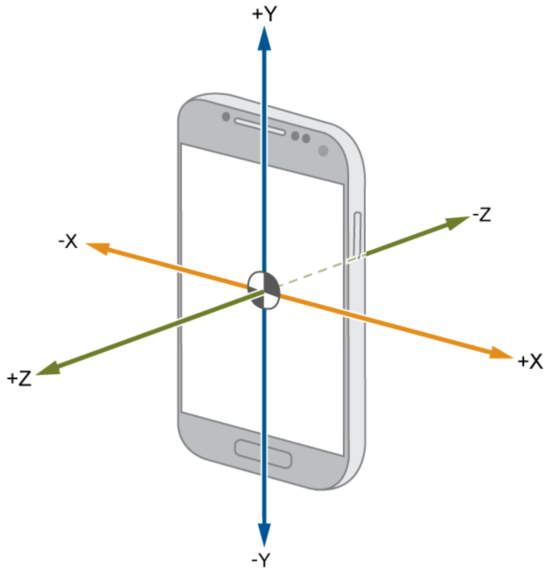


Fig. 4 The directions of the accelerometer's axes.[4]

⊻ Gravity

Gravity is a software sensor created via *sensor fusion* (combining two sensors together to produce a new sensor signal) using the gyroscope and accelerometer. This allows for the acceleration due to movement to be filtered out and leave only the pure gravity signal.

Typically, this sensor is used to determine the device's relative orientation in space.

This provides a much cleaner and more stable signal than the gyroscope alone and provides a three-dimensional vector indicating the direction each surface on your device is facing.

This is best used when precise or sustained values are desired, as well as fluid modulation between values.

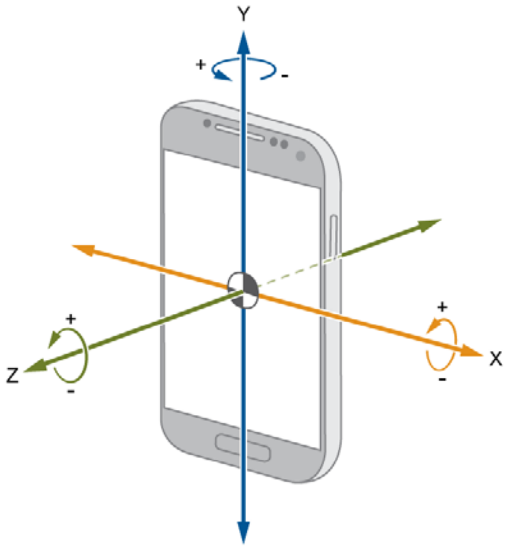


Fig. 5 The directions of the gravity sensor's axes.[4]

Quaternion

The quaternion represents the rotation of the device in 3D space. This is a more robust way to measure rotation than the gyroscope, as the quaternion can be used to rotate and scale the vector as movement occurs and thus produce a more accurate representation of the device's movement.

Unlike other sensors, this is represented in four axes: x, y, z, and w. The axis w is a product of the first three axes that allows for rotation and distance to be represented accurately.

This sensor is best used when wanting the closest representation of the device's motion in space by the mapped parameters. The four axes also provide the largest number of mapping possibilities of any sensor.

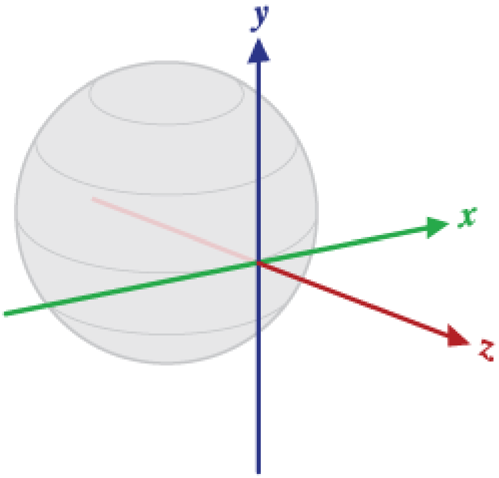


Fig. 6 The directions of the quaternion's real axes.[3]

Compass

The compass sensor measures the cardinal direction the device is pointing. Within A0.gest, this is measured through the degrees of the compass (0 – 360°). Because there is only a single axis of measurement, there is only one mappable parameter for compass.

