Co-funded by the
Erasmus+ Programme
of the European Union

DEEDU tutorials

This document gathers all tutorials produced under the Digital Environmental Education project.

Each device tutorial is followed by two activity tutorials, the first suitable for 9-12 yo, the second for 12 + yo.

Automated gardening kit - device tutorial **6**
Introduction 6
What does it control ? 6
What actions can be performed ? 6
List of parts 6
The kit 6
The Command & Control part 6
The use cases 7
Arduino code 9
The setup 15
Electronics 16
Automated gardening kit - 9-12 yo activity idea **17**
Introduction 17
Videos 17
Example of activity
"The wireless Phone" Per groups 18
Automated gardening kit - 12+ yo activity idea **19**
Introduction 19
Home automation with snap circuits - device tutorial **21**
Introduction 21
Setting up the snap 21
List of parts 21
3D printing the snap part 21
Putting it all together 21
Circuitry activity 23
List of parts 23
Electronic prototyping and coding activity 23
List of parts 23

Home automation and energy saving 25

Create a new project in the Blynk app 26

Choose Your Hardware 27

Auth Token 27

Program the ESP32 board 28

Set up the Blynk app 29

Home automation with snap circuits - 9-12 yo activity tutorial 31

Introduction 31

Setting up the miniature smart house 33

Home automation with raspberry pi - device tutorial 37

Home automation with raspberry pi - Empty room activity 37

Introduction 37

Creation of the environment 37

What you need? 37

How to create the environment 38

How to build the box? 38

How to build the device? 38

How to prepare the Blynk app? 39

Let’s play! 41

Environmental monitoring 41

Conclusions of the activity 42

Results 42

Home automation with raspberry pi - thermostat control activity 43

Introduction 43

Creation of the environment 43

What you need? 43

How to create the environment 44

How to build the box? 44

How to build the device? 45

How to set up the Blynk app? 46

Environmental monitoring 47

Description of the environment 47

Conclusion of the activity 49

Results 49

Home automation with raspberry pi - brightness control activity 50

Introduction 50

Creation of the environment 51

What you need? 51

How to create the environment 51

How to build the box? 51

How to build the device? 52

How to prepare the Blynk app? 53

Environmental monitoring 54

Conclusion of the activity 55

Results 56

EMI Probe - Device tutorial 57

Introduction 57

Assembling the EMI probe 57

List of parts 58

Program the EMI detector 60

Watch the EMI probe in action here. 62

EMI Probe - 9-12 yo activity tutorial 62

Introduction 63

Program the EMI detector 63

Activity Idea 64

EMI Probe - 12+ yo activity tutorial 65

Introduction 65

Building the EMI detector 65

List of parts: 65

Step by step instructions for the Arduino Uno shield 66

Step by step instructions for the Arduino Nano shield 68

Place the EMI detector inside its case 71

Program the arduino board 71

Activity Idea 72

Geiger Counter - device tutorial 73

Introduction 73

Soldering the geiger counter 73

Using the geiger counter with an arduino 73

Geiger Counter - 9-12 yo activity tutorial 75

Introduction 75

What is radiation? 75

Radiation sources in daily life 76

Using the geiger counter with an arduino 77

Risk of radiation 78

Geiger Counter - 12+ yo activity tutorial 79

Introduction 79

What is radiation? 79

Radiation sources in daily life 79

Soldering the geiger counter 80

Using the geiger counter with an arduino 80
Risk of radiation: 82

Water probe - device tutorial 83
Introduction 83
Assemble the probe 83
List of parts 83
Program the arduino board 86
Using the water probe 88
Water pollution 89

Water probe - 9-12 yo activity tutorial 90
Introduction 90
Program the arduino board 90
Using the water probe 92
Water pollution 93
Plotting data on a shared online map 93

Water probe - 12+ yo activity tutorial 96
Introduction 96
Step by step instructions 96
List of parts 97
Program the arduino board 99
Using the water probe 100
Water pollution 101
Plotting data on a shared online map 102
Launching a citizen science project 104

Connected weather station - device tutorial 106
Setting up the weather station 106
List of parts 106
Wiring up the components 107
Program the ESP32 board 107
Monitor the weather station remotely via Blynk App 110
Create a new project in the Blynk app 110
Choose Your Hardware 110
Auth Token 111
Set up the Blynk app 112
Program the ESP32 board 112

Connected weather station - 9-12 yo activity tutorial 114
List of parts 114
Setting up the weather station 115
Monitoring the values recorded by the station via Blynk 115
Setting up the flying station 115

Connected weather station - 12+ activity tutorial **117**

List of parts 117

Setting up the weather station 117

Monitoring the values recorded by the station via Blynk 117

Setting up the flying station 117

Plotting the values recorded by the station on UMap 118

Automated gardening kit - device tutorial

Introduction

The automated gardening kit is a system than allow control and automatic watering of a small plant. This system is developed for an educational purpose, the plant shall stay at different pupils house to be controlled. The user will take datas every weeks from the control panel to create an database following the growth of the connected plant

What does it control ?

- The moisture of the soil, this value will be directly used to water or not the plant
- The CO2 gas concentration
- The intensity of the light
- The temperature and the humidity of ambient air

What actions can be performed ?

- A water pump will get water from a small reservoir to put it on the plant
- A screen will display the informations needed and 2 buttons to control what info to be displayed

List of parts

The kit

The kit consists of:

- the electronic control and command part
- the water tank
- pot for plant
- box for electronic (https://www.thingiverse.com/thing:4106140)
- 3d printed light sensor support (https://www.thingiverse.com/thing:3986667)
- 3d printed sprinkler (https://www.thingiverse.com/thing:3986672)

The Command & Control part

The material used for the Command and Control are the following:

- An arduino UNO board
- A CO2 Gas sensor
- A moisture sensor
- A DHT11 sensor for temperature and the humidity of the ambient air
- A photo Resistor to measure light intensity
- A DC relay to control the water pump
- A small water pump
- 2 push buttons

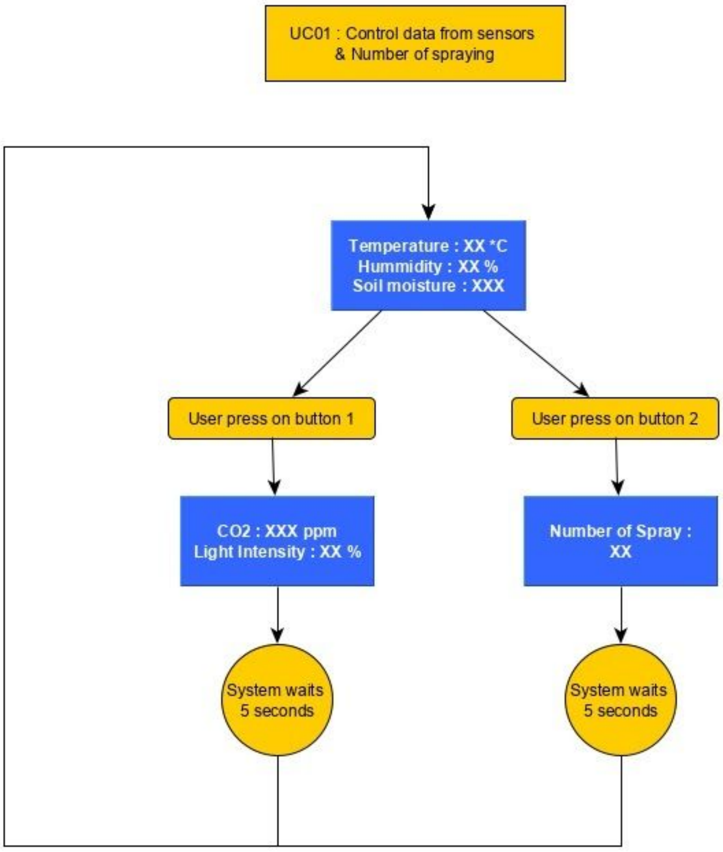
- A LED to warn user about the need to refill the water tank
- 3 x 1k Ohm resistor for the buttons and the LED

List of links to buy components:
https://drive.google.com/open?id=1LSXfRO9uICLiE4qot_-Eyt0NwxpdHKMAKepLOG5v-bs

The use cases

In this sections we present how to use to automated gardening kit.

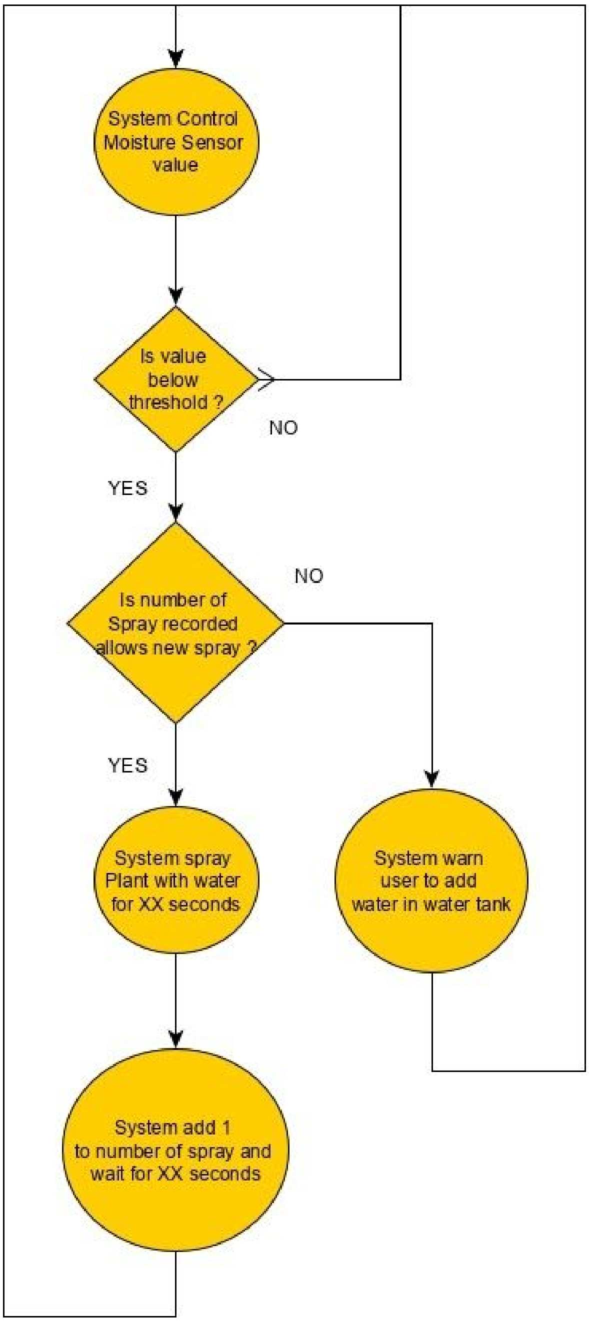
UC01 : Control data from sensors & Number of spraying
By using one button and looking at the LCD screen, the user shall be able to monitor the values recorded by all sensors in real time.



UC02 : Spray the plant when needed & Water tank level control

The system shall be able to water the plant when needed (The moisture sensor value indicates when watering is needed).
The water pump is tricky to start (if no water is present in tube, it isn't powerful enough to start), so it's best to prevent complete emptiness of the water tank. An LED is available which warns the user when it's time to refill the water tank.

UC02 : Spray the plant when needed
& Water tank level control


System Control Moisture Sensor value
Is value below threshold ?
NO
YES
Is number of Spray recorded allows new spray ?
NO
YES
System spray Plant with water for XX seconds
System warn user to add water in water tank
System add 1 to number of spray and wait for XX seconds

Arduino code

Download the full code by clicking on the link below:

https://drive.google.com/open?id=1JZUIXMrRgDBNw2atqdtvAm1ruhZdMs13l-pPuLpy3P8

```
//import the library to control LCD display, this library has been added manually through ZIP file
#include <LiquidCrystal_I2C.h>

//import library for DHT sensor
#include "dht.h"
#define dht_apin A3 // Analog Pin sensor is connected to

//create DHT object
dht DHT;

// create LCD object
LiquidCrystal_I2C lcd(0x27,20,4);  // set the LCD address to 0x27 for a 16 chars and 2 line display

//declaration of variable for buttond
int ButtonOnePin = 10;  //pin for button 1
int ButtonOneState =0 ; //one variable to store current button state
int ButtonOnePreviousState = 0; //one variable to store previous button state
int ButtonTwoPin = 9; // same for button 2
int ButtonTwoState =0 ;
int ButtonTwoPreviousState = 0;

unsigned long timeBeginDisplay=0;
int timeDisplayPeriod = 6000; //period the 2nd and 3rd screen are displayed
int valToDisplay = 0; //variable to decide what information is to be displayed

int airTemp; //variable to store air temperature measured by DHT11 sensor
int airHumidity; //variable to store air humidity measured by DHT11 sensor

int LedPin =  13;

int waterPumpPin=7; // pin on which is connected the relay that command water pump
int sprayNum=0; //variable to store number of spraying
int pumpOn =false; //variable used for spraying, it stays ON during the spray timing
int MoistureHigh = 800;
int MoistureGood = 710;
int MoistureLow = 630; //threshold that should launch water spray of the plant
int valSoil ;  //variable to store value measured by moisture sensor
int soilPin = A0; // pin on wich is connected moisture sensor
int soilPower = 12;

int timePeriod = 1000; //variable to store timing in ms, period between each display refresh, a clear is called each time display is called
int humMeasurePeriod=0;
unsigned long currentMillis; //variable to store current timing
unsigned long debutMillis=0; //variable used to store timing at the debut of a period
int timePumpPeriod = 3000; //timing for water spraying
unsigned long timePumpDebut = 0; //variable used to store timing value for the debut of a spraying period

int gazPin = 1; //pin on which is connected gaz sensor
int valGaz ; //variable on which is stored gaz sensor value

int lightPin = 2; //pin on which is connected the photoresistor
int valLight ; //variable on which light value is stored

//function to read moisture sensor value
int readSoil()
{

    digitalWrite(soilPower, HIGH);//turn D7 "On"
    delay(10);//wait 10 milliseconds
    valSoil = analogRead(soilPin);//Read the SIG value form sensor
    digitalWrite(soilPower, LOW);//turn D7 "Off"
    return valSoil;//send current moisture value
}

//function called to refresh LCD screen, displaying wanted screen regarding value of variable "valToDisplay"
int toDisplay(){

  lcd.clear(); //clear the screen to remove character that might not be overwritten by new message

  if (valToDisplay==0){
    lcd.setCursor(0,0); //to put cursor on 1rst character of 1rst line
    lcd.print("Temp:");
    lcd.setCursor(6,0);
    lcd.print("Hum:");
    lcd.setCursor(11,0);
    lcd.print("Mois:");
    lcd.setCursor(0,1); //to put cursor on 1rst character of 2nd line
    lcd.print(airTemp);
    lcd.setCursor(2,1);
    lcd.print((char)223); //special character for temperature
    lcd.print("C");
    lcd.setCursor(6,1);
    lcd.print(airHumidity);
    lcd.setCursor(8,1);
    lcd.print("%");
    lcd.setCursor(11,1);
    lcd.print(valSoil);
  }
  else if (valToDisplay==1 && (currentMillis-timeBeginDisplay)<timeDisplayPeriod){
    lcd.setCursor(0,0);
    lcd.print("CO2 :");
    lcd.setCursor(0,1);
    lcd.print(valGaz);
    lcd.print(" ppm");
    lcd.setCursor(9,0);
    lcd.print("Light:");
    lcd.setCursor(9,1);
    lcd.print(valLight);
  }
  else if (valToDisplay==2 && (currentMillis-timeBeginDisplay)<timeDisplayPeriod){
    lcd.setCursor(0,0);
    lcd.print("Number of Spray :");
    lcd.setCursor(2,1);
    lcd.print(sprayNum);
  }

  if((currentMillis-timeBeginDisplay)>timeDisplayPeriod){ //display original screen when
display period is over
    valToDisplay=0;
    //we display in this IF condition the 1rst screen because before there was a blank
screen after the timing
    lcd.setCursor(0,0); //to put cursor on 1rst character of 1rst line
    lcd.print("Temp:");
    lcd.setCursor(6,0);
    lcd.print("Hum:");
    lcd.setCursor(11,0);
    lcd.print("Mois:");
    lcd.setCursor(0,1); //to put cursor on 1rst character of 2nd line
    lcd.print(airTemp);
    lcd.setCursor(2,1);
    lcd.print((char)223); //special character for temperature
    lcd.print("C");
    lcd.setCursor(6,1);
    lcd.print(airHumidity);
    lcd.setCursor(8,1);
    lcd.print("%");
    lcd.setCursor(11,1);
    lcd.print(valSoil);
  }

}

int readTempHum (){

    DHT.read11(dht_apin); //read value from DHT sensor

    airTemp = DHT.temperature; //store temperature value from sensor
    airHumidity = DHT.humidity; // store humidity value from sensor

    //delay(3000);//Wait 3 seconds before accessing sensor again, without timing, sensor
measurement is faulty. we commented this delay because this function isn't called often
}

int waterPump(){

  //the condition below to command or not the water pump
  if (valSoil<MoistureLow && pumpOn==false){
    timePumpDebut=currentMillis;
    pumpOn=true; //this value change, so next water pump cycle is launched
  }

  //once the waterpump cycle is launched, this if condition will turn ON the pump on the
third of the timePumpPeriod defined, the other 2 third, the water pump will be turned OFF
  if (currentMillis-timePumpDebut<timePumpPeriod && pumpOn==true){
    if((currentMillis-timePumpDebut)<(timePumpPeriod/3)){ //time proportion between
spraying and non spraying
      digitalWrite(waterPumpPin,HIGH); //turn ON relay so water pump
    }
    else{
      digitalWrite(waterPumpPin,LOW);  //turn OFF relay so water pump
    }
  }
  else if (currentMillis-timePumpDebut>=timePumpPeriod && pumpOn==true){
      pumpOn=false; //once pump cycle is over, variable value change to make water spraying dependant on moisture value again
      sprayNum++; //add +1 to the number of spraying performed
  }
}

int waterTankAlarm(){
  if (sprayNum>10){ //when number of spraying is above limit, turn on led to warn user
    digitalWrite(LedPin,HIGH);
  }
  else {
    digitalWrite(LedPin,LOW);
  }
}



void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  //defining wich pin are IN or OUT
  pinMode(LedPin, OUTPUT);
  pinMode(ButtonOnePin, INPUT);
  pinMode(ButtonTwoPin, INPUT);
  pinMode(waterPumpPin, OUTPUT);

  // initialize the lcd
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Waking up");

  //ready first temperature and humidity from DHT sensor to store value (since this function isn't called often)
  readTempHum();

  delay(2000);
  lcd.clear();


}

void loop() {

  //reading of sensor value, not DHT due to the delay needed to its measurement
  readSoil();
  valGaz = analogRead(gazPin);
  valLight = analogRead(lightPin);

  currentMillis=millis(); //measurement of current time

  //small function to avoid button rebound, when pressing once on the button, arduino
reads only one pressing...
  ButtonOneState=digitalRead(ButtonOnePin); //reading of button value
  if (ButtonOneState !=ButtonOnePreviousState){ //comparing with previous state
    if (ButtonOneState == HIGH){ //if previous state was 0 and current is 1, then take button
pressing in account
      delay(200);
      valToDisplay=1;
      timeBeginDisplay=currentMillis;
    }
    ButtonOnePreviousState=ButtonOneState;
  }
  //same function for button 2 as button 1
  ButtonTwoState=digitalRead(ButtonTwoPin);
  if (ButtonTwoState !=ButtonTwoPreviousState){
    if (ButtonTwoState == HIGH){
      delay(200);
      valToDisplay=2; //particular value for "valToDisplay"
      timeBeginDisplay=currentMillis;
    }
    ButtonTwoPreviousState=ButtonTwoState;
  }

  waterPump(); //calling for water pump function which launch spray cycle if needed

  waterTankAlarm(); //calling function to warn user about water tank level

  //small condition for screen refreshing and measurement of DHT sensor
  if (currentMillis-debutMillis>timePeriod){ //end of a period
    debutMillis=currentMillis; //define the begining of a new period
    toDisplay(); //refresh of the screen regarding "valToDisplay" value
    humMeasurePeriod++; //DHT sensor values are measured every 20 times the screens
is refreshed
    if (humMeasurePeriod>20){ //go to measure DHT values
        readTempHum();
        humMeasurePeriod=0;
    }

  }
```

```
}
```

The setup

Moisture Sensor calibration
The value of the moisture sensor is important because it is based on this number that the water pump will be activated.

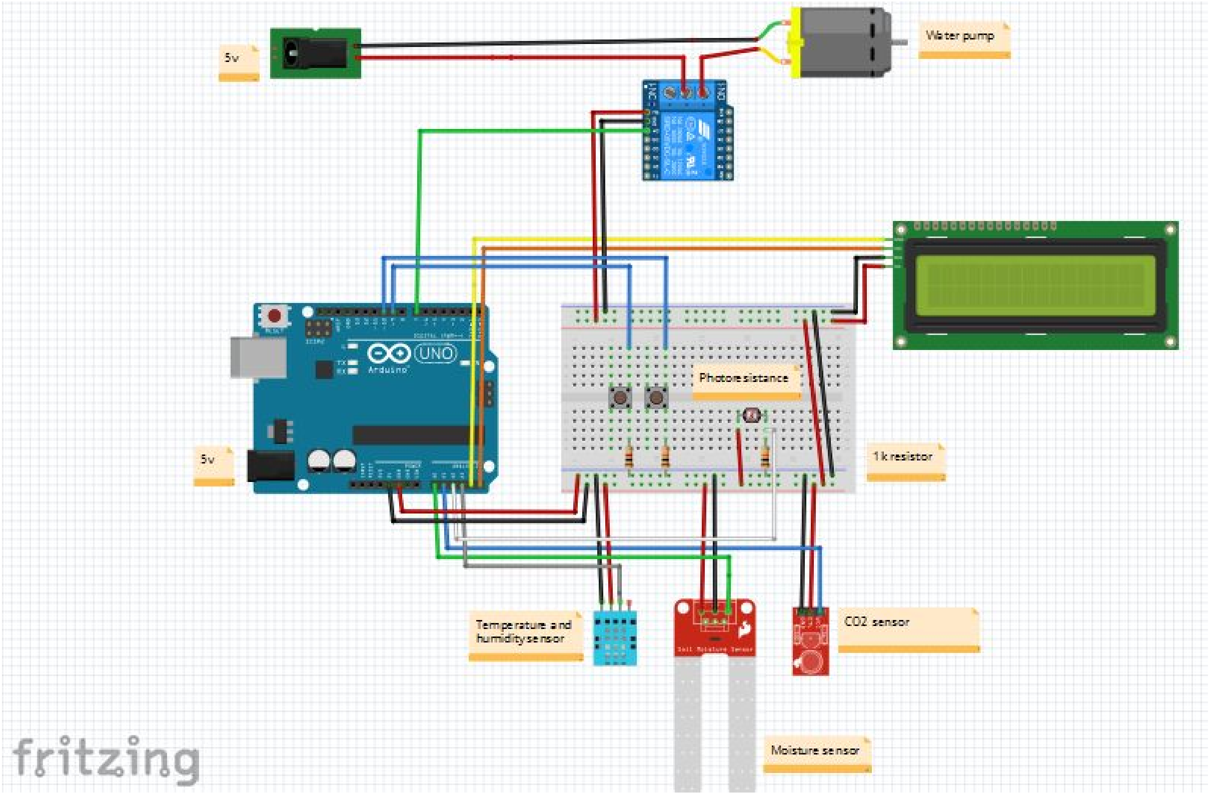
MoistureLow variable to be modified to change water pump behavior (no need to be modified, change it only regarding your project and the needs of the plant).

timePumpPeriod is the variable in milliseconds of the arduino code to be modified regarding the volume of water sprayed. (no need to be modified, change it only regarding your project and the needs of the plant).

timePeriod timing in millisecond that defines timing between each refresh of the screen

timeDisplayPeriod timing in milliseconds that define the length of time values are displayed on the LCD screen.

Electronics



Digital pins

Pin 7, relay
Pin 9, button 1
Pin 10, button 2
Pin 13, LED

Analog pins

Pin A0, Moisture sensor
Pin A1, Co2 sensor
Pin A2, Photoresistor
Pin A3, Temperature sensor
Pin A4, Screen Rx
Pin A5, Screen Tx

The Arduino is to be powered by the same device that powers the water pump. This way the whole system needs only one electric plug to work.

The Water pump is a 5V alimentation, the arduino can support that voltage but this leads to a warming of internal component of the arduino and possibly damage to the board with time. The easiest and fastest choice for now is two 5v power supply (one for arduino and one for the pump).

A page presenting the different solutions

Automated gardening kit - 9-12 yo activity idea

Introduction

The purpose of this activity is to exchange about plants which are connected to an automated gardening kit. Groups of kids will connect to and engage with different other groups of youth from across Europe. The goal of the activity is to provide this groups of youth with basic knowledge and confidence that will then enable them to exchange more thoroughly about electronics, environmental science, citizens science, etc.

The activity can be implemented using smartphone on which kids will be viewing a series of videos. The educator can alternatively use a video projector to surf on YouTube and show the videos to the whole classroom.
Videos will be accessible through QR codes, laser engraved on wooden items.

Videos

the plant kingdom
https://www.youtube.com/watch?v=IYxfz1PSfZ0&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=2
The trees
https://www.youtube.com/watch?v=5I7u5FMQxHA&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=7
The ground
https://www.youtube.com/watch?v=fnLmTwK6hyg&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=6
Water
https://www.youtube.com/watch?v=bGWr5jXJfbs&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=10
The sun
https://www.youtube.com/watch?v=RzkJkEKV8Yk&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=5
Photosynthesis
https://www.youtube.com/watch?v=UPBMG5EYydo&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=1
The Atmosphere
https://www.youtube.com/watch?v=LGvcwk5d-zM&list=PLD7uTVLDEOKvvJGoISPELs4pePjJHnutw&index=3

The video should be watched in group and the teacher should encourage kids to later describe their contents with their own words, to evaluate the extent to which they understand the topic of the video.

At a later time kids will be invited to observe the connected plant available in their classroom.

Key words

Roots - the part of a plant which attaches it to the ground and are used to take water for soil.
Water - is a liquid present on 70% of our planet and the first need for life
Sun - is a star from who our planet is turning around
Light - is created by lamp or sun to make things visible
Soil - ground of earth
Moisture - water in a solid or condensed on a surface
Plant - is a living organism like trees, herbs...
Leaf - is a flat part of a plant who make plants breathing
Tree - is a kind of plant
Co2 - carbon dioxide is a gas (created by human in breathing and cars, planes, boats…)
Temperature - measured with degree to know if it’s cold or warm
Wood - is an material who compose the stem of a tree
Stem - stem is the structural part of the tree
Seeds - are small reproducing parts created by plants to reproduce themselves
Crate - a crate in a box with the top open
Pot - is an recipient
Screen - is an electrical component where we can see or read values
Sensor - are electrical component to take information
...

Example of activity

"The wireless Phone"

The teacher can give a video to watch on smartphone by on kids of the group with an headphone, this video will be in english after one or two watchings, the kid 1 can tell to a second kid the content of the video and explain in english using the lexical what he watch, during this time the other person of the group should put their hands on their ears.
After 1,2,3,4 kids the content will be degraded but the last kids will perform an presentation in front of the group or class what he learned. Then the video can be watched by everyone.
In this activity we want to involve the smartphone as a tool, the communication in english and speaking in front of public.

Automated gardening kit - 12+ yo activity idea

Introduction

The objective of this activity is to gain an understanding of plants and the environment by observing and recording data from a plant connected to electronic sensors. We suggest participants take the level 1 activity first. This way, kids will already have an awareness of how a plant is growing and will have observed how plants grow in different countries/settings. The first activity will also provide introduce them to technical words related to plants and electronic sensors.
In this activity, participants will be compiling a form relative to their own plant. They will then share these data with groups of youth based in different countries and working with the same automated gardening kit.

The information compiled by the form are:

-the status of the plant alive or not: Important for the rest of the test

-the number of spreading water: With the number of spreading we can observe the needs of water of the plant and the soil, and determitated if plant have maybe to much sun or an too warm room who dry the soil.

-The values of humidity in the ground: This number is one of the most complicated to understand is expressed in Ohm and it’s the ‘’resistance’’ of the ground, 1000 is full water conductivity the electricity goes easy without resistance and losing intensity, 0 means that the electricity can’t go because it’s totally dry

-the value of the light sensor: The light (Sun) is an important component of the photosynthesis operation because plant need sun to change CO2. the sun is also an part of the evaporation of water from soil.

-the value of the CO2 sensor (PPM particle per meter): More CO2 is an sign of quality in air and indicator of pollution, plants are taking CO2 and transform in Oxygen using sun

-the actual temperature (in degree celsius): Temperature is an value important for the evaporation of water and the life of the plant because some plants like to have an medium temperature to live well.

-Picture of the plant: the last step in the form is the picture who is an good way to observe the plant, how much leaves, are there green or brown ?

Link Poland:
https://drive.google.com/open?id=1IX100fPAd3YvFqwG2p3ea-qEHPtrWBNkZUbHxvGP7rE
Link Italia:
https://drive.google.com/open?id=1PBVMe7RaCgHcP3JjQk-qpPTddIVCY83IQSGuJS7d2uE
Link France:
https://drive.google.com/open?id=1IAivMgdjOcWVDZqCcx5EISk3xZ7Ezl66t2mk11vk8g8
Link Czech republic:
https://drive.google.com/open?id=1WQokUqUYCMyWSu4uWCyDs4SyBx7fpaaJ3rlGzKKIQog

Access to spreadsheet (here you can consult results)
Poland:
https://drive.google.com/open?id=0B7VAaBIM_k8hflp2YWpoSUs0N01NVE44ZVJFNllaZjNoTDc1VjNmSVU2U0M1dUktVzJhamc
Italia:
https://drive.google.com/open?id=0B7VAaBIM_k8hfnMzSkNSSDJ0X28tYWFCdW14UDU1ZWg0T3h5MnNrLUZ1SjA3WjFHdGhXVjQ
France:
https://drive.google.com/open?id=0B7VAaBIM_k8hfkd6MHFpMi1VQndDdVpIQXR4LXVldjRFUHVIMVVLWHRrcE1GekxWeDZzNXM
Czech republic:
https://drive.google.com/open?id=0B7VAaBIM_k8hfldtV2piTUNYcEhOejRWTGtiR0FVa1BKWVlJTTM2UkNQZDlKWXpualQxajA

At every moment all the classes will have access to the form to observe (with numbers) the growth of their plant.

Every month classes can have a video call to exchange in english the value that they observed and the size of the plant.

If one plant is growing more we can use the data from the spreadsheets to observe that one maybe had more lights or an higher temperature and then develop less or more...This activity and data can be used by teacher to sensibilize student about citizen science, global warming , deforestation or other topics related to environment.

This plant will grow on the long term and students will observe development of it week after week, an introduce time needed to plants for growing.

Home automation with snap circuits - device tutorial

Introduction

Snap circuits are a fun support to introduce kids to circuitry and electronic prototyping. They can also be used to tackle topics related to energy saving.

In this tutorial, you will learn how to create your own snap circuits embedding electronic components of different kinds, and how to implement educational activities revolving around this support: activities on circuitry, electronic prototyping and programming, energy saving and home automation.

Setting up the snap

List of parts

1x 3D printed snap support
1x electronic component (ex. Led, buzzer, mini vibrating motor)
2x 12x6mm magnets

3D printing the snap part

First, you will need to 3D print the snap part. The stl file ready for 3D printing is available here.
You can also tweak this design on tinkercad and make it suitable to hold 3-legged components (ex. temperature sensor, light sensor).

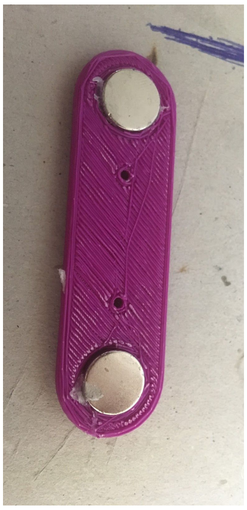
A timelapse of the 3D printing process is available here.

Putting it all together

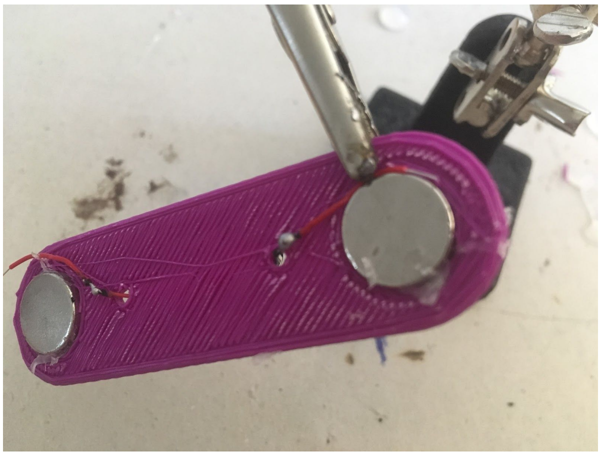
A timelapse of the assembling process is available here.

Put some hot glue around the perimeter of each hole.

Place the 2 magnets on the snap part, one per each hole. Make sure each magnet is securely fixed to the snap via the hot glue.



Next put your component into place and solder each leg onto one magnet.



You can thus create snap components containing LEDs, buzzers and vibrating motors. That's just 3 examples, however, in principle any electronic component can be mounted on a snap support.

Circuitry activity

List of parts

- several snap supports
- 1 power supply (3V should be enough)
- crocodile cables

The circuitry activity is an introduction to electricity and circuits, enabled by the snap supports.
You can use different snap parts to create simple circuits, arranged in series and parallel.
To create a simple series circuits, place two snap parts (ex. LED snap and mini dc motor snap) as pictures below. Then power the circuit by plugging a battery pack (3V is enough to power a few electronic components). one end of the circuit goes to the + of the battery pack, the other end goes to the -. Beware of the polarity of the LED (anode and cathode must be connected to positive and negative of the battery pack respectively), otherwise the LED snap won't light up.

To create a simple parallel circuit, place two snap components as in the picture below.

Then power the circuit by plugging the ends of one snap part to a battery pack. Again, beware of the polarity of the LED snap. Connect the anode to the + of the battery pack, the cathode to the -.

Electronic prototyping and coding activity

List of parts

- 1x Arduino board (arduino Uno or nano or any other version are Ok) + USB power cable
- crocodile cables
- jumper wires (male-male)
- snap components

In this activity, you'll be using the snap components to build an electronic system and subsequently program it via a visual coding software.

