- silent spectacle -

a [noise] pollution instalation

| final documentation |

- silent spectacle -

a [noise] pollution instalation

final documentation

„New nature in park at the Ilm”
Introductory Project-Module
Advisor: Dr.-Ing. Sabine Zierold

Bauhaus University Weimar
faculty of architecture & urbanism
Master Degree Program
MediaArchitecture

wintersemester 2020/2021

Berchtold Patricia
mtr.nr.: 122606
Paola Ferrari
mtr.nr.: 123224

| outline |

- silent spectacle -

The installation “silent spectacle” at the “Park an der Ilm” in Weimar seeks to create a new nature atmosphere, in which humans and nature co-live in balance. The installation therefore proposes a space where humans can realize their impact and distance themselves from our exploratory relation with nature. To achieve that, the “silent spectacle” turns natural data input of the park into a visible and interactive installation. In the project area, the Roman-House-Hill, a large difference in noise levels was observed at different points. The hill acts as a sound barrier, blocking the city’s sound pollution.

[Noise] pollution can deeply impact a balanced ecosystem, but this problem is almost always overviewed since it’s not very tangible in daily life. Therefore, our project wants to show the invisible effects of noise pollution by making it viewable through fog and light.

The amount of fog and light will be directly corresponding to the amount of sound pollution at the Roman-House-Hill, making the association clear and direct to pedestrians.
While the fog creates an ephemeral and romantic atmosphere, changing the pedestrians’ relation with the park, the light gives a playful and direct response to human impact.

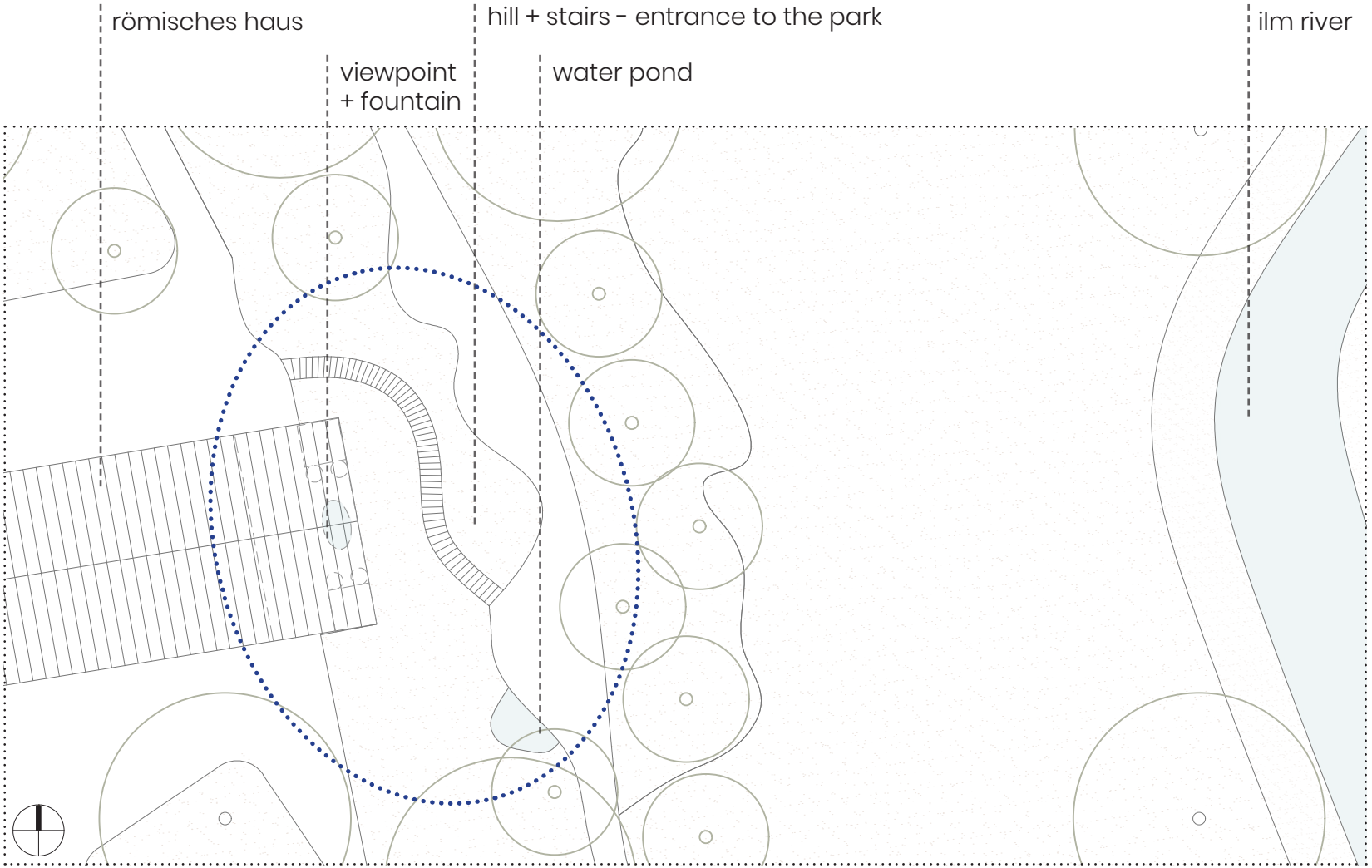
In addition to proposing this new relation with nature, “silent spectacle” proposes a response to the problem of global warming in the park as it creates a cool atmosphere for the plants that are already suffering with heat waves.

“Silent spectacle” as-a-result is an interactive installation that returns something back to nature and at the same time makes the human impact to the environment visible.

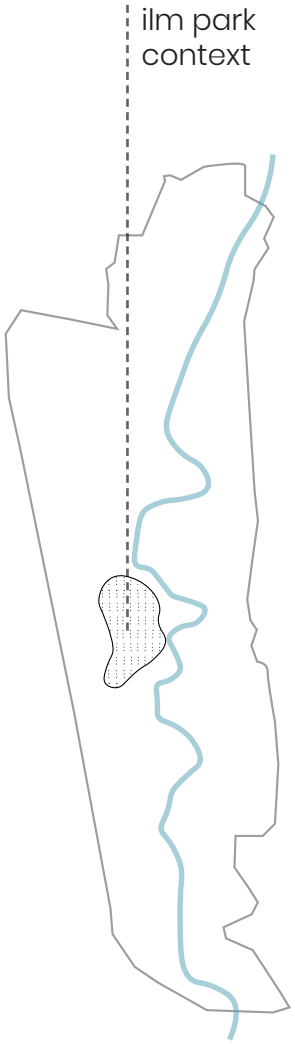
| 2. place |

The 48-hectare park on the Ilm is a unique landscape garden on the edge of Weimar's old town. Duke Carl August and Johann Wolfgang Goethe realized their horticultural ideas here. They created a walk-in work of art with varied landscapes, park architecture and seating areas, which to this day serves for recreation, education and aesthetic enjoyment of nature.
Goethe and Carl August together planned the first grounds in the new English taste between the town, the palace and Goethe's garden house.
Starting in 1778, the design of the western Ilmhang with its wooded backdrops, walkways and park architecture was created.

The construction of the Roman House, completed in 1797, marked the high point of the park's development after the large-scale expansion of the park to the south and the integration of older palace gardens.
We chose the Park-facing side of the Roman House as our projects place, including the viewpoint over the Ilm valley and the hill in front of it. A Staircase forms its way down the hill functioning also as a sound barrier to the city noises and allows the visitor to gain a quieter entry to the park.
The Poem at the end of the staircase was written in 1790 by Goethe and shows how he imagined this place to let the people come to more observing stillness.



site plan 1: 300

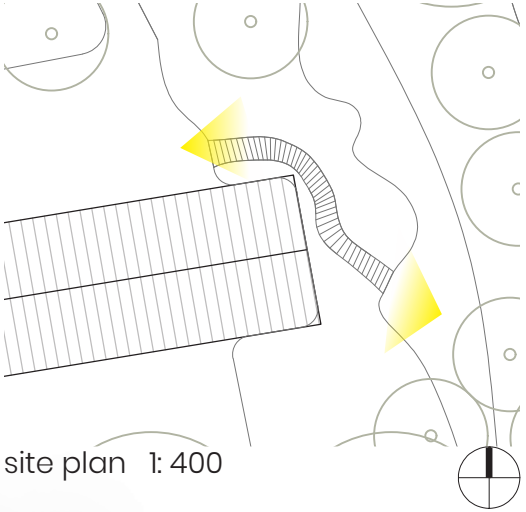


site map n / scale

staircase upstairs:
entry to the park with
view over the ilm river
traffic sounds behind

birds eye view
over chosen area:
hill, staircase, Roman House

staircase downstairs:
arrival in the park-
quieter with poem
from Goethe

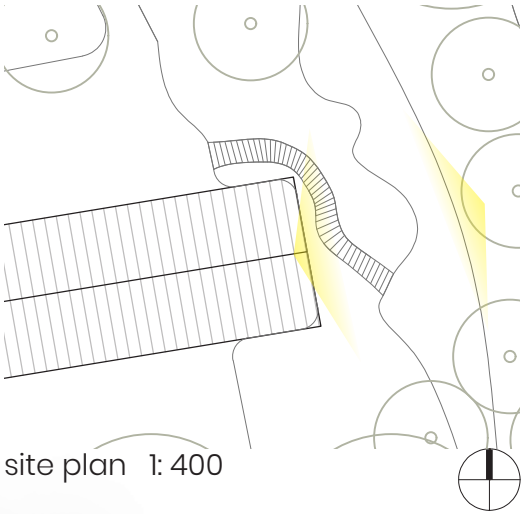


site plan 1: 400

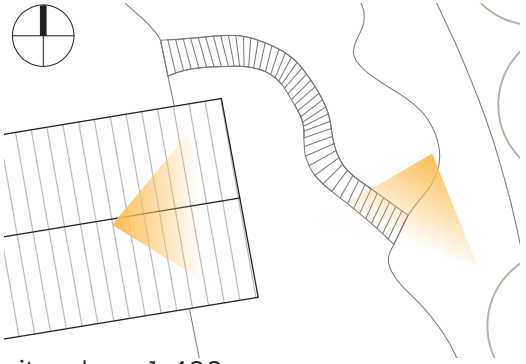
09

panoramic view:
taken from the standing point in front of the pond

panoramic view:
taken from downstairs in front of the hill viewing the whole hill and staircase Roman House with pond on top



site plan 1: 400



site plan 1: 400

Another interesting arranged topic around the Roman House is water. A fountain or small water well was placed at the shielded location between four columns from which point a beautiful view over the park is guaranteed. The sound you hear from there is now connected to the Ilm river in front of you.

Furthermore, the pictures on the right show a small pond at the end of the staircase. It collects the down streaming water of the hill, gathers it and sends it back to the river. This creates an invisible water path from the artificially created architecture to the existing river and therefore forms communication with nature.

