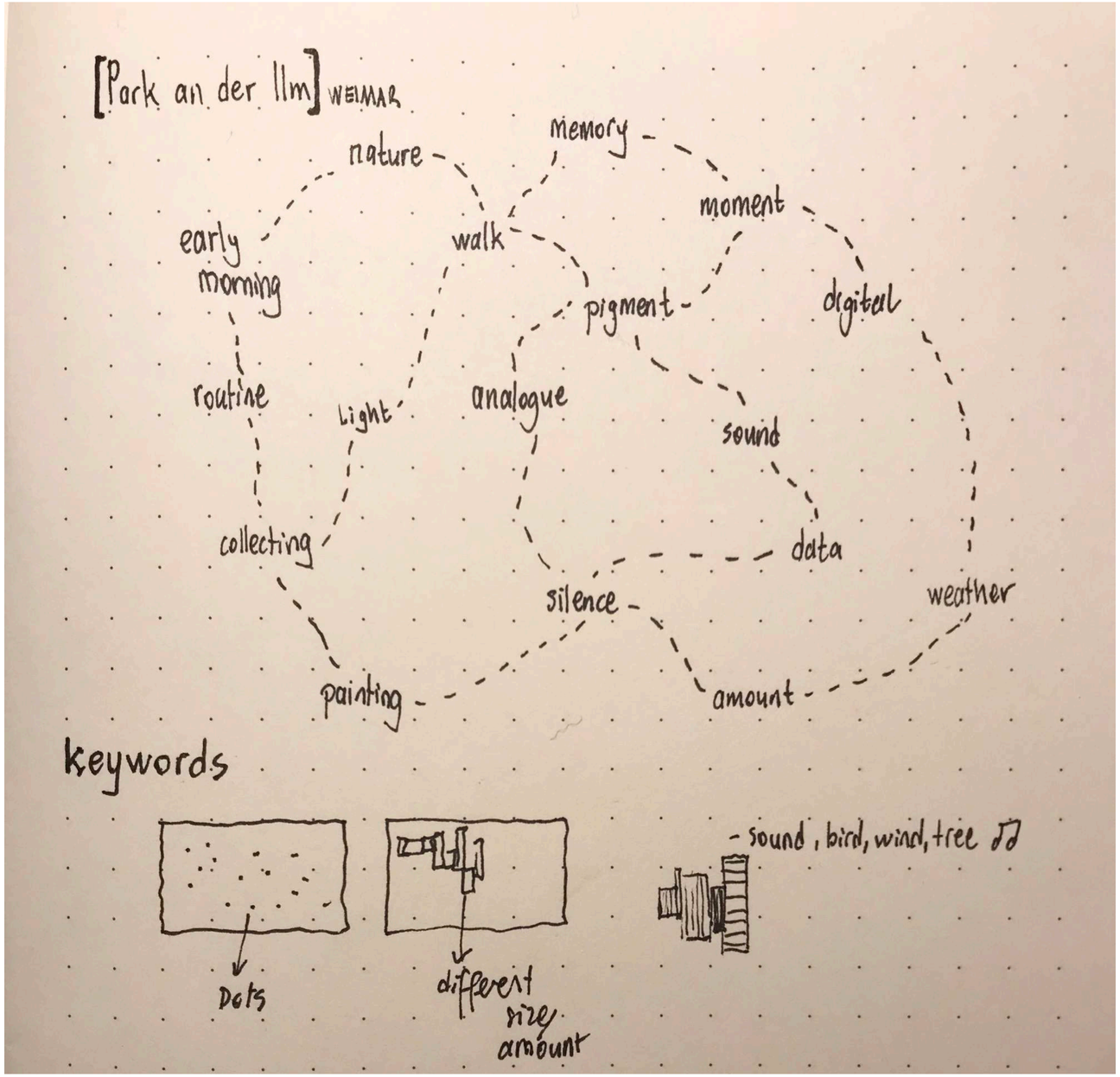
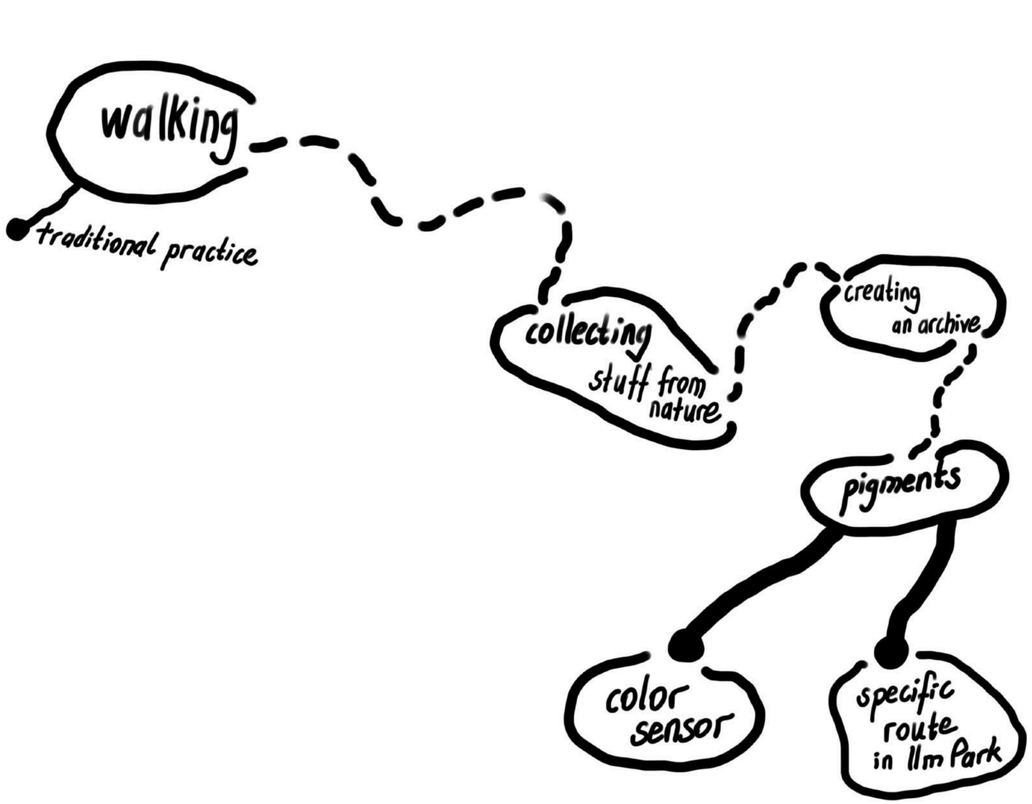