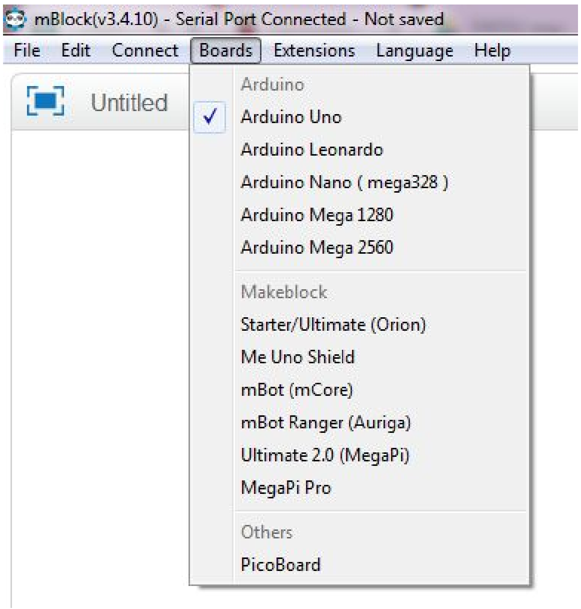
Download mBlock 3, not the latest version.

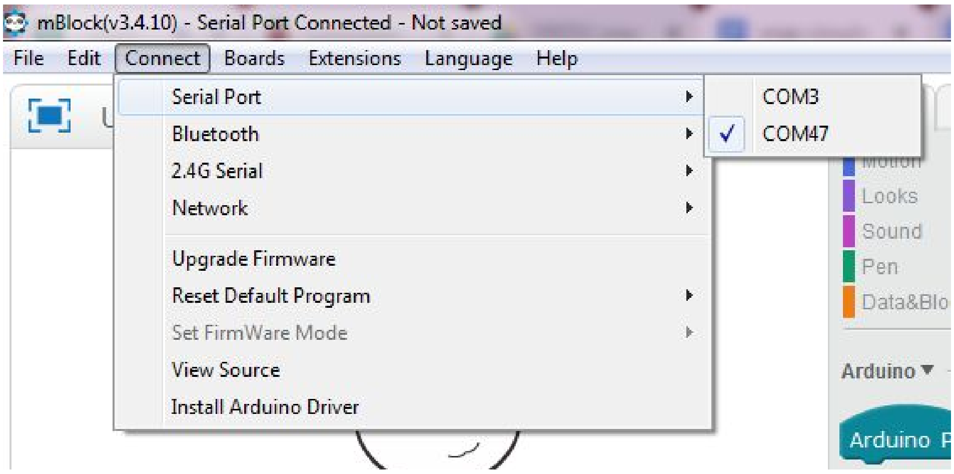
You can plug the snap components to the arduino board, as if it were regular electronic components. Remember that some components (ex. LED) have polarity, so make sure to connect the anode with a digital pin and the cathode to GND.

Try for example to have a snap Led blink.

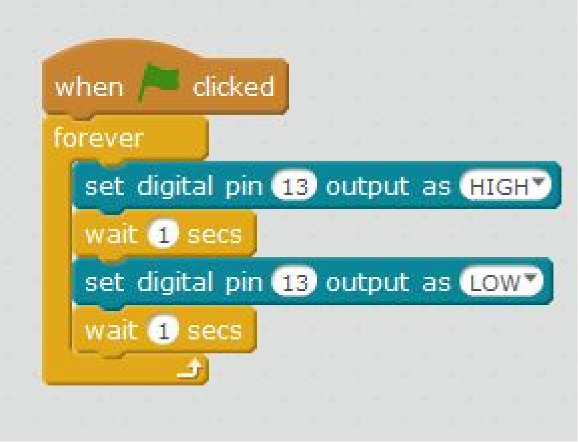
First of all, wire the snap component to the arduino board, and plug the board to your pc.

Next, launch mBlock, select the board that you re using under “Boards”, and connect to it by clicking on “Connect” and selecting the correct port (in the example below COM47).





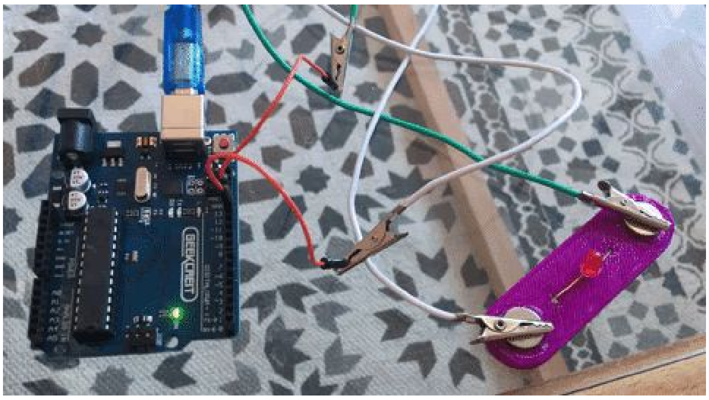
Move the available programming blocks around to obtain this:



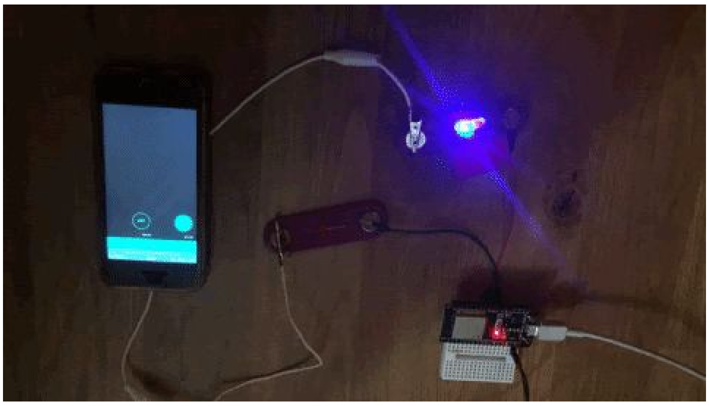
```
when 🏁 clicked
forever
    set digital pin (13) output as (HIGH ▾)
    wait (1) secs
    set digital pin (13) output as (LOW ▾)
    wait (1) secs
```

We’ve connected the Led snap to pin 13, if you choose a different pin don’t forget to select the correct pin in the code too.

To launch the program just hit the Green Flag.



Home automation and energy saving



Snap components can be used to run activities on connect objects. It is possible for example to arrange miniature electrical appliances in a miniature house, and control them remotely. Being able to remotely control one s appliances gives the user the obvious advantage of being able to choosing when they're running and when they aren’t, thus contributing to saving energy and making the miniature house as energy efficient as possible.

We've designed a number of 3D printable miniature electronic appliances that can be placed on top of a snap component. You can for example imagine to place the miniature oven on top of a Led or a miniature 3D printer on top of a mini vibrating motor snap, thus emulating real-life operations of those appliances.

Find all appliances available for 3D printing by clicking on the links below:

snap circuit tv

snap circuit stove

snap circuit 3D printer

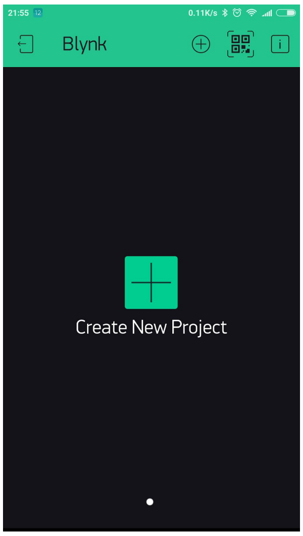
snap circuit mixer

snap circuit washing machine

This activity will require the Blynk application. So, first download Blynk on your smartphone.

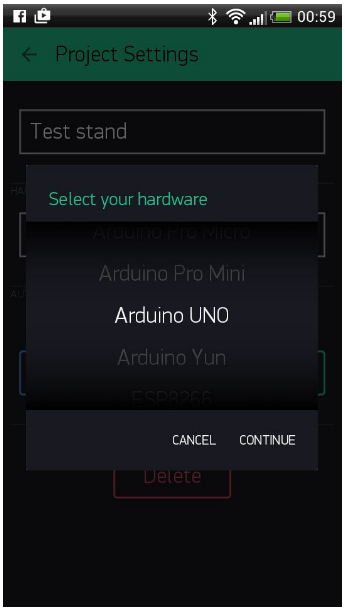
Create a new project in the Blynk app

After you’ve successfully logged into your account, start by creating a new project.



Choose Your Hardware

Select the hardware model you will use. If you are following this tutorial you ll probably be using an ESP32 board.



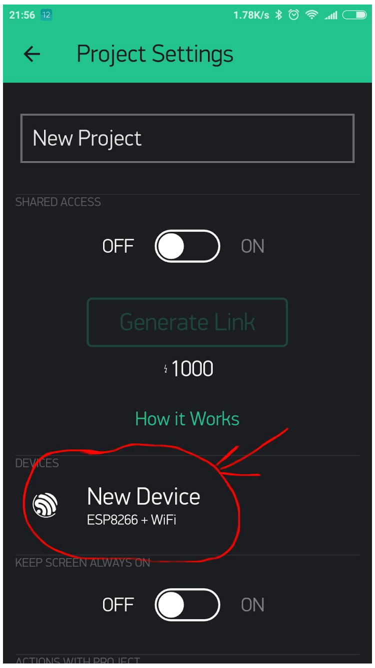
Project Settings

Test stand

Select your hardware

Auth Token

Auth Token is a unique identifier which is needed to connect your hardware to your smartphone. Every new project you create will have its own Auth Token. You'll get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device:



Project Settings

New Project

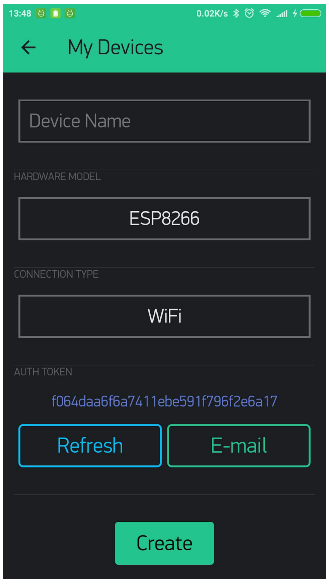
SHARED ACCESS: OFF

DEVICES

New Device
ESP8266 + WiFi

KEEP SCREEN ALWAYS ON: OFF

And you’ll see token :



My Devices

Device Name

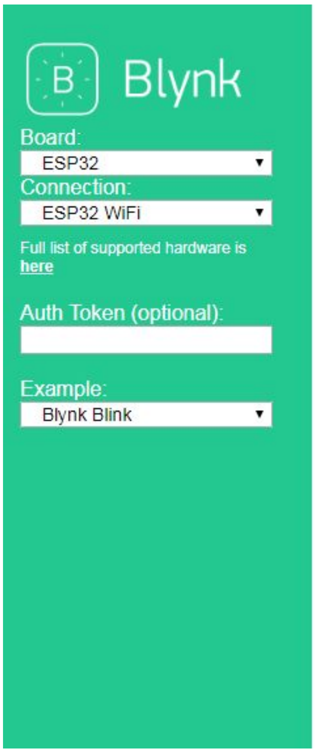
HARDWARE MODEL
ESP8266

CONNECTION TYPE
WiFi

AUTH TOKEN
f064daa6f6a7411ebe591f796f2e6a17

Program the ESP32 board

Head to this website, select your hardware, the connection mode (ex. wi-fi) and choose the Blynk Blink example.



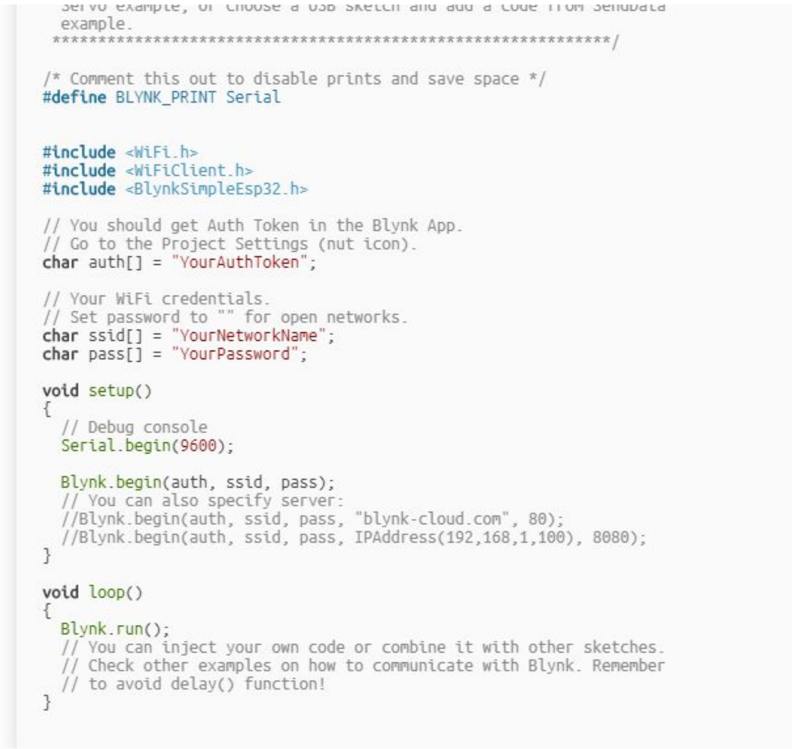
Board: ESP32

Connection: ESP32 WiFi

Full list of supported hardware is here

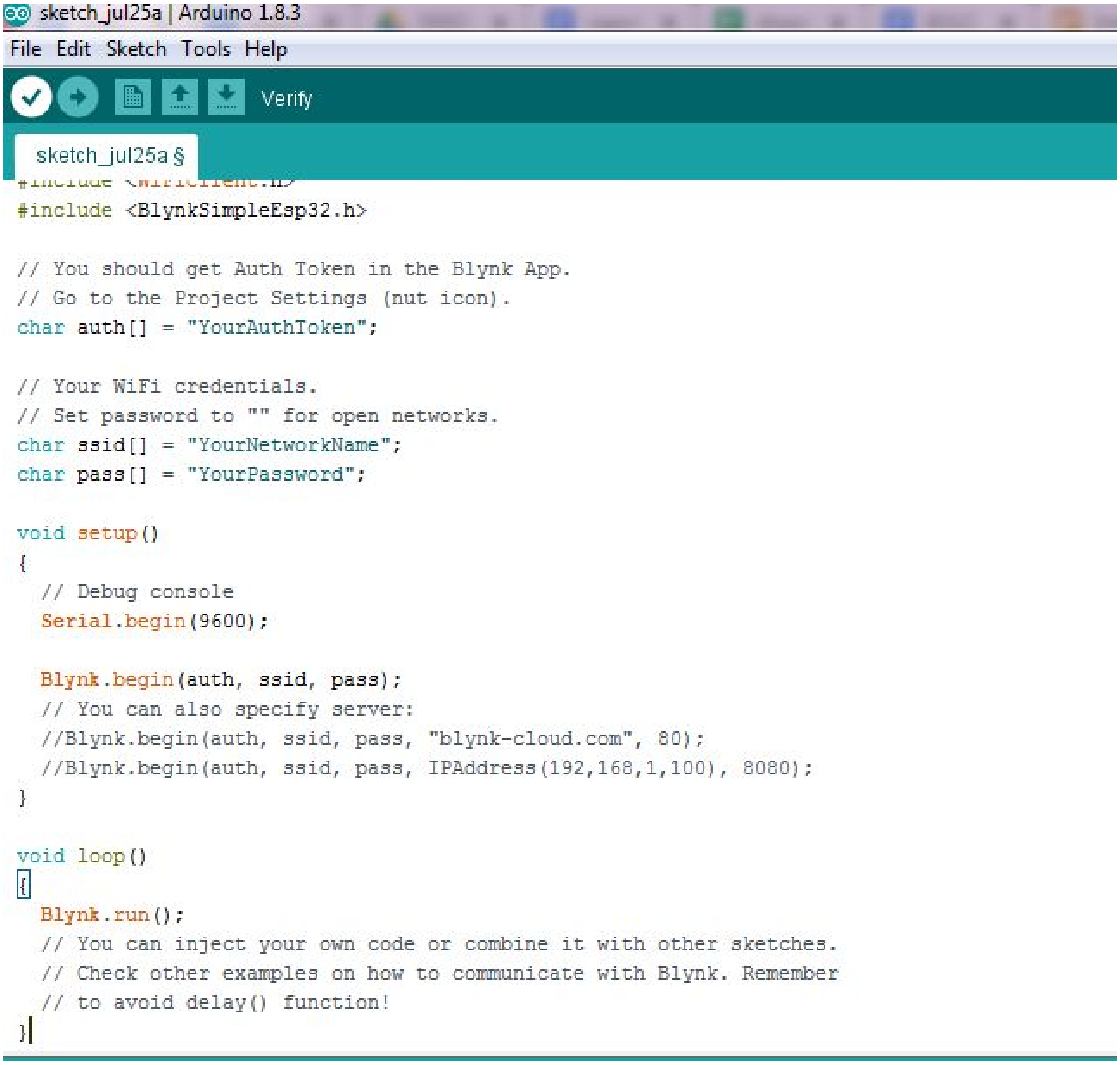
Auth Token (optional):

Example: Blynk Blink



```
  servo example, or choose a USB sketch and add a code from SendData
  example.
 *************************************************************/

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial


#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

Copy the code and paste it on Arduino IDE (prior to that, make sure you select the correct board and the correct port - under “Tools”-).



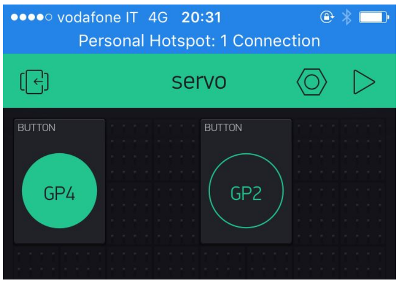
```
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

Replace “YourAuthtoken” with the token available on the app, replace “YourNetworkName” and “YourPassword” with your wi-fi credentials.

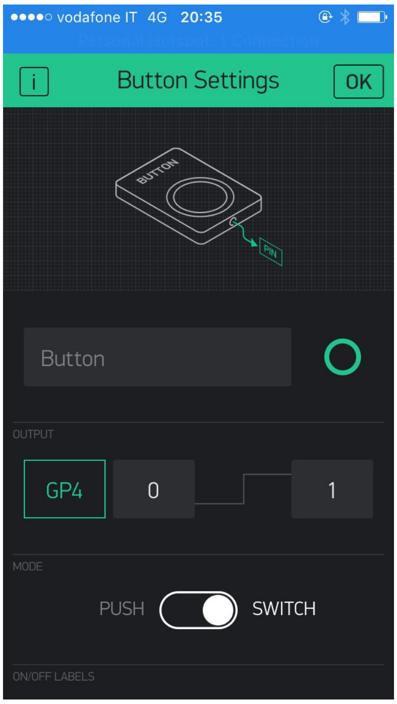
Finally, upload the code onto the board.

Set up the Blynk app

In your Blynk project, choose button widgets, as many buttons as you have snaps to control remotely. In our example we’ll add two buttons widgets since we have two snap parts to control (both are LEDs).



Next select the first button and, under output, choose the port to which one of your snap is connected to the ESP32 board (ex. GP4). Make sure to have 0 and 1 next to GP4, just like in the picture below. You can also choose whether the button will function in mush or switch mode.



Button

OUTPUT

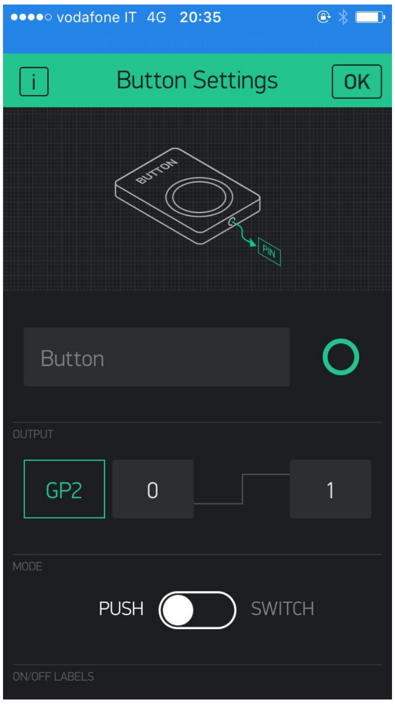
GP4 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

Do the same for the second button, only this time connect to the relevant ESP32 pin (ex. GP2).



Button Settings OK

Button ○

OUTPUT

GP2 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

Finally, launch the app by clicking on the Play symbol. If everything goes OK, you ll be notified that your project is online, and you shall be able to remotely control your snaps.

Home automation with snap circuits - 9-12 yo activity tutorial

This activity is about setting up a miniature house and maximise energy efficiency.
Kids will be integrating miniature appliances connected to snap circuits, and program the snaps via arduino.
The activity is suitable for groups of children aged 9 to 14 yo.

Introduction

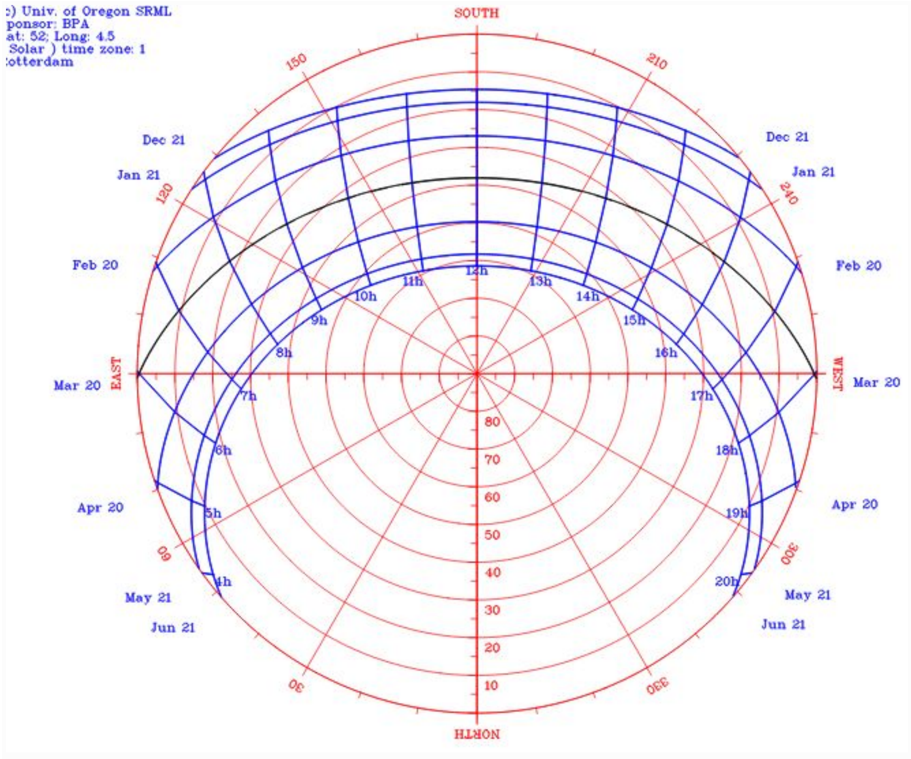
First of all, let's get clear on what energy efficiency is.
A warm living room in winter or a sports stadium brightly illuminated at night – we use energy to achieve a specific benefit. Energy efficiency is a means of measuring the energy-expenditure required to achieve a certain benefit. The lower the losses in energy to achieve a specific purpose are, the higher is the degree of energy efficiency.

Energy demand is increasing worldwide. The energy market situation is heating up and energy prices are on the rise. Instabilities in many exporting and transit countries are a cause for concern and the increased combustion of fossil energy sources is accelerating climate change. An expansion of energy supply options is costly and will take time. On the other hand, increasing energy efficiency curbs energy prices, reduces dependency on energy imports, counteracts energy distribution conflicts and cuts climate-damaging carbon dioxide emissions.

So how can we achieve energy efficiency in our own living environment?

Consider the Known Conditions: Sun, Wind and Light

The orientation of your home is one of the factor that can impact on energy efficiency. Obtain a sun path diagram for your site's location. This will help you determine the orientation of your home by giving a visual of where the sun travels in the sky throughout the day. Generally speaking, the best tip for minimizing energy consumption is to orient the home facing south to capture solar gain in the winter and block solar gain in the summer.



So how do you read a sun path diagram?

The image above is a sun path diagram for the city of Rotterdam, in the Netherlands.
We can tell that in Rotterdam, on June 21st, at 20h, the sun is located at 310°, which is to say very close to the geographical north. On April 20th, at 18h, the sun is located at about 280°, which is to say towards the geographical west.

Also, it's good to keep in mind that the sky is generally speaking three times brighter at the zenith than at the horizon, so if you live in a predominantly overcast area this is the perfect environment to let in that bright light from above and maximize free illumination.

Other factors to take into consideration in order to maximize energy efficiency are directly related to the appliances you use.

Here's a few tips:

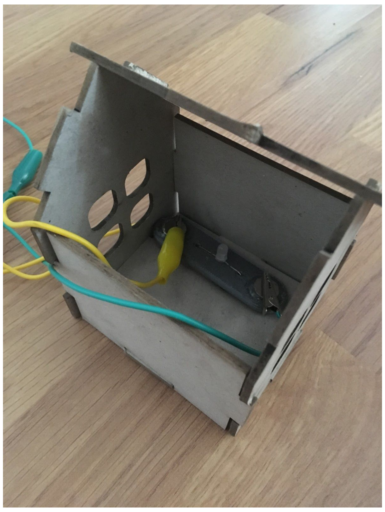
- use smart appliances, for example lights bulbs that go on at night and automatically turn off during the day
- use smart plugs that can be programmed to turn on and off at specific times, that can be remotely controlled via wi-fi, or that display the energy consumption (ex. in kwh) in real time.

Setting up the miniature smart house

Kids will be using arduino uno boards and snap components to work on the energy efficiency of their miniature house.

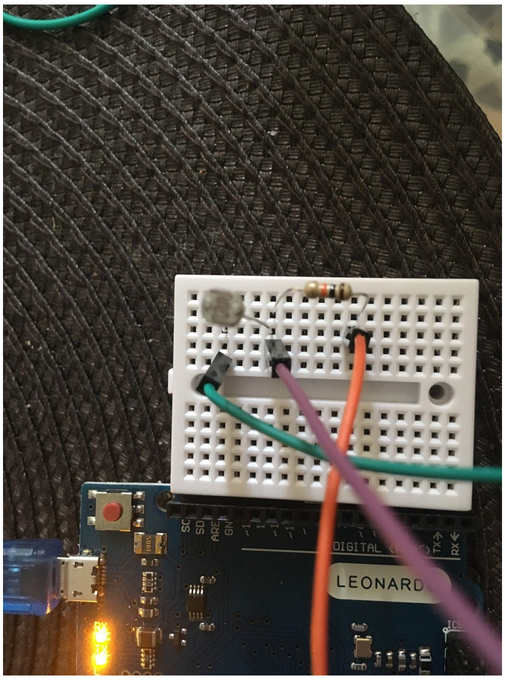
First of all, have the kids set up their miniature houses. They can build one using cardboard, or you can laser cut them in advance, using for example a 3mm thick MDF board. Here's the design of a miniature house, ready for laser cut.

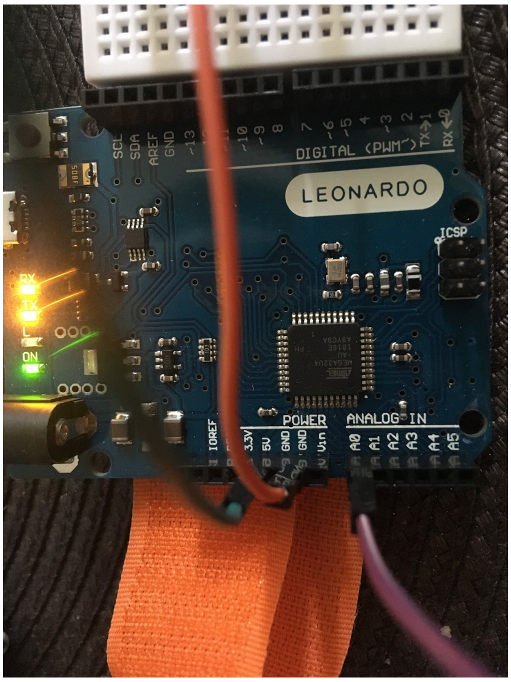
Next, ask the participants to place at least one snap LED component inside the house.



At this stage, they will need to program the LED(s) they have installed in the house via arduino, specifically program them to go **on during the day** (that is if a certain light threshold is attained) and **off at night** (that is to say if the amount of light in the environment falls below a given threshold).

They'll need to use an LDR in connection with the LEDs. The LDR can be mounted on a snap support, or simply used as it is.

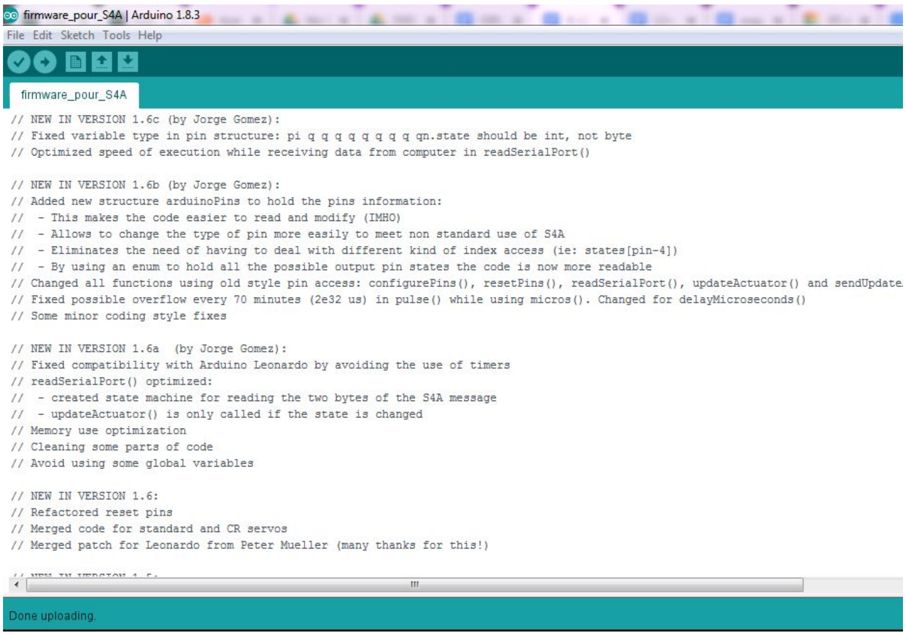




Instruct the kids that they need to Use Scratch for Arduino to achieve this programming part.

First of all, we need to prepare the arduino board. To do so, you need to upload the S4A firmware onto the board, via arduino IDE.

Launch Arduino IDE and open this code.

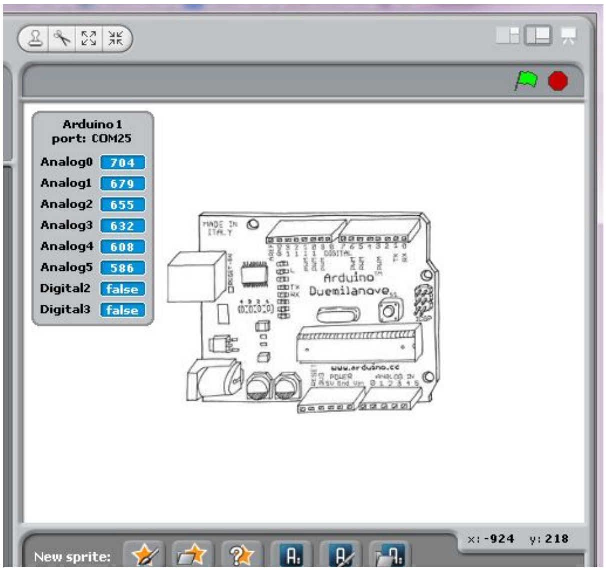

firmware_pour_S4A | Arduino 1.8.3
File Edit Sketch Tools Help
firmware_pour_S4A

```
// NEW IN VERSION 1.6c (by Jorge Gomez):
// Fixed variable type in pin structure: pi q q q q q q q q qn.state should be int, not byte
// Optimized speed of execution while receiving data from computer in readSerialPort()

// NEW IN VERSION 1.6b (by Jorge Gomez):
// Added new structure arduinoPins to hold the pins information:
//  - This makes the code easier to read and modify (IMHO)
//  - Allows to change the type of pin more easily to meet non standard use of S4A
//  - Eliminates the need of having to deal with different kind of index access (ie: states[pin-4])
//  - By using an enum to hold all the possible output pin states the code is now more readable
// Changed all functions using old style pin access: configurePins(), resetPins(), readSerialPort(), updateActuator() and sendUpdate
// Fixed possible overflow every 70 minutes (2e32 us) in pulse() while using micros(). Changed for delayMicroseconds()
// Some minor coding style fixes

// NEW IN VERSION 1.6a  (by Jorge Gomez):
// Fixed compatibility with Arduino Leonardo by avoiding the use of timers
// readSerialPort() optimized:
//  - created state machine for reading the two bytes of the S4A message
//  - updateActuator() is only called if the state is changed
// Memory use optimization
// Cleaning some parts of code
// Avoid using some global variables

// NEW IN VERSION 1.6:
// Refactored reset pins
// Merged code for standard and CR servos
// Merged patch for Leonardo from Peter Mueller (many thanks for this!)
```

```
Done uploading.
```

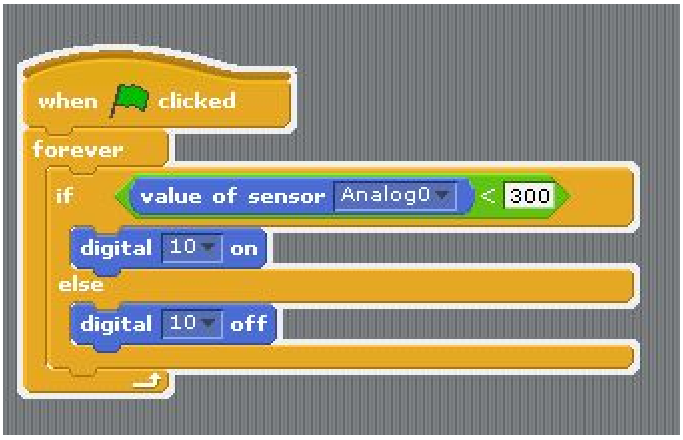
Upload the code onto the board.