This sensor is best used when wanting to emphasize the direction you are facing. This is particularly useful for performances, where facing toward or away from the audience can produce radically different sonic results.

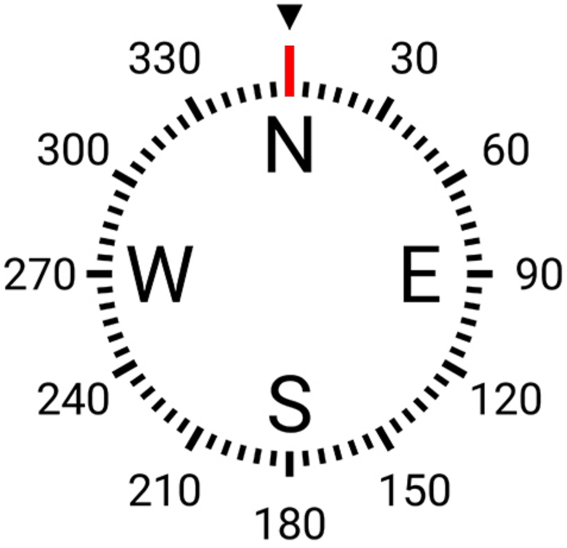


Fig. 7 The degrees of the compass corresponding to cardinal directions.

◉ Touch

The touch sensor detects the position of 2D touch on the device's screen. The x-axis corresponds to the horizontal position of the touch, and the y-axis to the vertical position.

The visualizer for touch maps directly to the size and position of the screen, so that the bottom corresponds directly to the bottom of the screen, the top corresponds to the top of the screen, etc.

The touch sensor is great for getting precise values, dialing in parameters that are subsequently locked in regardless of motion, and modulating any two parameters as a matrix.

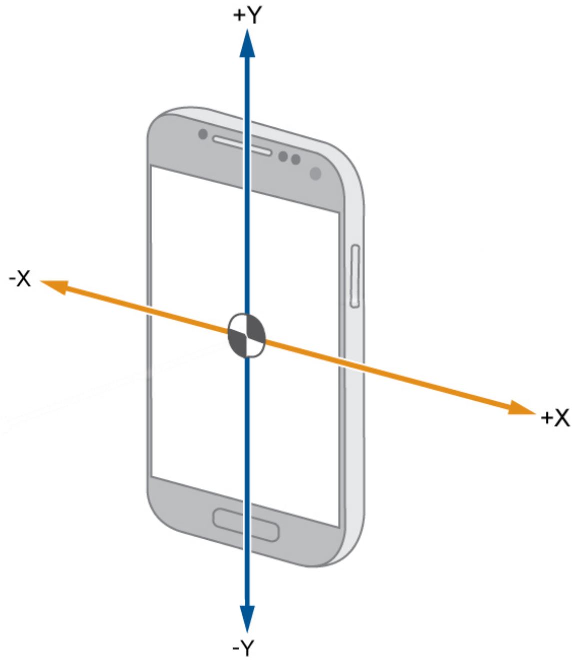


Fig. 8 The directions of the 2D touch sensor across the device's screen.