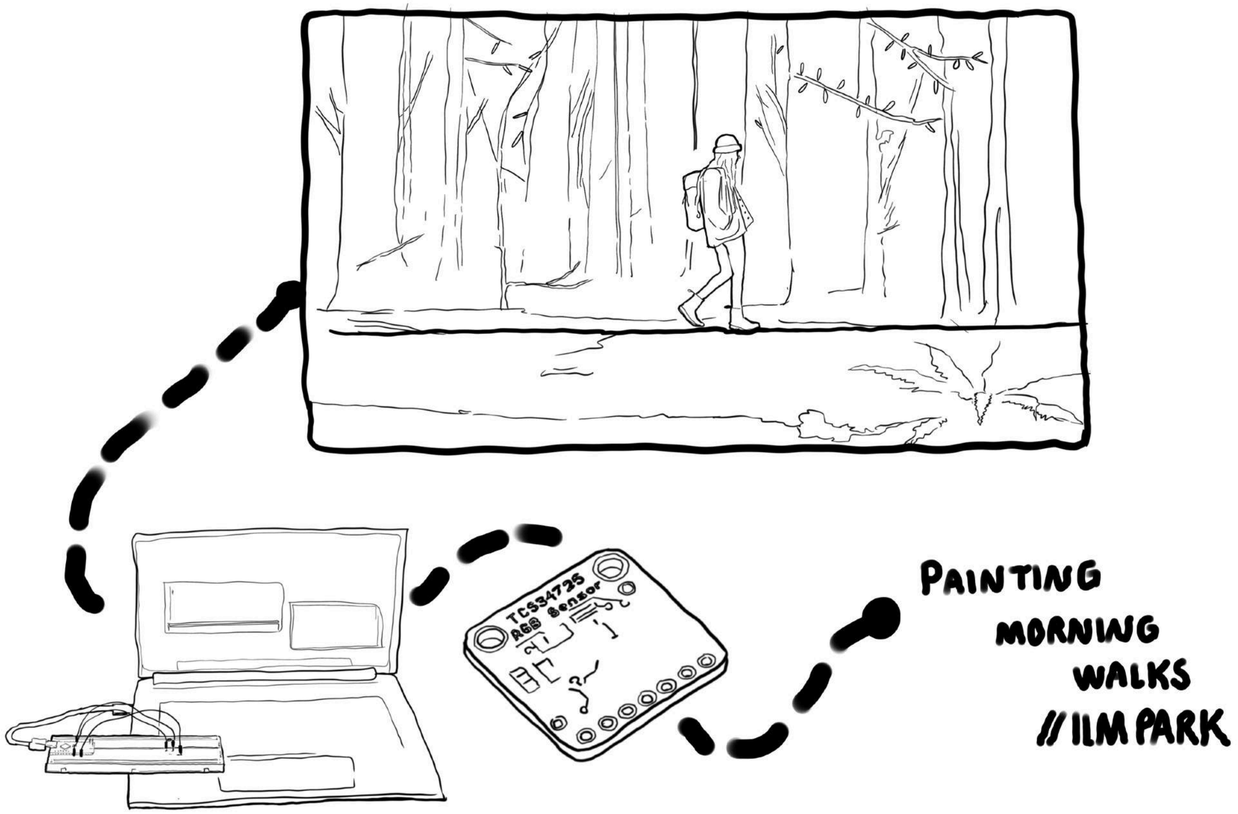
Speculative Atmosphere II