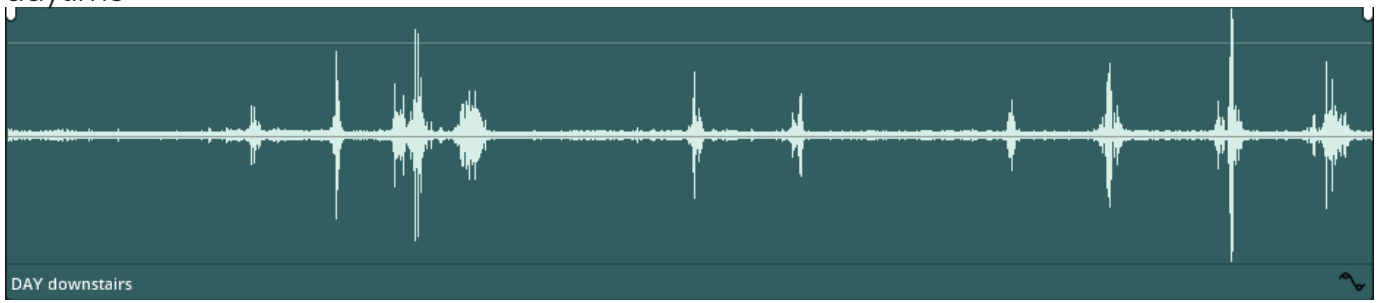
| 2. place | sound

duration: 20 minutes

morning

a lot of birds
river sound
few joggers and dogwalkers on the pathways
barks of dogs
only few stair-sounds
a chirchbell sound

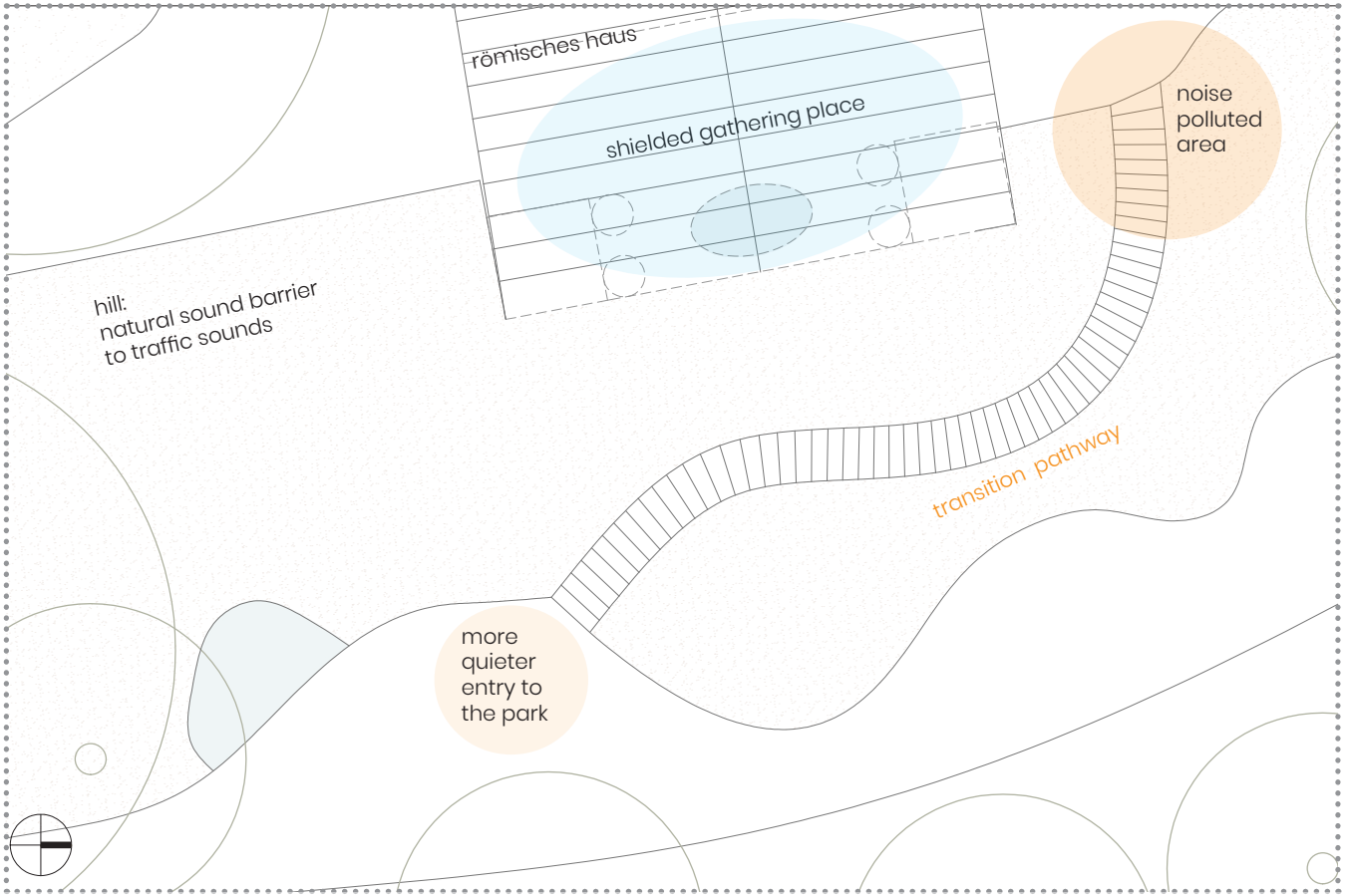
daytime



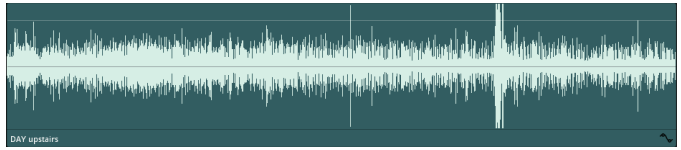
many steps of strollers and joggers, on the pathways and stairs
conversations of people passing by, meeting and staying
few dogwalkers
bikes and cars
a plane and landmower
few birds and wind

night

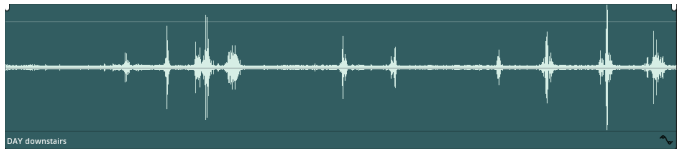
river sound
soft wind
highway from the distace
some clicking noises



site plan 1: 75



daytime upstairs

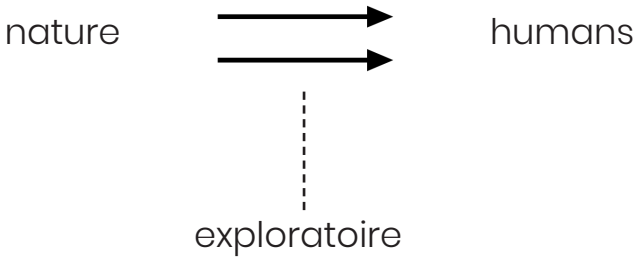


daytime downstairs

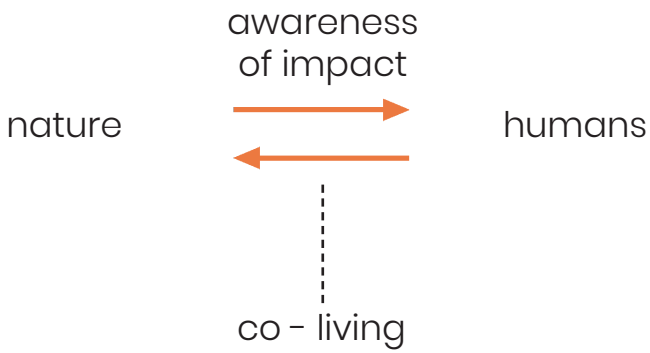
unopposed traffic and wind sounds
--> overshadowing the natural noises
--> hill as a sound barrier

| 3. concept |

I today"s human relation with nature I



I what could a new nature could be I



what could a new nature be ?

humans are using the park

better
co - living
conditions
for all

aware of their impact in nature

what could a new nature be ?

How do we impact nature?

Do we notice all the changes we make in the natural environment by only doing our daily routine? The “silent spectacle” installation aims at discussing the constant impact we have in nature, and that we are not aware.

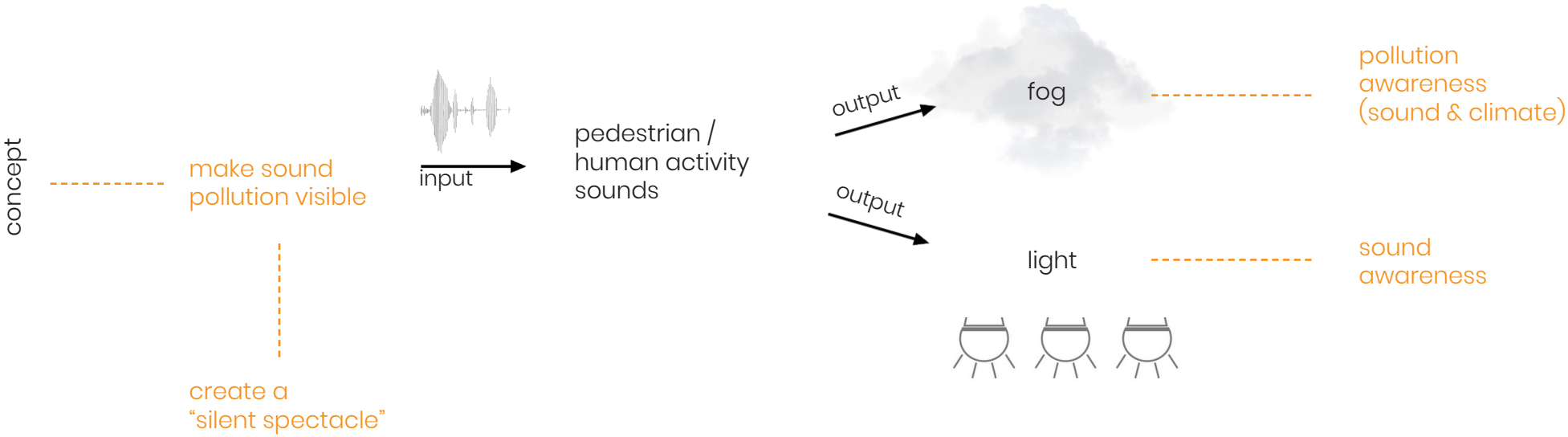
The project believes that being aware is the first step towards a “new nature”, one in which humans and nature create a egalitarian environment, where humans are not only using natural resources, but also using knowledge and technology to return to nature.

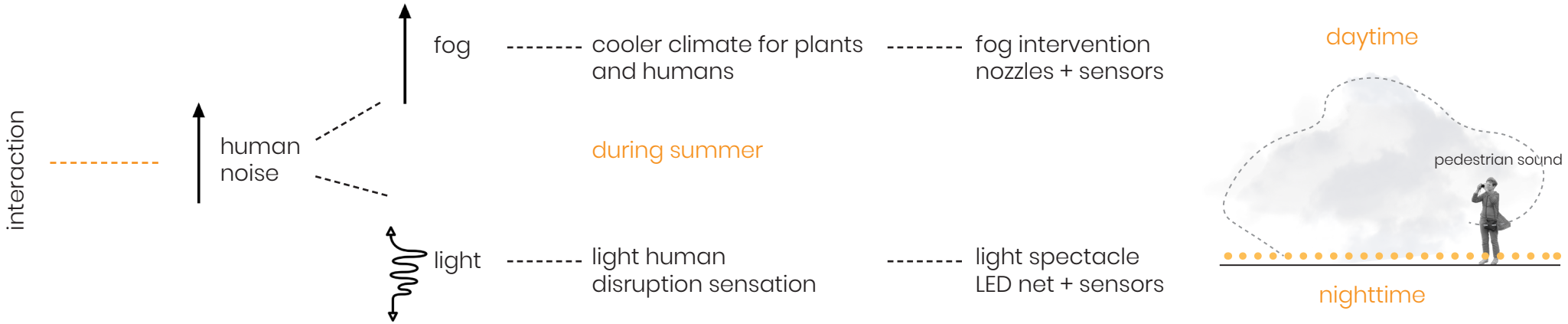
“Silent spectacle” seeks to show that co-living can be the key factor in protecting the environment and, therefore, our society.