Next, open Scratch for arduino and wait until the software recognizes the board.


Arduino1
port: COM25

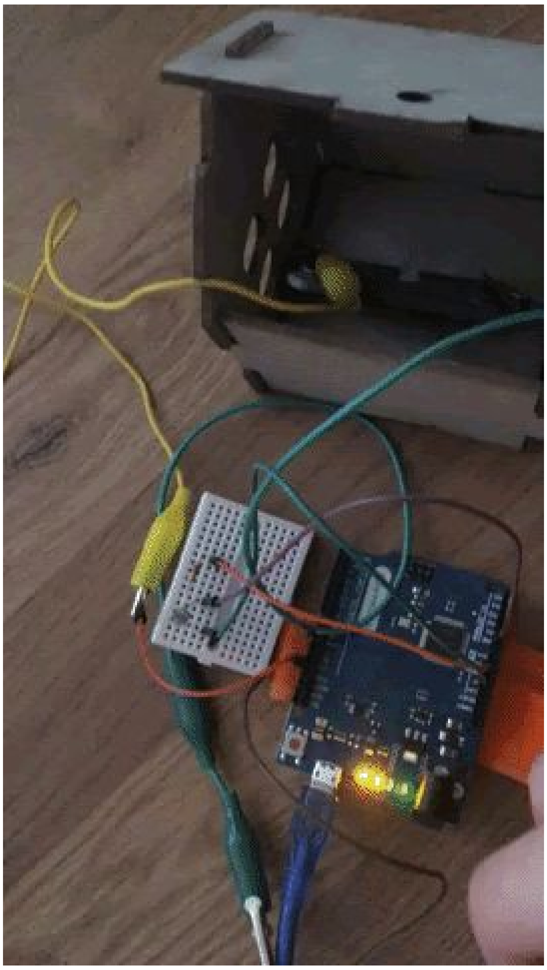
Analog0	704
Analog1	679
Analog2	655
Analog3	632
Analog4	608
Analog5	586
Digital2	false
Digital3	false

Now program Scratch for Arduino to obtain something like this:

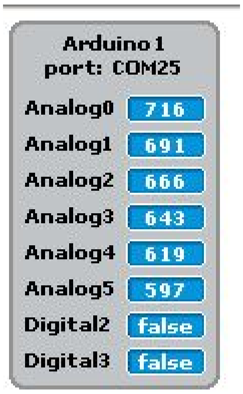


```
when [flag] clicked
forever
  if value of sensor Analog0 < 300
    digital 10 on
  else
    digital 10 off
```

The Led inside the miniature house should go on if the LDR records an amount lower than 300. Otherwise it should stay off.



n.b. you can visualize in real time the amount of light detected by the LDR just by looking at analog 0 under this dialogue window



Arduino 1
port: COM25

Analog0	716
Analog1	691
Analog2	666
Analog3	643
Analog4	619
Analog5	597
Digital2	false
Digital3	false

At this stage, kids can add extra appliances in their house, for example a miniature Aircon device. The main support for the miniature Aircon device can be a vibrating motor snap or an LED snap. Have them place the device inside the miniature house. Next, instruct them to use a temperature /humidity sensor to program the device so that the aircon goes on whenever the temperature rises above, say 28 degrees, or the humidity exceeds say 70%.

Home automation with raspberry pi - device tutorial

The following Instructables presents a "Smart Home device" developed under the scope of the EU project Digital Environmental Education. The original purpose of this device, however, is not only to develop a sort of remotely control of devices; the project has an educational goal that is to sensitize teenagers towards "Green Thinking", teaching them tips to save energy.

Introduction

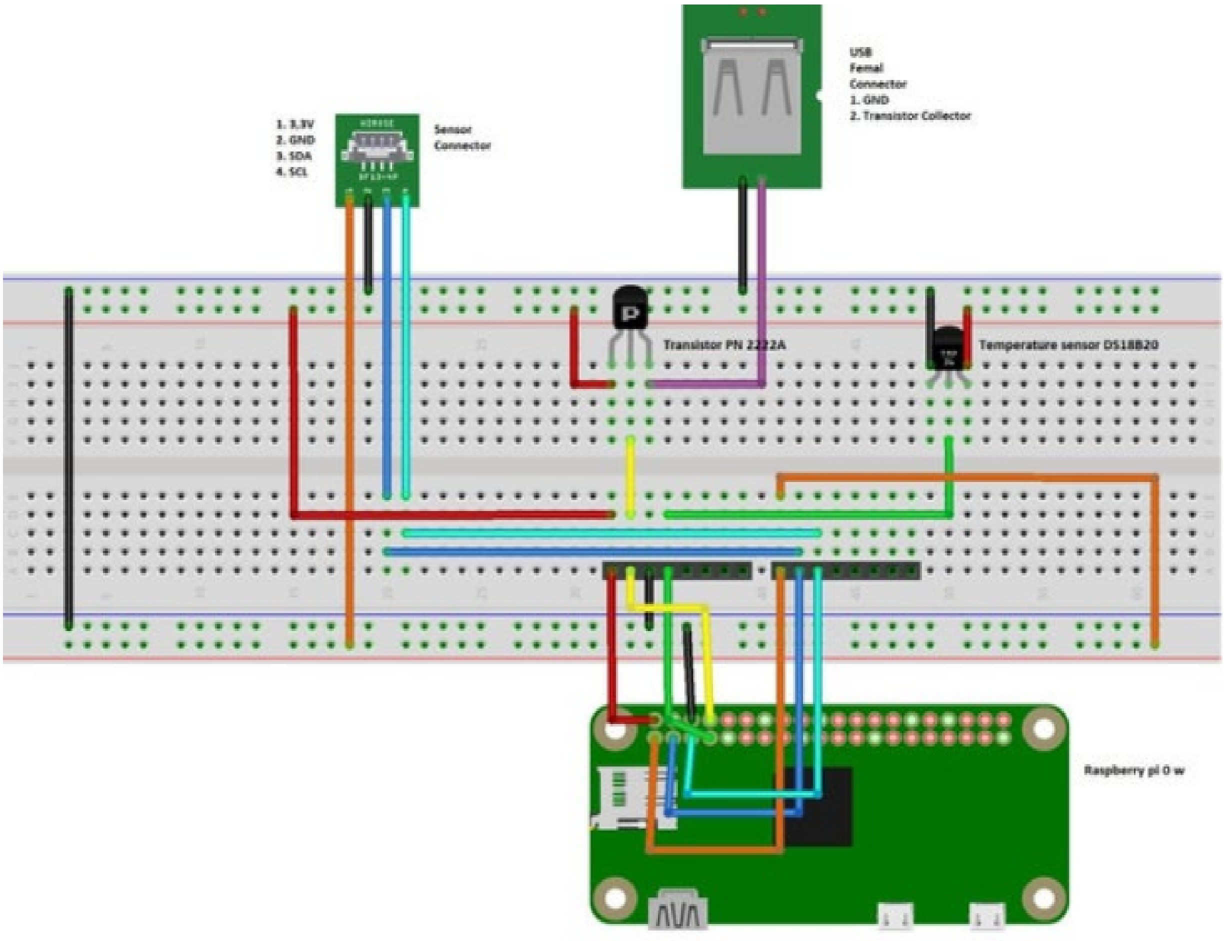
In the following we describe how to control a device receiving notifications about its status. In particular, we show two cases

First-> A led lamp coupled with a luminosity sensor: when the light is switched on and the luminosity in the environment does not change, the user gets a "green notification" with the request to turn the lamp ou.

Second-> A fan coupled with a temperature sensor: when the fan is switched on the temperature does not change, the user gets a "green notification" with the request to turn the Fan

The control and the notification service has been developed using BLYNK, a well known smartphone application. What you need:

1. Raspberry Pi 0 w (or another Raspberry Pi device with WiFi Connection),
2. SD card (minimum size 8GB) with an image of Raspbian,
3. Nodered (application for Raspbian, included in the full installation),
4. A pin strip compatible with Raspberry,
5. A power supply (5V 2A possibly),
6. A solder (with solder past, obviously),
7. A transistor (we used PN2222A),
8. A temperature sensor (we used DS18B20),
9. An USB female connector,
10. A light sensor (we used ADPS9301),
11. An usb powered lamp,
12. Various jumpers (MM, FF, MF depends on the pins you use),
13. A 3d printed enclosure case,
14. ... (patience) ...
15. and, ofc a smartphone with BLYNK



Soldering

The first step towards the assembly of the complete device is to have well soldered pins. Problems in the soldering lead to sneaky failures very hard to identify. Do not underestimate the importance of this step.

Choose accurately the type of pin you solder on the board. We soldered male pins on the board, so we will use female jumpers. You are free to solder female connectors, as well you use male jumpers.

In the attached photos, you find the soldered pins on the Raspberry Pi 0 w and an example of light sensor, not soldered yet.

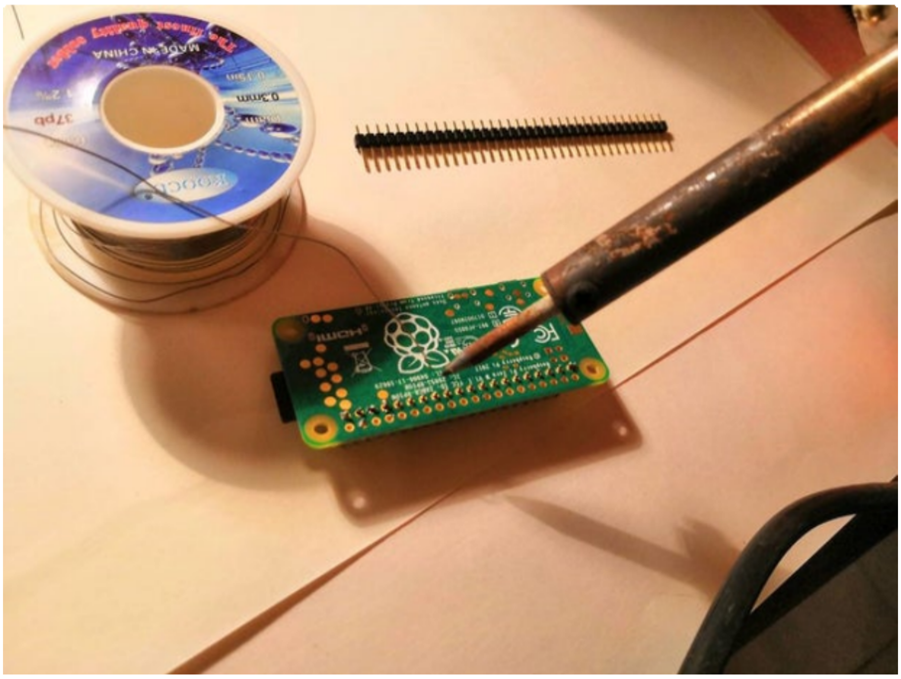
The solderings have to be done according to the schematic presented in figure 3. In the picture, a dummy i2c Light Sensor has been used. Connections with the Raspberry pi 0 w have been done using as reference the website pinout.xyz

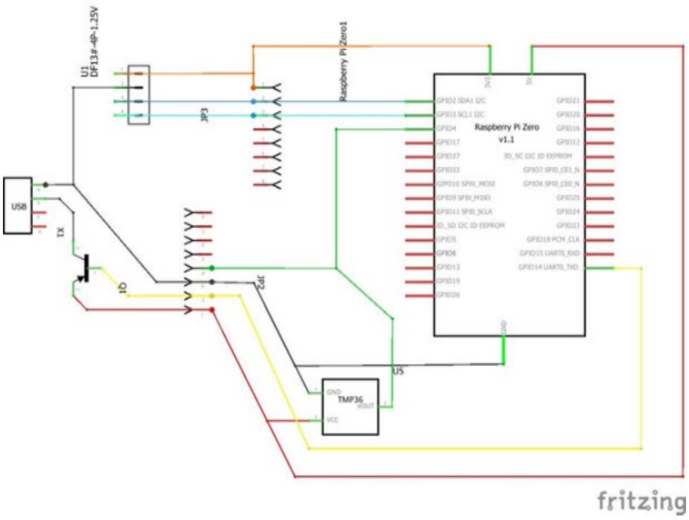
(In picture 4 you find the schematic with a generic connector for a generic sensor that works with i2c protocol. Note that different sensors may have a different order in the pins, so check your sensor before you solder your pcb).

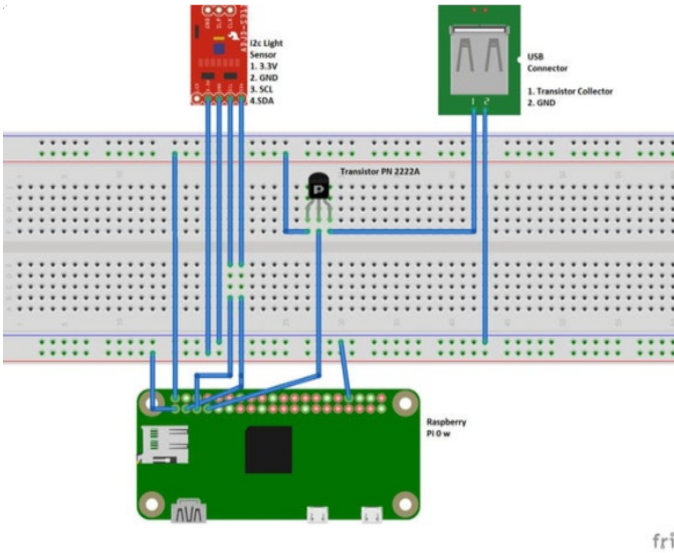
The following videos could be helpful.

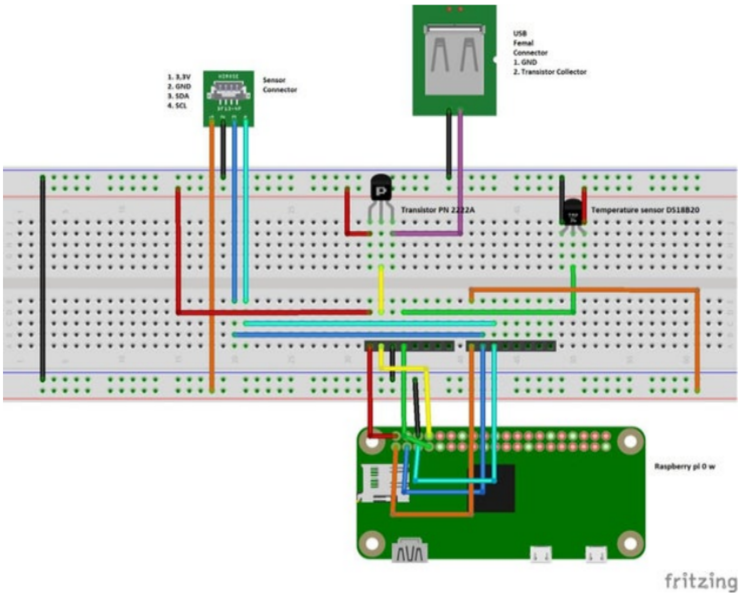
//www.youtube.com/embed/UEqjpMs15jo

An USB socket is great in all cases similar to the this one. It allows to use several devices and it makes highly reusable all systems described in this tutorial. You find a video attached showing how to solder a usb socket.









Raspberry Software Side

Be sure your sd card has enough space for the (quite large, actually) Raspbian Image; 8 gb should be ok. We recommend to install the version that includes the recommended software. In this way, there souldn't be problems with Node.js and NodeRed.

Set up your device, connecting it to a screen and to an input device.

Ready? Connect the power supply - turn on your Raspberry pi 0 w- , log into the system and connect to a wifi network. Update your software at the first run, not to have problem in the future.

(Optional step: you could set up a SSH connection and working from another terminal. This could let you save a lot of time since the management of the inputs result into high computational burden for the board. As other alternative, you can use VNC.)

From the Raspberry software side, we just need to run some commands on the terminal:

```
>> sudo apt-get install nmp
```

```
>> npm install --global node-red-contrib-blynk-ws
```

(this let you download and install the libraries for the communication between Raspberry and the BLINK app)

```
>> sudo systemctl enable nodered.service
```

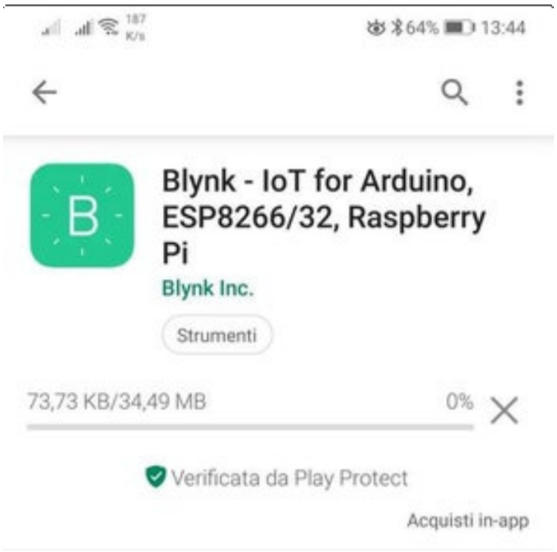
(this adds nodered to the startup services. This is very useful: the process starts when the device is turned on)

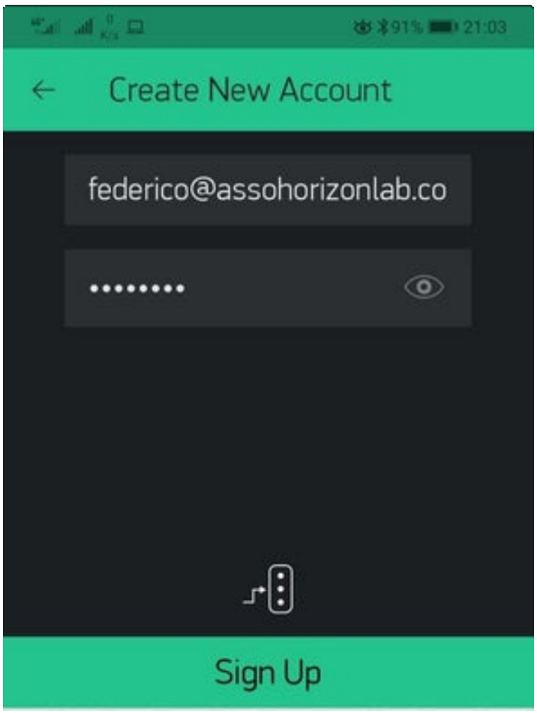
```
>> node-red start
```

Blynk on Your Smartphone

Download Blynk on your smartphone, run it and create your account (see images 1-3).

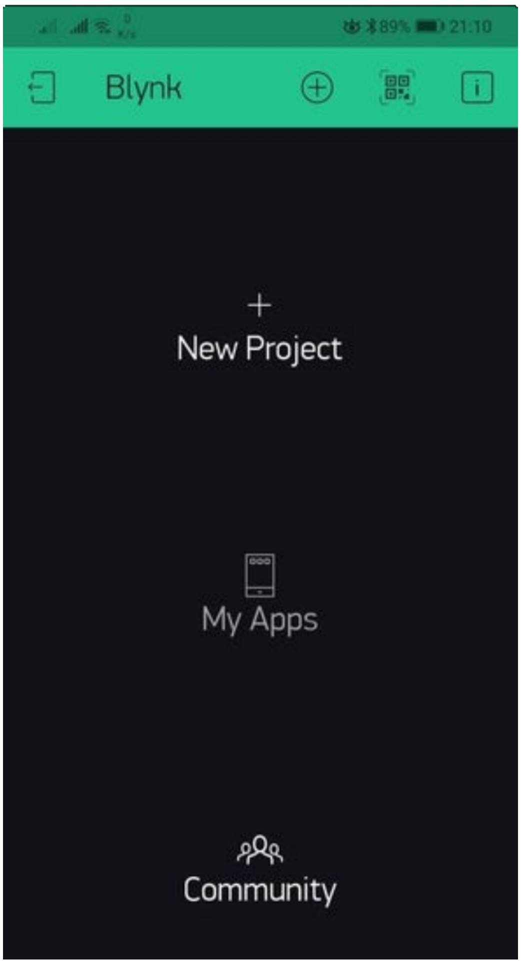
Create a new project selecting as hardware Raspberry Pi 2 and connection type WiFi (pictures 4 & 5). Once the project is created you will receive a mail containing the authorization token for your project (pics 6 & 7). On your smartphone you have a blank blynk project (pic 8). Note that BLYNK ouers you some free features for the project. By now, do not add anything! Adding useless stuff might affect your chance to create a free project. Of course, if you re interested into managing some bigger applications, you can pay to have unlimited access.

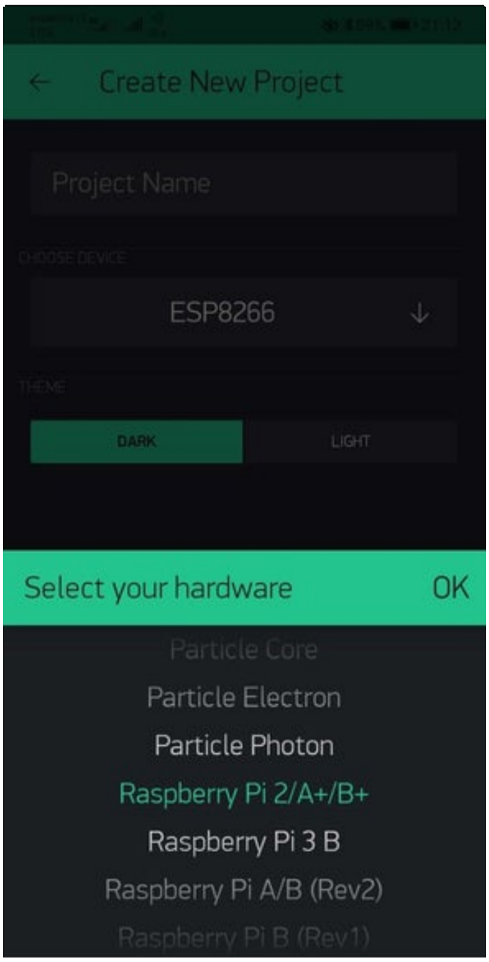

64% 13:44
Blynk - IoT for Arduino, ESP8266/32, Raspberry Pi
Blynk Inc.
Strumenti
73,73 KB/34,49 MB
0%
Verificata da Play Protect
Acquisti in-app



federico@assohorizonlab.co

•••••••••




Create New Project

Project Name

CHOOSE DEVICE

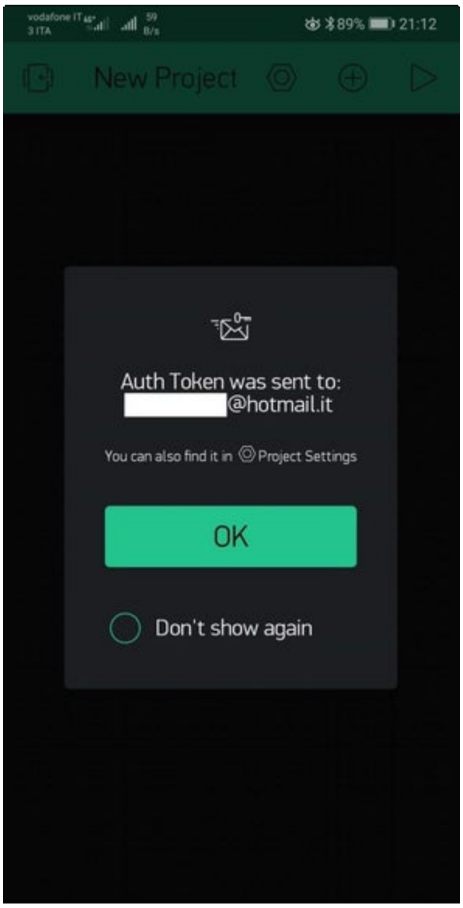
ESP8266

THEME

DARK LIGHT

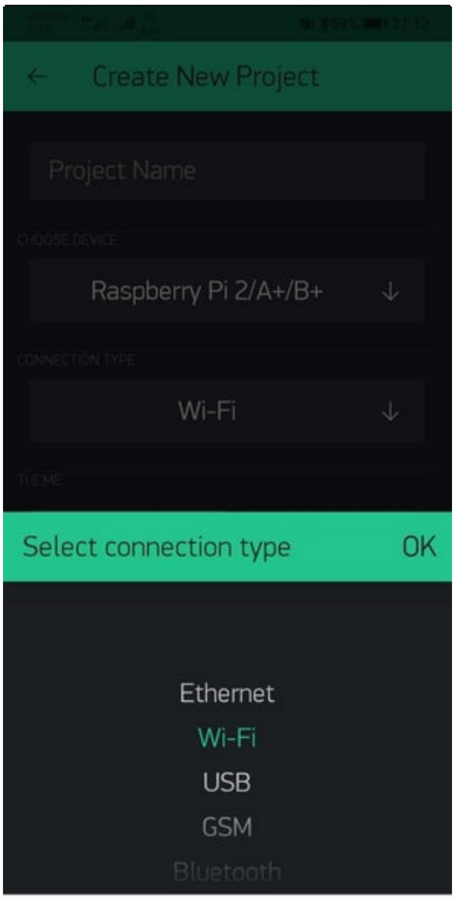
Select your hardware OK

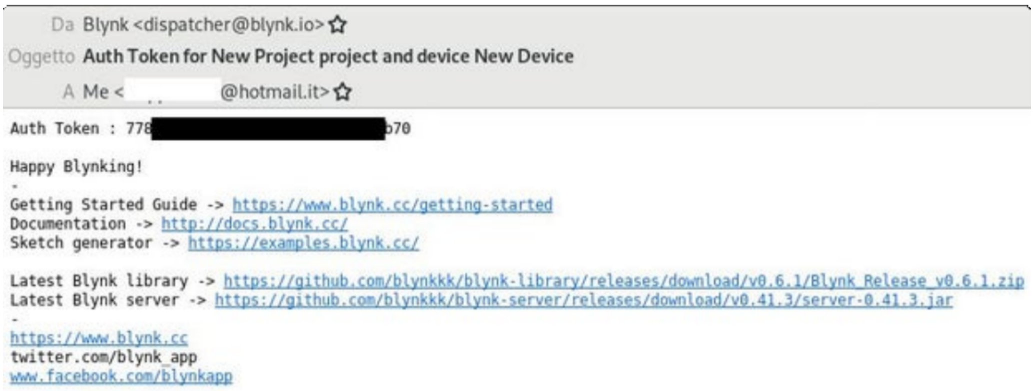
Particle Core
Particle Electron
Particle Photon
Raspberry Pi 2/A+/B+
Raspberry Pi 3 B
Raspberry Pi A/B (Rev2)
Raspberry Pi B (Rev1)



Auth Token was sent to:
[redacted]@hotmail.it

You can also find it in Project Settings



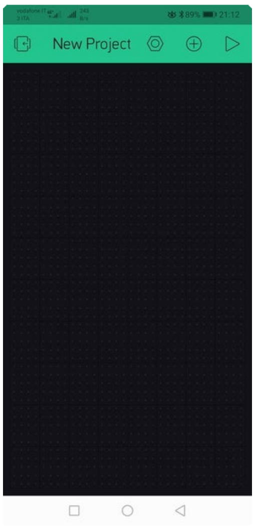


Da Blynk <dispatcher@blynk.io>

Oggetto Auth Token for New Project project and device New Device

A Me <[redacted]@hotmail.it>

```
Auth Token : 778[redacted]b70

Happy Blynking!
-
Getting Started Guide -> https://www.blynk.cc/getting-started
Documentation -> http://docs.blynk.cc/
Sketch generator -> https://examples.blynk.cc/

Latest Blynk library -> https://github.com/blynkkk/blynk-library/releases/download/v0.6.1/Blynk_Release_v0.6.1.zip
Latest Blynk server -> https://github.com/blynkkk/blynk-server/releases/download/v0.41.3/server-0.41.3.jar
-
https://www.blynk.cc
twitter.com/blynk_app
www.facebook.com/blynkapp
```



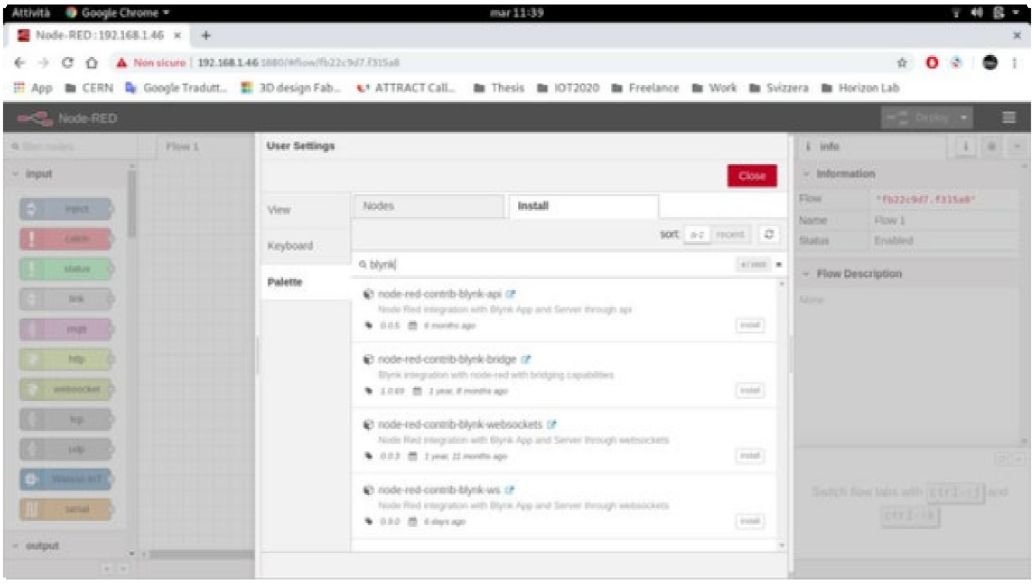
Node Red Console Setup

It's time to access Node-Red! You can do it via a common web browser (shown in the attached image).

Note that usually Node-Red is provided with Raspbian, in case your OS has not nodered installed, you can easily follow this tutorial. https://nodered.org/docs/getting-started/local If you are working locally (same subnet) on your Raspberry, you can access to Node-Red console simply typing

127.0.0.1:1880 in your address bar.
(If you set up a SSH connection, you can connect from the terminal changing part of the address with the ip address of your device)

Once you access Node-Red, click on the three lines in the upper right corner and then on Manage Palette. Go to the tab install and search for "blynk". Install the palette "node-red-contrib-blynk-ws". For the i2c communication, install the palette "node-red-contrib- i2c", while for the ui "node-red-dashboard".



User Settings

View | Keyboard | Palette

Nodes | Install

sort: a-z recent

blynk

node-red-contrib-blynk-api
Node Red integration with Blynk App and Server through api
0.0.5 · 6 months ago · install

node-red-contrib-blynk-bridge
Blynk integration with node-red with bridging capabilities
1.0.49 · 1 year, 8 months ago · install

node-red-contrib-blynk-websockets
Node Red integration with Blynk App and Server through websockets
0.0.3 · 1 year, 11 months ago · install

node-red-contrib-blynk-ws
Node Red integration with Blynk App and Server through websockets
0.9.0 · 6 days ago · install

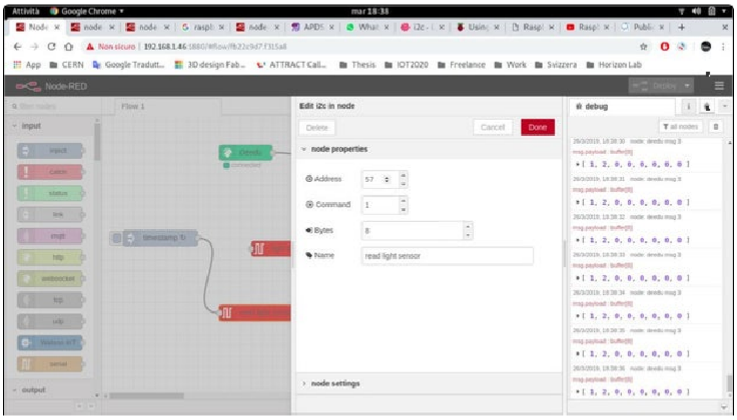
NodeRed Flow

Once the palettes are installed, you can import the code you find attached to this step to create the flow. Just click on menu (top right) -> import -> from clipboard. Then, in the following window, paste the code copied from the file attached here.

Now, it is time to insert your personal data. In particular, you must insert the authorization token in the blocks "DEEDU1" ad "DEEDU2", changing the parameter "InsertYourTokenHere".

When done, click on deploy.

In case of problems, you could download an iso image here. Be careful, because it needs to be customized putting the blynk token generated on your smartphone.



Edit i2c in node

node properties

Address: 57
Command: 1
Bytes: 8
Name: read light sensor

node settings

```
msg.payload : buffer[8]
▸[ 1, 2, 0, 0, 0, 0, 0, 0 ]
```

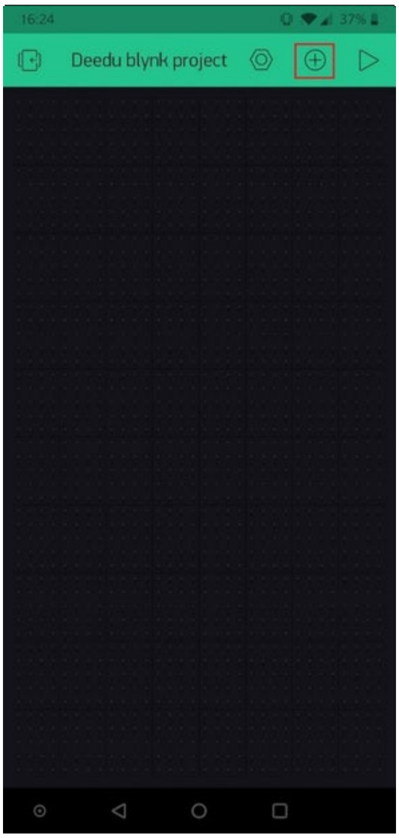
Coming Back to BLYNK

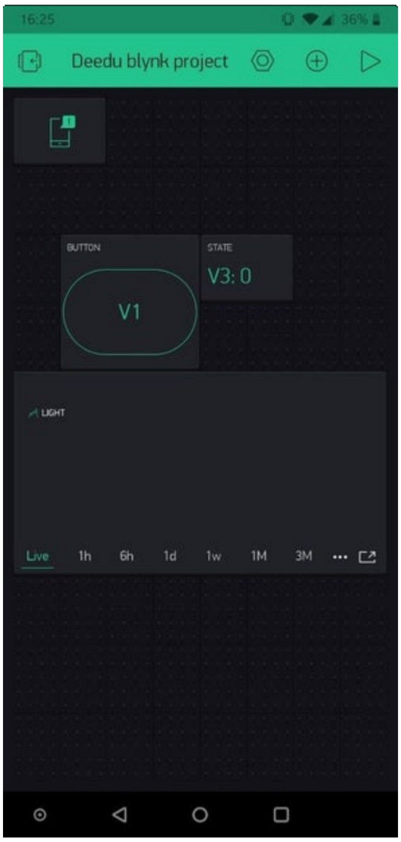
Now it is time to create the BLYNK project. We previously left the black project, as in Figure 1. Tap on '+' and select from the list the objects to re-create the project shown in Figure 2. You will need a Button, a Notification, a Chart and a Display tiles.