[Park an der Ilm] WEIMAR
nature
memory
moment
early morning
walk
pigment
digital
routine
Light
analogue
sound
collecting
data
silence
weather
painting
amount
keywords
- sound, bird, wind, tree
Dots
different size amount


walking
traditional practice
collecting stuff from nature
creating an archive
pigments
color sensor
specific route in Ilm Park


TCS34725 RGB Sensor
PAINTING MORNING WALKS // ILM PARK

Is there a possibility that the activities or habits we do in our daily life can turn into meaningful data?

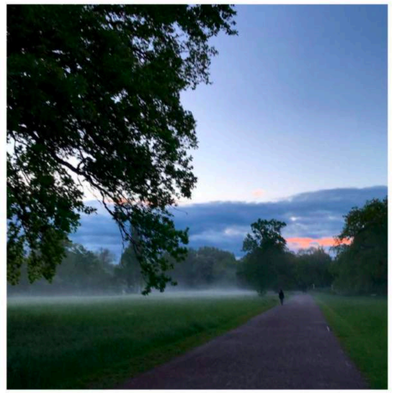
What do we encounter when this data is collected from nature with the digital method and transformed into an art practice with the traditional method?

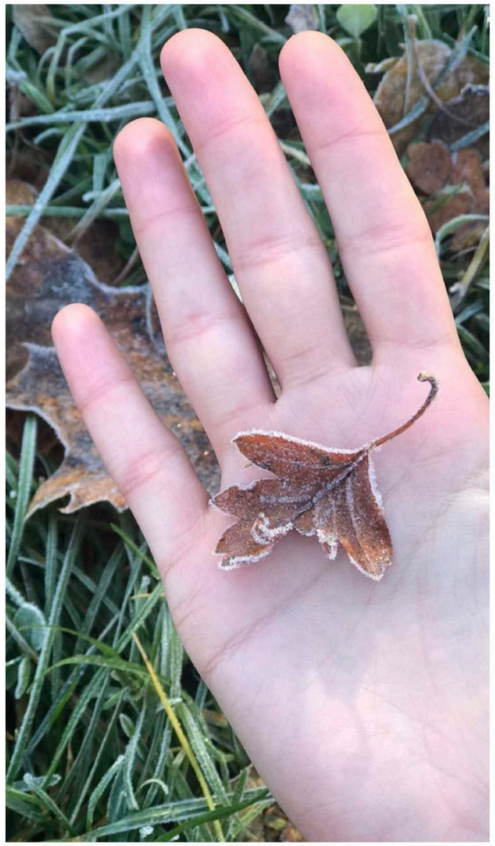
Project Method:

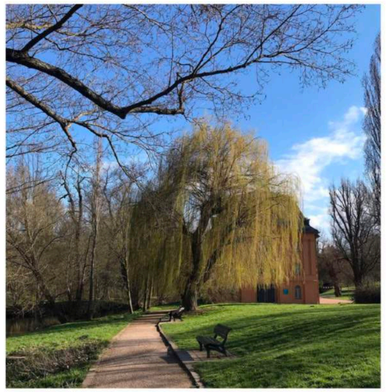
An Arduino with a color sensor records the data and this data (variety of pigments) will be use for to paint morning walks at Ilm Park.

A color will be archived while walking every morning on the specific route determined in the park.

Pigments collected from Ilm park will be detected digitally and this data will be archived according to their size and time of discovery.