Giving nature a “voice” through the installation, showing environmental data in a visible and tangible installation, could be one of the ways of constituting this co-living environment, where nature is also heard and comprehended.

what could a new nature be ?

how can we propose a new nature ?





how is the interaction in this new nature ?

"silent spectacle"

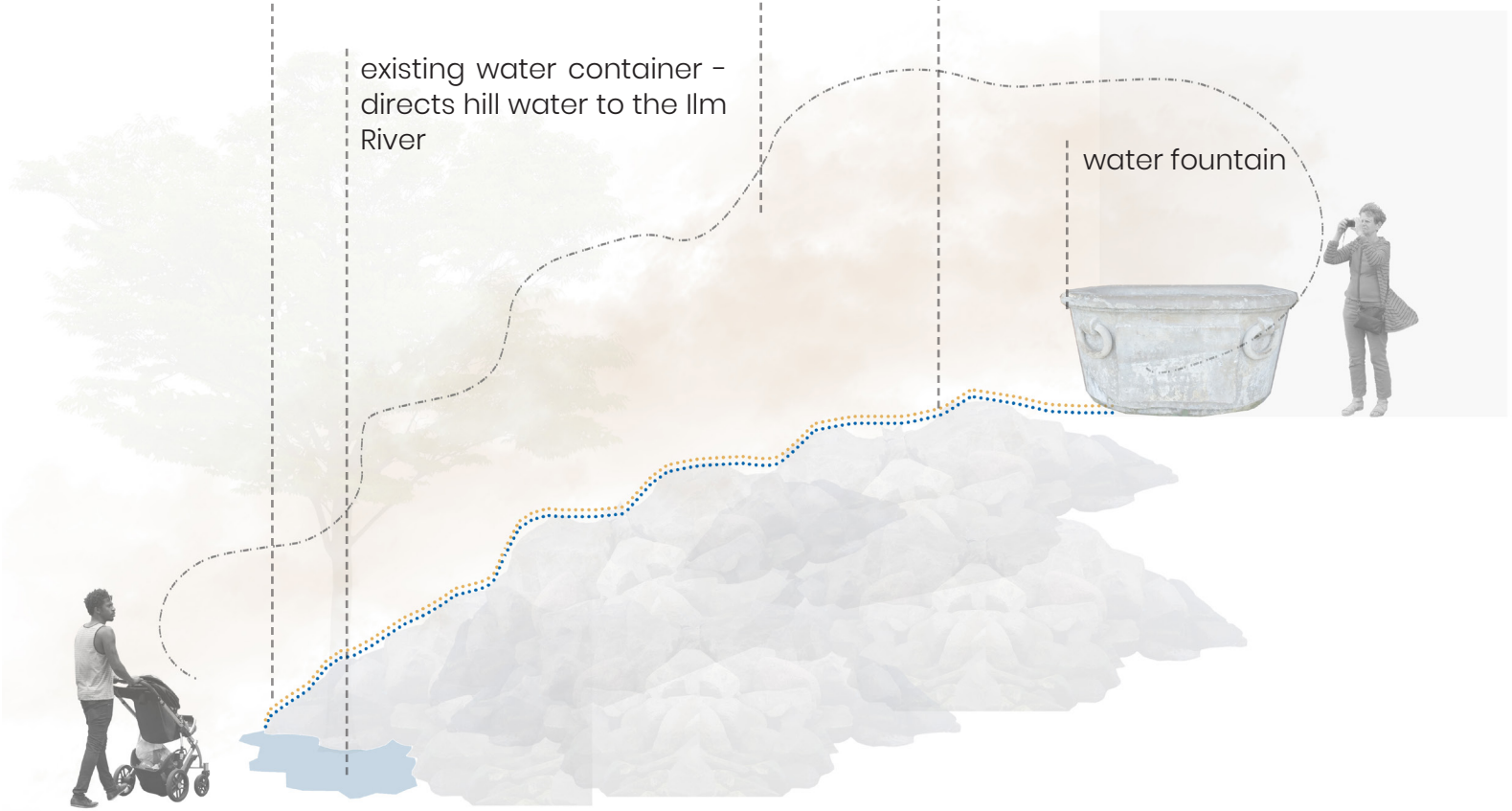
a fog nozzle pathway
connects two
pre-existing water sources
"invisible water path"

with more sound pollution ,
a fog-cloud atmosphere is
created

the LED's string gives a instant
noise response

existing water container -
directs hill water to the Ilm
River

water fountain




37

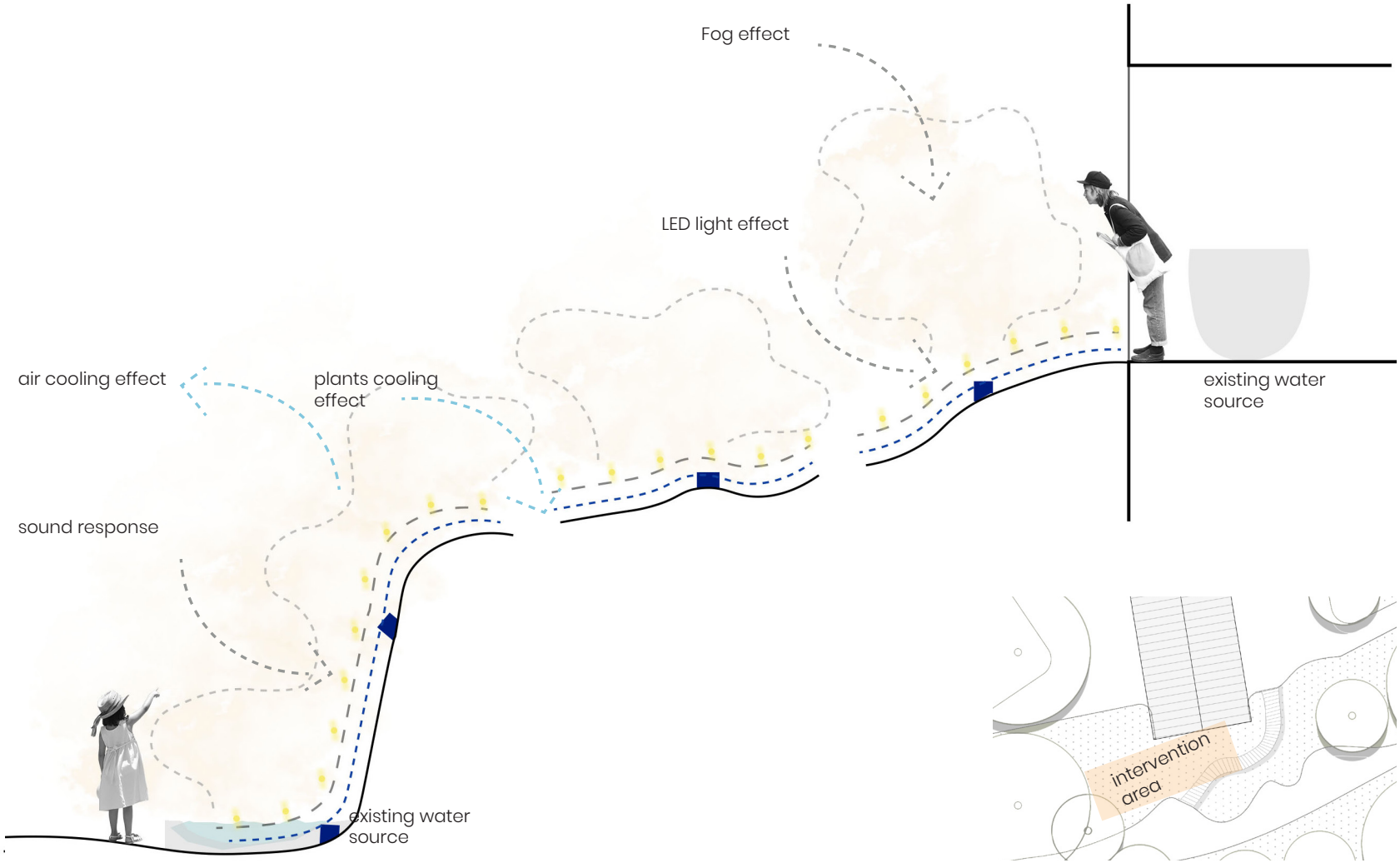

38

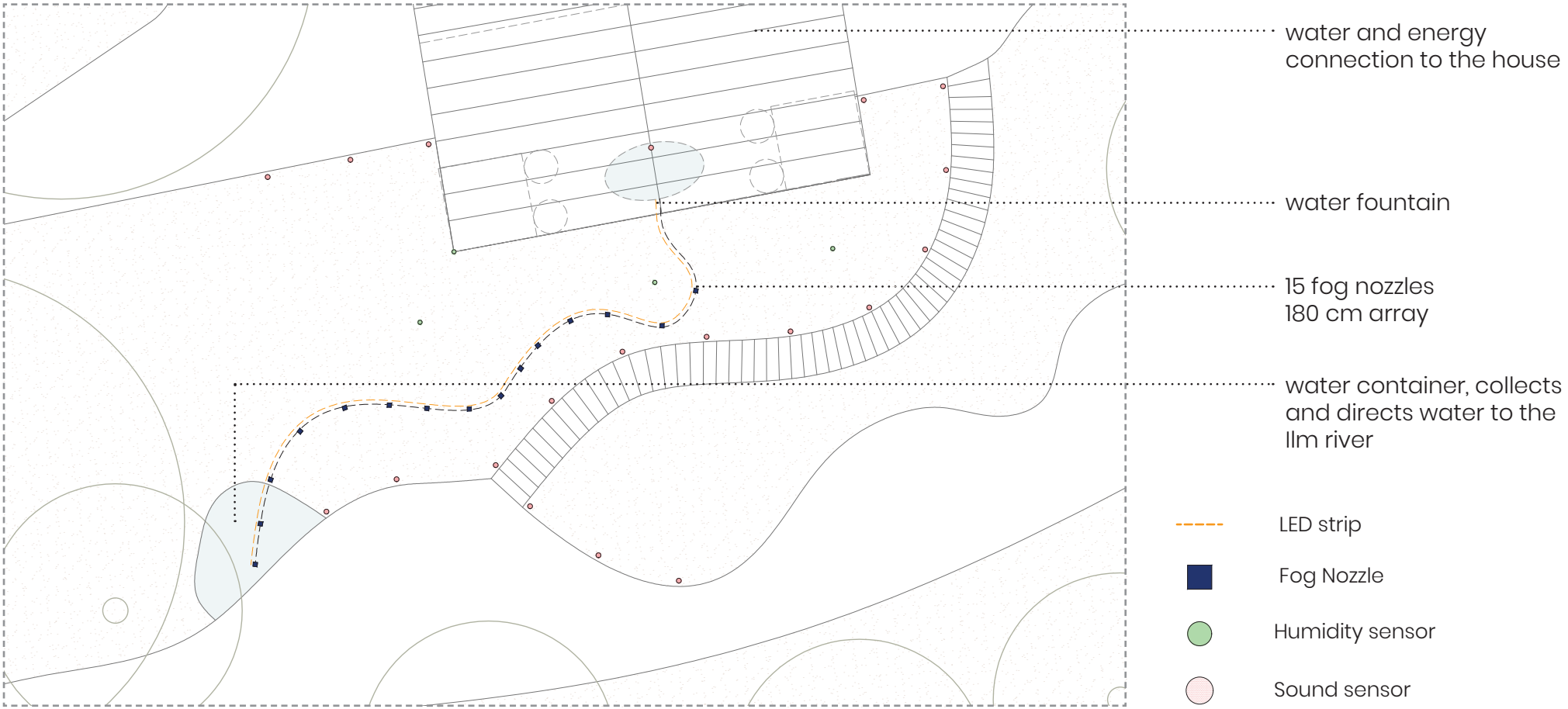

39

to make [noise] pollution visible

| 4. design |

The final concept of “silent spectacle” connects the pre-existing water sources of the Roman House: the fountain on top of the hill and the water pond on the bottom. A line of fog-nozzles create a cooling effect for both nature and humans. The combined LED-line forms a direct sound response and invites the pedestrians to stop and re-evaluate their own emission of sound pollution.