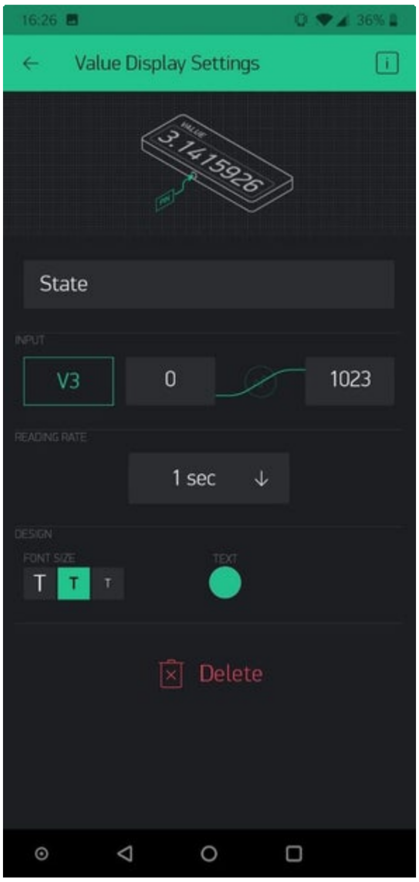
Button must be set as shown in Figure 3, Display as in Figure 4 and Notifications as in Figure 5. To set up the Chart, see the order shown in figure 6: tapping on the option tile, you need to set the same parameters shown in figure 7.

When finished, open the projects settings and tap on Email all.

That's it! By touching the button, your Raspberry should turn the light on and, in case the luminosity does not change too much, you will get your notification.







16:26 36%

Value Display Settings

State

INPUT

V3 0 1023

READING RATE

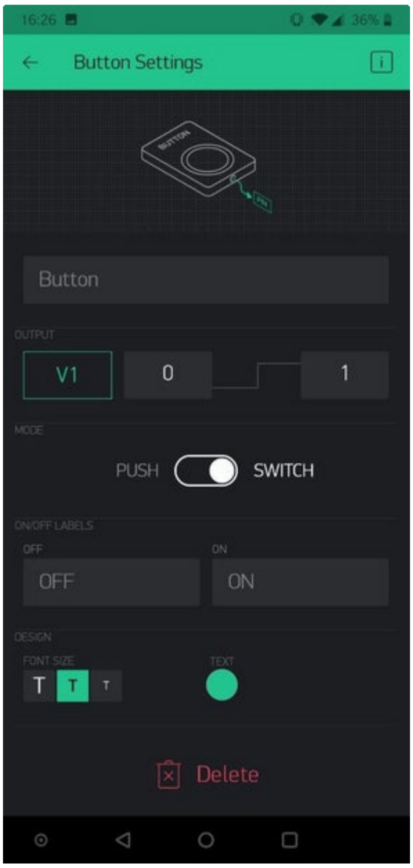
1 sec

DESIGN

FONT SIZE T T T

TEXT

Delete



16:26 36%

Button Settings

Button

OUTPUT

V1 0 1

MODE

PUSH SWITCH

ON/OFF LABELS

OFF: OFF

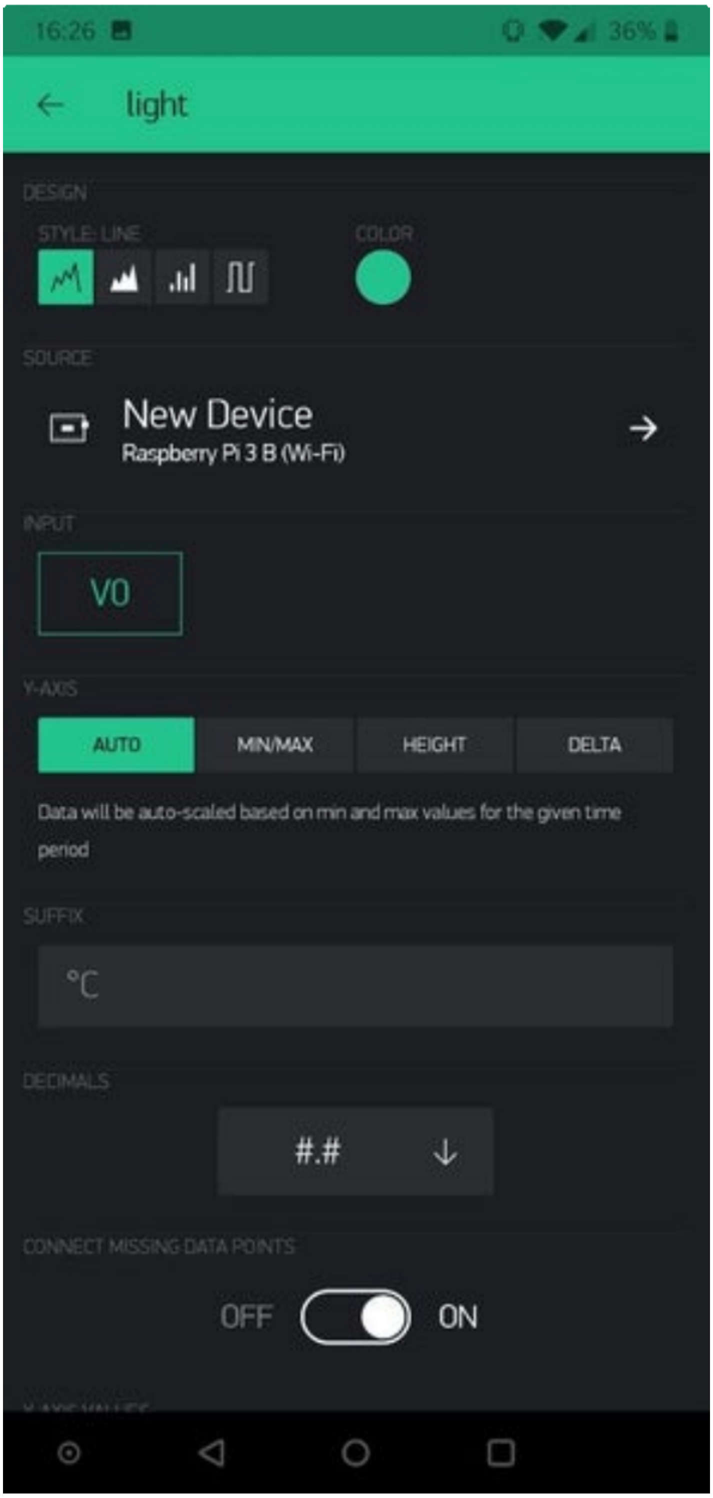
ON: ON

DESIGN

FONT SIZE T T T

TEXT

Delete



16:26 36%

light

DESIGN

STYLE: LINE

COLOR

SOURCE

New Device

Raspberry Pi 3 B (Wi-Fi)

INPUT

V0

Y-AXIS

AUTO MIN/MAX HEIGHT DELTA

Data will be auto-scaled based on min and max values for the given time period

SUFFIX

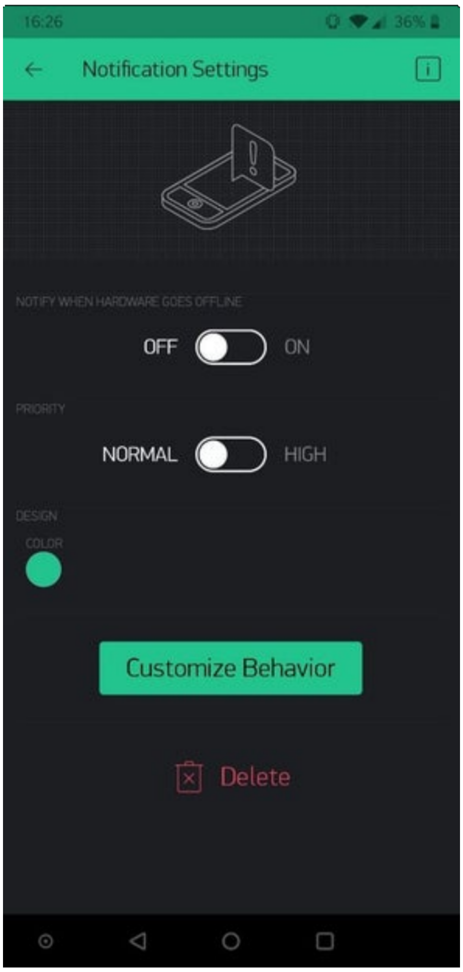
°C

DECIMALS

#.#

CONNECT MISSING DATA POINTS

OFF ON



16:26 36%

Notification Settings

NOTIFY WHEN HARDWARE GOES OFFLINE

OFF ON

PRIORITY

NORMAL HIGH

DESIGN

COLOR

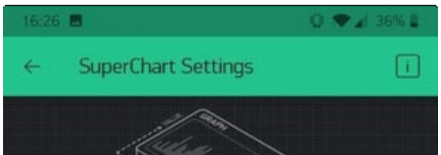
Customize Behavior

Delete

Final Assembly

A 3D printable enclosure case is designed and showed above. You find it attached as .stl file. You can easily print the file using a common 3D printer with standard settings.

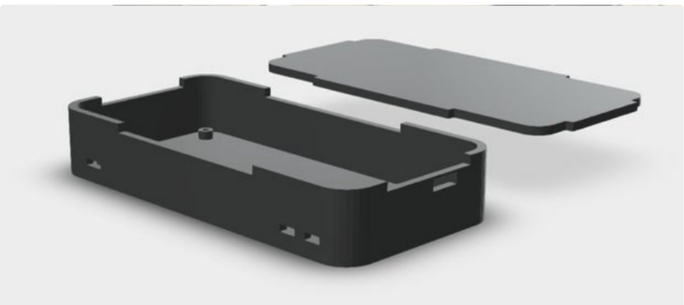
https://www.thingiverse.com/thing:4062244



16:26 36%

SuperChart Settings

In the attached images you it is possible have a view of what the device should look like.



Home automation with raspberry pi - Empty room activity

Introduction

The purpose of this activity is to increase the user's sensitivity to energy consumption for temperature control.
This activity is designed for children over the age of 12, who are able to read and understand the indicative temperature numbers and the information that appears on the displays as well as having a perception of orientation on satellite maps.
The use of electricity in the home is called "domestic consumption".
Energy efficiency is the ability to make intelligent use of resources, minimizing waste as much as possible. For example, a room that is not frequented by anyone may not need air conditioning or artificial light.
So how can we achieve energy efficiency in our living environment?
Many small steps can be taken to get better use of consumption.
In this activity we want to teach the user how to set a geolocation-based control on the smartphone to turn off any users that are mistakenly left on in the home.
The user will trace a perimeter of coordinates within which, if not found, the smartphone would automatically send a signal to the Deedu device to switch off any utilities left on.
At the end of this activity, the user is expected to have increased personal sensitivity in the use he makes of domestic resources.

Creation of the environment

What you need?

1 - A remote control device #Deedu;

2 - Environments Nodered and Blynk;
3 - A box whose purpose is to abstract the concept of the home environment.

How to create the environment

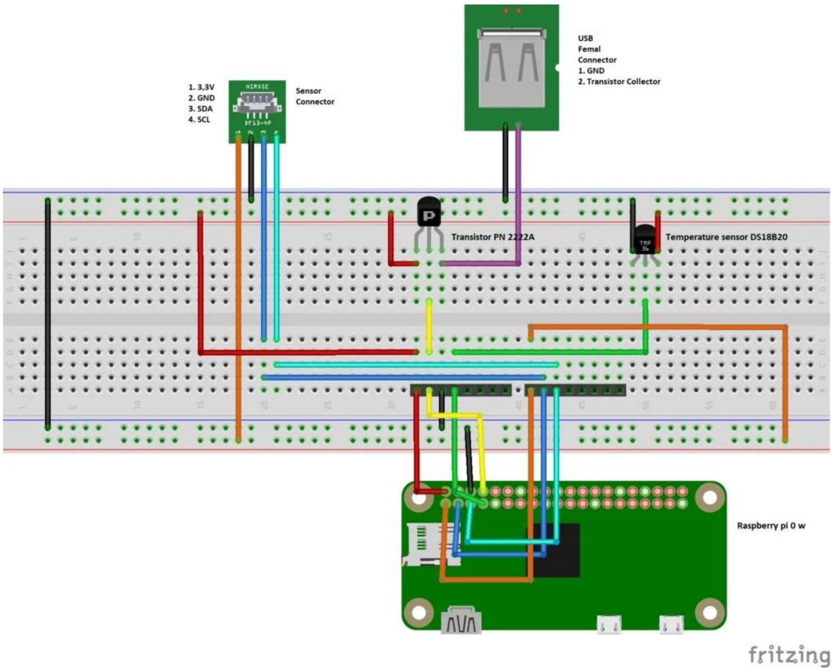
Here we have to describe how the miniature house is created, the environment to be simulated and the things you need to do it:

How to build the box?

- We take a box of shoes that we no longer use;
- With the help of a pair of scissors with a rounded tip, let's cut one of the shorter sides of the box. From here we could observe inside the box itself what will be simulated;
- Let's cut the other minor side in the same way. From this we will insert the small fan;
- By inserting the device inside the box, we created our miniature room and we are ready for the experiment.

How to build the device?

It is necessary that the pcb is well soldered so that the circuits are resistant and that they do not disconnect when you are going to put everything in the box. once the pcb is finished, it will have to be positioned over the raspberry for it to be put into operation.



To check if everything works well, switch the raspberry on and insert the pcb above it. With the help of a tester you can check that all the connections have been made well, just check that the voltage reaches all the desired points. A more thorough test can then be rerun when the whole device is ready.

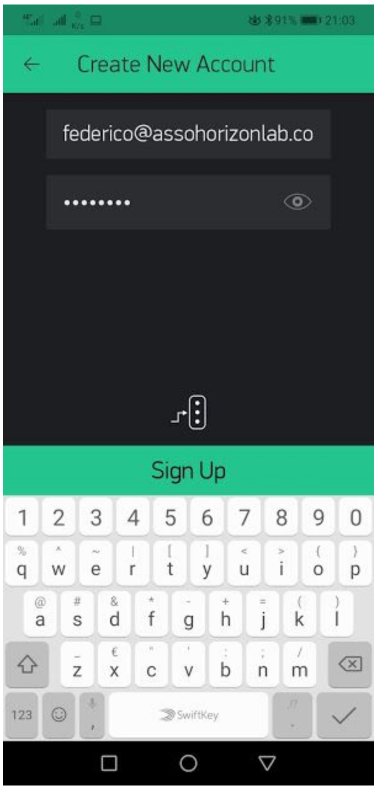
The attached diagram also indicates the temperature sensor as the activity in question is part of a package of pedagogical tools to improve the ecological sensitivity of the user. The temperature sensor is then used in another activity but the creation of a pcb for setting the device is preferable both as a single one. In this way you can reuse the same device on all available activities.

For the construction of the device, consult the guide at the following link
https://www.youtube.com/watch?v=UEqjpMs15jo

To close everything in a wrapper, it may be useful to 3D print the suitable box whose source can be downloaded at the following link.
https://www.thingiverse.com/thing:4062244

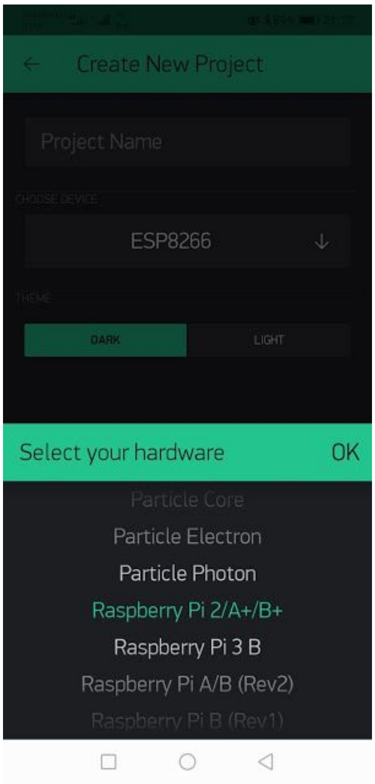
How to prepare the Blynk app?

To set up the software system via Blynk, follow the guide:
https://www.instructables.com/id/Digital-Environmental-Education-Domotics/



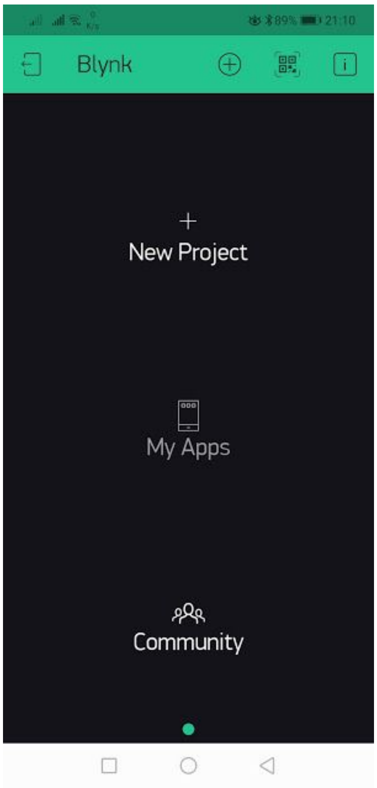
federico@assohorizonlab.co

••••••••



ESP8266

Particle Core
Particle Electron
Particle Photon
Raspberry Pi 2/A+/B+
Raspberry Pi 3 B
Raspberry Pi A/B (Rev2)
Raspberry Pi B (Rev1)



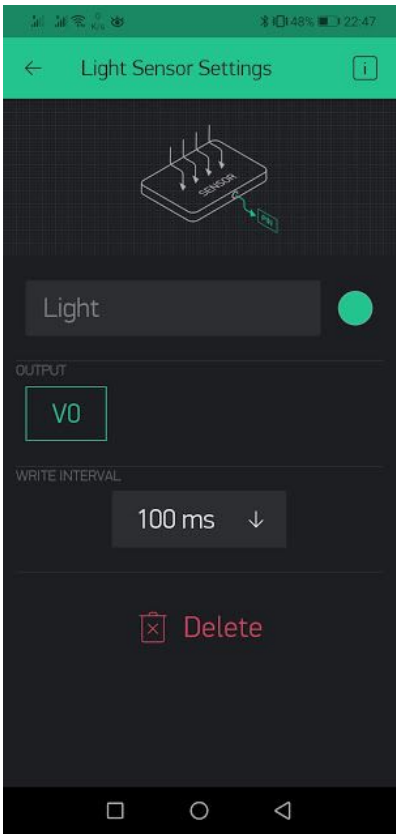
Once the application has been downloaded from the store, it is necessary to register a Blynk account by creating a profile, after which a new project must be created and a token generated. The token is an element that acts as a shared key, that is, it is a word (a token) that uniquely identifies the project and allows you to connect the device to the control terminal.
The app can be run on Android and Ios tablets and smartphones. This allows a wider audience of people to run and build the remote controller.

The Blynk app allows you to easily customize and change the interface of the remote controller. This guide explains how to make a version of the remote controller but nothing prevents you from adding new features to make it even more useful and nice.

Unlike the other activities, you need to add some extra widgets this time.
We insert the brightness sensor of the mobile phone, the widget for events and the gps sensor. We configure the 3 widgets as in the following images. So that when we move away from the home and leave the perimeter, the user switches off. Or when the brightness increases and exceeds the set threshold, the lamp switches off.

Let’s play!

We let the user enter and leave the traced perimeter and understand that when you are away from the home environment there is no need to leave the households on.
In the same way, we let the user understand that when he is outdoors, it is useless to leave the home lamp on.



Light Sensor Settings

Light

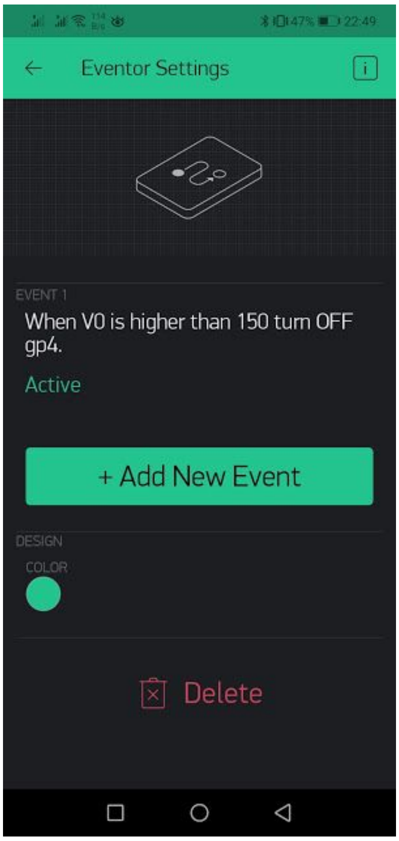
OUTPUT

V0

WRITE INTERVAL

100 ms

Delete



Eventor Settings

EVENT 1

When V0 is higher than 150 turn OFF gp4.

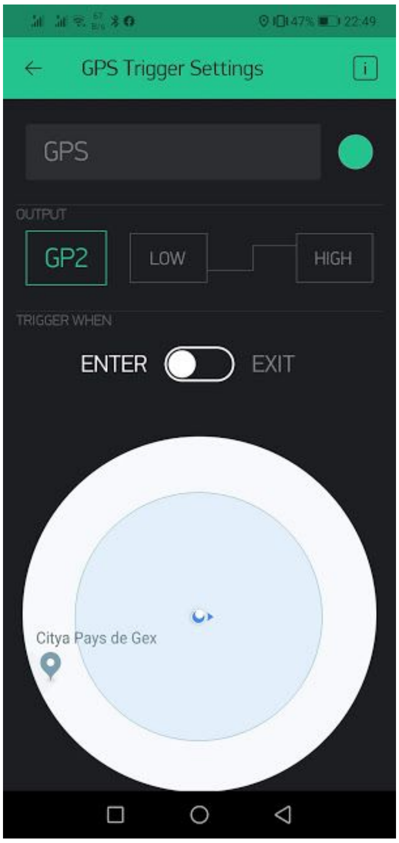
Active

+ Add New Event

DESIGN

COLOR

Delete



GPS Trigger Settings

GPS

OUTPUT

GP2 LOW HIGH

TRIGGER WHEN

ENTER EXIT

Environmental monitoring

A closed environment in which there is nobody is an environment that in the vast majority of cases does not need to be illuminated or cooled. Likewise, it is often completely unnecessary to keep many devices such as televisions, heaters or electronic consoles turned on.
From the Blynk app, we allow the user to familiarize themselves with turning on or off an electrical appliance.

The user must learn about what type of environment he is exercising control over. A bedroom, for example, does not need to be illuminated in the morning if you are at school or at work and therefore it is good to turn off the lights that may have been left on. Likewise, it is easy to guess that a bathroom does not need to be heated when you are away from home.

The user must learn about what type of environment he is exercising control over. A bedroom, for example, does not need to be illuminated in the morning if you are at school or at work and therefore it is good to turn off the lights that may have been left on. In the same way it is easy to guess that a bathroom does not need to be heated when you are away from home.
The user will therefore have to hypothesize multiple application cases:

1. Light on in a bedroom during the day or in the evening
2. Heating of a bathroom day or night
3. Fan in a kitchen day or night

For each case, it is necessary to ask the user to reflect on the need to keep a device controlled through the remote control #deedu on or off.
In this way, the user will make multiple attempts and metabolize the principles and advantages of a conscious use of electricity. In addition, remote control ensures that the user can remotely check whether home users are used correctly.

Conclusions of the activity

At the end of the activity it is thought to have the children elaborate a page of diary, asking them to tell the experience they have carried out by highlighting the strengths and weaknesses of the device they have used and to provide any advice to the creators.
This will serve administrators, as well as creators of the device, on multiple fronts.
It will undoubtedly be useful to creators to work on weak points, and therefore to improve.
On the other hand, it will serve as a sort of database. In fact, this diary page will be kept by the creators in an archive, in order to be always accessible.
Furthermore, if the critical issues emerge, once they are improved, the creators could think of carrying out this activity again.
Therefore, the latter becomes essential for the formation of an archive and to be able to compare the results if the activity is proposed again later.

Results

Sometimes we act wrongly not out of negligence or indifference towards our planet, but because some issues and some important precautions are ignored.
We adults have a moral obligation to inform children, sensitize them, allow them to have all that useful information to be a good citizen and to love their world (and here we do not focus only on the environmental aspect!)
In fact, working with children allows you to promote initiatives that aim to change the wrong lifestyles that we may have due to bad, or completely absent, information.

By carrying out this activity that strongly recalls reality, children understand that this device can really be applied in any home environment, including their own home.

The purpose of the activity is to stimulate the user to make conscious use of energy resources in the home. The abstraction of the home environment by means of the box is aimed at arousing a connection in the user's mind with a real daily case. In this way, the user assimilates the advantages of using digital technology to make more efficient use of domestic resources. The perception of greater efficiency shows the user the sense of a reduction in waste.

Home automation with raspberry pi - thermostat control activity

Introduction

The purpose of this activity is to increase the user’s sensitivity to energy consumption for temperature control.

This activity is designed for children over the age of 9, who are able to read and understand the indicative temperature numbers and the information that appears on the displays.

The use of electricity in the home is called "domestic consumption".

Energy efficiency is the ability to make intelligent use of resources, minimizing waste as much as possible. For example, a room that is not frequented by anyone may not need air conditioning or artificial light. Adapting the ignition or adjusting the intensity of the users of this environment allows for a more intelligent use of resources, therefore a more efficient use in which waste is reduced to a minimum.

The lower the energy loss to achieve a specific purpose, the higher the degree of energy efficiency.

Energy demand is on the rise worldwide. The situation on the energy market is heating up and energy prices are rising.

There is a strong correlation between energy demand and social tensions in the world. Furthermore, there is also a strong correlation between the massive use of mineral resources, pollution, the increase in particular of diseases, desertification and global warming.

Many of the world's most important problems are related to the use of energy. A reduction in energy demand would lead to a confusion of many global problems.

So how can we achieve energy efficiency in our living environment?

Many small steps can be taken to get better use of consumption. The purpose of this activity is to make the user learn the main problems related to the domestic world and learn to adopt all the solutions to prevent the inefficient use of energy.

Creation of the environment

What you need?

- A pair of scissors
- An old shoe box
- Pastels / glue / colored sheets

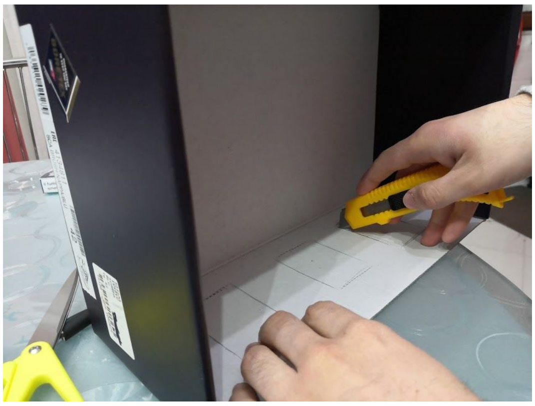
- A small fan

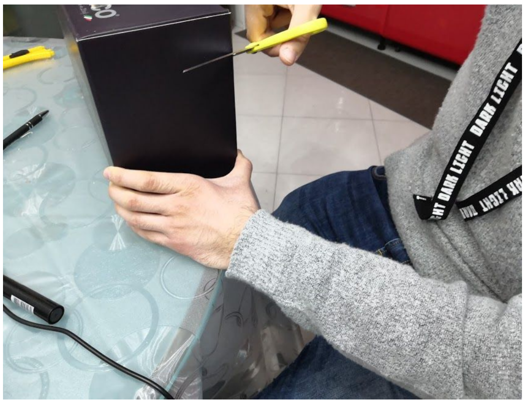
How to create the environment

As in the previous activity, the children have a box that will simulate the home environment. The moderator can leave room for children to decorate their boxes and express their creativity.
Even in this case, children will have to cut the smaller sides: on one side they will be able to look into the box, on the other they can insert a small fan.
The device, inserted in the box, will detect if there is the right temperature.
By accessing the fan, the environment will become colder, although there is no need.
Thanks to the signaling of the device, children will understand that the fan can be turned off.

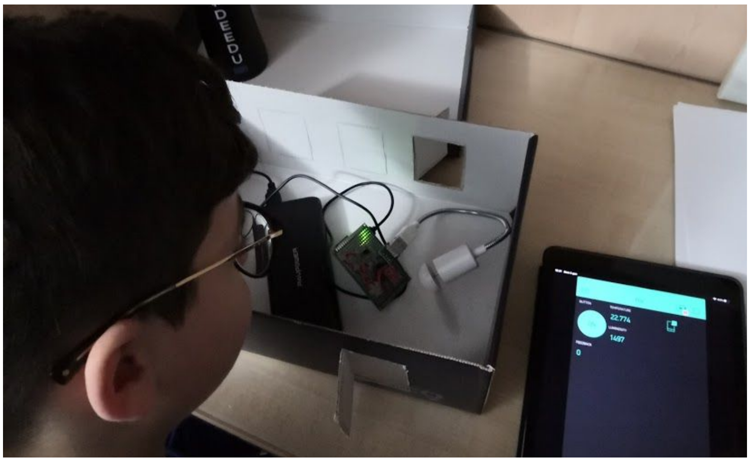
How to build the box?

- We take a box of shoes that we no longer use;
- With the help of a pair of scissors with a rounded tip, let's cut one of the shorter sides of the box. From here we could observe inside the box itself what will be simulated;





- Let's cut the other minor side in the same way. From this we will insert the small fan;



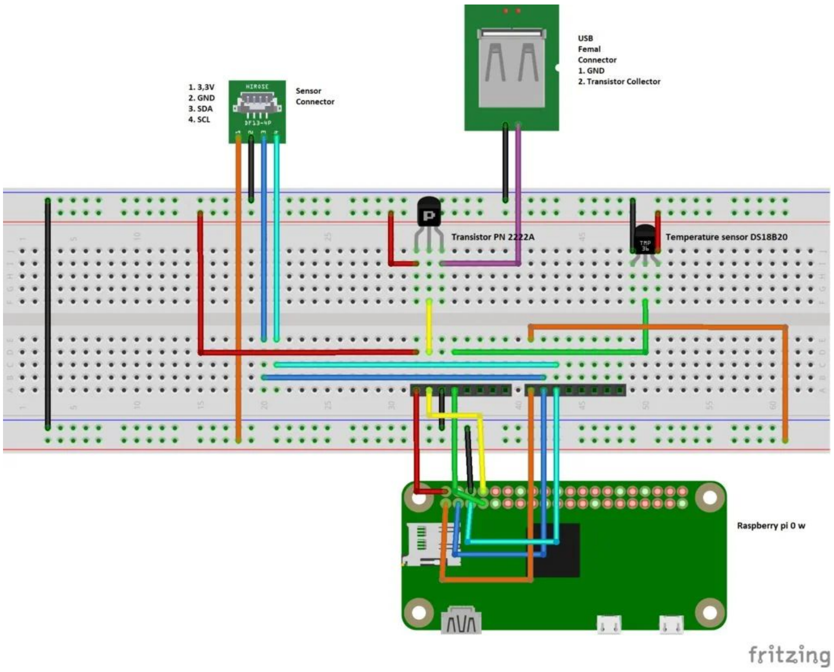
- By inserting the device inside the box, we created our miniature room and we are ready for the experiment.

How to build the device?

For the construction of the device, consult the guide at the following link:

https://www.instructables.com/id/Digital-Environmental-Education-Domotics/

It is necessary that the pcb is well soldered so that the circuits are resistant and that they do not disconnect when you are going to put everything in the box. once the pcb is finished, it will have to be positioned over the raspberry for it to be put into operation.



To check if everything works well, switch the raspberry on and insert the pcb above it. With the help of a tester you can check that all the connections have been made well, just check that the voltage reaches all the desired points. A more thorough test can then be rerun when the whole device is ready.

To close everything in a wrapper, it may be useful to 3D print the suitable box whose source can be downloaded at the following link:

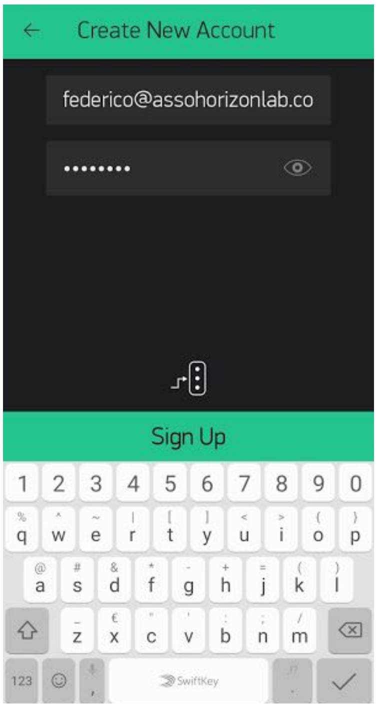
https://www.thingiverse.com/thing:4062244

How to set up the Blynk app?

To set up the software system via Blynk, you need to follow the link guide once again:

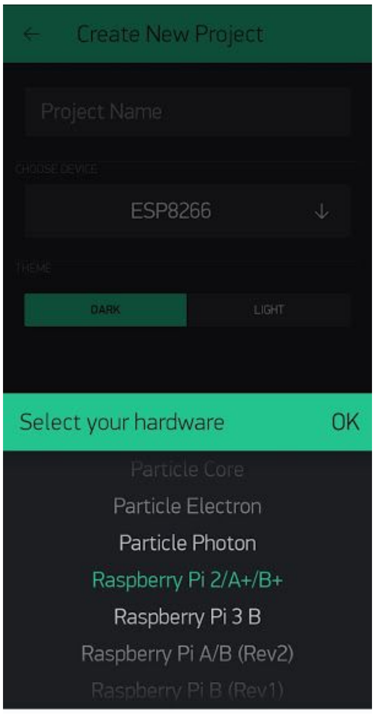
https://www.instructables.com/id/Digital-Environmental-Education-Domotics/

Once the application has been downloaded from the store, it is necessary to register a Blynk account by creating a profile, after which a new project must be created and a token generated. The token is an element that acts as a shared key, that is, it is a word (a token) that uniquely identifies the project and allows you to connect the device to the control terminal.