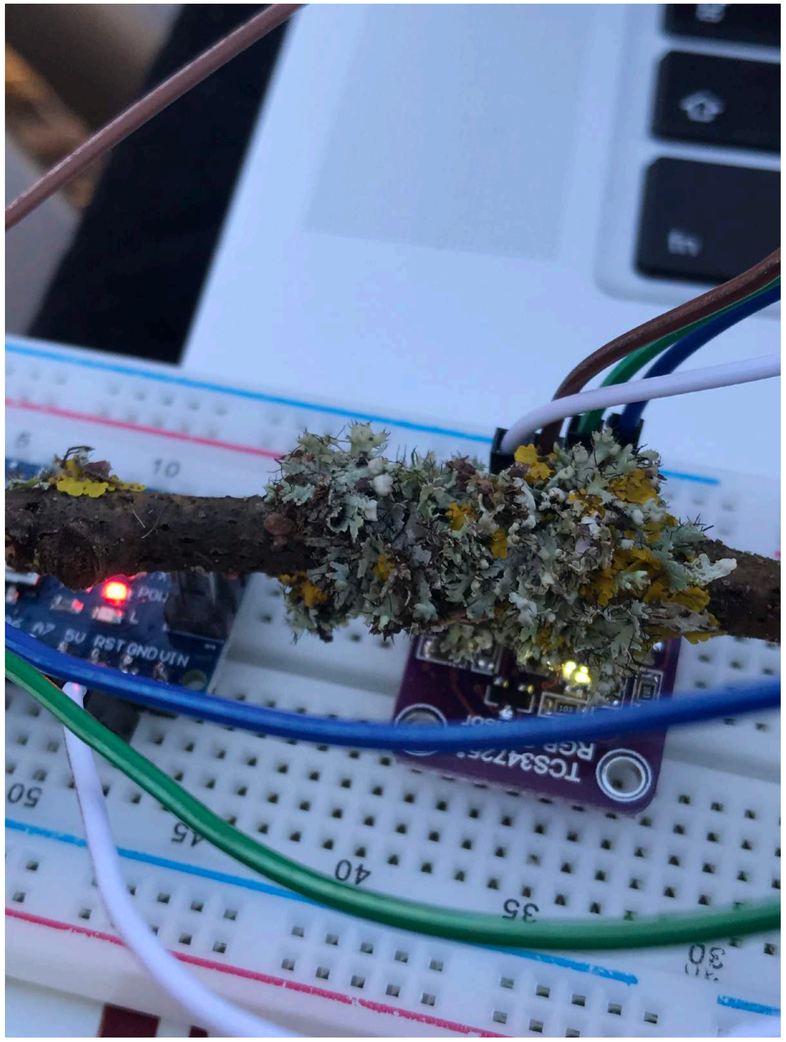



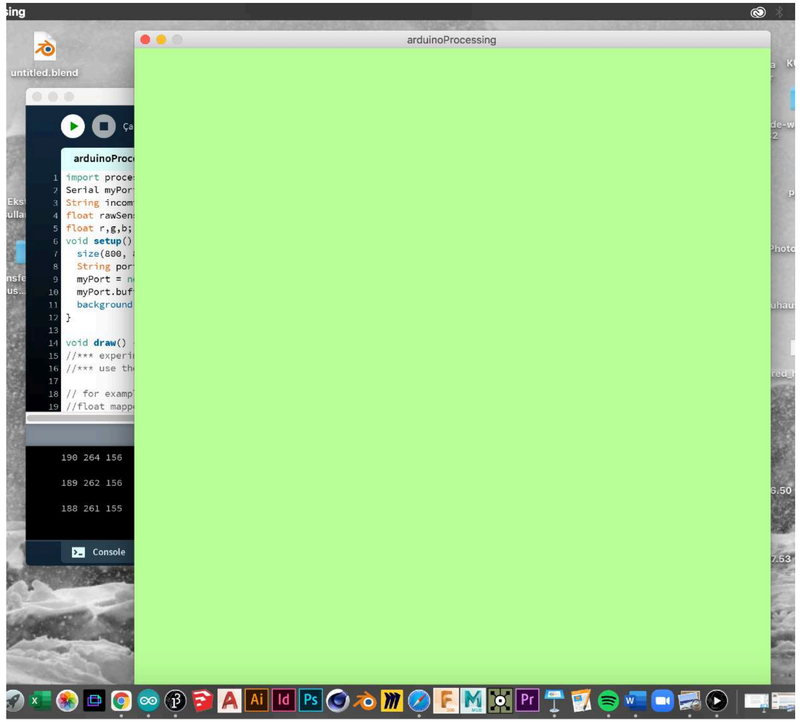