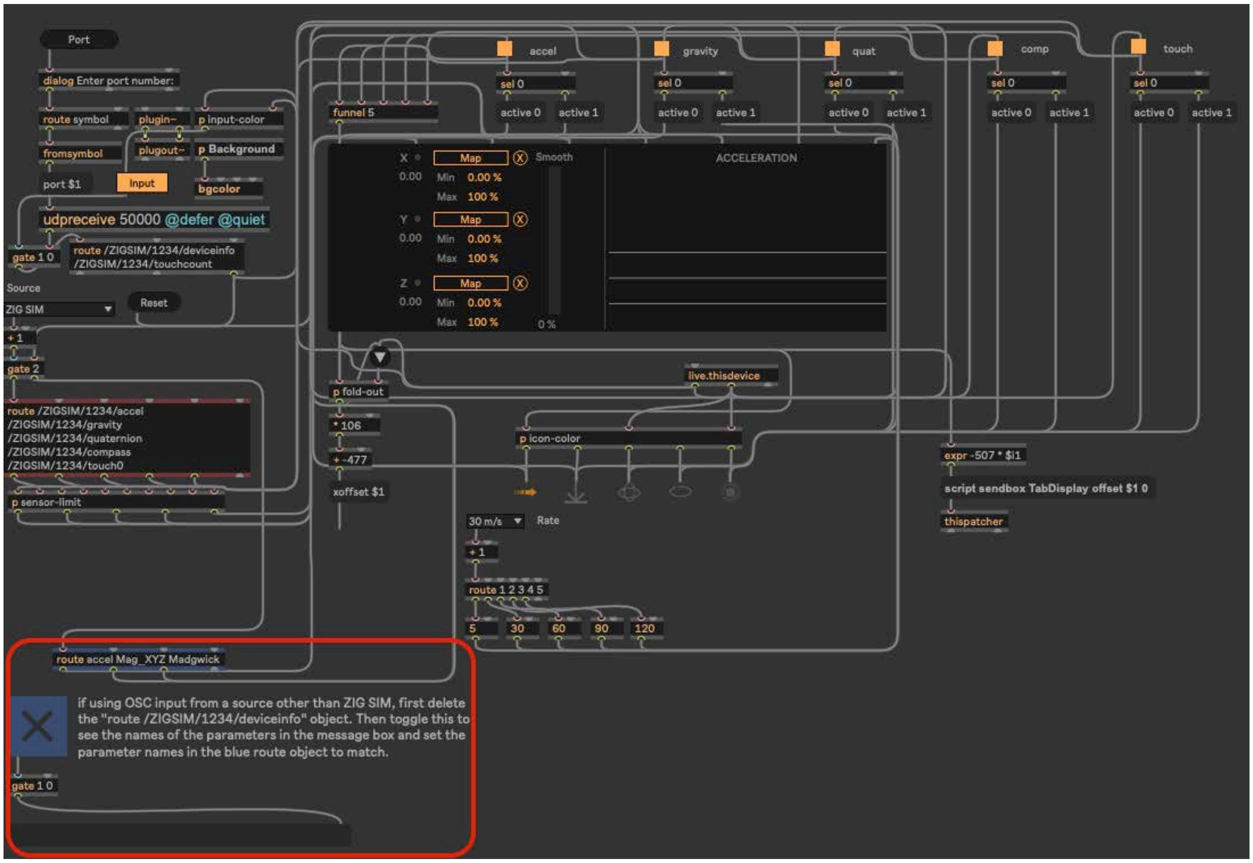
Troubleshooting

The device does not display input while ZIG SIM is sending:

1. Check that the device has correct network settings.
    a. Ensure the IP address being sent from the device is the same as the computer A0.gest is running on.
        i. On Windows, open command prompt and type "**ipconfig**".
        ii. Copy the address next to **IPv4 Address** to the sending device.
    b. Ensure the port being used is free and matches between the sending and receiving devices.
        i. On **Windows**, open command prompt and type **netstat -an**.
            a. Look for the ports marked **UDP** and ensure the default port 50000 is listed. If it is, move to step 2.
            b. If port 50000 is not listed, pick a different UDP port. Change the port in ZIG SIM listed under **Port Number** in the **Settings** tab, as well as in A0.gest by clicking the **Port** button on the left of the device and entering the new port in the dialog.
        ii. On **Mac**, open Terminal and type "**sudo lsof -i -n -P | grep UDP"**
            a. Look for the ports marked **UDP** and ensure the default port 50000 is listed. If it is, move to step 2.
            b. If port 50000 is not listed, pick a different UDP port. Change the port in ZIG SIM listed under **Port Number** in the **Settings** tab, as well as in A0.gest by clicking the **Port** button on the left of the device and entering the new port in the dialog.
2. Check firewall settings. Both **Max (32-bit)** and/or **Max (64-bit)** AND **Ableton Live 11 Suite** / **Ableton Live 10 Suite** need to be unblocked in your computer's firewall.
3. Use a new port in ZIG SIM and A0.gest.
    a. Try other UDP ports listed in netstat.
    b. Try entering random 5-digit numbers (11111, 12345, etc.).