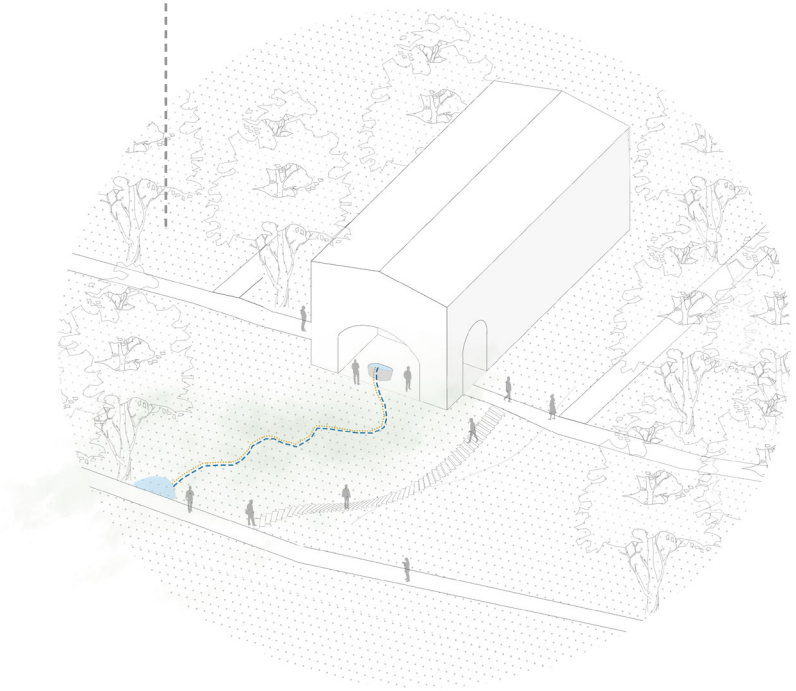


romisches haus terrasse plan 1:75

normal condition
small amount of sound polution

fog: small amount for plant cooling

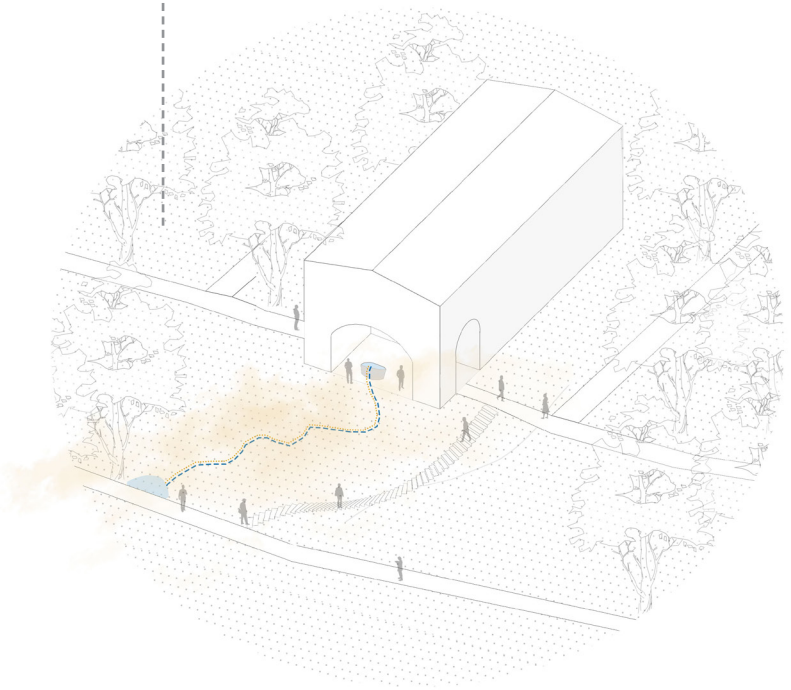
light: “silent spectacle”



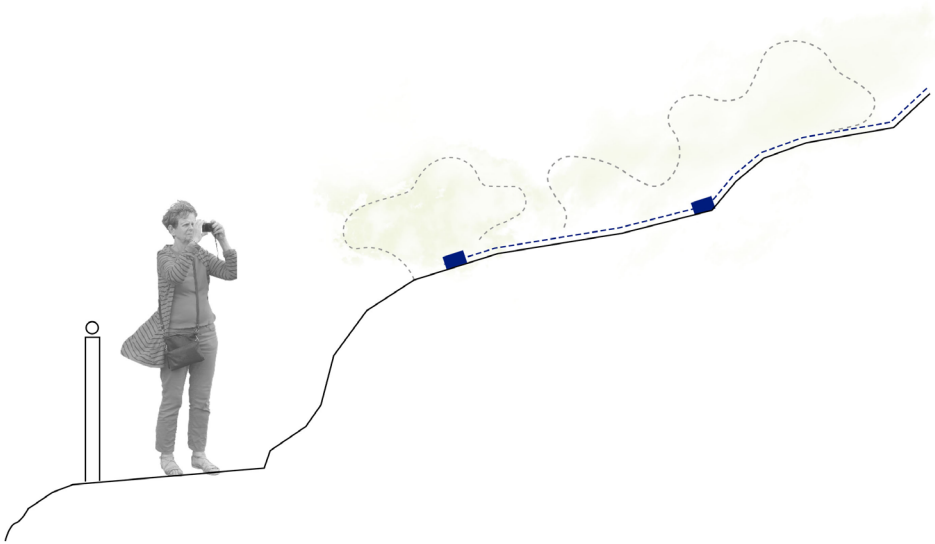
loud condition
high amount of sound polution

fog: “cloud atmosphere”

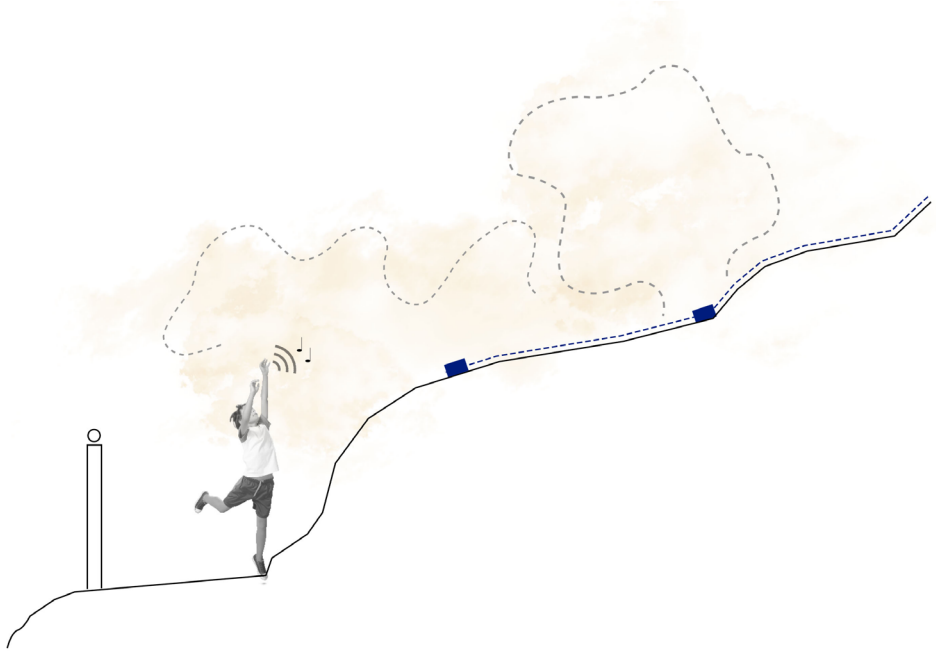
light: disrupted



the cloud atmosphere



smaller and stable amount of “fog pollution”
creates a cooled space for the nature during summer

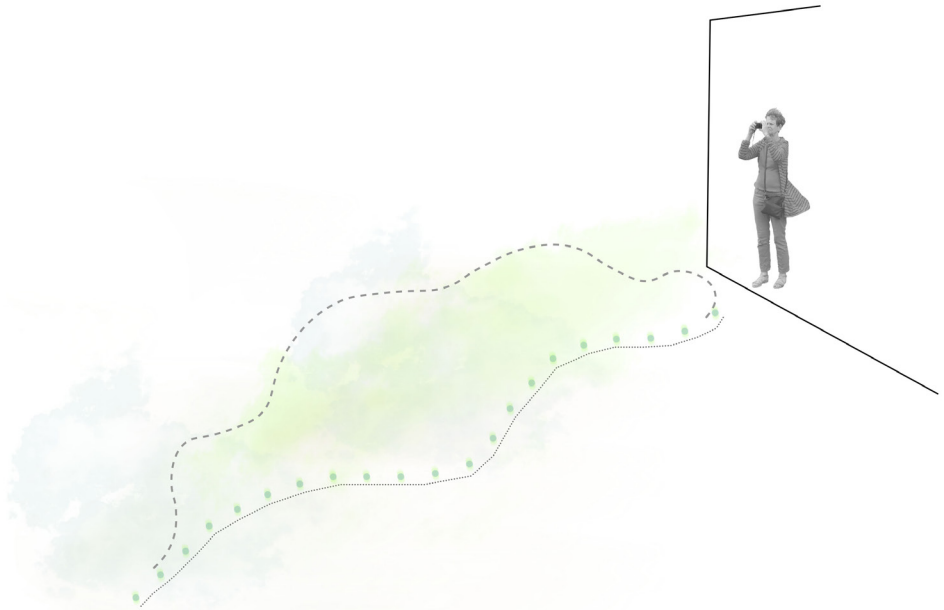


raising amount of “fog pollution” activated by human sound
creates a fog cloud - disrupts the view, cooling effect rises