Photos from Morning Walks at Ilm Park

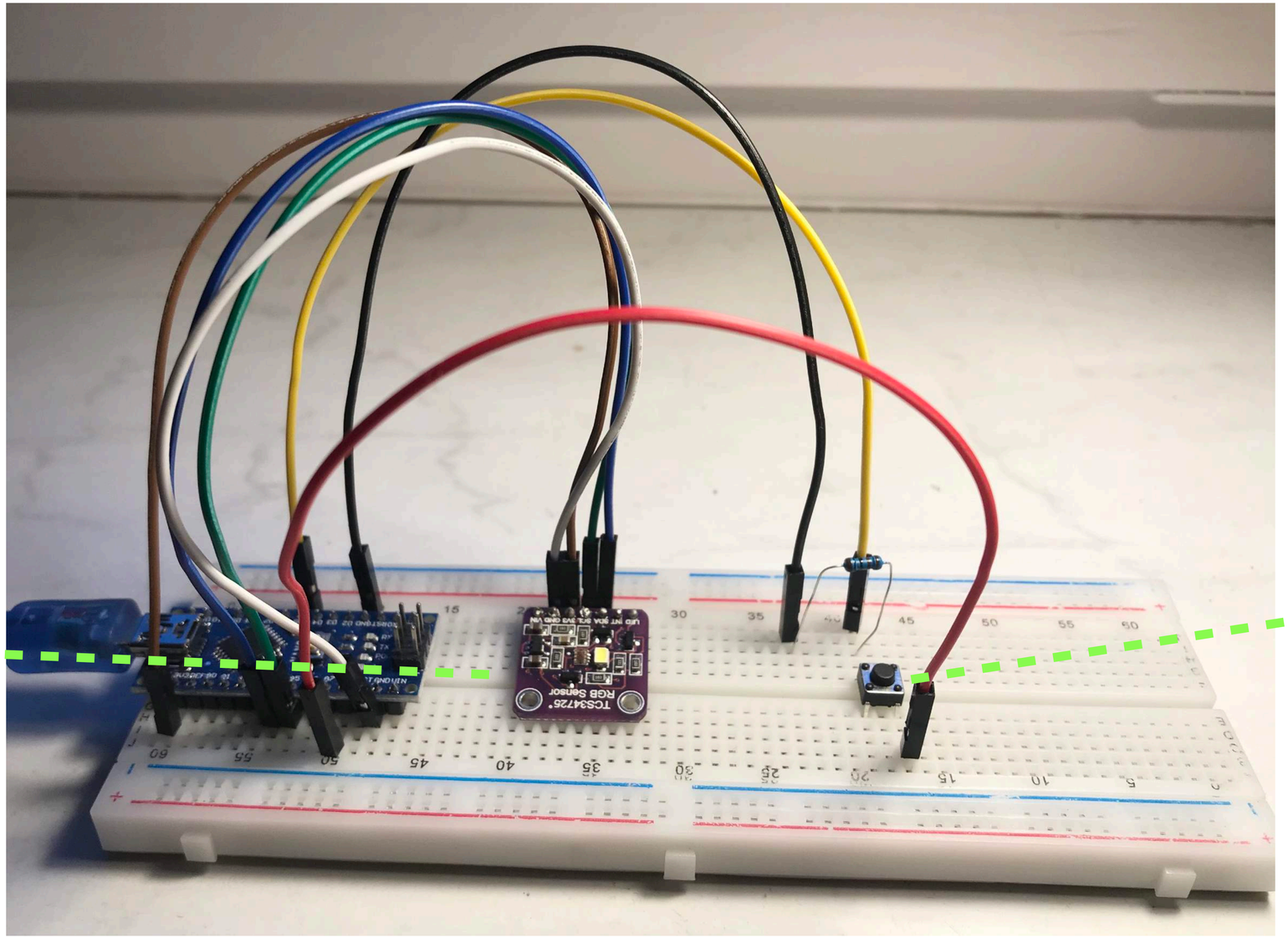
Materials

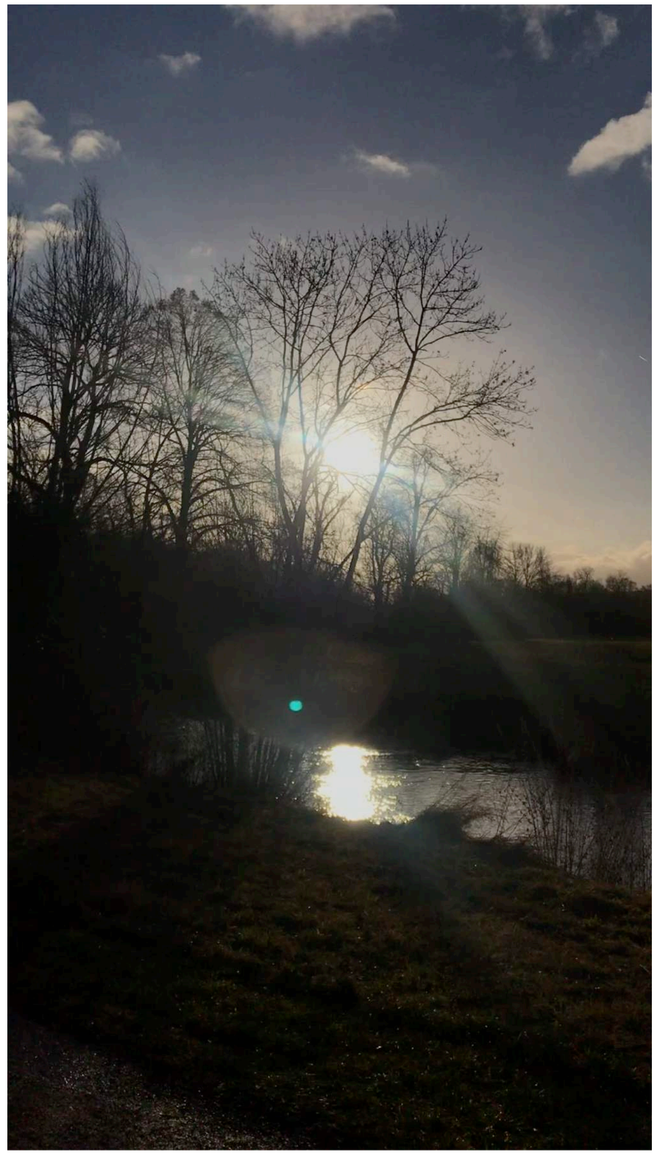


