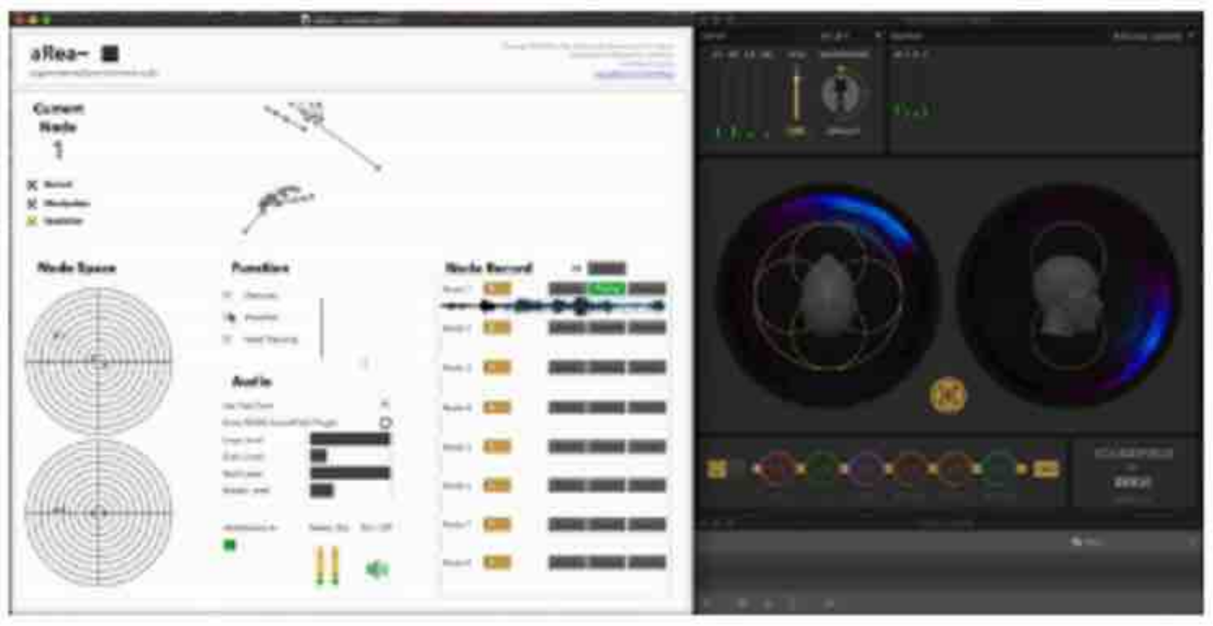
# Home

Sam Bilbow edited this page last week · 9 revisions

## `area~`



**Exploring Real and Virtual Environments Through Gestural Ambisonics and Audio Augmented Reality**

Platform Mac | Environment Max | Publication SonicScope | Guide Wiki | Project Blog | Discord XRt Space

The area~ system, which stands loosely for 'augmented reality environmental audio' aims to afford users the ability to spectro-morphologically manipulate sounds from their environment into a virtual audio environment. Through bone conduction headphones and head tracking, this sound field is heard in synchronicity with their actual environment. The system was created in order to explore and reveal the relationship between real and virtual environments. The following were the results:

- The blending of real and virtual auditory environments to create a third, augmented environment that was greater in experiential nature than the sum of its parts (not simply a combinatorial layering)
- The ability to spectromorphologically manipulate sounds in real-time in this third environment with the body
- The potential for creating believable illusions of real-world sound sources from these manipulated and spatialised virtual sounds.

To get started with `area~`, follow the guide shown on the sidebar of this page.




## Citation

Bilbow, S. (2021). The area~ System: Exploring Real and Virtual Environments Through Gestural Ambisonics and Audio Augmented Reality. Sonic Scope: New Approaches to Audiovisual Culture. https://doi.org/10.21428/66f840a4.b74711a8

or with BibTeX

Pages 7

## `area~` Guide

- Getting Started

### Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

### Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

### Musicking

- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

Clone this wiki locally

`https://github.com/sambilbow/ar`

# Software Requirements

Sam Bilbow edited this page last week · 8 revisions

## Compatibility

- Tested with macOS 10.15 (Catalina)
- Tested with Max 8.0.8
- Tested with LeapMotion SDK 2.3.0