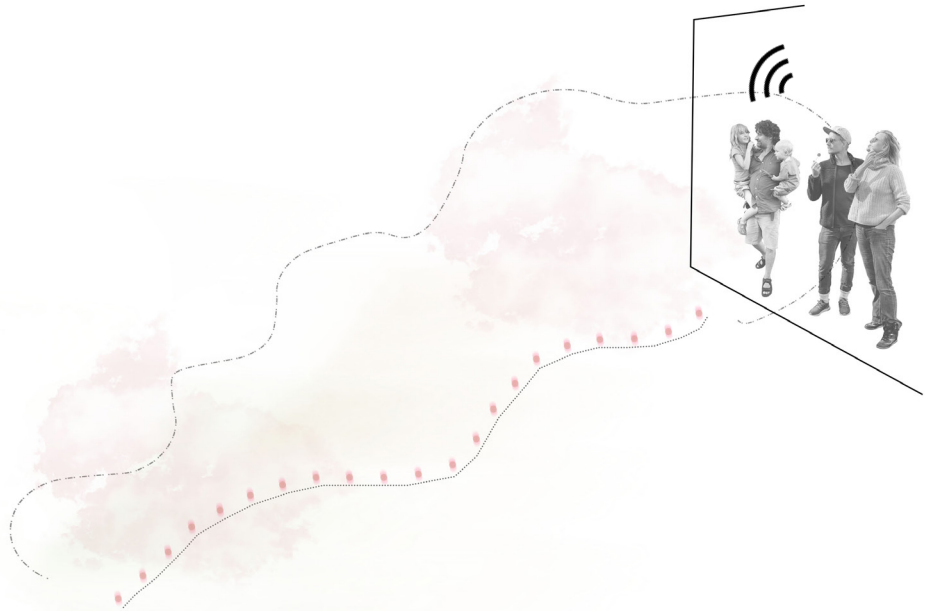
- experience more present during daytime
- cooling effect in summer for humans & environment

fog

the silent spectacle



light spectacle in base-state:
- natural colours of the Ilm Park
- slow stream of movements



light spectacle in interfered-state:
- human made colours of the römisches Haus
- disrupted shaking movements

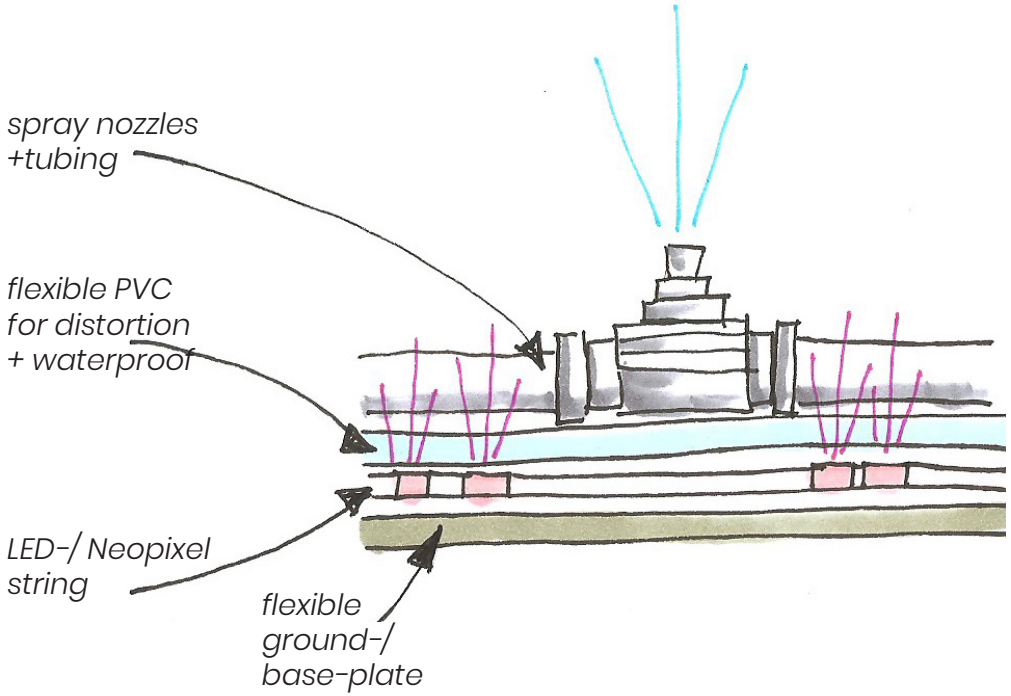
• experience more present during nighttime
• “quiet spectacle” effect for humans & environment

light

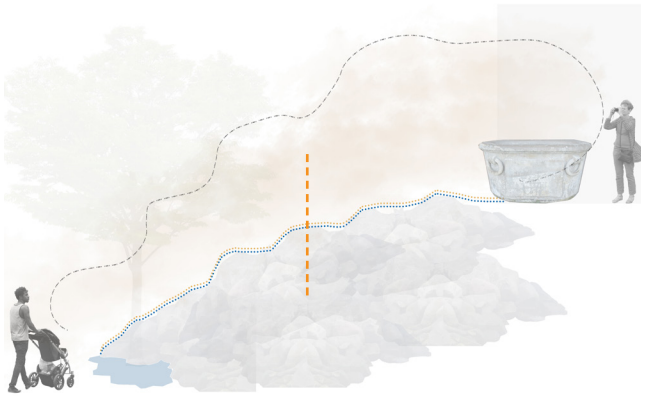
| 5. real implementation |

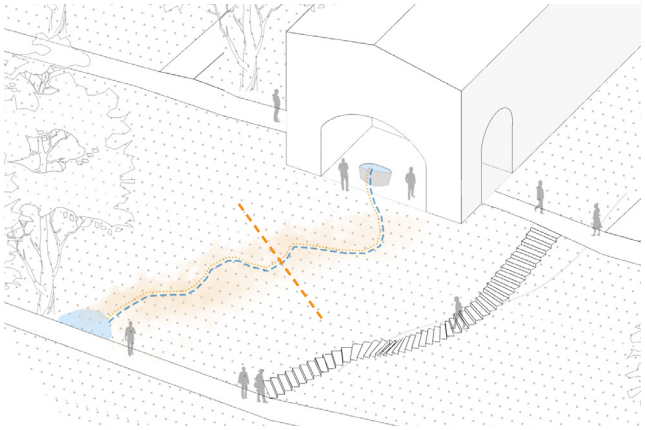
system components

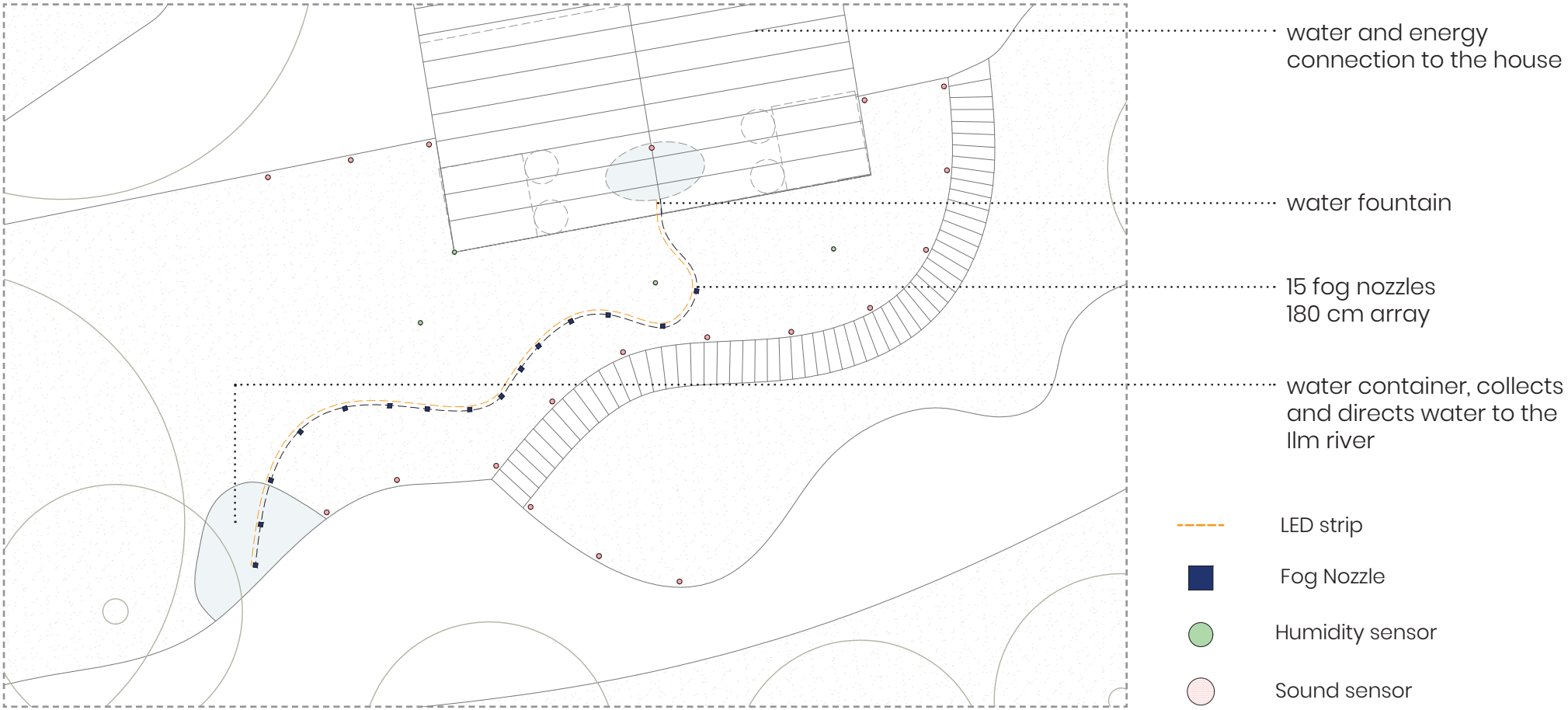
For the real installation, we proposed a string system, formed by ultra fine mist fog nozzles, LED RGB strings, and intermediate materials to impermeabilize and not interfere directly on the soil. The string system will be connected in existing water and energy outlets, and the fog nozzles pressure will be controlled by an outside pumping system. The fog and light quantity will be created in response to the audio input, sensed by sound sensors, and regulated according to humidity sensors, that can shut off the fog if the humidity can damage the local ecosystem.



In conclusion, it's a 36-meter string system that follows the hill topography to create this almost unseen string, that creates an ephemeral atmosphere with fog and light.







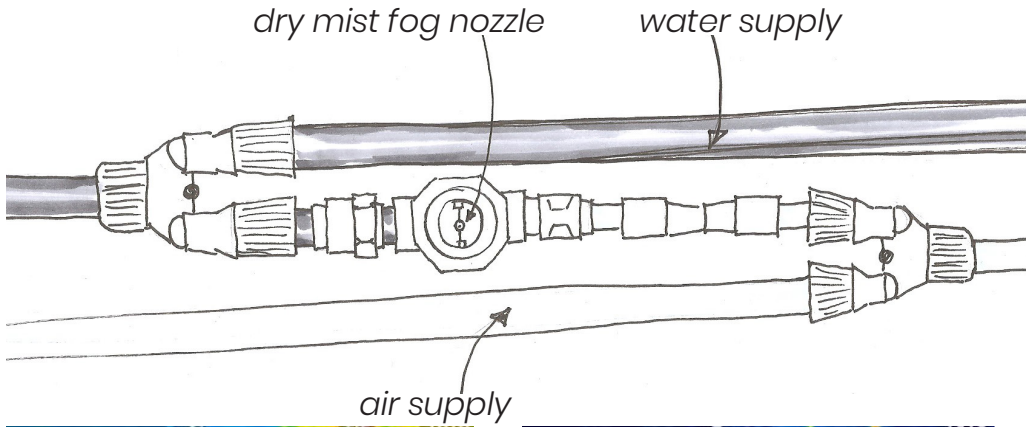
romisches haus terrasse plan 1:75

| 5. real implementation | sensors

The use of live data sensors are essential to create the responsiveness of the installation, reacting to the sound pollution of the environment, while also measuring the local humidity in order to allow the fog to be turned off when necessary for the plants. The sound sensors will be located throughout the pedestrian paths in the hill - stairs and park paths - which allows everyone in the area to contribute to the installation and notice it's sound pollution impact. On the other hand, the humidity sensors will be located directly on the hill to monitorate the humidity of the existent ecosystem, formed by mosses, ground vegetation and some bushes.

the cloud atmosphere

The fog nozzle system is inspired by the Panasonic system “45 Green AC Flex” Solution, that uses a two-fluid mist nozzles solution in order to create a fine dry water mist without need to use other fog production methods, such as warming or ultrasonic sound. This innovative system has been used in artistic installations and also in cooling down cities that are facing heating waves, and therefore seems like the ideal solution for our installation proposal.




cooler climate for humans...