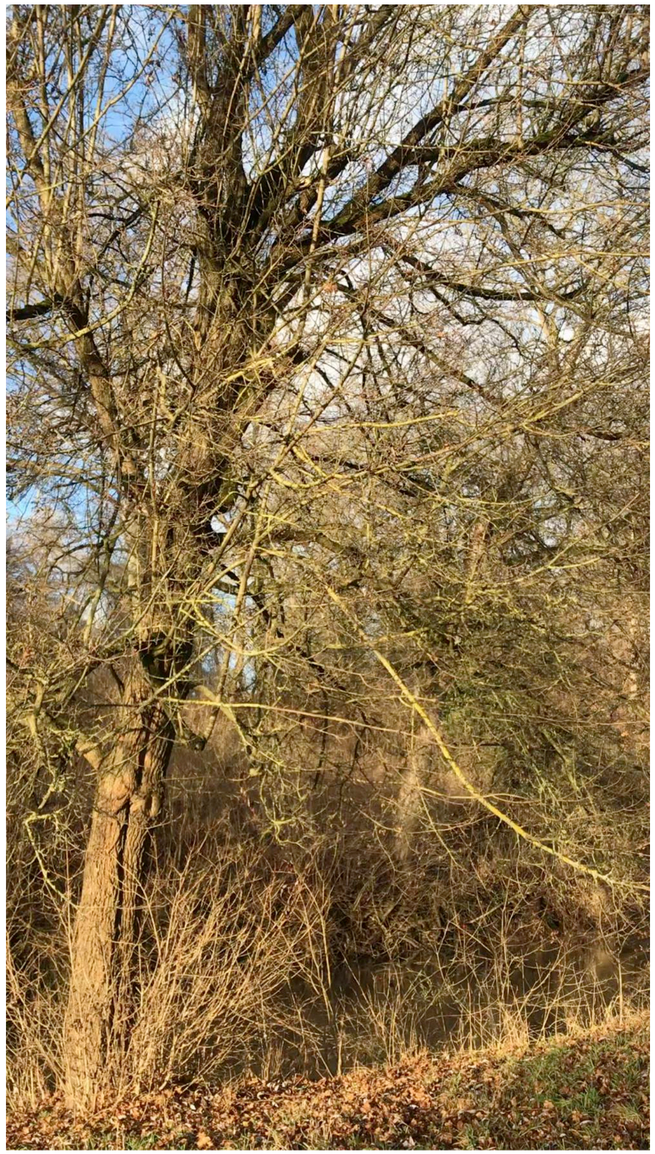


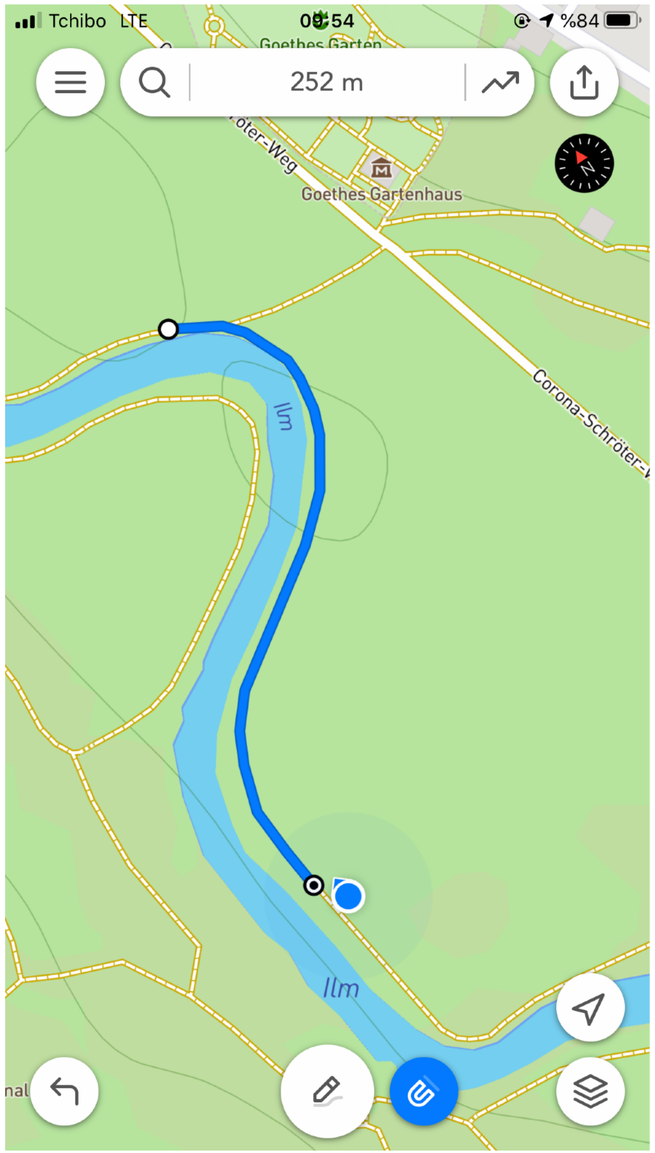
RGB colours are coming from the sensor,
And we are only reading it when the button is pressed .