## Installation

- Download the repository as a zip or
- `git clone https://github.com/sambilbow/area` in your terminal emulator.
  - ~~You will need `git-lfs` installed to do this due to the size of `test.wav`~~
  - `test.wav` deprecated due to lfs pricing. Contact me if you require an ambisonic test file, or download one online (e.g. link)

## Max 8

### Abstractions Included

- `nodelooper.mxo` (looping patch using on karma~)
- `envlooper~.mxo` (ambisonic looper built using nodelooper and mc)
- `serialparse.mxo` (takes care of parsing the serial stream from the ESP32)
- `rgrain.mxo` (modified rgrain.maxpat from C'74 Examples)
- `granulator.mxo` (modified rgrano.maxpat from C'74 Examples)
- `rchoose.mxo` (in C'74 Examples)
- `transratio.mxo` (in C'74 Examples)

### Externals

- karma~ by Rodrigo Constanza available on GitHub - version 1.0
  - `karma~.mxo`
- Ambisonics Externals by ICST available on zhdk.ch - version 2.3.2
  - `ambidecode~.mxo`
  - `ambiencode~.mxo`
  - `ambimonitor.mxo`
- Leapmotion for Max by Jules François available on GitHub
  - Requires LeapMotion SDK 2.3.0 available from their V2 archive
  - `leapmotion.mxo`

## Plugins

- RØDE Soundfield Plugin available on RØDE website

Pages 7

**area~** Guide

- Getting Started

Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

Musicking

- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

Clone this wiki locally

`https://github.com/sambilbow/ar`

# Hardware Requirements

Sam Bilbow edited this page last week · 4 revisions

## Hand Tracking

- Leap Motion Controller (info)

## Audio Input

- Ambisonic Microphone (used RØDE NT-SF1)
- 4 Channel Audio Interface with 48V and AmbiX output

## Audio Output

- Bone Conduction Headphones (used Aftershokz Aeropex, now Shokz Openrun)

## Head Tracking

- ESP32 - Devkit C (info)
- 18650 Lithium Ion Battery (info)
- 18650 Wemos Battery Shield with Charging Unit (example)
- USB cable for charging the Battery
- SPDT Toggle Switch (info)
- MPU9250 IMU (info)
- 2m of stranded wiring for IMU
- 1x JST-XH 4pin socket-plug pair
- 1x JST-XH 2pin socket-plug pair
- Stripboard or Breadboard
- Assorted coloured solid wiring for stripboard connections