...and nature

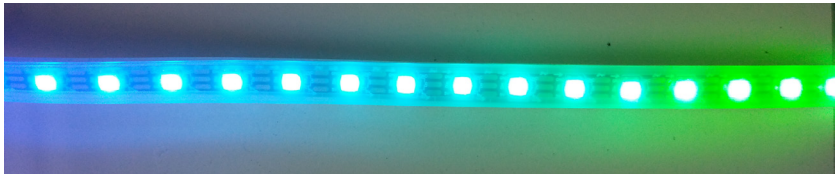
thermo viewer without mist

with mist

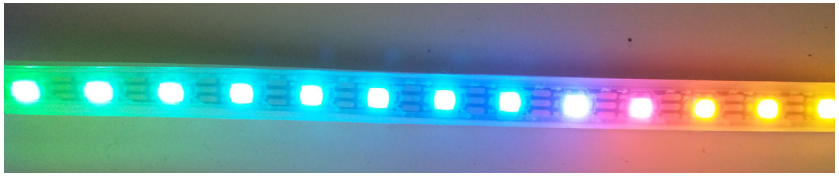

Panasonic Newsroom :
Installation Art and Heat Countermeasure by Ultra-fine Dry Mist
https://www.youtube.com/watch?v=8aK1mYhOZ4g

fog

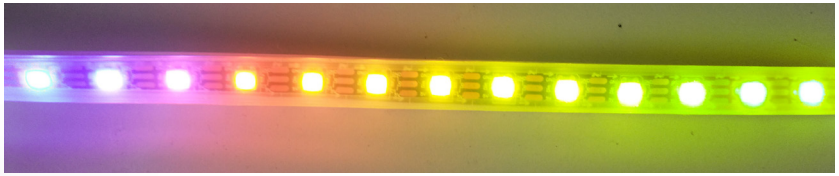
01_Light State : low sound pollution = natural colors



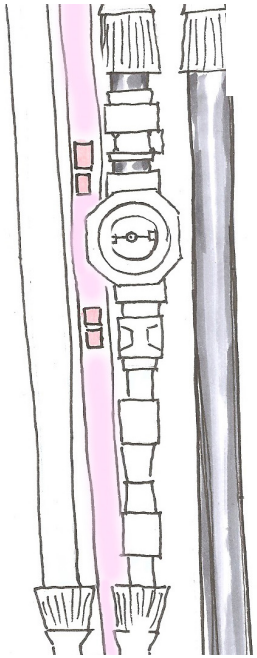
02_Light State : medium sound pollution = disturbed natural colors

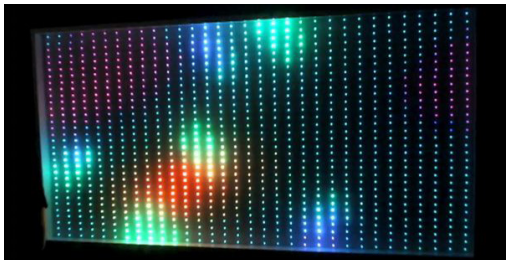


03_Light State : high sound pollution = disturbed/ human made colors



The LED String system is responsible to provide different light patterns and colors according also to the sound input captured by the sound sensors. It's a waterproof string, which allows it to be near the fog system without being damaged. The LED string is composed of individual LEDs that can be addressed by code. For this project, a code was developed in Arduino to distinguish the sound input in 3 amount phases - low, high, medium - and create a correspondent light color pattern, that also symbolizes our project concept of new nature and human interference, the first one being represented by natural colors, such as green and blues, and the last one by more human-made colors, also present in the Romisches Haus, such as red and oranges.










For having a real and measurable perception of how the fog system and LED would work in it's outdoor context, we turned to commercial products which could allow a first perception of the components behaviour. We tested a LED RGB String to see how visible it was in daylight, and had positive responses, as seen in the pictures on the side. In addition, we tested a commercial version of the fog nozzles, used for garden irrigation, to see how the fog would react in the real environment, and in the experiment we noticed the different patterns and atmospheres that the fog can create.







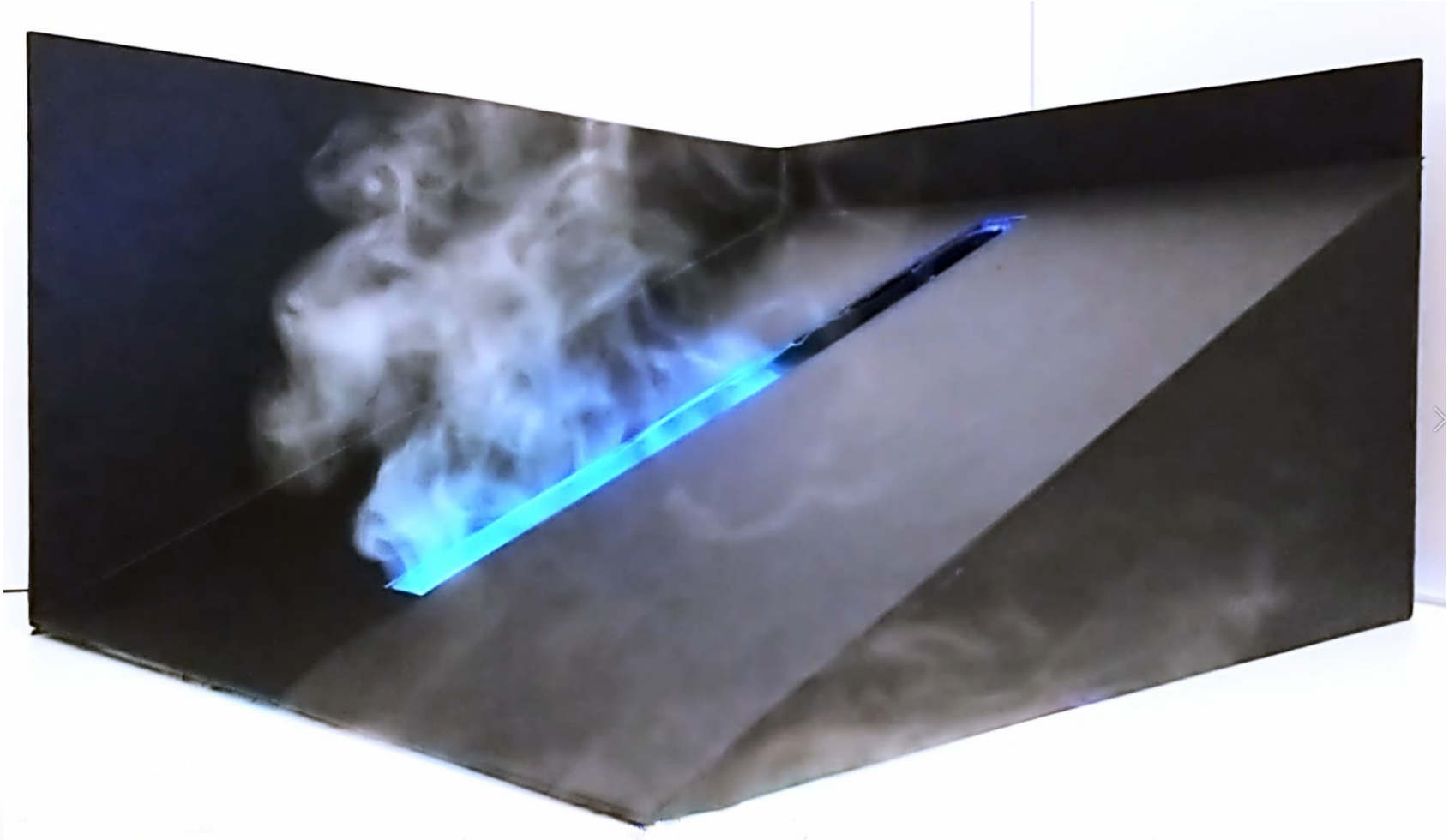

Techinical component	Quantity	Provider	Costs? €	etc
Nebel Düsenanlage	15 Nozzles + Tubing + air&water pressure system --- 1 nozzle / meter	CoolCloud	3.000	Renting possibility - in contact with company
LED String Adressable	95 euros / meter --- --- 40 meters in total	Adafruit	678	
Eletrical instalations conection to the house energy outlet	20 meters	Nexans - Amazon	128	
Rented computer to run softwares	1 month	Grover	50	
Humidity Sensors	3 units / 39,95 each	Reichelt - LoRaWAN temperature & humidity sensor	119,85	
Soundsensors	20 sound sensors/ 122 each		2440	
Instalation estimated cost - in euros				6.416

what are the next possible steps in the project development?

- Improve the sound analysis in the code
  - Differ sound income, sound range/pitch, sound producer
- Develope code to measure the humidity in the hill and adjust according to the plant needs
- Develope collaborations with fog nozzle system

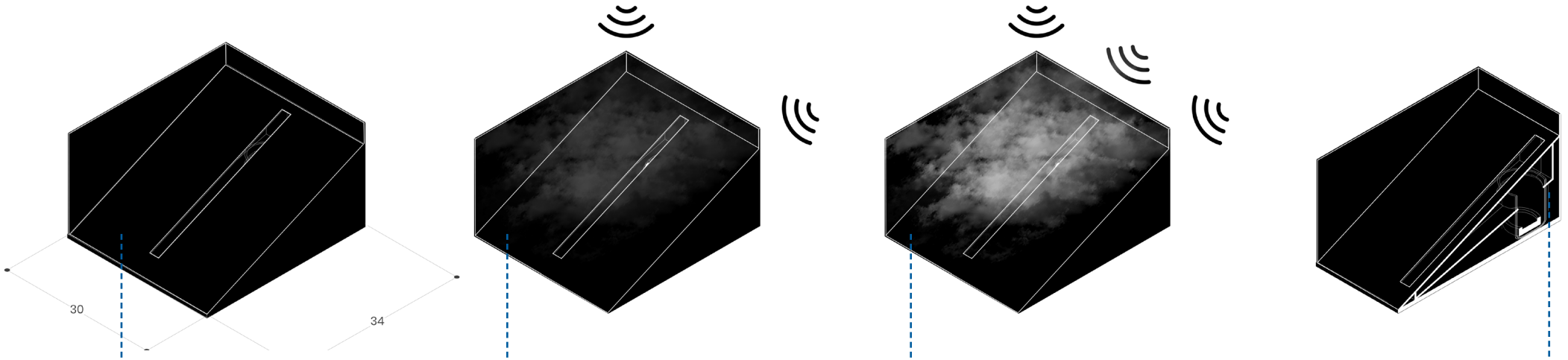
The development of the “silent spectacle” required us to conduct real life experiments and research to adapt the project outcome to the best solution that represents our new nature concept and atmosphere ideas.
Some challenges were faced while developing the prototype and code, but that were important to create a direct and clear installation concept. We can already see some further developments of the project that could optimize and sophisticate it’s potential.
The first development point concerns further writing the code so it can distinguish different sound range input, such as natural sounds or human sounds, and create an installation that responds only to human sound pollution.
Another point would be creating a code that can measure the local humidity and assess the perfect condition for the local ecosystem and plants, in order to not danificate them.
Finally, for real implementation, it would be ideal to develop a partnership with fog nozzle producers, in order to implement the best and more suitable system and spreading the possibilities of this system use.

| 6. prototype |



The prototype was developed to allow the testing of components that would be used in the installation, such as the sound sensor, fog production and LED String.
Furthermore, we wanted to create an abstract model of how the fog and LED would look in real life.
A inclined shell, therefore, was created for covering the components and allowing the fog to spread as it would in the Romisches Haus.
This allowed us to test the components while also evaluating the real life impact of the output that we proposed, fog and light.