Create New Account

federico@assohorizonlab.co

••••••••


Create New Project

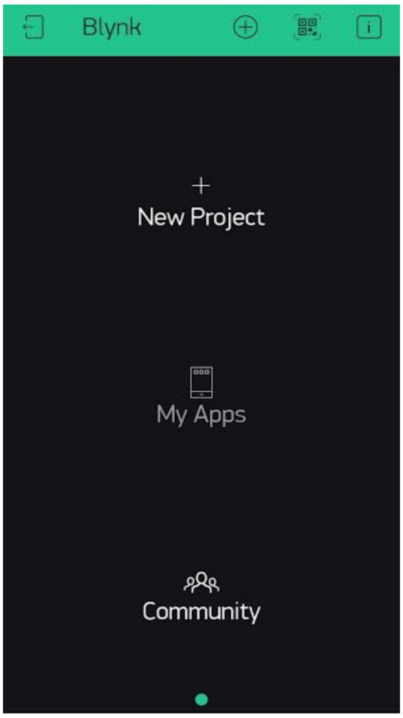
Project Name

CHOOSE DEVICE: ESP8266

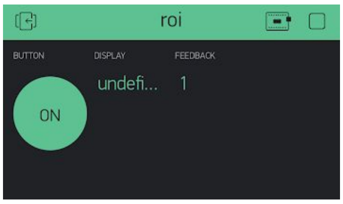
THEME: DARK LIGHT

Select your hardware OK

Particle Core
Particle Electron
Particle Photon
Raspberry Pi 2/A+/B+
Raspberry Pi 3 B
Raspberry Pi A/B (Rev2)
Raspberry Pi B (Rev1)



The app can be run on Android and Ios tablets and smartphones. This allows a wider audience of people to run and realize the remote controller.


roi

BUTTON: ON

DISPLAY: undefi...

FEEDBACK: 1

The Blynk app allows you to easily customize and change the interface of the remote controller. This guide explains how to make a version of the remote controller but nothing prevents you from adding new features to make it even more useful and nice.

Environmental monitoring

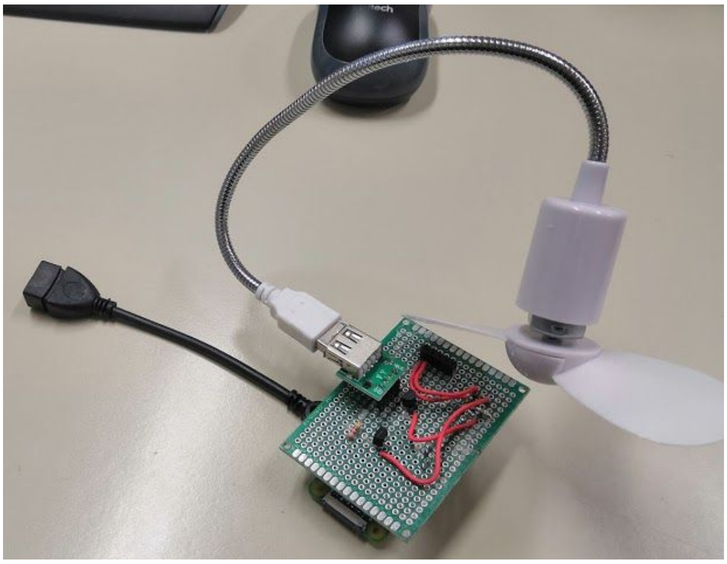
In order to check the conditions of the environment at runtime, just follow the trend of the values that are printed on the screen on the terminal you are using (tablet, smartphone, etc.)

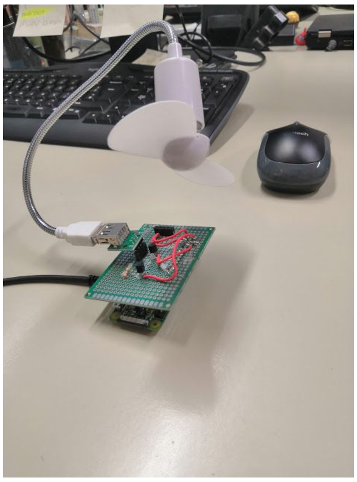
The user must therefore follow the variation of the measurement in order to be able to account for the use he is making of the electric user.

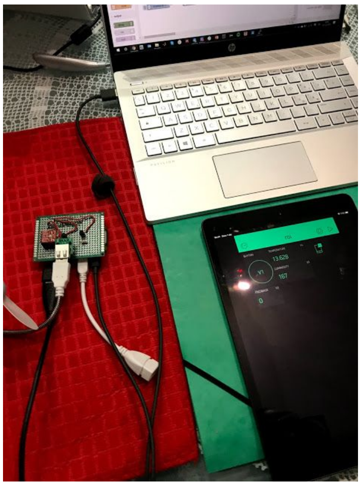
https://www.youtube.com/watch?v=iwO0tae45k8

Description of the environment

Recall that the purpose of the activity is to familiarize the user with the concept of energy saving. The box in which the device is inserted is an abstraction of a room. The connection between the components of the box and a domestic context will be well explained to the user.
We will have to simulate multiple situations. The user must understand the use of an electrical user in completely inadequate and suitable cases.







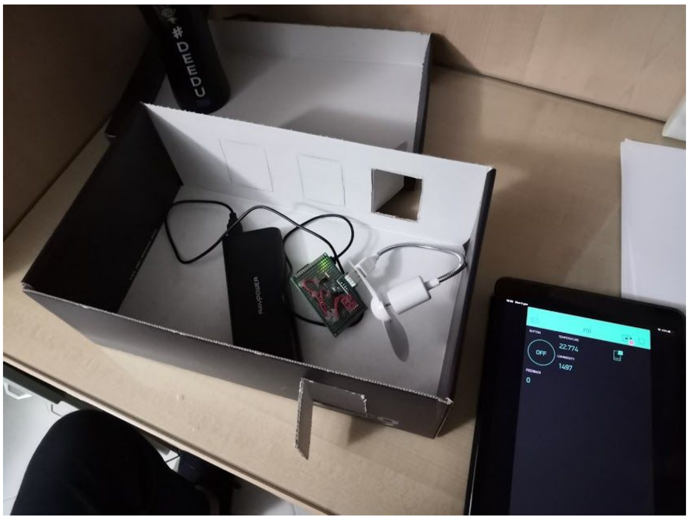
We ask the user to try to understand if the use of a fan is necessary based on the temperature that reads on the display. If the environment is already sufficiently cool, the user will have to learn that a fan (or in any case a cooling device) is inappropriate for the context. It should be turned off.

We often change configuration so that the user becomes familiar with all contexts and learns well the benefits of efficient use of the device.

https://www.youtube.com/watch?v=iwO0tae45k8

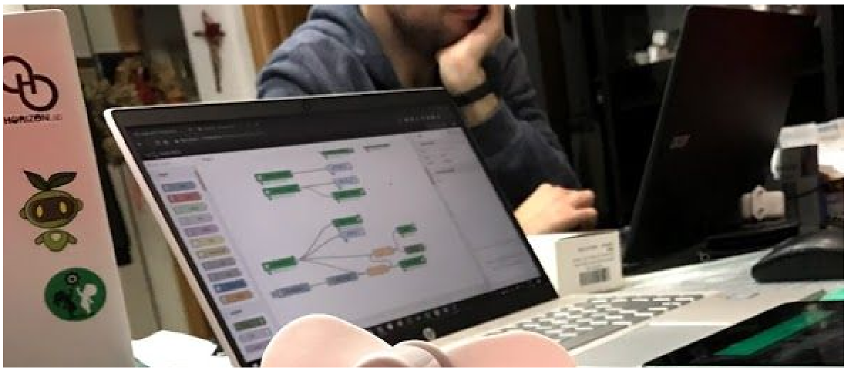
Conclusion of the activity

At the end of the activity it is thought to have the children elaborate a page of diary, asking them to tell the experience they have carried out by highlighting the strengths and weaknesses of the device they have used and to provide any advice to the creators.
This will serve administrators, as well as creators of the device, on multiple fronts.
It will undoubtedly be useful to creators to work on weak points, and therefore to improve.
On the other hand, it will serve as a sort of database. In fact, this diary page will be kept by the creators in an archive, in order to be always accessible.
Furthermore, if the critical issues emerge, once they are improved, the creators could think of carrying out this activity again.
Therefore, the latter becomes essential for the formation of an archive and to be able to compare the results if the activity is proposed again later.



Results

Sometimes we act wrongly not out of love or indifference to our planet, but because some issues and some important precautions are ignored.



As adults, we have a moral obligation to inform children, sensitize them, allow them to have all that useful information to be a good citizen and to love their world (and here we do not focus only on the environmental aspect!)
In fact, working with children allows you to promote initiatives that aim to change the wrong lifestyles that we may have due to bad, or completely absent, information.
At the end of the activity, the children should have refined their awareness of the environment. They are expected to be more attentive, more responsible towards the environment and also in the use of devices and home comforts. By raising children's awareness, the aim is to trigger chain awareness.
If the children have actually introjected these attentions, they will also be more sensitive at the end of the activity. Children who pay more attention to a light or a heated radiator at home or at school will be able to sensitize those around them too (friends, classmates, family members).
So, precisely through play, the aim is to "turn the spotlight" on these issues, to stimulate interest, a sense of responsibility and to arouse a critical perspective of the little ones.
In addition, the proposed game allows you to stimulate creativity and imagination. Cooperative learning and problem solving skills are called into play.

Children should be able to use the device. This will also be useful for sponsoring the device itself. In fact, anything accessible to children is defined as simple to use. For this reason, people who are less familiar with technologies, such as older people, will not be discouraged from using the device but will consider it accessible.
In addition, several studies have shown how the use of technologies by older people improves their cognitive and mnemonic abilities, helping the brain to remain trained.

By carrying out this activity that strongly recalls reality, children understand that this device can really be applied in any home environment, including their own home.

Home automation with raspberry pi - brightness control activity

Introduction

The purpose of this activity is to increase the user's sensitivity to energy consumption for brightness control.
This activity is designed for children over the age of 9, who are able to read and understand the indicative brightness measurements and the information that appears on the displays.

The use of electricity in the home is called "domestic consumption".
Energy efficiency is the ability to make intelligent use of resources, minimizing waste as much as possible. For example, a room that is not frequented by anyone may not need air conditioning or artificial light. Adapting the ignition or adjusting the intensity of the users of this environment allows for a more intelligent use of resources, therefore a more efficient use in which waste is reduced to a minimum.
The lower the energy loss to achieve a specific purpose, the higher the degree of energy efficiency.
Energy demand is on the rise worldwide. The situation on the energy market is heating up and energy prices are rising.
There is a strong correlation between energy demand and social tensions in the world. Furthermore, there is also a strong correlation between the massive use of mineral resources, pollution, the increase in particular of diseases, desertification and global warming.
Many of the world's most important problems are related to the use of energy. A reduction in energy demand would lead to a confusion of many global problems.
So how can we achieve energy efficiency in our living environment?
Many small steps can be taken to get better use of consumption.
The purpose of this activity is to make the user learn the main problems related to the domestic world and learn to adopt all the solutions to prevent the inefficient use of energy.

Creation of the environment

What you need?

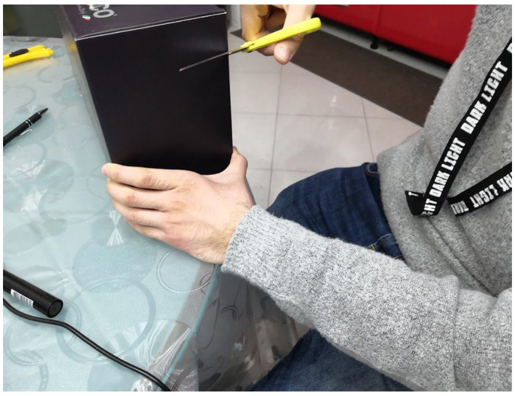
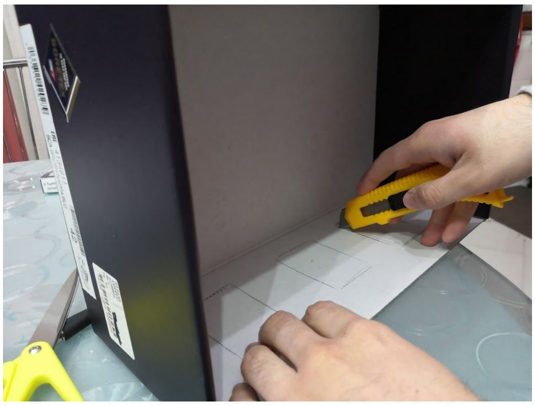
1 - A remote control device #Deedu;
2 - Environments Nodered and Blynk;
3 - A box whose purpose is to abstract the concept of the home environment.

How to create the environment

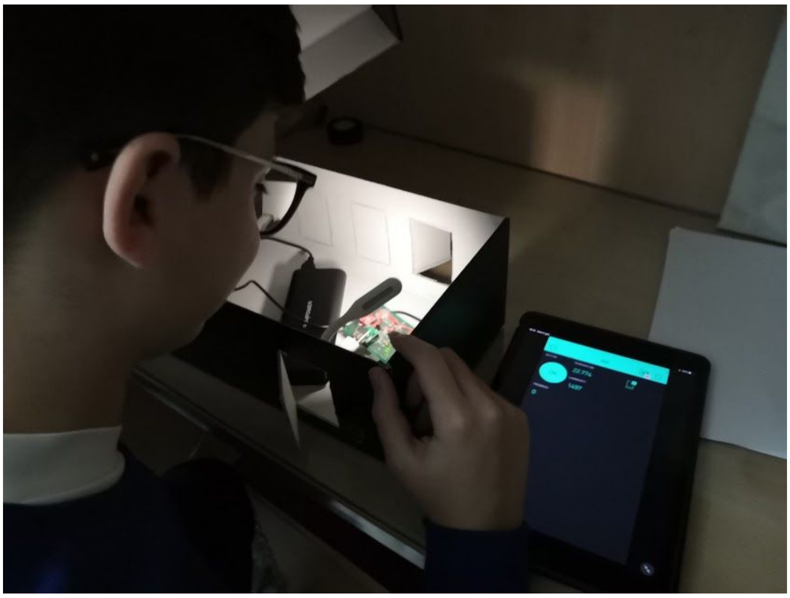
Here we have to describe how the miniature house is created, the environment to be simulated and the things you need to do it:

How to build the box?

- We take a box of shoes that we no longer use;
- With the help of a pair of scissors with a rounded tip, let's cut one of the shorter sides of the box. From here we could observe inside the box itself what will be simulated;

- Let's cut the other minor side in the same way. From this we will insert the small fan;
- By inserting the device inside the box, we created our miniature room and we are ready for the experiment.

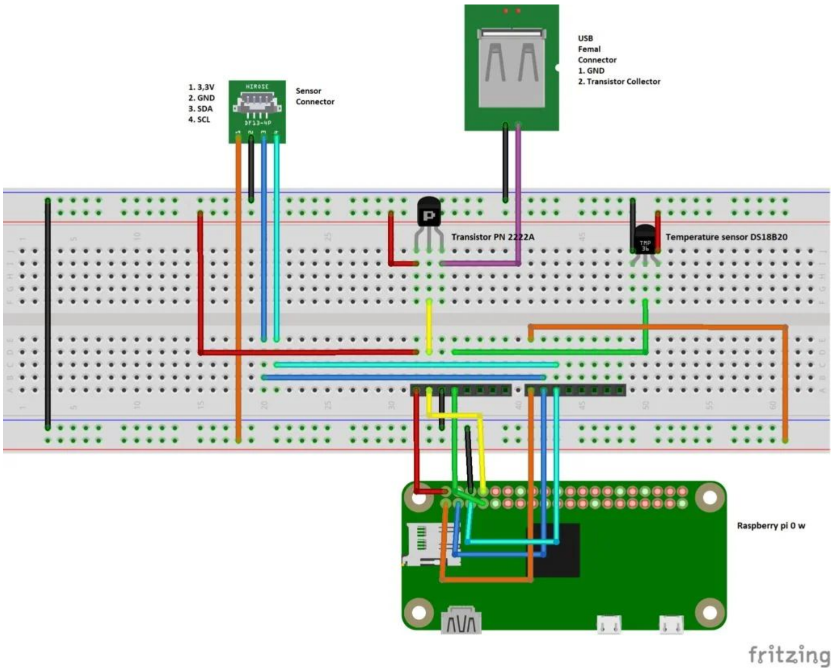


How to build the device?

For the construction of the device, consult the guide at the following link:

https://www.instructables.com/id/Digital-Environmental-Education-Domotics/

It is necessary that the pcb is well soldered so that the circuits are resistant and that they do not disconnect when you are going to put everything in the box. once the pcb is finished, it will have to be positioned over the raspberry for it to be put into operation.



To check if everything works well, switch the raspberry on and insert the pcb above it. With the help of a tester you can check that all the connections have been made well, just check that the voltage reaches all the desired points. A more thorough test can then be rerun when the whole device is ready.

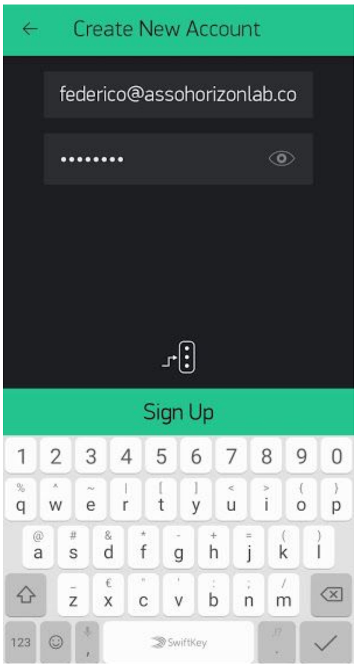
The attached diagram also indicates the temperature sensor as the activity in question is part of a package of pedagogical tools to improve the ecological sensitivity of the user. The temperature sensor is then used in another activity but the creation of a pcb for setting the device is preferable both as a single one. In this way you can reuse the same device on all available activities.

How to prepare the Blynk app?

To set up the software system via Blynk, you need to follow the link guide once again:

https://www.instructables.com/id/Digital-Environmental-Education-Domotics/

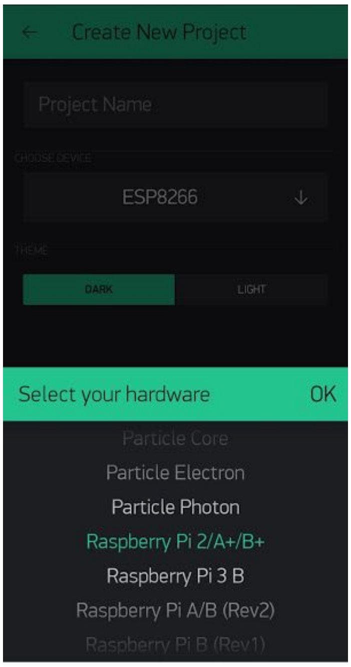
Once the application has been downloaded from the store, it is necessary to register a Blynk account by creating a profile, after which a new project must be created and a token generated. The token is an element that acts as a shared key, that is, it is a word (a token) that uniquely identifies the project and allows you to connect the device to the control terminal.



Create New Account

federico@assohorizonlab.co

••••••••



Create New Project

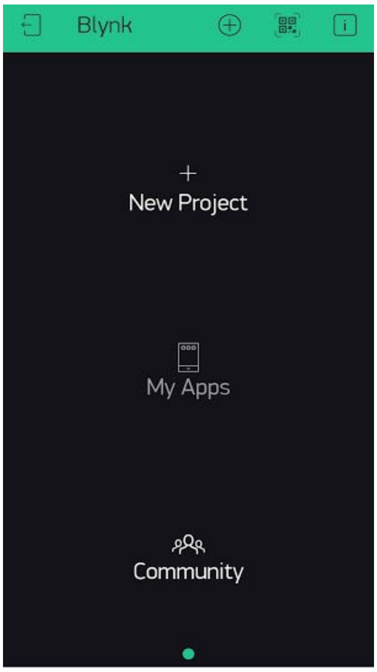
Project Name

CHOOSE DEVICE: ESP8266

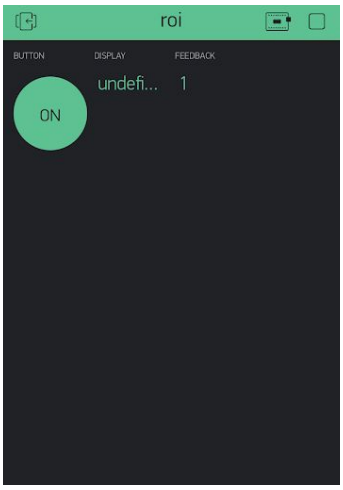
THEME: DARK LIGHT

Select your hardware OK

Particle Core
Particle Electron
Particle Photon
Raspberry Pi 2/A+/B+
Raspberry Pi 3 B
Raspberry Pi A/B (Rev2)
Raspberry Pi B (Rev1)



The app can be run on Android and Ios tablets and smartphones. This allows a wider audience of people to run and realize the remote controller.



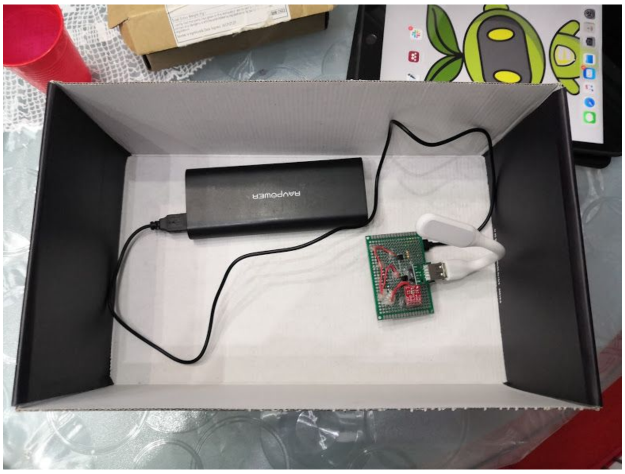
The Blynk app allows you to easily customize and change the interface of the remote controller. This guide explains how to make a version of the remote controller but nothing prevents you from adding new features to make it even more useful and nice.

To close everything in a wrapper, it may be useful to 3D print the suitable box whose source can be downloaded at the following link.

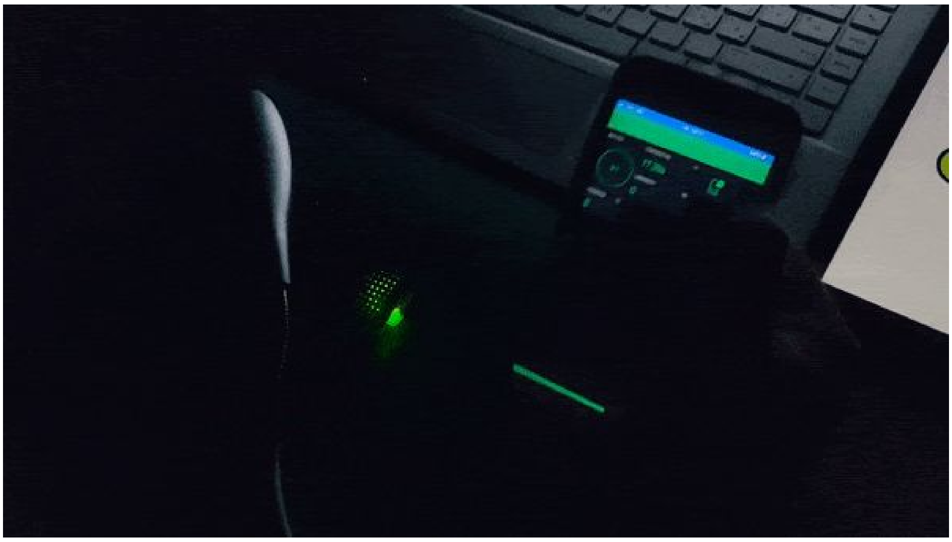
https://www.thingiverse.com/thing:4062244

Environmental monitoring

To check the ambient light conditions, just follow the values that can be read on the display. Once everything is ready, the brightness value that the device measures at runtime should be readable, the value varies with the variation of the ambient brightness.



If we put it outdoors on a sunny day, it will mark a decidedly high value, if we keep it in a room in the dark it will bring back a value close to zero, if not zero. By turning the usb lamp on and off, using the usb port controlled by the device, we will read a variation of the measured brightness.



Conclusion of the activity

At the end of the activity it is thought to have the children elaborate a page of diary, asking them to tell the experience they have carried out by highlighting the strengths and weaknesses of the device they have used and to provide any advice to the creators.

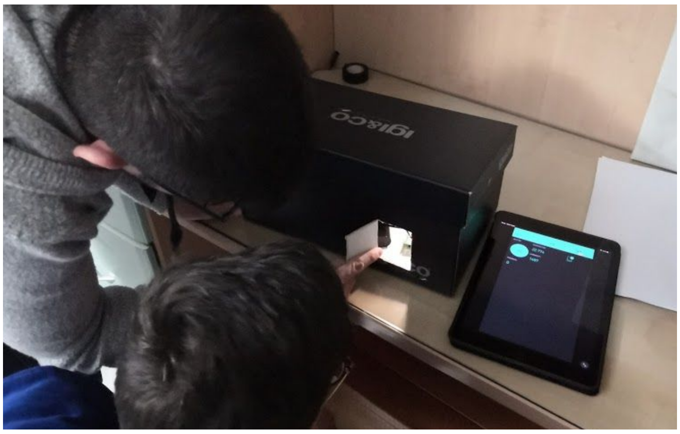
This will serve administrators, as well as creators of the device, on multiple fronts.
It will undoubtedly be useful to creators to work on weak points, and therefore to improve.
On the other hand, it will serve as a sort of database. In fact, this diary page will be kept by the creators in an archive, in order to be always accessible.
Furthermore, if the critical issues emerge, once they are improved, the creators could think of carrying out this activity again.
Therefore, the latter becomes essential for the formation of an archive and to be able to compare the results if the activity is proposed again later.

Results

By carrying out this activity that strongly recalls reality, children understand that this device can really be applied in any home environment, including their own home.

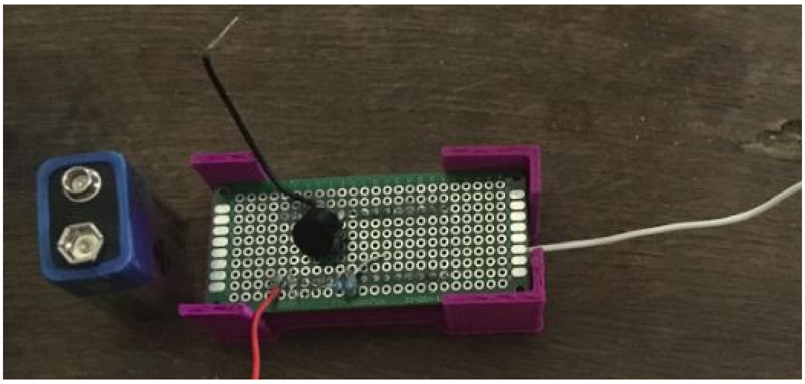
At the end of the activity, the children should have refined their awareness of the environment.

The purpose of the activity is to stimulate the user to make conscious use of energy resources in the home. The abstraction of the home environment by means of the box is aimed at arousing a connection in the user's mind with a real daily case. In this way, the user assimilates the advantages of using digital technology to make more efficient use of domestic resources. The perception of greater efficiency shows the user the sense of a reduction in waste.



EMI Probe - Device tutorial

In this tutorial you will learn how to assemble an EMI (electromagnetic interference) probe.



Introduction

EMI is a form of electromagnetic radiation: a combination of electric and magnetic waves traveling outward from anywhere that an electrical power signal is changing or being turned on and off rapidly

Where this gadget excels is spotting “phantom” or “vampire” energy loads.
More correctly called standby power, this is the amount of electricity that constantly flows through some electronic devices, even when they’re supposedly switched off or in standby mode. Devices use standby power on features such as digital clocks, remote control reception, and thermometers.
Relatively weak energy efficiency regulations in the United States result in many devices drawing far more wattage than they need in standby mode.

the EMI detector works by capturing the electrical energy coming into the arduino's analog port, and turning it into a sound via the speaker.

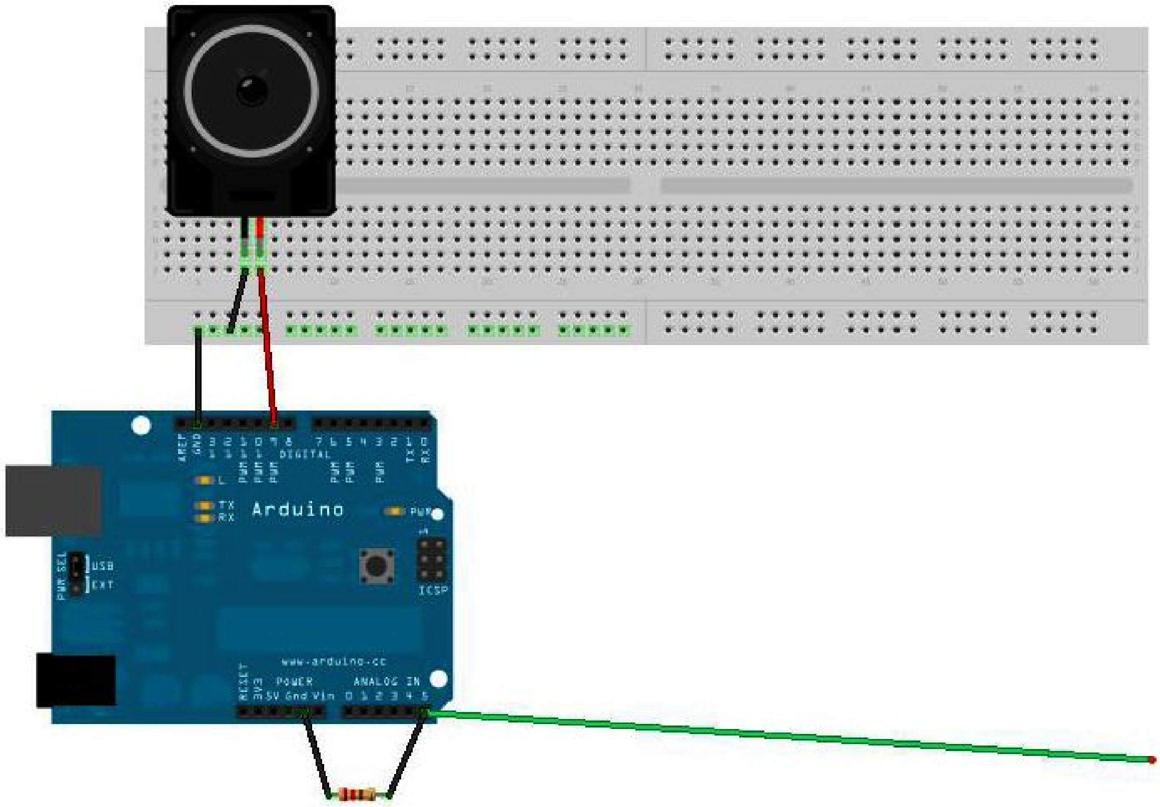
Assembling the EMI probe

It is possible to assemble an EMI probe using an arduino Uno or an arduino nano.

Here is a timelapse of the assembly process of an EMI probe based on arduino nano.
Here is a video of the assembly process of an EMI probe based on arduino uno

List of parts

- 1x Arduino uno or arduino nano + USB cable
- 1x 1MOhm resistor
- some single core hook up wire
- 1x 4x6cm PCB
- a few arduino male headers
- 1x piezo speaker
- link to the digital design of a case for your EMI detector (suitable if you are using an arduino nano)

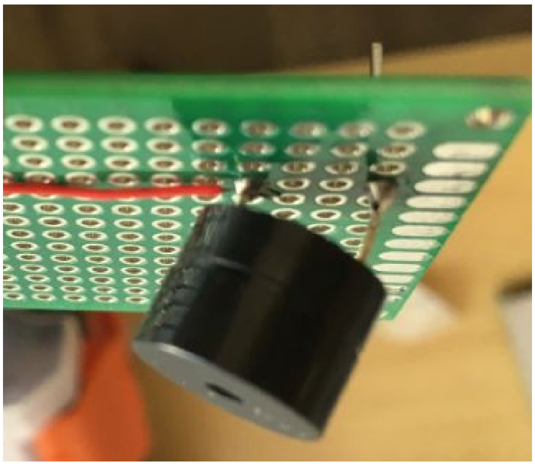


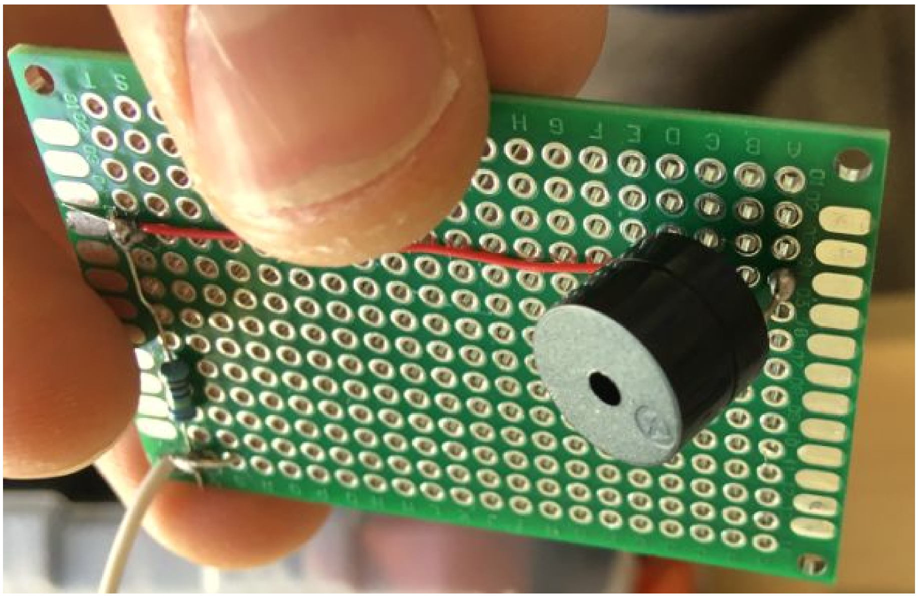
wiring diagram for the EMI probe

To begin with, solder 3 male headers on the PCB. When you will plug the PCB onto the arduino board, the headers will have to go into pin 9, GND, and Analog5.