Pages 7

### area~ Guide

- Getting Started

### Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

### Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

### Musicking

- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

**Clone this wiki locally**

https://github.com/sambilbow/ar

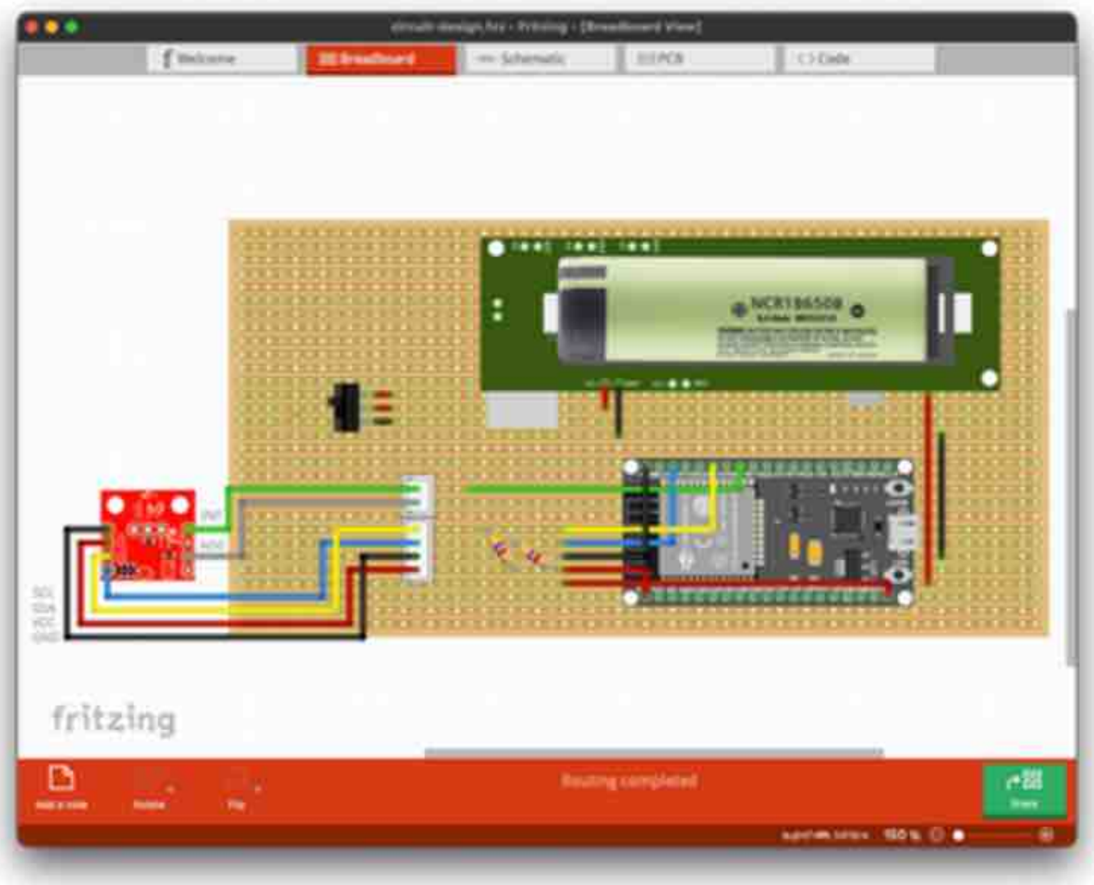
# Building Hardware

Sam Bilbow edited this page last week · 6 revisions

## IMU Connections

1. Form 2 metres of cabling, I heatshrunk the cables together, you'll need 6 in total.
2. Crimp JST-XH plugs onto the cables. I chose to bundle the VCC, GND, SCL, SCA into a 4 pin connector, and the AD0 and INT into a 2 pin connector.
3. Solder the cabling onto the IMU, taking care to note down which cable is connected to each pin
4. Solder the JST-XH corresponding sockets to your stripboard.
5. Solder resistor connectors
6. Plug in the IMU cables.
7. Solder solid core cabling from the strips connected to each JST-XH socket pin, via resistors if specified, to the following ESP32 pins.

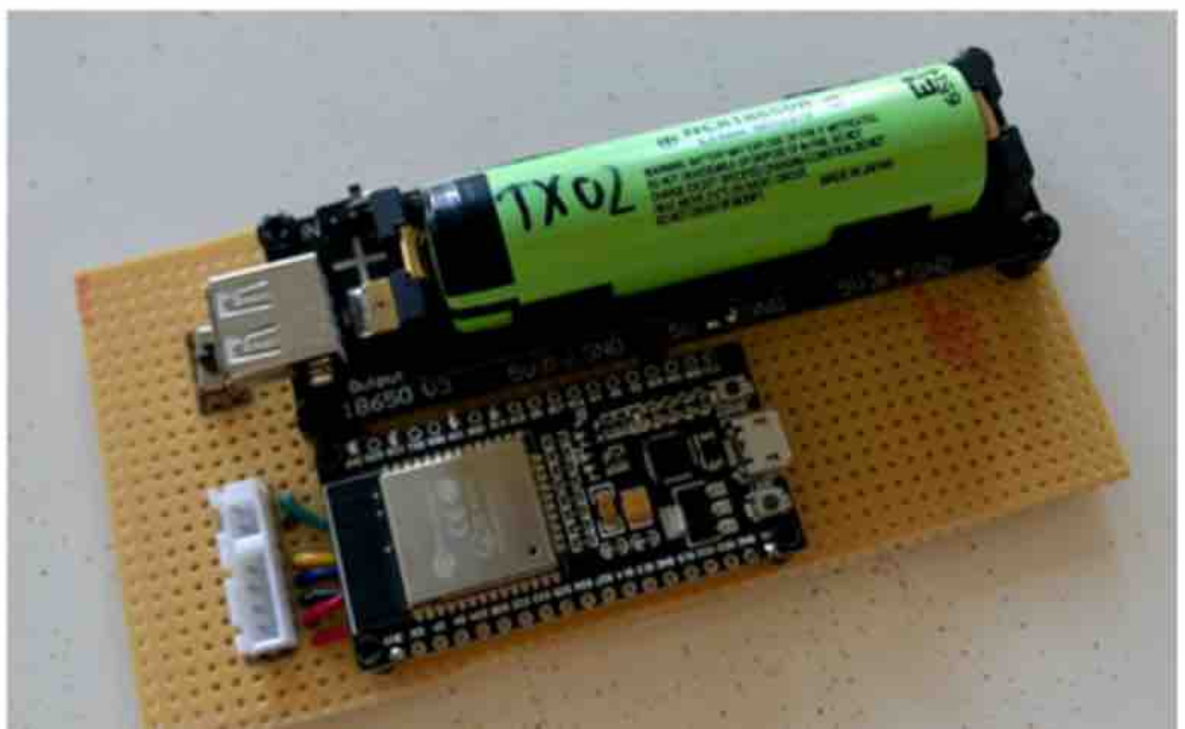
| IMU Pin | ESP32 Pin |
|---|---|
| VCC | 3V3 |
| GND | GND |
| SCL/SCLK | GPIO22 via 4.7 kΩ to 3V3 |
| SDA/SDI | GPIO21 via 4.7 kΩ to 3V3 |
| AD0/SDO | None |
| INT | GPIO19 |