--------downscale: prototype ----------



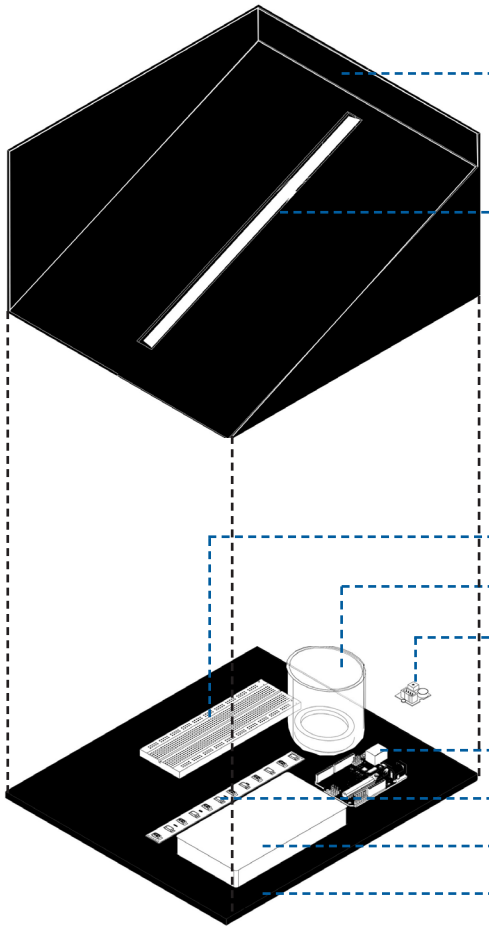
abstract model + prototype

low data input - less fog

high data input - more fog

technical components
under topography shell

downscale: prototype



Abstract model in impermeable material

Openess to let the fog go up
– produced by Arduino and Atomizer

Breadboard

Fog atomizer inside water cup

Groove sound sensor

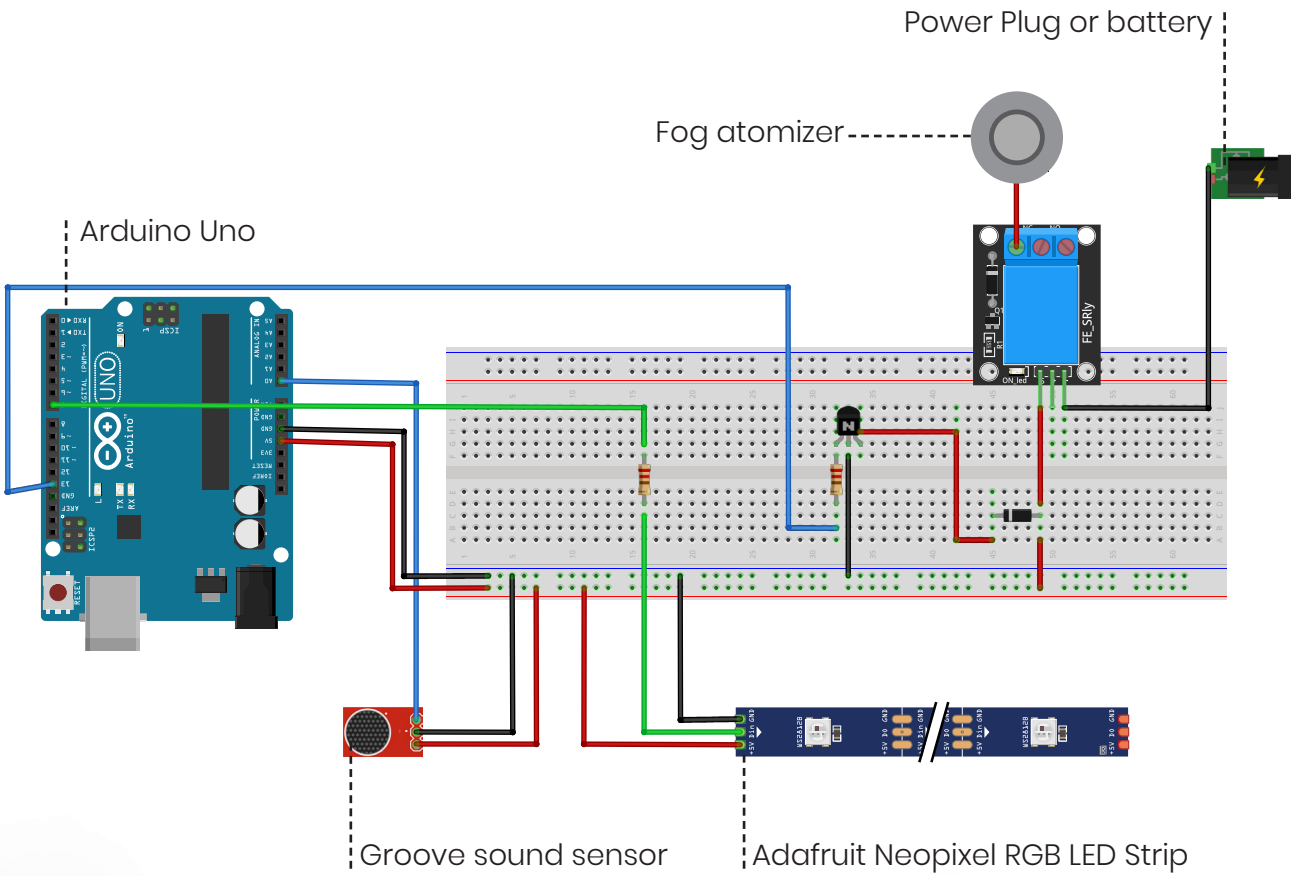
Arduino

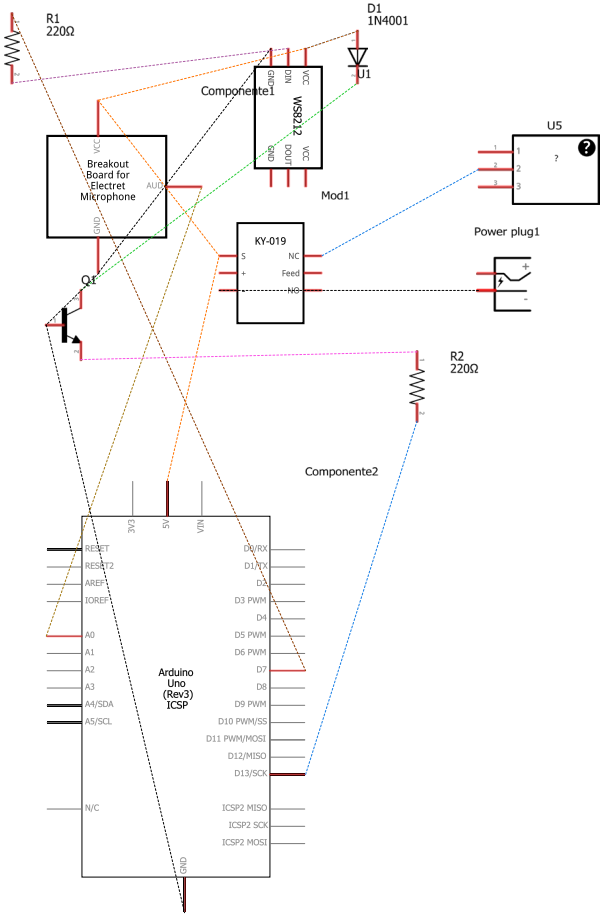
LED- / neopixel-strip

Battery

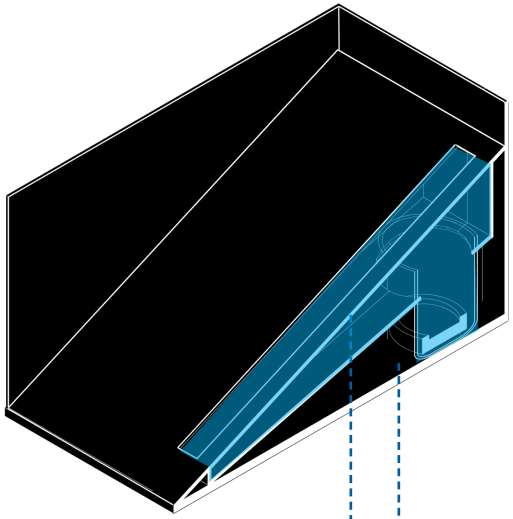
base plate

arduino schematics





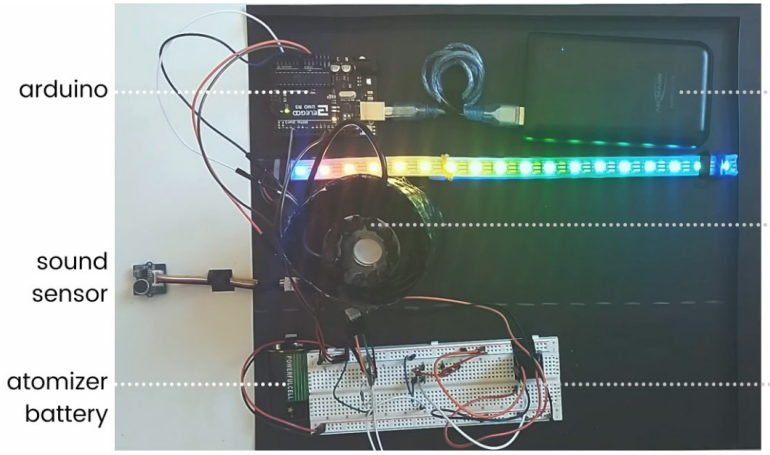
shield components



waterproof chamber in wich atomizer & fog are

technical components under topography shell

For the prototype it was necessary to develop impermeabilisation that allowed the electrical components and fog production with water to be in the same place. A sealing chamber was created for the fog atomizer, so that the condensed water would not spill on the other electrical components. All the prototype walls were waterproof with black acryglas and black silicon, which sealed the model joints. Furthermore, the use of batteries was proposed so that the model could stand itself with no need of being connected to power or computer, since the code is also encrypted in the arduino, located inside the model.