Solder the speaker onto the PCB. The positive leg of the speaker need to be connected to the male header going into pin 9 of the arduino board.

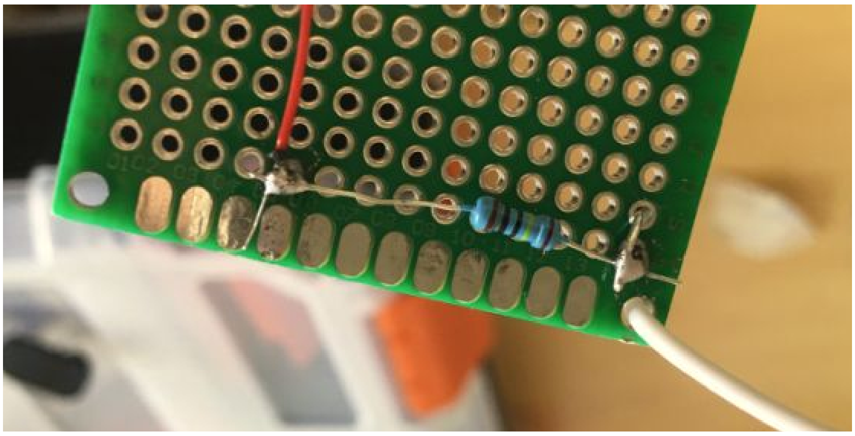
The other leg (negative leg) of the speaker needs be connected to one end of the resistor (via some hook up wire).

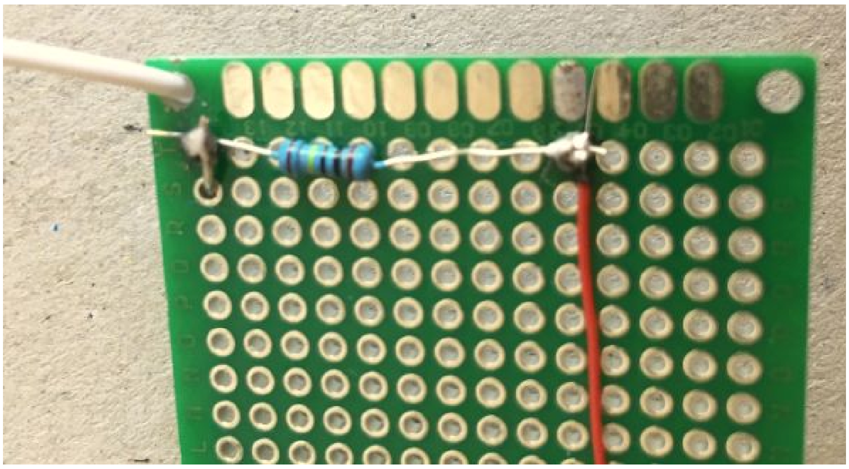




Now, solder the resistor onto the PCB. Connect one end of the resistor to the male header going into GND on the arduino board. Connect the other end to the header going into A5.

Grab a piece of solid core wire about 20 cm long, and solder one end in correspondence with the male header going into A5.





Your EMI probe is ready.

Program the EMI detector

Whether you re using an arduino uno or a nano, the code that you ll need to upload in order for the probe to function correctly is basically the same.
Just make sure to program the correct digital pin for the piezo speaker. In the instructions above, we connected the speaker on D9 on an arduino uno, and D3 on an arduino nano.

```
// Arduino Electromagnetic interference detector
// Code modified by Patrick Di Justo, based on
// Aaron ALAI EMF Detector April 22nd 2009 VERSION 1.0
// aaronalai1@gmail.com
//
// This outputs sound and numeric data to the 4char
#include <SoftwareSerial.h>
#define SerialIn 2
#define SerialOut 7
#define wDelay 900
int inPin = 5;
int val = 0;
SoftwareSerial mySerialPort(SerialIn, SerialOut);
void setup()
{
pinMode(SerialOut, OUTPUT);
pinMode(SerialIn, INPUT);
mySerialPort.begin(19200);
mySerialPort.print("vv");
mySerialPort.print("xxxx");
delay(wDelay);
mySerialPort.print("----");
delay(wDelay);
mySerialPort.print("8888");
delay(wDelay);
mySerialPort.print("xxxx");
delay(wDelay);
Serial.begin(9600);
}
void loop()
{
val = analogRead(inPin);
Serial.println(val);
dispData(val);
val = map(val, 1, 100, 1, 2048);
tone(9,val,10);
}
void dispData(int i)
{
if ((i<-999) || (i>9999))
{
mySerialPort.print("ERRx");
return;
}
char fourChars[5];
sprintf(fourChars, "%04d", i);
mySerialPort.print("v");
mySerialPort.print(fourChars);
}
```

The full arduino code is also available here.

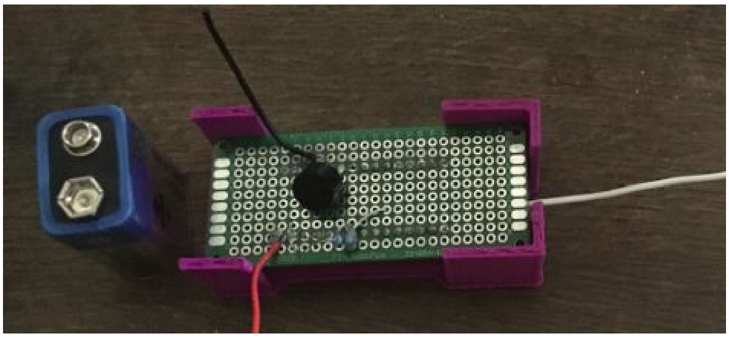
Because Arduino is connected by a USB cable to your computer, it is receiving a flood of electromagnetic interference from the computer. Even worse, that EMI is being pumped into Arduino via the USB cable. To make this detector really work, **we've got to go mobile**.
A fresh 9-volt battery should be enough to get this gadget running.

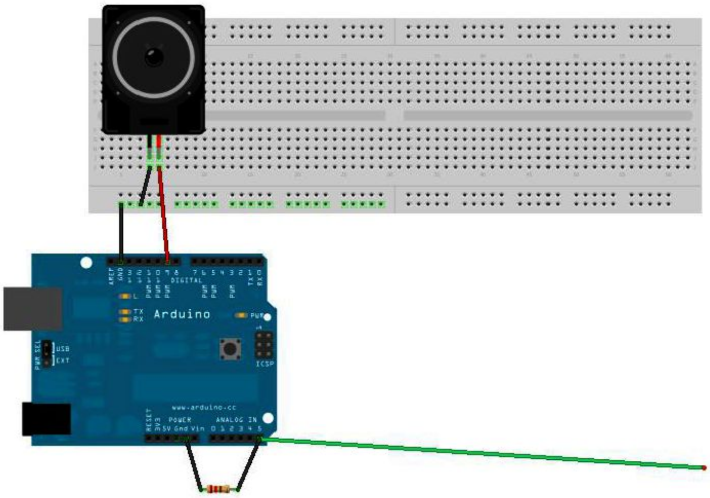
Your Arduino should start up normally: the LEDs mounted on the Arduino board should flash, and within a few seconds the EMI code should be up and running.

Watch the EMI probe in action here.

EMI Probe - 9-12 yo activity tutorial

In this tutorial you will learn how a DIY EMI detector works.
You will also understand what electromagnetic interference is, and why it's important to be aware of it.





Introduction

EMI is a form of electromagnetic radiation: a combination of electric and magnetic waves traveling outward from anywhere that an electrical power signal is changing or being turned on and off rapidly

Where this gadget excels is spotting “phantom” or “vampire” energy loads.
More correctly called standby power, this is the amount of electricity that constantly flows through some electronic devices, even when they’re supposedly switched off or in standby mode. Devices use standby power on features such as digital clocks, remote control reception, and thermometers.
Relatively weak energy efficiency regulations in the United States result in many devices drawing far more wattage than they need in standby mode.

the EMI detector works by capturing the electrical energy coming into the arduino's analog port, and turning it into a sound via the speaker.

Program the EMI detector

Whether you re using an arduino uno or a nano, the code that you ll need to upload in order for the probe to function correctly is basically the same.
Just make sure to program the correct digital pin for the piezo speaker. In the instructions above, we connected the speaker on D9 on an arduino uno, and D3 on an arduino nano.

```
// This outputs sound and numeric data to the 4char
#include <SoftwareSerial.h>
#define SerialIn 2
#define SerialOut 7
#define wDelay 900
int inPin = 5;
int val = 0;
SoftwareSerial mySerialPort(SerialIn, SerialOut);
void setup()
{
pinMode(SerialOut, OUTPUT);
pinMode(SerialIn, INPUT);
mySerialPort.begin(19200);
mySerialPort.print("vv");
mySerialPort.print("xxxx");
delay(wDelay);
mySerialPort.print("----");
delay(wDelay);
mySerialPort.print("8888");
delay(wDelay);
mySerialPort.print("xxxx");
delay(wDelay);
Serial.begin(9600);
}
void loop()
{
val = analogRead(inPin);
Serial.println(val);
dispData(val);
val = map(val, 1, 100, 1, 2048);
tone(9,val,10);
}
void dispData(int i)
{
if ((i<-999) || (i>9999))
{
mySerialPort.print("ERRx");
return;
}
char fourChars[5];
sprintf(fourChars, "%04d", i);
mySerialPort.print("v");
mySerialPort.print(fourChars);
}
```

The full arduino code is also available here.

Because Arduino is connected by a USB cable to your computer, it is receiving a flood of electromagnetic interference from the computer. Even worse, that EMI is being pumped into Arduino via the USB cable. To make this detector really work, **we've got to go mobile**.
A fresh 9-volt battery should be enough to get this gadget running.
Your Arduino should start up normally: the LEDs mounted on the Arduino board should flash, and within a few seconds the EMI code should be up and running.

Activity Idea

You can use the EMI probe to compare and contrast EMI radiations deriving from different electronic appliances.
Hold the probe next to a stereo system or a TV whilst these devices are in standby mode, and you ll probably get a similar reading to a laptop when this is turned on.
Once you ve found out which electronic appliances radiate the biggest amount of EMI when in standby mode, you can learn to plug these off to save energy.

EMI Probe - 12+ yo activity tutorial

Introduction

In this tutorial you ll learn how to build you own DIY electromagnetic interference detector. You ll understand what EMI is, and why it’s important to be aware of it.

EMI is a form of electromagnetic radiation: a combination of electric and magnetic waves traveling outward from anywhere that an electrical power signal is changing or being turned on and off rapidly

Where this gadget excels is spotting “phantom” or “vampire” energy loads.
More correctly called standby power, this is the amount of electricity that constantly flows through some electronic devices, even when they’re supposedly switched off or in standby mode. Devices use standby power on features such as digital clocks, remote control reception, and thermometers.
Relatively weak energy efficiency regulations in the United States result in many devices drawing far more wattage than they need in standby mode.

the EMI detector works by capturing the electrical energy coming into the arduino's analog port, and turning it into a sound via the speaker.

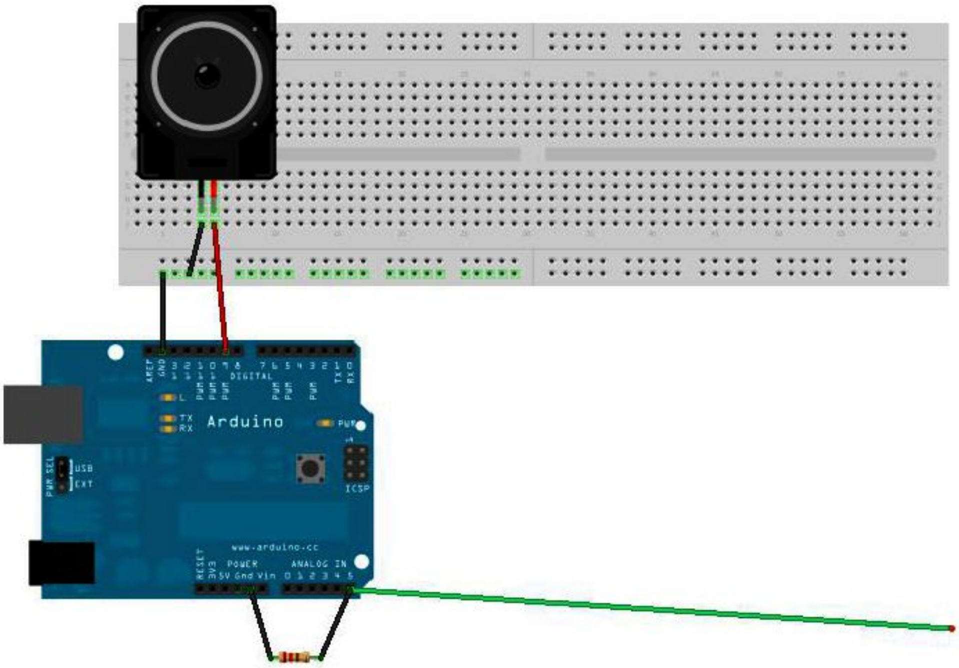
Building the EMI detector

The EMI detector comes in two forms: the gadget is mounted on a shield suitable for an arduino uno board, or the detector is embedded on a shield on which an arduino nano is mounted.
We will start by building the shield for arduino uno

List of parts:

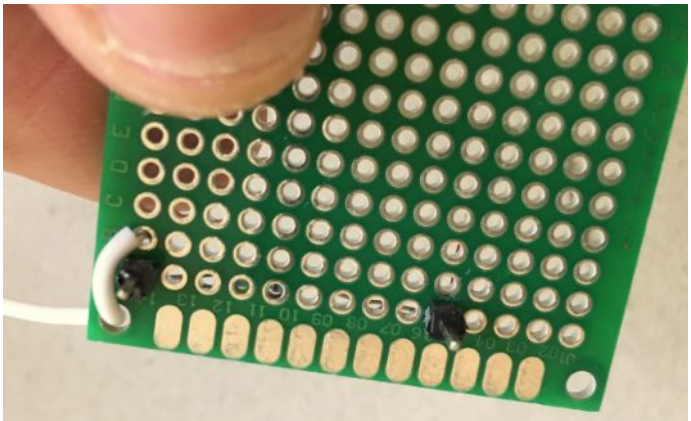
1x 8ohm speaker
1x arduino uno board
1x 40x60 PCB
1x 1Mohm resistor
single core electric wire
male pin header strip for arduino
1x arduino uno or nano board

here's the wiring diagram for the EMI detector



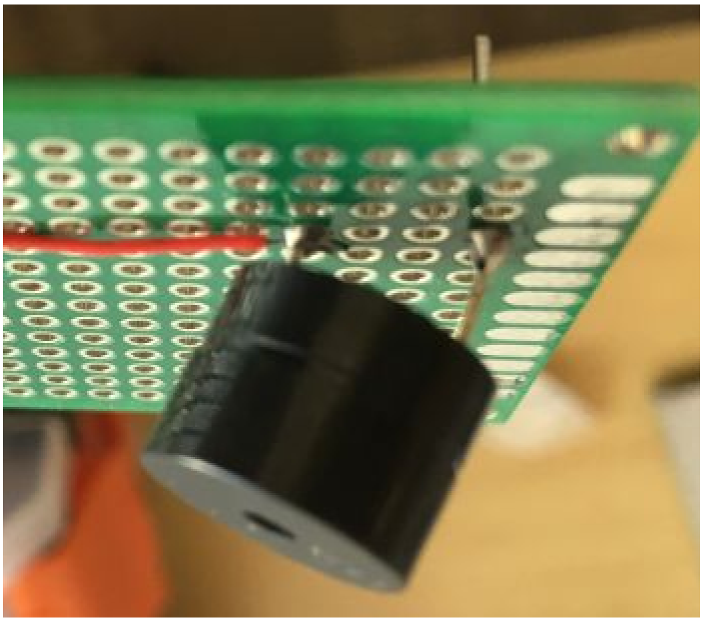
Step by step instructions for the Arduino Uno shield

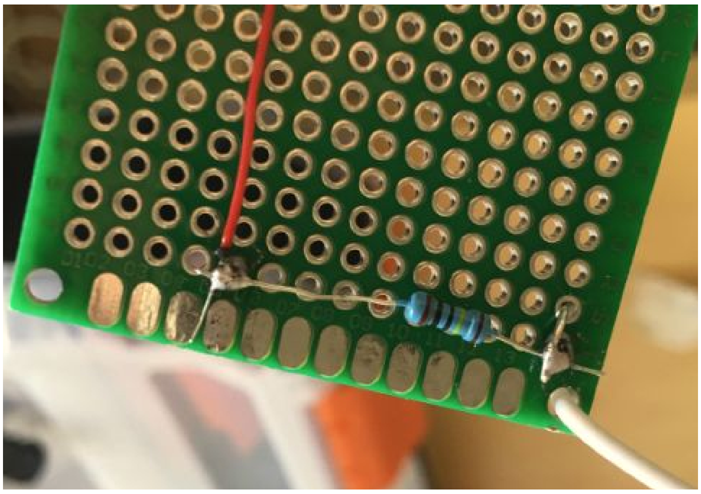
First, solder at least a couple of pins to the PCB. These will go into GND and Analog 5 in th arduino uno board.



Next, solder an extra pin on the opposite side of the PCB. This will connect to Digital 9 on the arduino board.

The speaker will also be soldered onto the PCB. Solder the positive end of the speaker to the pin which goes into analog 9 on the arduino.
Solder the negative end of the speaker into the PCB.
Then connect a short (5 cm max) piece of electric wire to the negative end of the speaker.
The other end of the cable is soldered on the pin which goes into GND.

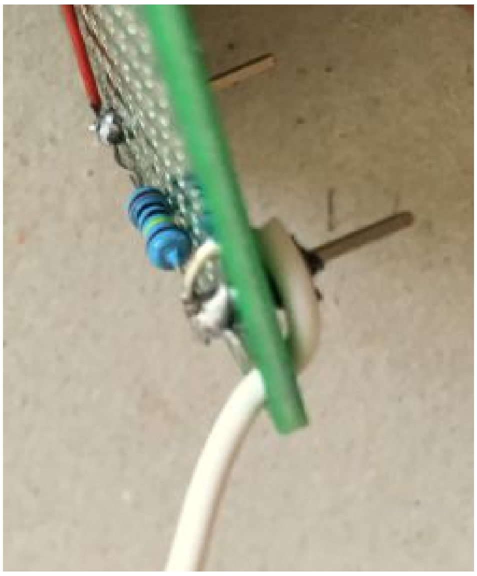




Use a 1Mohm resistor to connect the pin which goes into GND and the one that goes into Analog 5 on the PCB (see photo above).

It’s now time to add the antenna of your EMI detector.

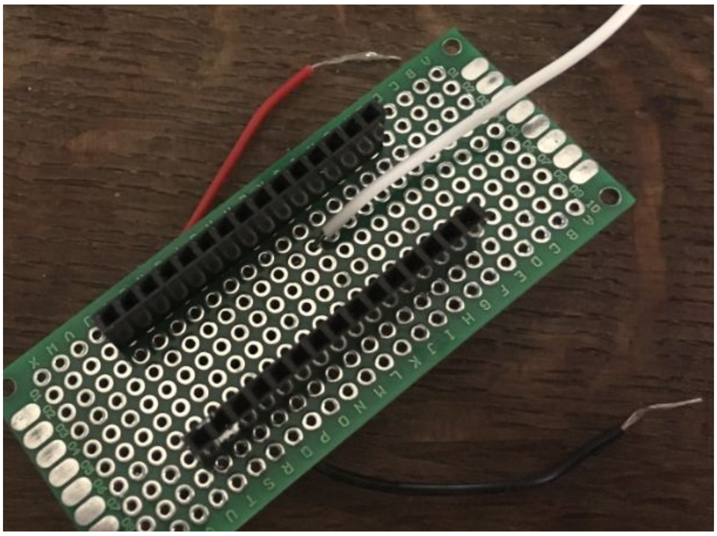
Take about 20 cm of solid core wire, and solder one end of it on to the PCB, precisely to the pin that goes into Analog 5 on the board.



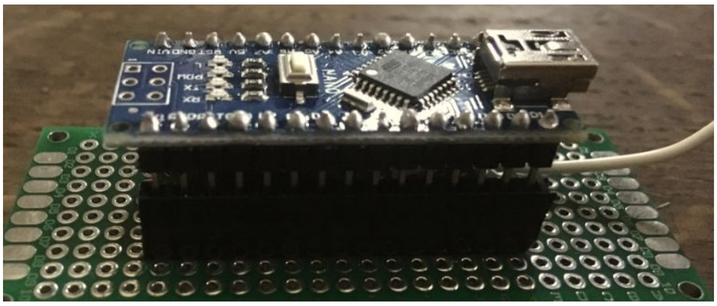
Step by step instructions for the Arduino Nano shield

An timelapse is available here

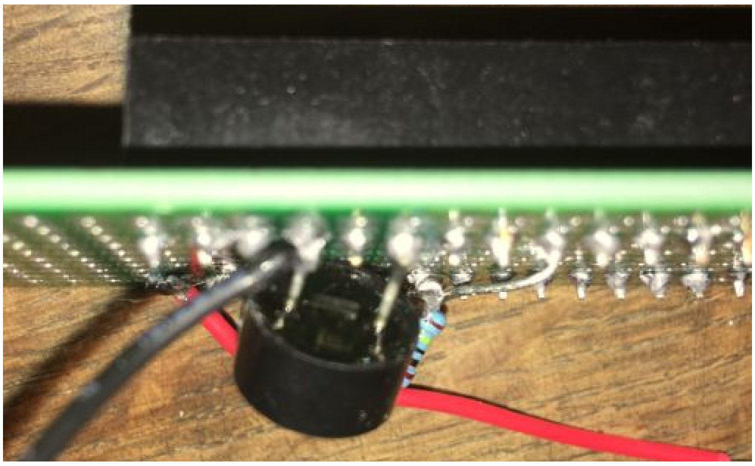
Solder two strips of female headers onto a PCB (3cm x 7cm)



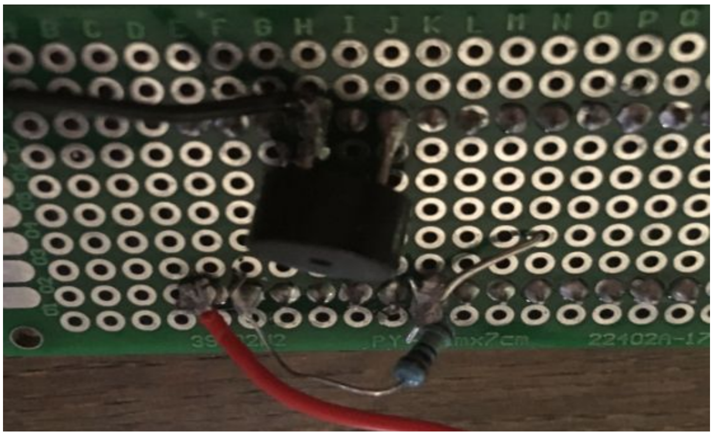
You will need to be able to arrange the arduino nano onto these strips of female headers.



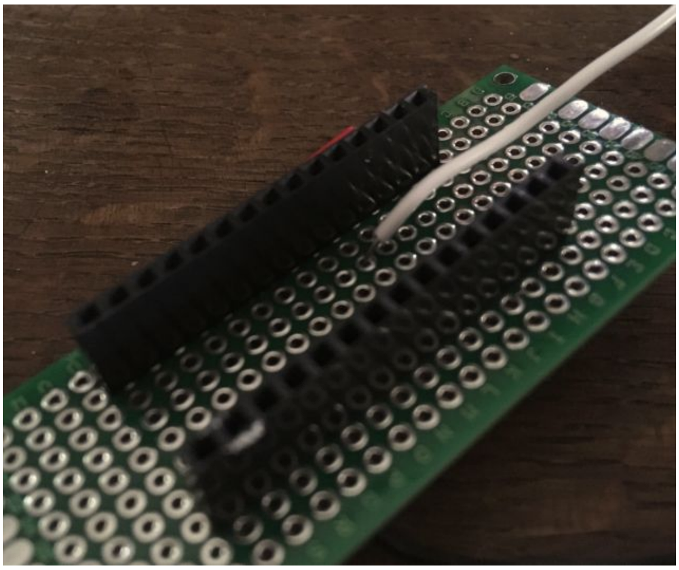
Solder the positive side of the speaker to the PCB, in correspondence with the D3 pin. Solder the other end of the speaker onto the PCB, in correspondence with GND pin of the arduino nano.

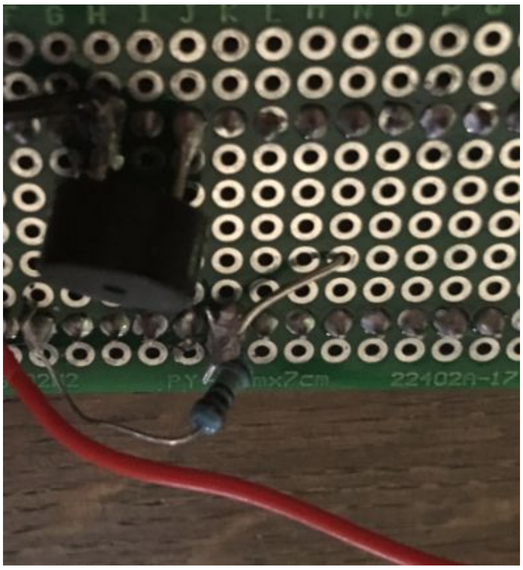


Next, grab the 1Mohm resistor, and solder one end to the PCB pin which leads to A5 on the board, the other end to the PCB pin which goes into GND.

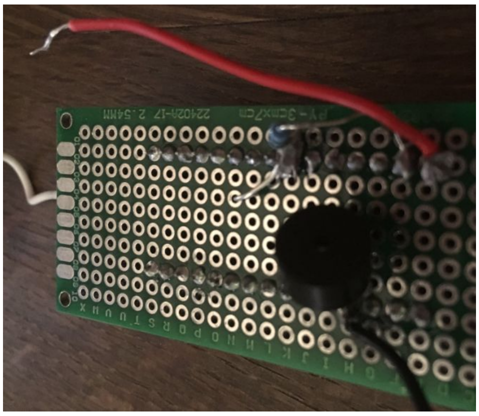


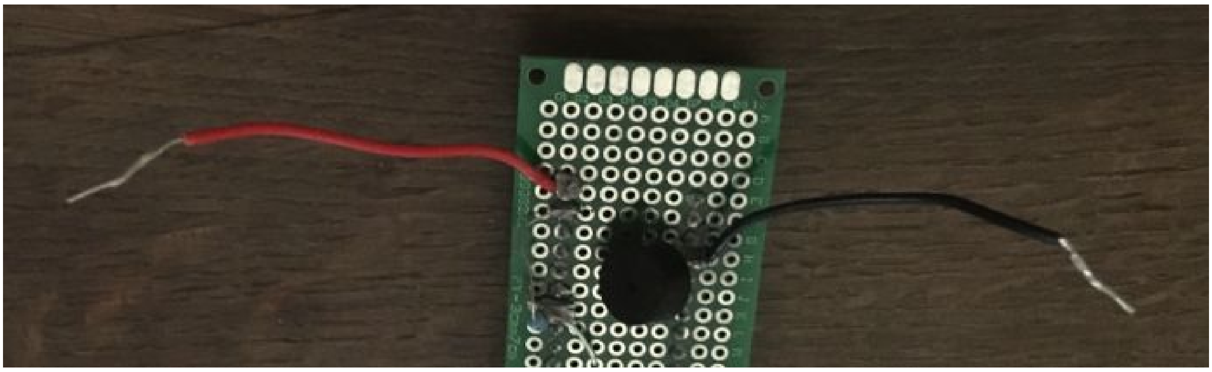
To make the antenna of your device, take a piece of solid core wire (about 15 cm long), and solder one end of it to the PCB pin which leads to GND on the arduino nano.





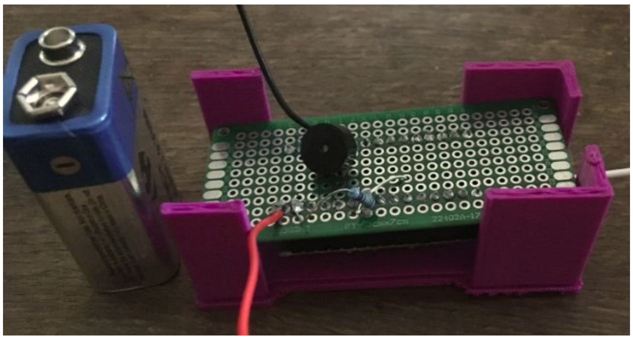
Finally, grap two short pieces of electric wire. You will use them to connect a 9V battery to the arduino nano and power the board. Solder one end of the first cable to VIN on the arduino nano, solder one end of the other cable to GND.

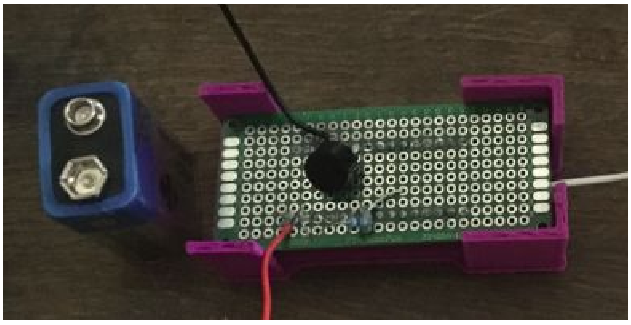




Place the EMI detector inside its case

You can download the stl file and 3D print the case. The stl file is available here.





Program the arduino board

Whether you re using an arduino uno or a nano, the code that you ll need to upload in order for the probe to function correctly is basically the same.
Just make sure to program the correct digital pin for the piezo speaker. In the instructions above, we connected the speaker on D9 on an arduino uno, and D3 on an arduino nano.

```
// Arduino Electromagnetic interference detector
// Code modified by Patrick Di Justo, based on
// Aaron ALAI EMF Detector April 22nd 2009 VERSION 1.0
// aaronalai1@gmail.com
//
// This outputs sound and numeric data to the 4char
#include <SoftwareSerial.h>
#define SerialIn 2
#define SerialOut 7
#define wDelay 900
int inPin = 5;
int val = 0;
SoftwareSerial mySerialPort(SerialIn, SerialOut);
void setup()
```

```
{
pinMode(SerialOut, OUTPUT);
pinMode(SerialIn, INPUT);
mySerialPort.begin(19200);
mySerialPort.print("vv");
mySerialPort.print("xxxx");
delay(wDelay);
mySerialPort.print("----");
delay(wDelay);
mySerialPort.print("8888");
delay(wDelay);
mySerialPort.print("xxxx");
delay(wDelay);
Serial.begin(9600);
}
void loop()
{
val = analogRead(inPin);
Serial.println(val);
dispData(val);
val = map(val, 1, 100, 1, 2048);
tone(9,val,10);
}
void dispData(int i)
{
if ((i<-999) || (i>9999))
{
mySerialPort.print("ERRx");
return;
}
char fourChars[5];
sprintf(fourChars, "%04d", i);
mySerialPort.print("v");
mySerialPort.print(fourChars);
}
```

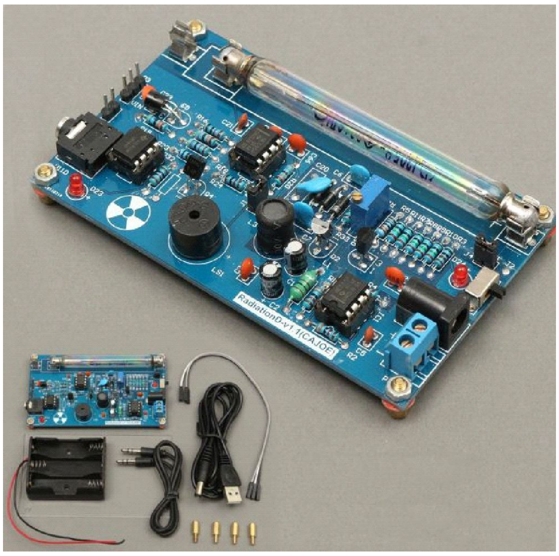
The full arduino code is also available here.

Activity Idea

You can use the EMI probe to compare and contrast EMI radiations deriving from different electronic appliances.
Hold the probe next to a stereo system or a TV whilst these devices are in standby mode, and you ll probably get a similar reading to a laptop when this is turned on.

Geiger Counter - device tutorial

In this tutorial you will learn how to assemble a nuclear radiation detector
You can purchase the Geiger Counter Kit here



Introduction

A Geiger counter is an instrument used for detecting and measuring ionizing radiation. Also known as a Geiger–Mueller counter (or Geiger–Müller counter), it is widely used in applications such as radiation dosimetry, radiological protection, experimental physics, and the nuclear industry.

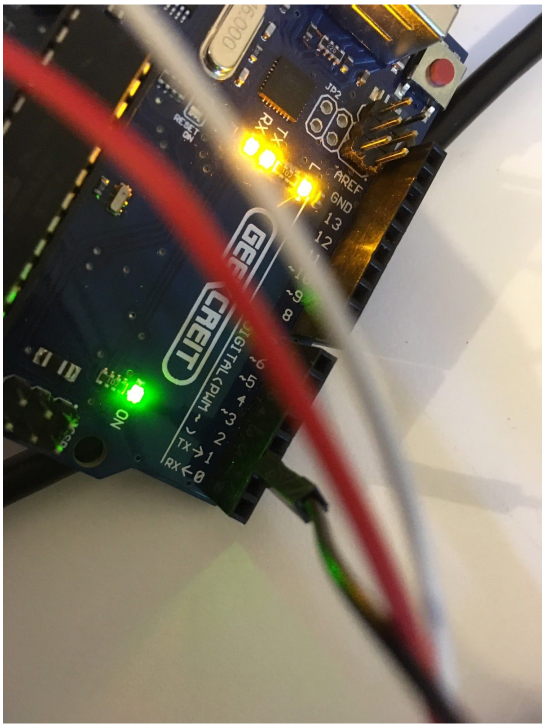
Geiger counters are used to detect radioactive emissions, most commonly beta particles and gamma rays. The counter consists of a tube filled with an inert gas that becomes conductive of electricity when it is impacted by a high-energy particle.