| Soldering in Progress | Fritzing Schematic from /area/pcb/ |
|---|---|
|  |   |

## Battery Connections

1. Connect the battery shield to the ESP32
2. Use an SPDT toggle switch if you wish to be able to toggle the power, and conserve charge when not using
3. Put the 18650 cell in to the shield

| 18650 Shield | ESP32 Pin |
|---|---|
| 5V | 5V |
| GND | GND |



Pages 7

area~ Guide

- Getting Started

Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

Musicking

- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

Clone this wiki locally

`https://github.com/sambilbow/ar`

## Arduino Sketch

`under (re)construction`

1. Due to data corruption, the actual sketch has been lost. It drew on this sketch by Kris Winer, that uses open source sensor fusion algorithms to increase the stability and accuracy of the sensor. Adapting the sketch involved removing redundant measurements from the serial print calls. At a later date I may have time to create the sketch again.
2. The only data needed from the IMU is the head rotation. This can tested by sending all three rotation values via Serial, rotating your head whilst the sensor is attached to the bone condudction headphones, and selecting the measurement which corresponds with the axis that experiences the largest delta.
3. The sketch used the onboard Bluetooth capabilities of the ESP32 to send the serial data via BT.
4. Flash the ESP32 via USB (you will require your ESP32's Arduino Library, as well as esptool installed).

## Headphones / Wearable

1. Place circuit inside a wearable pouch - I used a belt pouch for a PDA I found online
2. Orient circuit so that USB B Micro on the Battery Shield is accessible for charging.
3. Connect the IMU to the bone conduction headphones. I did this via velcro, so that it was somewhat adjustable while using.

# Running Software

Sam Bilbow edited this page last week · 4 revisions

## Prerequisites

1. You have all the Software Requirements
2. The ambisonic microphone is plugged in, powered by 48V and selected in Max's audio options
3. The wearable IMU / bone conduction headphones are on, connected to the host computer, and transmitting data.
4. The bone conduction headphones are selected as the audio output in Max's audio options

## Max 8 patch

1. Open `area~.maxpat` in Max 8
2. Connect to the appropriate serial device
   i. Leave presenter view (cmd-option-e)
   ii. Open `[p serial]`
   iii. Enter edit mode (cmd-e)
   iv. Change `[serial b 115200]` to reflect your serial device's index
   v. Leave edit mode (cmd-e)
   vi. Close `[p serial]`
   vii. Enter presenter view (cmd-option-e)
3. Set up the patch audio settings
   i. Turn on the Stereo Out `[ezdac~]`
   ii. Loop Level `[slider]`
   iii. Grain Level `[slider]`
   iv. Notification Level `[slider]`
   v. Main Level `[slider]`
   vi. Check the RØDE Plugin to ensure signal from the microphone is being received
   vii. (optional) If not using a microphone, place an ambisonic file called `test.wav` in project root folder, and toggle on `Use Test Track [x]`
4. Turn on the function toggles
   i. `[x] Gestures`
   ii. `[x] Visualiser` (optional)
   iii. `[x] Head Tracking`
      a. The `ambimonitor` on the left side of the patch should now show two points rotating as your head moves

Pages 7

## area~ Guide

- Getting Started

### Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

### Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

### Musicking

- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

# Gesture Workflow

Sam Bilbow edited this page last week · 2 revisions

## Global Controls

Start recording your composition (optional) by clicking the `[o] (bang)` in the top left section of the patch, next to `area~`. This should record to the project root, two binaural `.wav` files, containing the real and virtual audio capture.