I’m using a device other than a smartphone, or an app other than ZIG SIM and the device does not display input for one or more parameters:

1. Check that the sensor data being sent is in the same format as ZIG SIM.
    a. Open the patcher window by clicking the (edit Button) at the top of the device.
    b. Enter patching mode by clicking the ❄ (freeze button) and (patching mode button) in the bottom toolbar of the patch. Ensure the patch is unlocked (🔓) in the bottom toolbar as well.
    c. Delete the **route /ZIGSIM/1234/deviceinfo** object and reconnect the path from **gate 1 0** to the route object’s previous destinations.
    d. Click the **blue toggle** in the bottom left of the patch. This is provided to allow you to view the parameter names and formatting of any incoming OSC messages.
        i. If no messages are appearing in the connected message box, move on to step 2.
    e. Follow the instructions next to the toggle box to change the parameter names in the **route** object to fit the incoming OSC messages.



if using OSC input from a source other than ZIG SIM, first delete the "route /ZIGSIM/1234/deviceinfo" object. Then toggle this to see the names of the parameters in the message box and set the parameter names in the blue route object to match.

Fig. 9 The area for debugging the sensor input is indicated in the red square.

2. Check that the device has correct network settings.
    a. Ensure the IP address being sent from the device is the same as the computer A0.gest is running on.
        i. On Windows, open command prompt and type **ipconfig**.
        ii. Copy the address next to **IPv4 Address** to the sending device.
    b. Ensure the port being used is free and matches between the sending and receiving devices.
        i. On Windows, open command prompt and type **netstat -an**.
        ii. Look for the ports marked **UDP** and ensure the default port 50000 is listed. If it is, move to step 3.
        iii. If port 50000 is not listed, pick a different UDP port. Change the port in your sending device, as well as in A0.gest by clicking the **Port** button on the left of the device and entering the new port in the dialog.
3. Check firewall settings. Both **Max (32-bit)** and/or **Max (64-bit)** AND **Ableton Live 11 Suite** / **Ableton Live 10 Suite** need to be unblocked in your computer's firewall.
4. Use a new port in ZIG SIM and A0.gest.
    a. Try other UDP ports listed in netstat.
    b. Try entering random 5-digit numbers (11111, 12345, etc.).
5. Change the device UUID in ZIG SIM Settings.
    a. Change the UUID to a new random number (1, 12345, etc.).

My CPU usage in Ableton Live is peaking / A0.gest is stuttering:

This is likely due to older hardware. To reduce CPU load:

1. Set **Rate** to a higher value, such as 60, 90, or 120 messages / second.
2. Lower the sending message rate:
    a. Open ZIG SIM → Settings.
    b. Under **“Message Rate(Per Sec)”**, lower the rate to the next lowest option until stuttering stops.
3. Set a higher buffer size in Ableton.
    a. **Live → Settings → Audio.**
    b. Set **Buffer Size** to a larger value.
4. Consider freezing or bouncing tracks to reduce the overall CPU load of the project file.

Credits

A0.gest User Manual Version 1.2
for Windows and Mac

A0.gest created, designed, and programmed by Aleph Null
User Manual by Aleph Null

Selected diagrams and reference material from:

[1] Stanford CCRMA OpenSoundControl

[2] Android Developers Manual

[3] Apple Developer Documentation

[4] Simulink Support Package for Android Devices

https://a0alephnull.github.io

aleph.a0.null@gmail.com