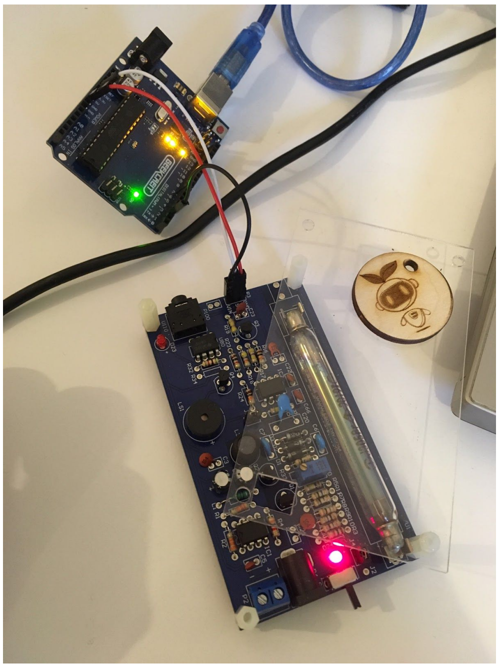
Soldering the geiger counter

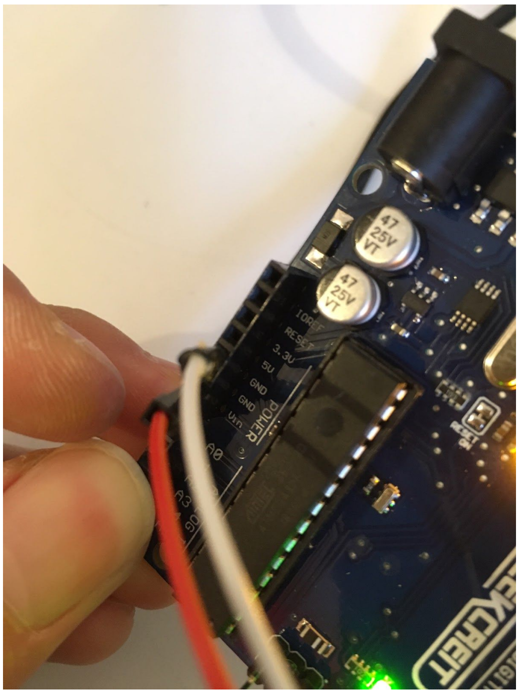
This is a timelapse of the assembly process of the geiger counter

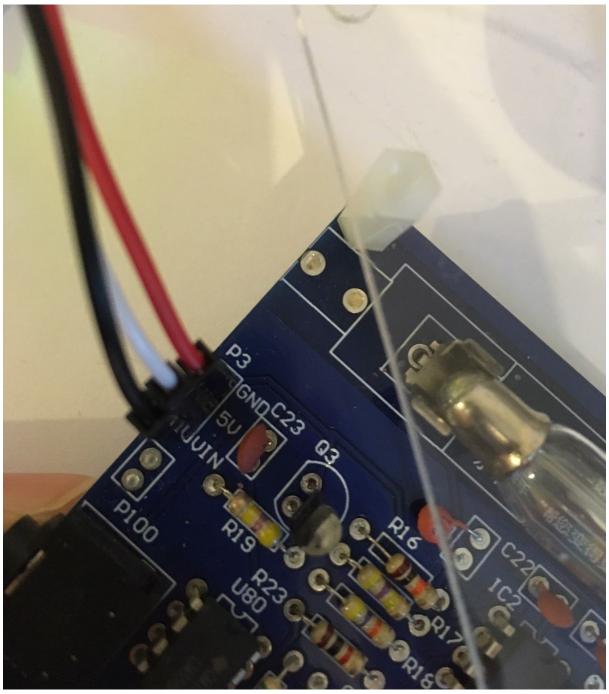
Using the geiger counter with an arduino

Connect the P3 Pin GND, 5V, VIN to arduino GND, 5V, Digital 2 respectively.
Then in the arduino software open the file: spi_rad_logger.ino which you could find here
https://drive.google.com/open?id=1BBhsOjpKFHZ5vheR6OtunmriIw6JLzbc

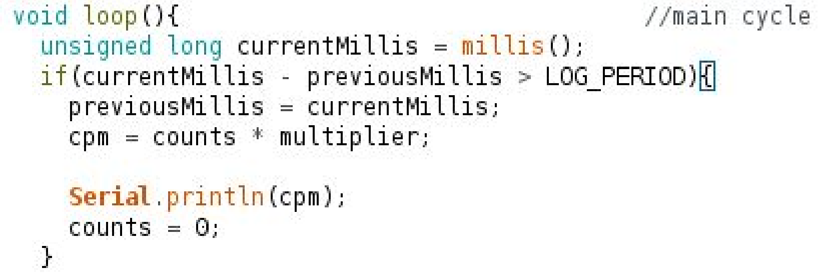






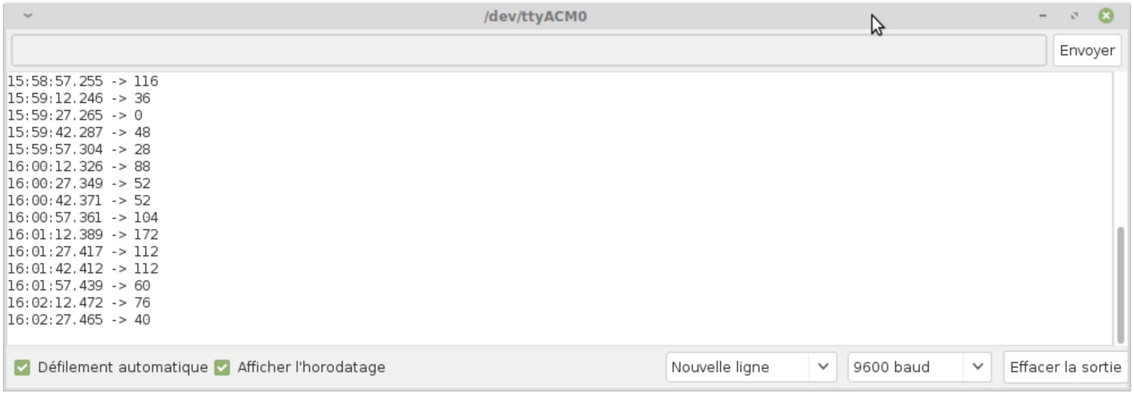


Be sure to change the Serial.print(cpm) command to Serial.println(cpm) in the void loop(){} for better readability.



```
void loop(){                                  //main cycle
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis > LOG_PERIOD){
    previousMillis = currentMillis;
    cpm = counts * multiplier;

    Serial.println(cpm);
    counts = 0;
  }
```

Download the program and open the serial port window by clicking on the scope on the upper right corner.



```
15:58:57.255 -> 116
15:59:12.246 -> 36
15:59:27.265 -> 0
15:59:42.287 -> 48
15:59:57.304 -> 28
16:00:12.326 -> 88
16:00:27.349 -> 52
16:00:42.371 -> 52
16:00:57.361 -> 104
16:01:12.389 -> 172
16:01:27.417 -> 112
16:01:42.412 -> 112
16:01:57.439 -> 60
16:02:12.472 -> 76
16:02:27.465 -> 40
```

☑ Défilement automatique ☑ Afficher l'horodatage Nouvelle ligne 9600 baud Effacer la sortie

Then we'll get the radiation value displayed in CPM, counter per minutes which could be converted to uSv/h with the index 151(151CPM=1uSv/h for M4011 GM
Tube).

Geiger Counter - 9-12 yo activity tutorial

In this tutorial you will learn how to assemble a nuclear radiation detector
You can purchase the Geiger Counter Kit here

Introduction

A Geiger counter is an instrument used for detecting and measuring ionizing radiation. Also known as a Geiger–Mueller counter (or Geiger–Müller counter), it is widely used in applications such as radiation dosimetry, radiological protection, experimental physics, and the nuclear industry.

Geiger counters are used to detect radioactive emissions, most commonly beta particles and gamma rays. The counter consists of a tube filled with an inert gas that becomes conductive of electricity when it is impacted by a high-energy particle.

What is radiation?

https://world-nuclear.org/nuclear-basics/what-is-radiation.aspx

Radiation is energy travelling through space.

Sunshine is one of the most familiar forms of radiation. It delivers light, heat and suntans. While enjoying and depending on it, we control our exposure to it.

Beyond ultraviolet radiation from the sun are higher-energy kinds of radiation which are used in medicine and which we all get in low doses from space, from the air, and from the earth and rocks.

Radiation sources in daily life

https://www.euradcom.org/top-5-sources-of-radiation-in-everyday-life/

1. ***Television***

The average American over the age of 2 watches 4.5 hours of TV daily. The electrical conductivity in TV sets and computer monitors gives off a minimal amount of X-rays: 1 mrem per year to the typical consumer. However, there are more urgent health hazards such as obesity if you pass several hours per day immobile in front of a screen.

2. ***Radon***

A colorless, odorless gas given off by decaying uranium seeps into the foundation of one out of 15 American homes and takes up residency in their basements. Luckily, you can test your house for high levels of radon and take the necessary steps to protect your family from this gas by consulting www.epa.gov.

3. ***Medical Imaging***

Obviously one does not undergo medical imaging procedures on a daily basis, but as the most common source of exposure for Americans beyond normal background radiation, medical imaging bodes mentioning. Medical imaging procedures such as dental or chest X-rays send 10 mrem to the patient. Mammograms log in at 138 mrem per image, and CT scans can deliver up to 1,000. An even higher dosage procedure, the colonography, produces 10,000 mrem, which increases your risk of cancer by 1%. However, if your doctor recommends any of these procedures, you’re better off taking the radiation risk than declining the procedure.

4. ***Cell phones***

Cell phones emit radiofrequency waves, a non-ionizing form of radiation, albeit at a low enough dose that there are no established health effects.

Here you can find out more about how to avoid radiation from cell phones.

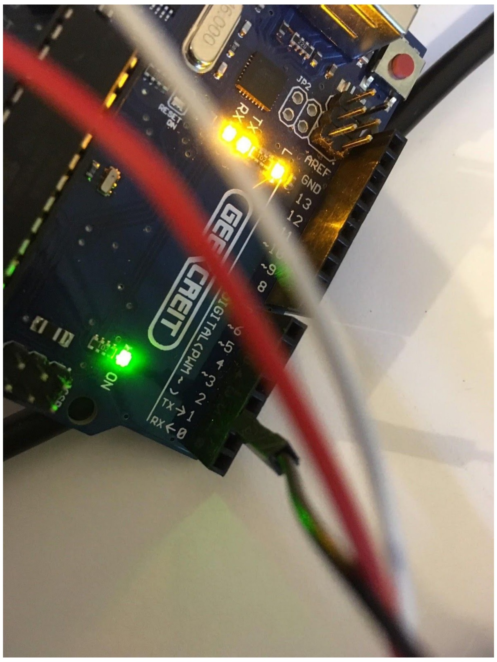
5. ***Smoking***

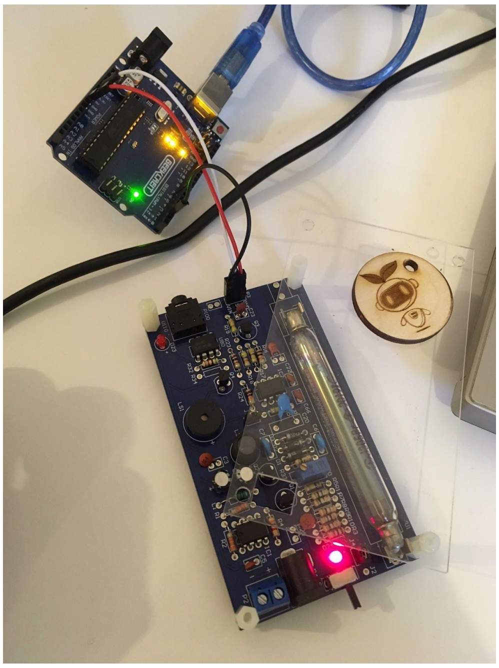
It should come as no surprise that cigarettes causes health problems even beyond the carcinogens in the tar component of smoke your body takes in with each inhale. Heavy smokers increase their radiation exposure by 870 mrem per year – more than doubling or even tripling their exposure compared to non-smokers.

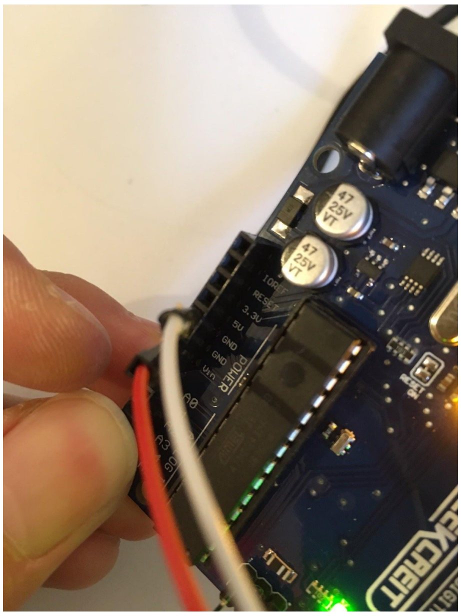
Keep in mind that most these quotidian objects and personal habits expose you to what, in the end, is a minimal amount of radiation. To learn more about the sources and risks of radiation, consult the International Atomic Energy Agency's findings on radiation in everyday life.

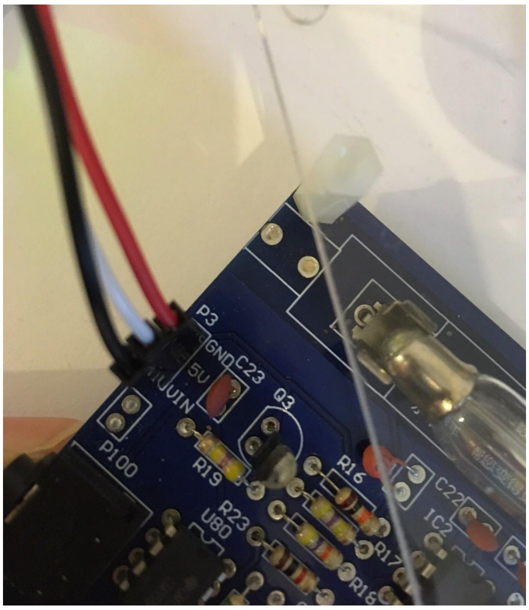
Using the geiger counter with an arduino

Connect the P3 Pin GND, 5V, VIN to arduino GND, 5V, Digital 2 respectively.

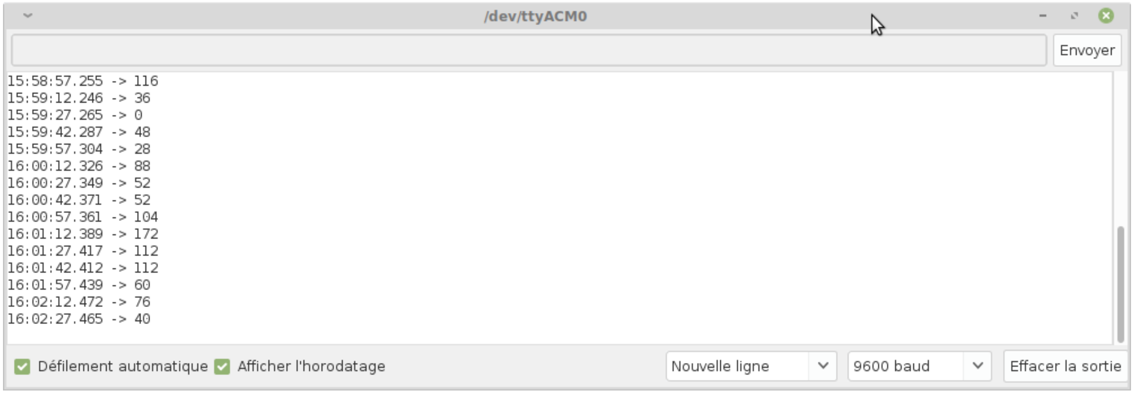








Open the serial port window by clicking on the scope on the upper right corner.



```
15:58:57.255 -> 116
15:59:12.246 -> 36
15:59:27.265 -> 0
15:59:42.287 -> 48
15:59:57.304 -> 28
16:00:12.326 -> 88
16:00:27.349 -> 52
16:00:42.371 -> 52
16:00:57.361 -> 104
16:01:12.389 -> 172
16:01:27.417 -> 112
16:01:42.412 -> 112
16:01:57.439 -> 60
16:02:12.472 -> 76
16:02:27.465 -> 40
```

Then we'll get the radiation value displayed in CPM, counter per minutes which could be converted to uSv/h with the index 151(151CPM=1uSv/h for M4011 GM Tube).

Risk of radiation

https://fr.search.yahoo.com/yhs/search?hspart=ddc&hsimp=yhs-linuxmint&type=__alt__ddc_linuxmint_com&p=dangerous+dose+of+radiation

https://www.reuters.com/article/us-how-much-radiation-dangerous-idUSTRE72E79Z20110315
* There is documented evidence associating an accumulated dose from two or three CT scans with an increased risk of cancer. The evidence is reasonably convincing for adults and very convincing for children.

* Large doses of radiation or acute radiation exposure destroys the central nervous system, red and white blood cells, which compromises the immune system, leaving the victim unable to fight off infections.

For example, a single one sievert (1,000 mSv) dose causes radiation sickness such as nausea, vomiting, hemorrhaging, but not death. A single dose of 5 sieverts would kill about half of those exposed to it within a month.

* Exposure to 350 mSv was the criterion for relocating people after the Chernobyl accident, according to the World Nuclear Association.

Geiger Counter - 12+ yo activity tutorial

In this tutorial you will learn how to assemble a nuclear radiation detector.
You can purchase the Geiger Counter Kit here.

Introduction

A Geiger counter is an instrument used for detecting and measuring ionizing radiation. Also known as a Geiger–Mueller counter (or Geiger–Müller counter), it is widely used in applications such as radiation dosimetry, radiological protection, experimental physics, and the nuclear industry.

Geiger counters are used to detect radioactive emissions, most commonly beta particles and gamma rays. The counter consists of a tube filled with an inert gas that becomes conductive of electricity when it is impacted by a high-energy particle.

What is radiation?

https://world-nuclear.org/nuclear-basics/what-is-radiation.aspx

Radiation is energy travelling through space.

Sunshine is one of the most familiar forms of radiation. It delivers light, heat and suntans. While enjoying and depending on it, we control our exposure to it.

Beyond ultraviolet radiation from the sun are higher-energy kinds of radiation which are used in medicine and which we all get in low doses from space, from the air, and from the earth and rocks.

Radiation sources in daily life

https://www.euradcom.org/top-5-sources-of-radiation-in-everyday-life/

1. ***Television***

The average American over the age of 2 watches 4.5 hours of TV daily. The electrical conductivity in TV sets and computer monitors gives off a minimal amount of X-rays: 1 mrem per year to the typical consumer. However, there are more urgent health hazards such as obesity if you pass several hours per day immobile in front of a screen.

2. ***Radon***

A colorless, odorless gas given off by decaying uranium seeps into the foundation of one out of 15 American homes and takes up residency in their basements. Luckily, you can test your house for high levels of radon and take the necessary steps to protect your family from this gas by consulting www.epa.gov.

3. *Medical Imaging*

Obviously one does not undergo medical imaging procedures on a daily basis, but as the most common source of exposure for Americans beyond normal background radiation, medical imaging bodes mentioning. Medical imaging procedures such as dental or chest X-rays send 10 mrem to the patient. Mammograms log in at 138 mrem per image, and CT scans can deliver up to 1,000. An even higher dosage procedure, the colonography, produces 10,000 mrem, which increases your risk of cancer by 1%. However, if your doctor recommends any of these procedures, you're better off taking the radiation risk than declining the procedure.

4. *Cell phones*

Cell phones emit radiofrequency waves, a non-ionizing form of radiation, albeit at a low enough dose that there are no established health effects.

Here you can find out more about how to avoid radiation from cell phones.

5. *Smoking*

It should come as no surprise that cigarettes causes health problems even beyond the carcinogens in the tar component of smoke your body takes in with each inhale. Heavy smokers increase their radiation exposure by 870 mrem per year – more than doubling or even tripling their exposure compared to non-smokers.

Keep in mind that most these quotidian objects and personal habits expose you to what, in the end, is a minimal amount of radiation. To learn more about the sources and risks of radiation, consult the International Atomic Energy Agency's findings on radiation in everyday life.

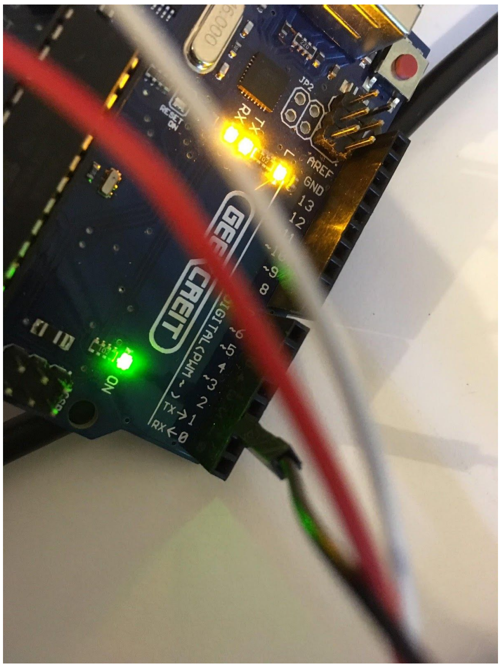
Soldering the geiger counter

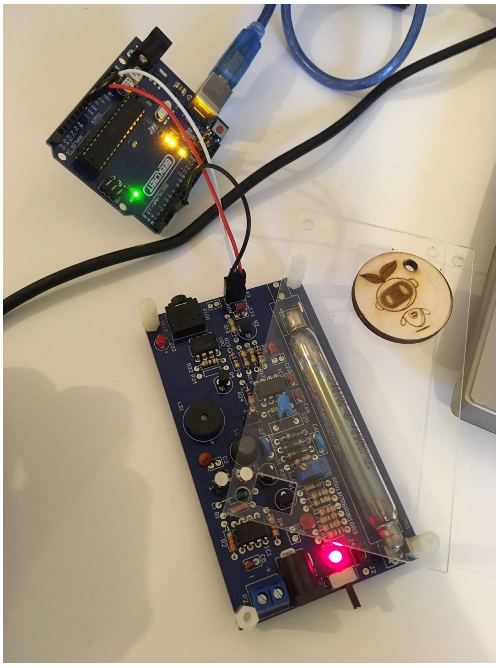
https://youtu.be/ImHbKBu3xus

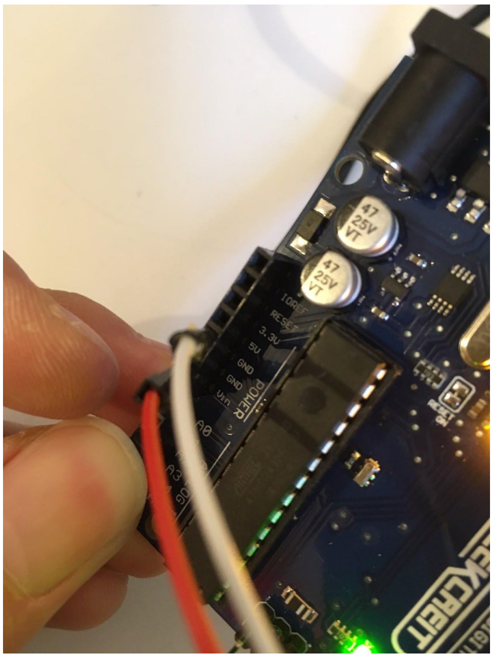
is a timelapse of the assembly process of an EMI probe based on arduino nano

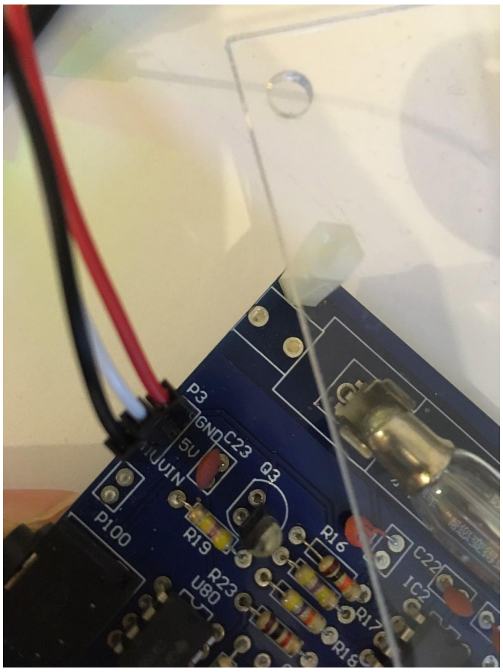
Using the geiger counter with an arduino

Connect the P3 Pin GND, 5V, VIN to arduino GND, 5V, Digital 2 respectively.
Then in the arduino software open the file: spi_rad_logger.ino which you could find here.

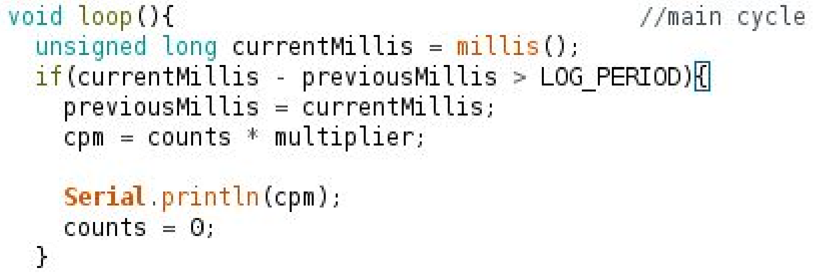






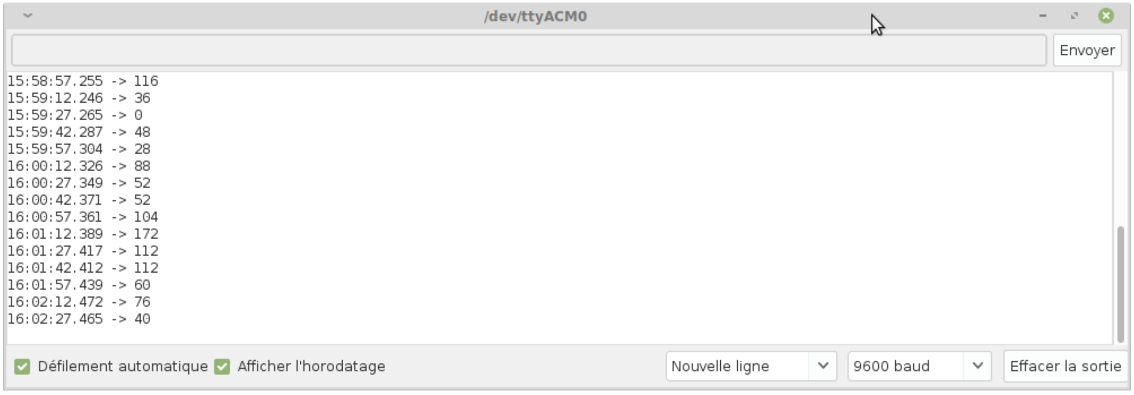


Be sure to change the Serial.print(cpm) command to Serial.println(cpm) in the void loop(){} for better readability.



```
void loop(){                                        //main cycle
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis > LOG_PERIOD){
    previousMillis = currentMillis;
    cpm = counts * multiplier;

    Serial.println(cpm);
    counts = 0;
  }
```

Download the program and open the serial port window by clicking on the scope on the upper right corner.



```
15:58:57.255 -> 116
15:59:12.246 -> 36
15:59:27.265 -> 0
15:59:42.287 -> 48
15:59:57.304 -> 28
16:00:12.326 -> 88
16:00:27.349 -> 52
16:00:42.371 -> 52
16:00:57.361 -> 104
16:01:12.389 -> 172
16:01:27.417 -> 112
16:01:42.412 -> 112
16:01:57.439 -> 60
16:02:12.472 -> 76
16:02:27.465 -> 40
```

☑ Défilement automatique ☑ Afficher l'horodatage Nouvelle ligne 9600 baud

Then we'll get the radiation value displayed in CPM, counter per minutes which could be converted to uSv/h with the index 151(151CPM=1uSv/h).

Risk of radiation:

https://fr.search.yahoo.com/yhs/search?hspart=ddc&hsimp=yhs-linuxmint&type=__alt__ddc_linuxmint_com&p=dangerous+dose+of+radiation

https://www.reuters.com/article/us-how-much-radiation-dangerous-idUSTRE72E79Z20110315
People are exposed to natural radiation of 2-3 mSv a year.

* In a CT scan, the organ being studied typically receives a radiation dose of 15 mSv in an adult to 30 mSv in a newborn infant.

A typical chest X-ray involves exposure of about 0.02 mSv, while a dental one can be 0.01 mSv.

* Exposure to 100 mSv a year is the lowest level at which any increase in cancer risk is clearly evident. A cumulative 1,000 mSv (1 sievert) would probably cause a fatal cancer many years later in five out of every 100 persons exposed to it.

* There is documented evidence associating an accumulated dose of 90 mSv from two or three CT scans with an increased risk of cancer. The evidence is reasonably convincing for adults and very convincing for children.

* Large doses of radiation or acute radiation exposure destroys the central nervous system, red and white blood cells, which compromises the immune system, leaving the victim unable to fight off infections.

For example, a single one sievert (1,000 mSv) dose causes radiation sickness such as nausea, vomiting, hemorrhaging, but not death. A single dose of 5 sieverts would kill about half of those exposed to it within a month.

* Exposure to 350 mSv was the criterion for relocating people after the Chernobyl accident, according to the World Nuclear Association.

Water probe - device tutorial

In this tutorial you will learn how to assemble your own DIY water probe to measure conductivity, hence the degree of pollution of any liquid.

Introduction

The water probe is a relatively simple device. Its workings rely upon the fact that pure water does not actually carry an electric charge very well. So what we're really doing with this device is assessing the concentration of conductive particles that are floating in the (mostly nonconductive) water.

Water is very seldom just the sum of its basic chemical formula: two atoms of hydrogen and one of oxygen. Typically, water is a mixture that also includes other substances that have dissolved into it, including minerals, metals, and salts. In chemistry, water is the solvent, the other substances the solutes, and combined they make a solution.
Solutes create ions: atoms that carry an electric charge. These ions are what actually move electricity through water.
That's why measuring conductivity is a good way to learn how pure (really, how impure) a water sample may be: the more stuff that's dissolved in the watery solution, the faster electricity will move through it.

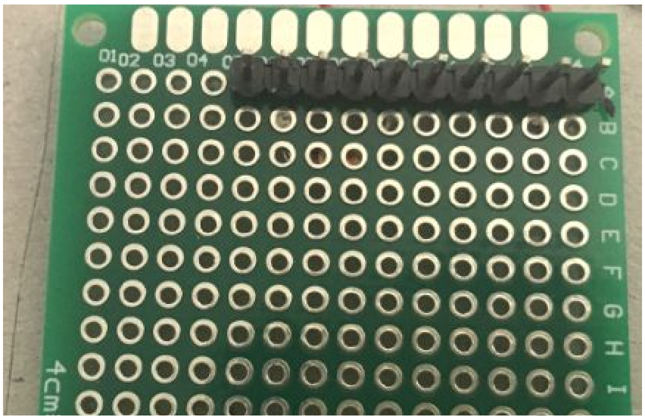
Assemble the probe

A video of the assembly process is available here.

List of parts

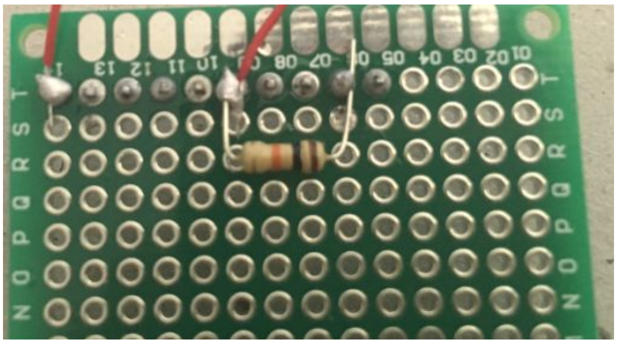
1x Arduino Uno board
1x 5x7cm PCB
1x Chassis mount binding post
Solid core wire
1x 10kOhm resistor
male headers strips for arduino

Solder a strip of male headers (about 10 pins) onto the PCB.

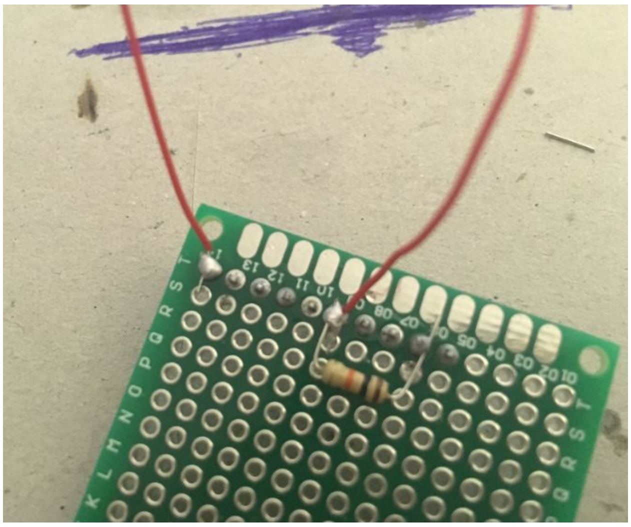


Beware that one pin needs to go into GND on the arduino board, another one into A5 and a third one into A0.

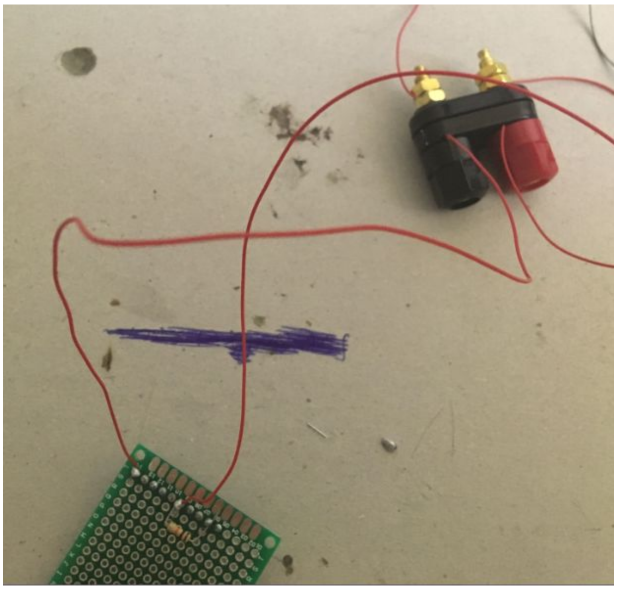
Grab the 10k Ohm resistor. Solder one end onto the header pin which goes into GND on the arduino board, the other end of the resistor onto the header pin which end on A0 in the arduino board. This way the resistor will basically create a bridge between GND and A0 on the arduino board.