More contextual information about musical parameter mappings can be found in the publication for this project. Moreover, an early visual guide to the gestures can be found in the explainer video provided, as well as the 360 binaural demonstration video with software overlay.

Publication SonicScope



## Recording

The recording or 'sampling' stage is initiated by making a left-hand grab above the LMC. The longer lasting the grab, the longer the portion of audio from the ambisonic palette is sampled. The three-dimensional coordinates of the hand above the LMC correlates with the location of audio recorded (this is achieved by mapping the hand coordinates to a virtual microphone inside the ambisonic palette), essentially allowing the user to record sounds around their person in three dimensions. Upon letting go of the grab gesture, the sample plays on repeat (using the karma~ Library) through the bone conduction headphones, thus setting up the session's virtual audio environment.

## Manipulating

The manipulation stage is automatically initiated after the ending of the previous grab gesture and uses translational (x, y, z) and rotational (roll, pitch) values from both hands when above the LMC. There are two audio effects being manipulated, with parameters from these effects mapped in different ways to the translation and rotation of the user's hands.

## Spatialising

The spatialise stage begins once the manipulation stage is ended by the user. The three-dimensional space above the LMC is mapped to the virtual audio environment, in which the user is currently listening to the sample that they have recorded. The user can use their right hand to move the sample around the virtual audio environment. For an example of the effect this has, moving the hand between the two extremes of the x-axis (left to right) results in hearing the sample move from ear to ear. The spatialise stage is ended by grabbing with the right hand.

To summarise, the patch can be categorised into having two inputs: audio from the user's environment and hand gesture, and one output: the virtual audio environment. In the background, this audio input is decoded into the ambisonic palette (inaudible), which is acted on by the user's hands to form one audible output: the virtual audio environment, which is comprised of up to 8 nodes. Through the choice of sensory overlay (bone conduction) and integration of head tracking, this virtual audio environment is experienced synchronously with the user's real, multisensory environment.