Lichen

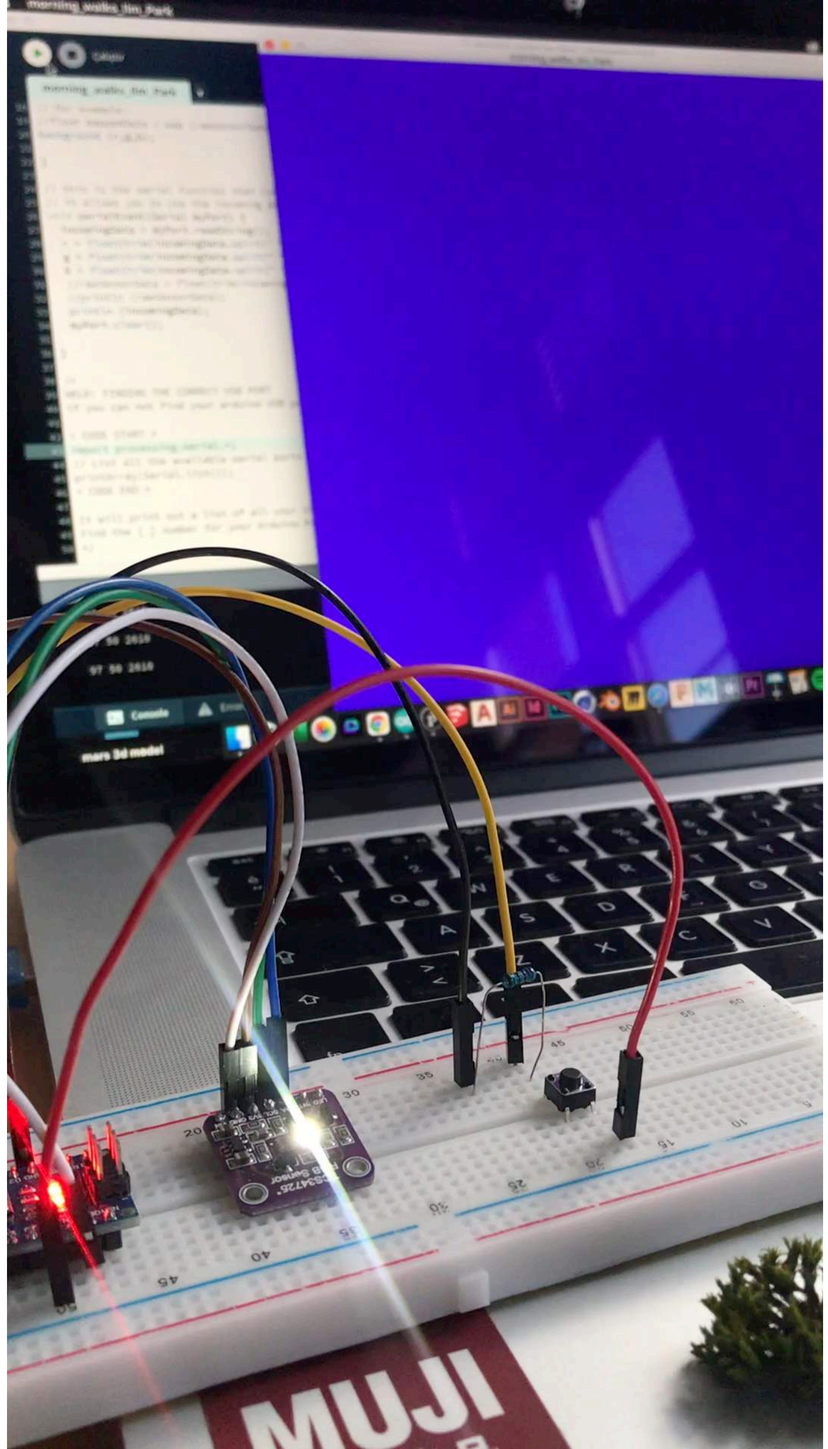

RGB Color Sensor
Button






Tchibo LTE
05:54
%84
252 m
Goethes Gartenhaus
Corona-Schröter-Weg
Ilm
Ilm

Route



-My next step is as factors like daylight and rain can negatively affect this experience while I am at while outside, (to be able to read colours with the sensor) I will make small black box and place it on top of the sensor. Colours can be read more clearly inside this black box.

```
import processing.serial.*;     // import Serial library
Serial myPort;                  // Create object from Serial class
String incomingData = null;     // create String "text" variable for incoming arduino data
float rawSensorData = 0;        // create float "number" variable for incoming arduino data
float r,g,b;
void setup() {
  size(800, 800);                           // define the size of the canvas
  String portName = Serial.list()[3];       // define the serial port. change the number in [] to a 1 or 2 etc. to match your Arduino USB port.
  myPort = new Serial(this, portName, 9600);  // create new serial object with baud rate (USB-speed) 9600 (must be the same in arduino!!!)
  myPort.bufferUntil('\n');                 // receive data until new line character comes up
  background (255);                         // make a white background. You can also put this in DRAW to refresh your canvas every draw loop.
}

void draw() {
//*** experiment here to visualise your sensor data
//*** use the variable – rawSensorData – to change things according to sensor changes

// for example:
//float mappedData = map (rawSensorData, 0, 1023, 0, 255);
background (r,g,b);

}

// this is the serial function that runs constantly in the background of your program
// it allows you to use the incoming data in the draw() function, you do not need to change this
void serialEvent(Serial myPort) {
  incomingData = myPort.readString();       // read the incoming data as String and save it in the "incomingData" variable
  r = float(trim(incomingData.split(" ")[0]));
  g = float(trim(incomingData.split(" ")[1]));
  b = float(trim(incomingData.split(" ")[2]));
  //rawSensorData = float(trim(incomingData));  // clean the incoming data String and convert it to a float data type (a number)
  //println (rawSensorData);                    // print the data to the console for inspection
  println (incomingData);
  myPort.clear();                           // clear the serial port for receiving new data

}

/*
HELP: FINDING THE CORRECT USB PORT
if you can not find your arduino USB port, make a new file and add this code:

* CODE START *
import processing.serial.*;
// List all the available serial ports
printArray(Serial.list());
* CODE END *

It will print out a list of all your USB connections.
Find the [ ] number for your Arduino Port and include it in line 8 of this code.
*/
```