arduino code structure

```
//=======================================
// ----------LIBRARIES--------------
////NEOPIXEL
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif
#define PIN 6                           // the LED Input Pin on the Arduino-board
Adafruit_NeoPixel strip = Adafruit_NeoPixel(30, PIN, NEO_GRB + NEO_KHZ800);
                                        // Parameter 1 = number of pixels in strip
                                        // Parameter 2 = Arduino pin number (most are valid)
                                        // Parameter 3 = pixel type flags, add together as needed:
                                        //   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
                                        //   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
// --------INTS & CONSTANTS ---------------
const int numReadings = 10;             // For the average sound Input Define the number of samples to keep track of.
                                        //  The higher the number, the more the readings will be smoothed,
                                        //  but the slower the output will respond to the input.
int readings[numReadings];              // the readings from the analog input
int readIndex = 0;                      // the index of the current reading
int total = 0;                          // the running total
int average = 0;                        // the average
int inputPin = A0;                      // the Sound Input Pin on the Arduino-board
const int pinAdc = A0;
const int atomizer =  13;               //Atomizer Input pin
const long soundInterval = 1000;        // interval at which to capture sound data (milliseconds)
const int NcycleInterval = 2500;        // number of millisecs between Neopixel Natural Cycle
const int DcycleInterval = 2500;        // number of millisecs between Neopixel Disturbed Cycle
const int HcycleInterval = 750;         // number of millisecs between Neopixel Human Cycle
const int NfogInterval = 2000;          // number of millisecs that Atomizer is on in Natural
const int DfogInterval = 1000;          // number of millisecs that Atomizer is on in Disturbed
const int HfogInterval = 500;           // number of millisecs that Atomizer is on in Human
```

arduino code structure

```
//------------ VARIABLES--------------------
unsigned long currentMillis = 0;        // stores the value of millis() in each iteration of loop()
unsigned long previousInputMillis = 0;  // will store last time Input reading was updated
unsigned long previousNcycleMillis = 0; // time when Natural cycle last checked
unsigned long previousDcycleMillis = 0; // time when Disturbed cycle last checked
unsigned long previousHcycleMillis = 0; // time when Human cycle last checked
unsigned long previousNfogMillis = 0;   // time when Natural fog last checked
unsigned long previousDfogMillis = 0;   // time when Disturbed fog last checked
unsigned long previousHfogMillis = 0;   // time when Human fog last checked
byte atomizer_State = LOW;              // used to record whether the atomizer is on or off  LOW = off

//________________________________________
//========================================
//========================================
void setup() {
 ////SOUNDSENSOR
  Serial.begin(115200);                         // initialize serial communication with computer
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }                                             // initialize all the readings to 0
  Serial.println("Grove - Sound Sensor Test...");
 ////NEOPIXEL
  strip.begin();                                // Starting Neopixels
  strip.setBrightness(50);                      // Defining brightness of LEDs
  strip.show();                                 // Initialize all pixels to 'off'
 ////ATOMIZER
  pinMode(atomizer, OUTPUT);
}


//========================================
//========================================
```

arduino code structure

```
//=======================================
void loop() {
  currentMillis = millis();     // capture the latest value of millis()
 ////SOUNDSENSOR
  readSound();                  // call soundsensor function
 ////NEOPIXEL
  cycle();                      // call Neopixel functions
 ////ATOMIZER
  fog();                       // call Atomizer functions
}



//=======================================
//=======================================
////SOUNDSENSOR
void readSound() {
  if (currentMillis - previousInputMillis >= soundInterval) {   // check to see if it's time to capture the input; that is, if t
                                                                // between the current time and last time capturing is bigger th
                                                                // the interval at which you want to capture. (e.g. one secound)
                                                          // save the last time capturing
    previousInputMillis = currentMillis;                  // store the time of this change
    total = total - readings[readIndex];                  // read Input with average: subtract the last reading
    readings[readIndex] = analogRead(inputPin);           // read from the sensor
    total = total + readings[readIndex];                  // add the reading to the total
    readIndex = readIndex + 1;                            // advance to the next position in the array
    if (readIndex >= numReadings) {                       // if we're at the end of the array...
      readIndex = 0;                                      // ...wrap around to the beginning
    }
    average = total / numReadings;                        // calculate the average
    Serial.println(average);                              // send it to the computer as ASCII digits
  }
}

//=======================================
```

arduino code structure

```
//=======================================
////NEOPIXEL
//    CYCLE
void cycle() {
    if (average <= 300){                          // determine the sound range of stage 1
    naturalCycle();                               // start natural cycle
    }
    if ((average > 300) && (average <= 450)) {    // determine the sound range of stage 2
    disturbedCycle();                             // start disturbed cycle
    }
    if (average > 450){                           // determine the sound range of stage 3
    humanCycle();                                 // start human cycle
    }
}
```

arduino code structure

```
// -----------  01   NATURAL  ------------
void naturalCycle() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousNcycleMillis >= NcycleInterval) {
    previousNcycleMillis += NcycleInterval;
    uint16_t i, j;
    for(j=0; j<256*1; j++) { // 1 cycles of green&blue colors on wheel
      for(i=0; i< strip.numPixels(); i++) {
      strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) + j) & 255));
      }
    strip.show();
    }
  }
}
uint32_t Wheel(byte WheelPos) {
  if (WheelPos > 126) {
 return strip.Color (0, min (255, 2 * abs (127 - WheelPos)), min (255, 256 - (2 * abs (WheelPos-127))));
  }
 else {
  return strip.Color (0, min (255, 2 * abs (127 - WheelPos)), min (255, 255 - (2 * abs (127-WheelPos))));
 }
}
// -------  02  DISTURBED ------------
void disturbedCycle() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousDcycleMillis >= DcycleInterval) {
    previousDcycleMillis += DcycleInterval;
    uint16_t i, j;
    for(j=0; j<256*1; j++) { // 1 cycles natural&human colors on wheel
      for(i=0; i< strip.numPixels(); i++) {
        strip.setPixelColor(i, Wheel2(((i * 256 / strip.numPixels()) + j) & 255));
      }
      strip.show();
    }
  }
}
```

arduino code structure

```
uint32_t Wheel2(byte WheelPos) {
  WheelPos = 255 - WheelPos;
  if(WheelPos < 170) {
    WheelPos -= 85;
    return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
  }
  WheelPos -= 170;
  return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}
// -------  03  HUMAN ------------
void humanCycle() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousHcycleMillis >= HcycleInterval) {
    previousHcycleMillis += HcycleInterval;
    uint16_t i, j;
    for(j=0; j<256*1; j++) { // 1 cycle of human&less natural on wheel
      for(i=0; i< strip.numPixels(); i++) {
        strip.setPixelColor(i, Wheel3(((i * 256 / strip.numPixels()) + j) & 255));
      }
      strip.show();
    }
  }
}
uint32_t Wheel3(byte WheelPos) {
  WheelPos = 255 - WheelPos;
  WheelPos -= 170;
  return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

//=======================================
```

arduino code structure

```
////ATOMIZER
//    FOG
void fog() {
    if (average <= 350){
    naturalFog();
    }
    if ((average > 350) && (average <= 600)) {
    disturbedFog();
    }
    if (average >600){
    humanFog();
    }
}
// -----------  01   NATURAL  ------------
void naturalFog() {
  unsigned long currentMillis = millis();
    if (currentMillis - previousNfogMillis >= NfogInterval) {
       previousNfogMillis = currentMillis;
       if (atomizer_State == LOW) {
       atomizer_State = HIGH;
       }
       else {
       atomizer_State = LOW;
      }
   // set the LED with the ledState of the variable:
    digitalWrite(atomizer, atomizer_State);
   }
}
```

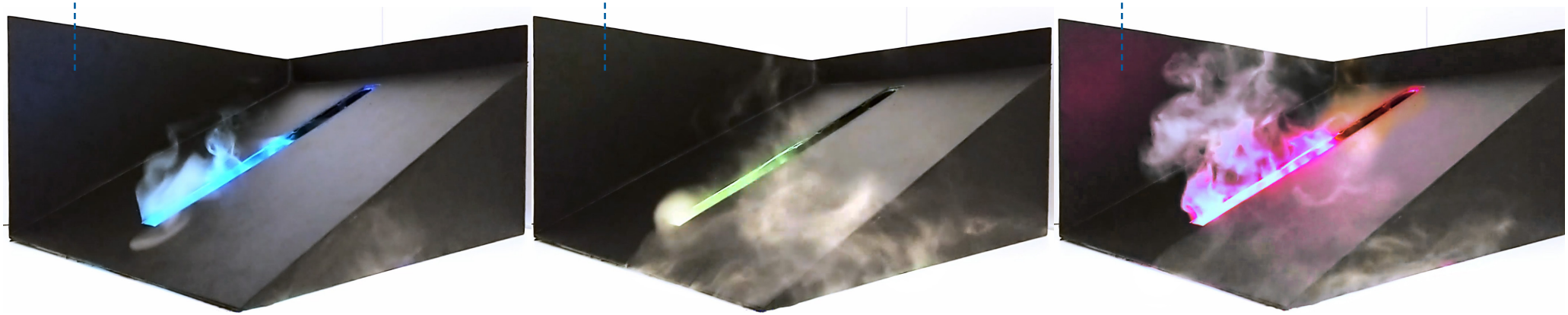
arduino code structure

```
// -------  02  DISTURBED ------------
void disturbedFog() {
unsigned long currentMillis = millis();
    if (currentMillis - previousDfogMillis >= DfogInterval) {
       previousDfogMillis = currentMillis;
       if (atomizer_State == LOW) {
       atomizer_State = HIGH;
       }
       else {
       atomizer_State = LOW;
      }
   // set the LED with the ledState of the variable:
    digitalWrite(atomizer, atomizer_State);
   }
}

// -------  03  HUMAN ------------
void humanFog() {
unsigned long currentMillis = millis();
    if (currentMillis - previousHfogMillis >= HfogInterval) {
       previousHfogMillis = currentMillis;
       if (atomizer_State == LOW) {
       atomizer_State = HIGH;
       }
       else {
       atomizer_State = LOW;
      }
   // set the LED with the ledState of the variable:
    digitalWrite(atomizer, atomizer_State);
   }
}

//=========================================
//=========================================
```

low sound pollution input

medium sound pollution input

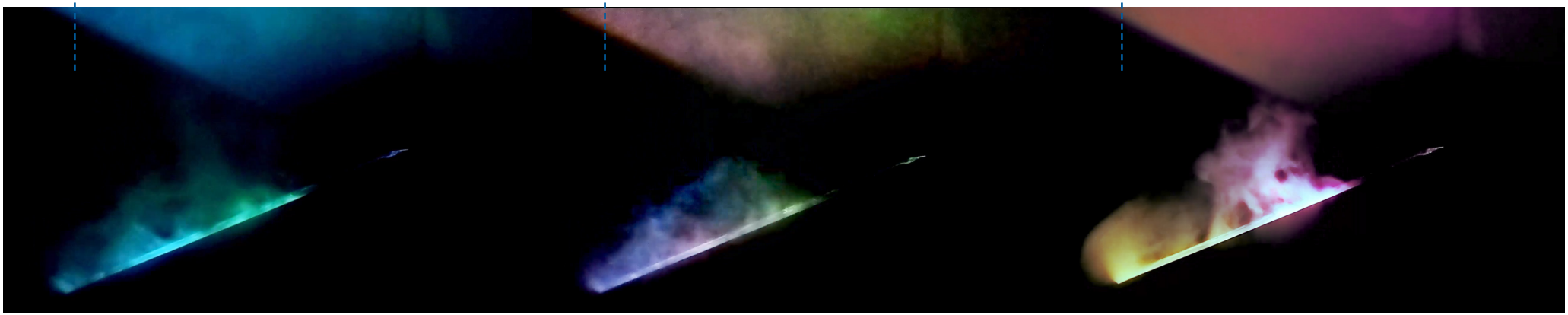
high sound pollution input



low sound pollution input

medium sound pollution input

high sound pollution input



| 7. atmosphere |

- silent spectacle -

a [noise] pollution instalation