Pages 7

### `area~` Guide

- Getting Started

### Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

### Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

### Musicking

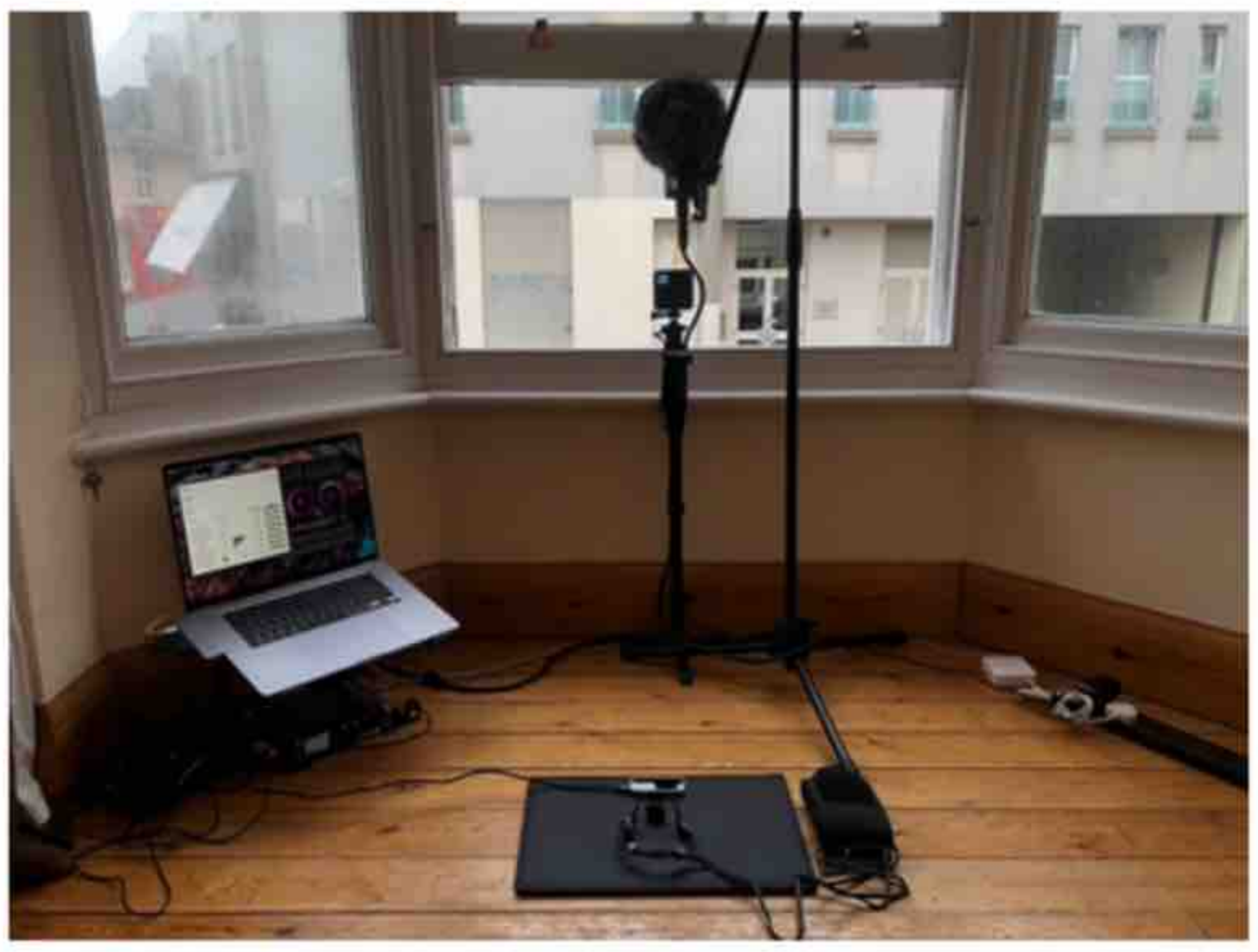
- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

Clone this wiki locally

`https://github.com/sambilbow/ar`

# Recording Setup

Sam Bilbow edited this page last week · 2 revisions



## Space

- Choose a space with an interesting sound palette!
- Monitor the audio for a while, get to know the sounds. Perhaps a deep listening exercise?
- I found it best to sit down, you're tethered by the length of your LMC USB cable anyway

## Visual

- I found that the recordings of compositions worked really well with an accompanying 360 video, I used a GoPro for this. I used GoPros tool to bounce down the two sides of 360 video into an equirectangular video.
- I also took a screen recording of the patch whilst composing, using QuickTime Player
- These were composited in Adobe Premiere Pro, with the screen view acting as a 'picture-in-picture'

## Audio

- The patch outputs (if enabled) two 4-channel B-Format `.wav` files to your user directory: `~/`, `realEnv.wav` and `virtualEnv.wav` containing the real and virtual audio capture from the patch.
- I added these to the Premiere Pro project as ambisonic audio
- I then mixed down the project to equirectangular, with ambisonic audio, for YouTube 360.

## Project Files

### Sources

- 360 footage `.mov` (converted from GoPro GBACK and GFRONT `.mp4` files)
- Screen Recording of Max 8 `.mov`
- B-Format Microphone Capture `realEnv.wav`
- B-Format Max 8 Capture `virtualEnv.wav`

### Output

- Equirectangular 360 / Ambisonic video `.mp4`

Pages 7

## `area~` Guide

- Getting Started

## Checking

- Software Requirements
  - Compatibility
  - Installation
  - Max 8
  - Plugins
- Hardware Requirements
  - Hand Tracking
  - Audio Input
  - Audio Output
  - Head Tracking

## Making

- Building Hardware
  - IMU Connections
  - Battery Connections
  - Arduino Sketch
  - Headphones / Wearable
- Running Software
  - Prerequisites
  - Max 8 Patch

## Musicking

- Gesture Workflow
  - Global Controls
  - Recording
  - Manipulating
  - Spatialising
- Recording Setup
  - Space
  - Visual
  - Audio
  - Project Files

### Clone this wiki locally

`https://github.com/sambilbow/ar`