Grab two pieces of solid core wire (about 30 cm long each) and strip both ends of each piece.
Solder one end of the first wire onto the header pin which ends in A5; solder one end of the second piece of wire onto the header pin which ends in A0 on the arduino board.

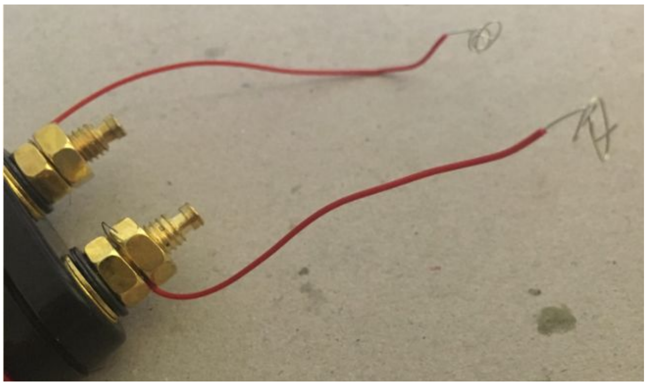


Connect the other ends of the pieces of solid core wire to the binding post. One end goes into the red part of the post, the other end goes into the black part of the binding post.

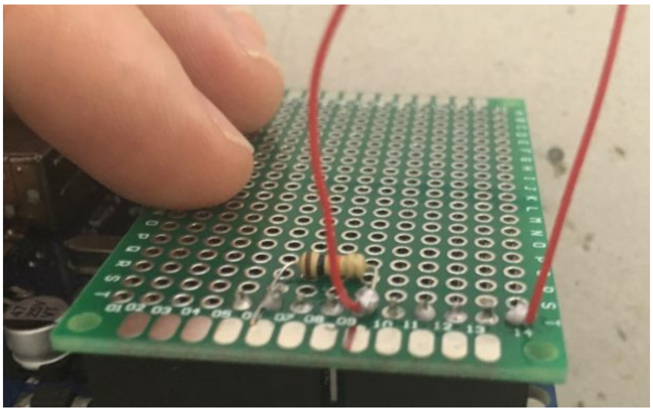


Now cut two pieces of solid core wire (about 10 cm long each), and strip both ends of each wire.

Connect one end of each piece of wire to the metal ends of the binding post. Use the bolts to secure the solid core wire in place. Curl the other ends.



Lastly, try placing the PCB on the arduino board, and make sure that one pin goes into GND, another into A0 and a third pin into A5.



Program the arduino board

To have a functioning water probe, you ll need to upload a specific program onto the arduino uno board.
Here's the sketch you need to upload:

```
/*
Water Conductivity Monitor
Sketch for an Arduino gadget that measures the electrical
conductivity of water.
This example code is based on example code that is in the public domain.
*/
const float ArduinoVoltage = 5.00; // CHANGE THIS FOR 3.3v Arduinos
const float ArduinoResolution = ArduinoVoltage / 1024;
const float resistorValue = 10000.0;
int threshold = 3;
int inputPin = A0;
int ouputPin = A5;
void setup()
{
```

```
Serial.begin(9600);
pinMode(ouputPin, OUTPUT);
pinMode(inputPin, INPUT);
}
void loop()
{
int analogValue=0;
int oldAnalogValue=1000;
float returnVoltage=0.0;
float resistance=0.0;
double Siemens;
float TDS=0.0;
while(((oldAnalogValue-analogValue)>threshold) || (oldAnalogValue<50))
{
oldAnalogValue = analogValue;
digitalWrite( ouputPin, HIGH );
delay(10); // allow ringing to stop
analogValue = analogRead( inputPin );
digitalWrite( ouputPin, LOW );
}
Serial.print("Return voltage = ");
returnVoltage = analogValue *ArduinoResolution;
Serial.print(returnVoltage);
Serial.println(" volts");
Serial.print("That works out to a resistance of ");
resistance = ((5.00 * resistorValue) / returnVoltage) - resistorValue;
Serial.print(resistance);
Serial.println(" Ohms.");
Serial.print("Which works out to a conductivity of ");
Siemens = 1.0/(resistance/1000000);
Serial.print(Siemens);
Serial.println(" microSiemens.");
Serial.print("Total Dissolved Solids are on the order of ");
TDS = 500 * (Siemens/1000);
Serial.print(TDS);
Serial.println(" PPM.");
if (returnVoltage>4.9) Serial.println("Are you sure this isn't metal?");
delay(5000);
}
```

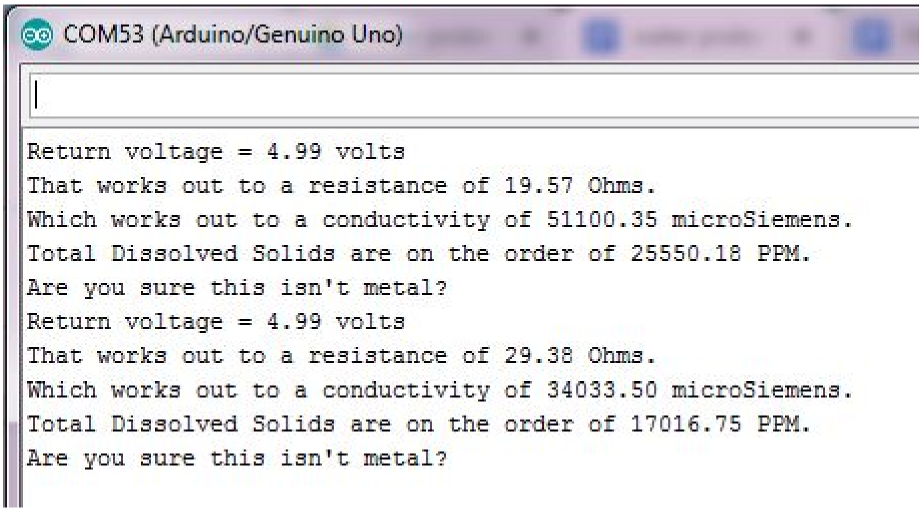
The complete code is also available here.

Using the water probe

After you've uploaded the code, dip the two curly ends of the water probe into a liquid and open the serial monitor.

You should be getting readings from the probe, which give you a rough idea of the resistance of the liquid, hence its conductivity.

You can easily test whether your probe is working properly, by just connecting the two curly ends to a piece of metal.
If the serial monitor returns the following message: “Are you sure this isn’t metal?”, you can be sure that the probe is giving you accurate readings.

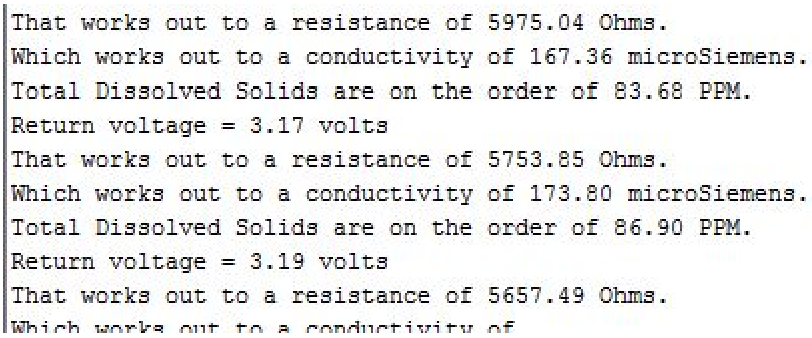


```
Return voltage = 4.99 volts
That works out to a resistance of 19.57 Ohms.
Which works out to a conductivity of 51100.35 microSiemens.
Total Dissolved Solids are on the order of 25550.18 PPM.
Are you sure this isn't metal?
Return voltage = 4.99 volts
That works out to a resistance of 29.38 Ohms.
Which works out to a conductivity of 34033.50 microSiemens.
Total Dissolved Solids are on the order of 17016.75 PPM.
Are you sure this isn't metal?
```

For tap water, you should be getting a conductivity of about 60 microSiemens.

Now try to add some washing up liquid to the water and see what readings you get.

This time, the conductivity of the liquid raises up to about 170 microSiemens.



```
That works out to a resistance of 5975.04 Ohms.
Which works out to a conductivity of 167.36 microSiemens.
Total Dissolved Solids are on the order of 83.68 PPM.
Return voltage = 3.17 volts
That works out to a resistance of 5753.85 Ohms.
Which works out to a conductivity of 173.80 microSiemens.
Total Dissolved Solids are on the order of 86.90 PPM.
Return voltage = 3.19 volts
That works out to a resistance of 5657.49 Ohms.
Which works out to a conductivity of
```

Water pollution

There is a straightforward connexion between water conductivity and water pollution. Since conductivity is an indication of the amount of foreign substances dissolved in water, it follows that the more conductive a liquid is, the more polluted it also is.

The consequences of water pollution are negative in many ways. One example is related to the concept surface tension.

Because of their polarity, water molecules are strongly attracted to one another, which gives water a high **surface tension**. The molecules at the surface of the water “stick together” to form a type of ‘skin’ on the water, strong enough to support very light objects. Insects that walk on water are taking advantage of this surface tension. Surface tension causes water to clump in drops rather than spreading out in a thin layer. It also allows water to move through plant roots and stems and the smallest blood vessels in your body – as one molecule moves up the tree root or through the capillary, it ‘pulls’ the others with it.

However, when foreign substances (ex. washing up liquid) are dissolved into water , this alters the surface tension of water altogether, causing a number of issues.

One experiment you can run at home will help illustrate surface tension and the consequences of polluting water.

Take a paper clip and delicately lower it onto a bowl full of water. The paper clip should then stay on the surface and float.

If, however, a single drop of washing up liquid or other chemical is introduced in the bowl of water, this will cause the paper clip to immediately sink.

The analogy here is between the paper clip and those insects that take advantage of the surface tension of water to walk on it. As foreign substances are introduced in a water reservoir (be this a lake, a stream, etc.) the surface tension is altered, and these insects will no longer be able to float on the surface. Ultimately this impacts on their lifecycle.

You can watch a video of this experiment here.

Water probe - 9-12 yo activity tutorial

In this tutorial you will learn how to use the DIY water probe.
You will find information about what the water probe is for, how it works, what experiments you can run with the water probe in order to understand what water pollution is.
Finally, you ll learn how to plot the data recorded by the water probe on a shared online map, so that your work will be visible to anyone.

Introduction

The conductivity meter is a relatively simple device. Its workings rely upon the fact that pure water does not actually carry an electric charge very well. So what we're really doing with this device is assessing the concentration of conductive particles that are floating in the (mostly nonconductive) water.

Water is very seldom just the sum of its basic chemical formula: two atoms of hydrogen and one of oxygen. Typically, water is a mixture that also includes other substances that have dissolved into it, including minerals, metals, and salts. In chemistry, water is the solvent, the other substances the solutes, and combined they make a solution.
Solutes create ions: atoms that carry an electric charge. These ions are what actually move electricity through water.
That's why measuring conductivity is a good way to learn how pure (really, how impure) a water sample may be: the more stuff that's dissolved in the watery solution, the faster electricity will move through it.

Program the arduino board

To have a functioning water probe, you ll need to upload a specific program onto the arduino uno board.
Here is the sketch you need to upload:

```
/*
Water Conductivity Monitor
Sketch for an Arduino gadget that measures the electrical
conductivity of water.
This example code is based on example code that is in the public domain.
*/
const float ArduinoVoltage = 5.00; // CHANGE THIS FOR 3.3v Arduinos
const float ArduinoResolution = ArduinoVoltage / 1024;
const float resistorValue = 10000.0;
int threshold = 3;
int inputPin = A0;
int ouputPin = A5;
void setup()
```

```
{
Serial.begin(9600);
pinMode(ouputPin, OUTPUT);
pinMode(inputPin, INPUT);
}
void loop()
{
int analogValue=0;
int oldAnalogValue=1000;
float returnVoltage=0.0;
float resistance=0.0;
double Siemens;
float TDS=0.0;
while(((oldAnalogValue-analogValue)>threshold) || (oldAnalogValue<50))
{
oldAnalogValue = analogValue;
digitalWrite( ouputPin, HIGH );
delay(10); // allow ringing to stop
analogValue = analogRead( inputPin );
digitalWrite( ouputPin, LOW );
}
Serial.print("Return voltage = ");
returnVoltage = analogValue *ArduinoResolution;
Serial.print(returnVoltage);
Serial.println(" volts");
Serial.print("That works out to a resistance of ");
resistance = ((5.00 * resistorValue) / returnVoltage) - resistorValue;
Serial.print(resistance);
Serial.println(" Ohms.");
Serial.print("Which works out to a conductivity of ");
Siemens = 1.0/(resistance/1000000);
Serial.print(Siemens);
Serial.println(" microSiemens.");
Serial.print("Total Dissolved Solids are on the order of ");
TDS = 500 * (Siemens/1000);
Serial.print(TDS);
Serial.println(" PPM.");
if (returnVoltage>4.9) Serial.println("Are you sure this isn't metal?");
delay(5000);
}
```

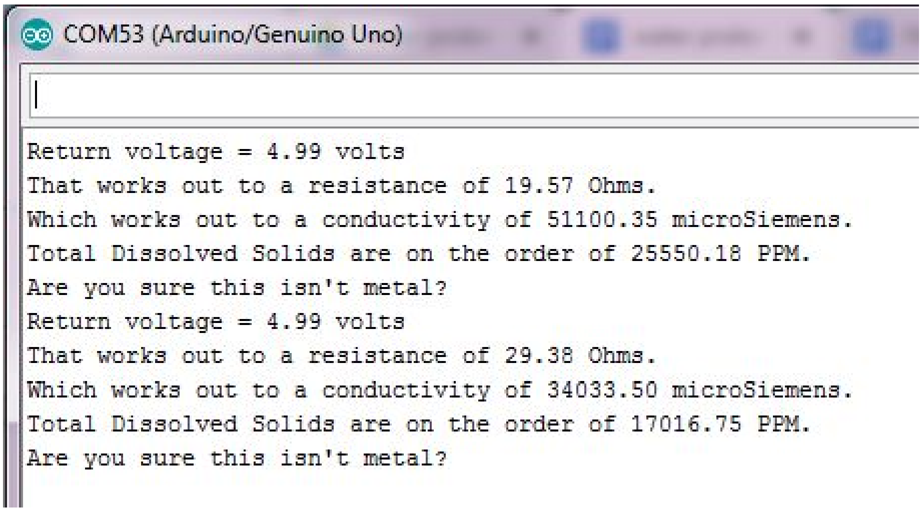
The complete code is also available here.

Using the water probe

After you've uploaded the code, dip the two curly ends of the water probe into a liquid and open the serial monitor.

You should be getting readings from the probe, which give you a rough idea of the resistance of the liquid, hence its conductivity.

You can easily test whether your probe is working properly, by just connecting the two curly ends to a piece of metal.
If the serial monitor returns the following message: “Are you sure this isn’t metal?”, you can be sure that the probe is giving you accurate readings.

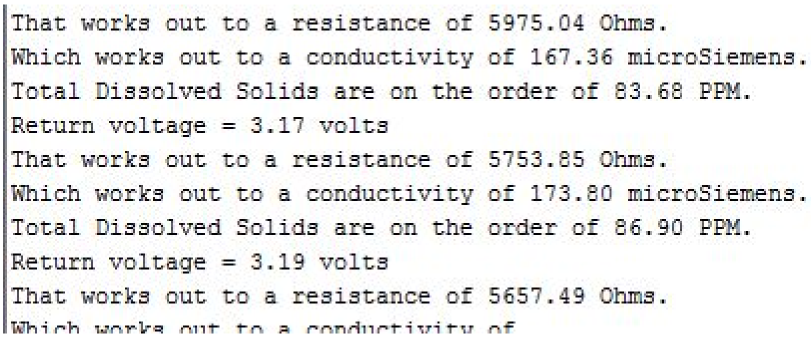


```
Return voltage = 4.99 volts
That works out to a resistance of 19.57 Ohms.
Which works out to a conductivity of 51100.35 microSiemens.
Total Dissolved Solids are on the order of 25550.18 PPM.
Are you sure this isn't metal?
Return voltage = 4.99 volts
That works out to a resistance of 29.38 Ohms.
Which works out to a conductivity of 34033.50 microSiemens.
Total Dissolved Solids are on the order of 17016.75 PPM.
Are you sure this isn't metal?
```

For tap water, you should be getting a conductivity of about 60 microSiemens.

Now try to add some washing up liquid to the water and see what readings you get.

This time, the conductivity of the liquid raises up to about 170 microSiemens.



```
That works out to a resistance of 5975.04 Ohms.
Which works out to a conductivity of 167.36 microSiemens.
Total Dissolved Solids are on the order of 83.68 PPM.
Return voltage = 3.17 volts
That works out to a resistance of 5753.85 Ohms.
Which works out to a conductivity of 173.80 microSiemens.
Total Dissolved Solids are on the order of 86.90 PPM.
Return voltage = 3.19 volts
That works out to a resistance of 5657.49 Ohms.
Which works out to a conductivity of
```

Water pollution

There is a straightforward connexion between water conductivity and water pollution. Since conductivity is an indication of the amount of foreign substances dissolved in water, it follows that the more conductive a liquid is, the more polluted it also is.

The consequences of water pollution are negative in many ways. One example is related to the concept surface tension.

Because of their polarity, water molecules are strongly attracted to one another, which gives water a high **surface tension**. The molecules at the surface of the water “stick together” to form a type of ‘skin’ on the water, strong enough to support very light objects. Insects that walk on water are taking advantage of this surface tension. Surface tension causes water to clump in drops rather than spreading out in a thin layer. It also allows water to move through plant roots and stems and the smallest blood vessels in your body – as one molecule moves up the tree root or through the capillary, it ‘pulls’ the others with it.

However, when foreign substances (ex. washing up liquid) are dissolved into water , this alters the surface tension of water altogether, causing a number of issues.

One experiment you can run at home will help illustrate surface tension and the consequences of polluting water.

Take a paper clip and delicately lower it onto a bowl full of water. The paper clip should then stay on the surface and float.

If, however, a single drop of washing up liquid or other chemical is introduced in the bowl of water, this will cause the paper clip to immediately sink.

The analogy here is between the paper clip and those insects that take advantage of the surface tension of water to walk on it. As foreign substances are introduced in a water reservoir (be this a lake, a stream, etc.) the surface tension is altered, and these insects will no longer be able to float on the surface. Ultimately this impacts on their lifecycle.

You can watch a video of this experiment here.

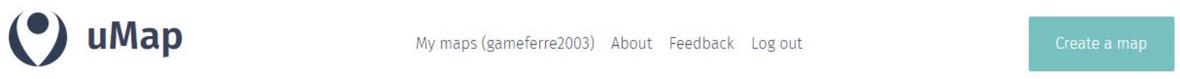
Plotting data on a shared online map

You will be sharing the data recorded by your water probe on UMap, which allows users to create their own maps embedding the data of their choice.

First, head to https://umap.openstreetmap.fr/fr/

You ll need to create an account to be able to edit your own maps.

Once you have successfully logged in, hit the “Create a map” button

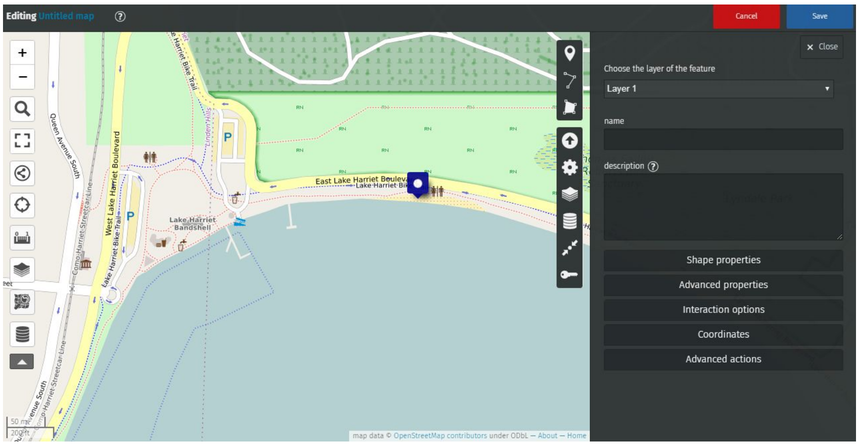


uMap

My maps (gameferre2003) About Feedback Log out

Create a map

You can now draw markers and enter the description you wish to publish.



Editing Untitled map

Cancel

Save

× Close

Choose the layer of the feature

Layer 1

name

description

Shape properties

Advanced properties

Interaction options

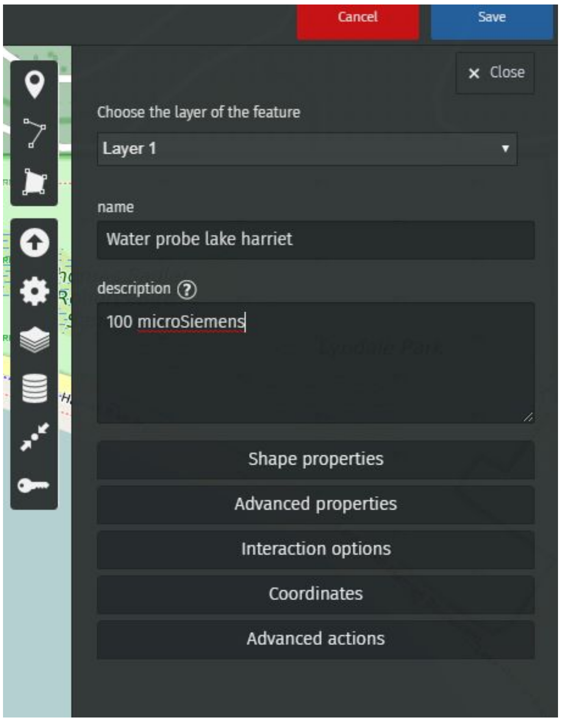
Coordinates

Advanced actions

East Lake Harriet Boulevard
Lake Harriet Bandshell
West Lake Harriet Boulevard
Queen Avenue South
map data © OpenStreetMap contributors under ODbL — About — Home

For example, let’s enter the value recorded by our water probe at Lake Harriet, in Minneapolis.

We’ve recorded 100 microsiemens, which by the way means that it is safe to bathe in Lake Harriet!



Cancel

Save

× Close

Choose the layer of the feature

Layer 1

name

Water probe lake harriet

description

100 microSiemens

Shape properties

Advanced properties

Interaction options

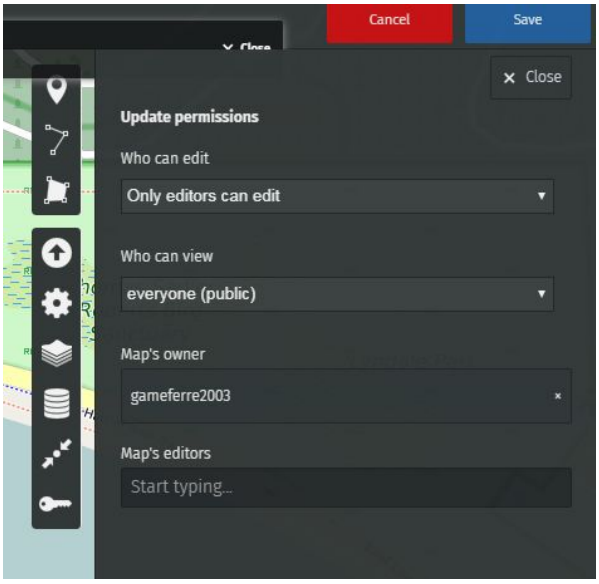
Coordinates

Advanced actions

Co-funded by the Erasmus+ Programme of the European Union

To share this map with anyone, you just need to provide them with the appropriate link.

Go to Update permissions and editors



Update permissions

Who can edit: Only editors can edit

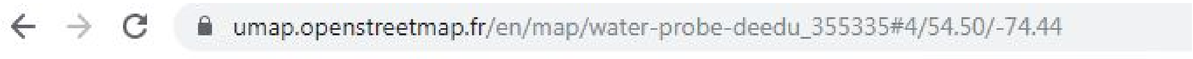
Who can view: everyone (public)

Map's owner: gameferre2003

Map's editors: Start typing...

You can choose who can view and who can edit this map.

To enable editors to edit the map, copy the link of your map (the web address of your map) and share it with whoever you want.



umap.openstreetmap.fr/en/map/water-probe-deedu_355335#4/54.50/-74.44

Water probe - 12+ yo activity tutorial

In this tutorial you will learn how to build your own DIY water probe.
You will be able to use this item to analyse water sample and determine their degree of purity.
Follow this tutorial and you ll also learn how to take readings with your water probe and how the probe works.
In the last section, you ll learn how to plot the values recorded by your water probe on a shared online map, and launch a citizen science project on zooniverse.com to communicate about your findings as well as to engage other people in your research.

Introduction

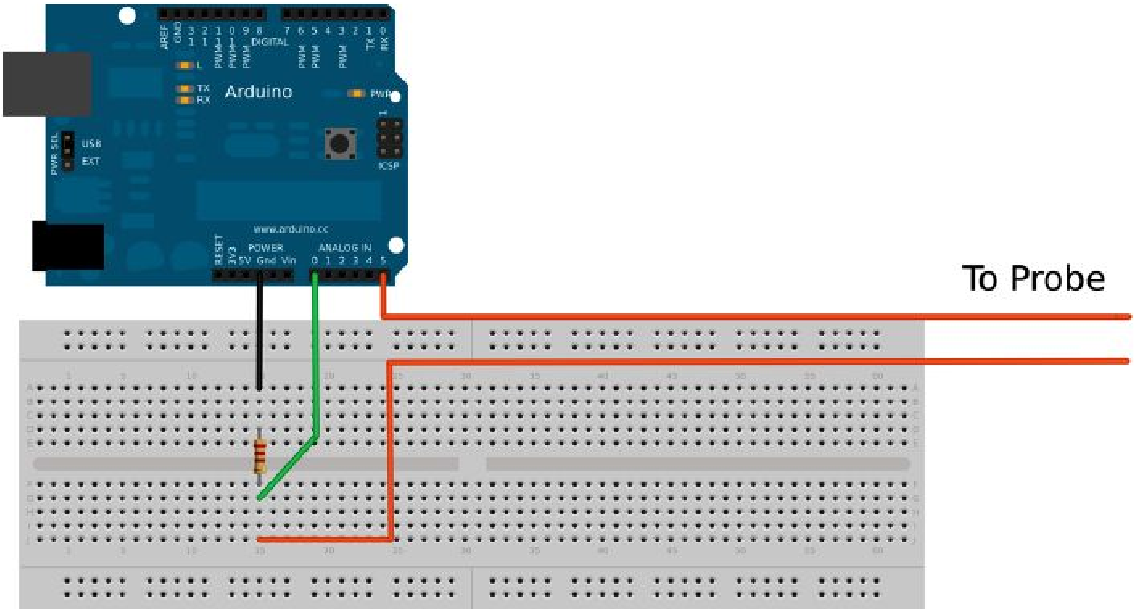
The conductivity meter is a relatively simple device. Its workings rely upon the fact that pure water does not actually carry an electric charge very well. So what we're really doing with this device is assessing the concentration of conductive particles that are floating in the (mostly nonconductive) water.

Water is very seldom just the sum of its basic chemical formula: two atoms of hydrogen and one of oxygen. Typically, water is a mixture that also includes other substances that have dissolved into it, including minerals, metals, and salts. In chemistry, water is the solvent, the other substances the solutes, and combined they make a solution.
Solutes create ions: atoms that carry an electric charge. These ions are what actually move electricity through water.
That's why measuring conductivity is a good way to learn how pure (really, how impure) a water sample may be: the more stuff that's dissolved in the watery solution, the faster electricity will move through it.

Step by step instructions

Here's a timelapse of the assembling process.

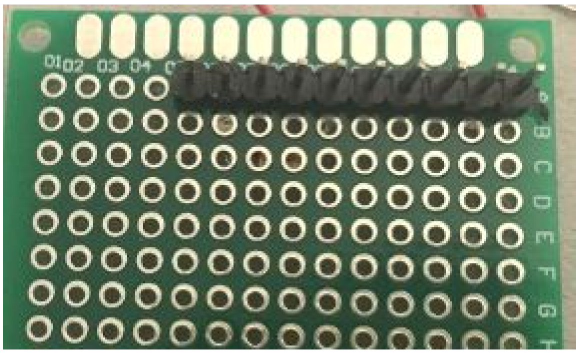
Here's the wiring diagram:



List of parts

1x arduino Uno board
1x 40x60mm PCB
electric wire
1x chassis mount dual female binding post
1x 10Kohm resistor
1 male pin header strip for arduino

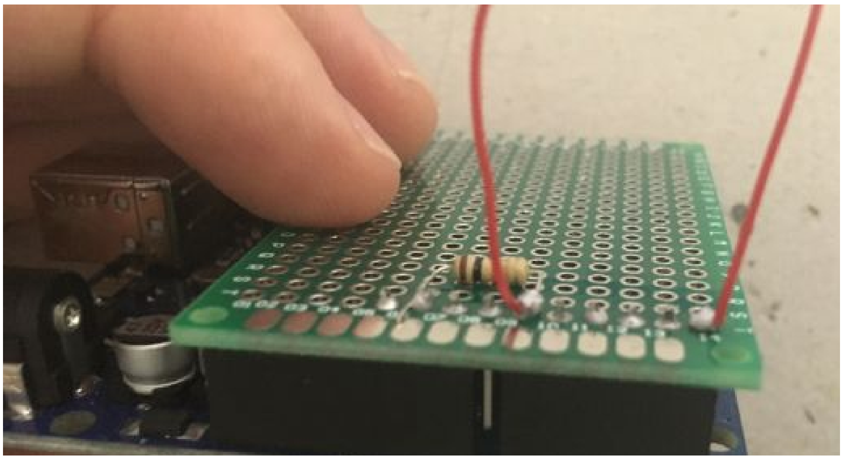
Begin by soldering the male pin header strip onto the PCB board. You need to cover at least 10 points of the PCB (see figure below).



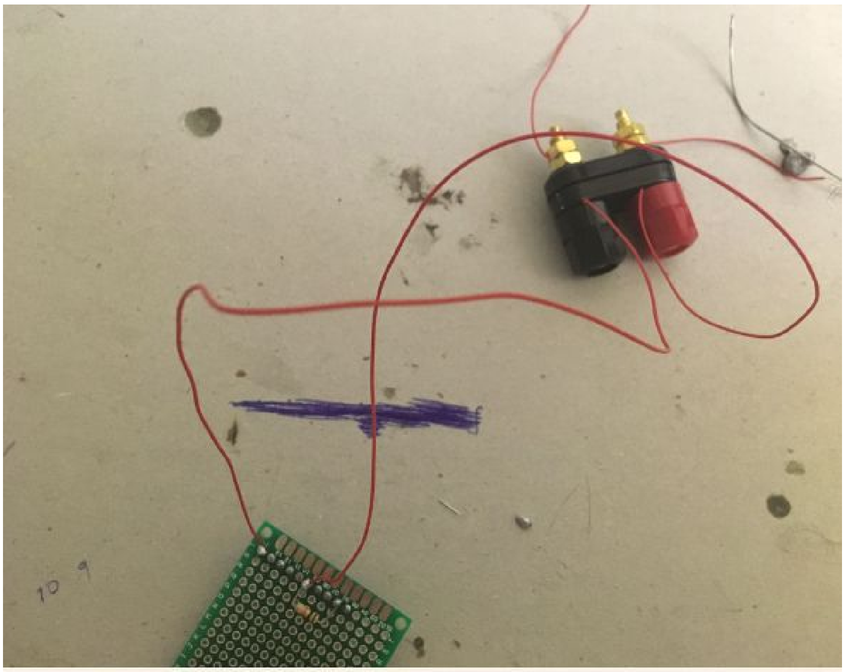
Next, solder the 1kohm resistor to the PCB. We soldered one end to n°6, the other end to n°9. This way, when the PCB will be mounted on the arduino uno board, one end of the resistor will connect to A0, the other one to GND.

Next, cut two pieces of electric wire, each about 20 cm long. The length doesn't really matter, however, longer wires will enable you to distance the arduino board further away from liquids.

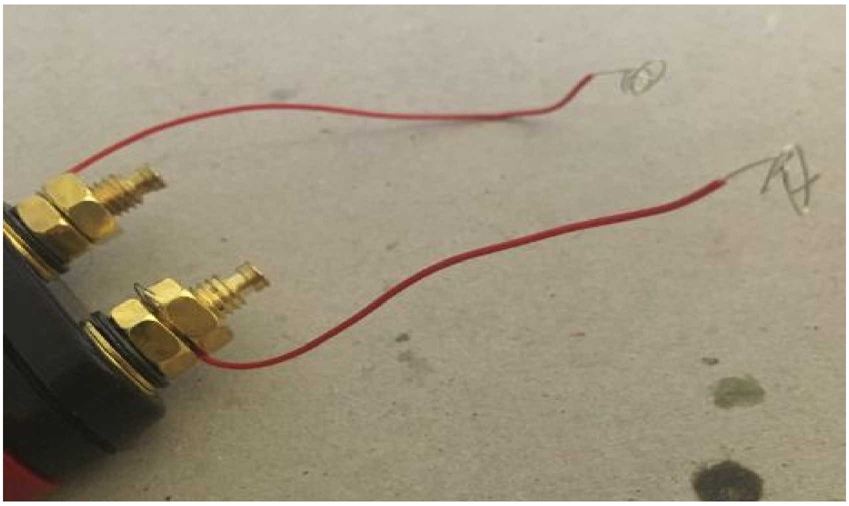
Solder one end of one wire to one end of the resistor. This will connect to A0 on the board. Solder the other end to n° on the PCB. The latter will then connect with A5 on the arduino uno board.



Strip off the other ends of the electric wire you just soldered onto the PCB, and connect them to the top of the binding post.



Now cut two short pieces of electric wire (about 10 cm long each), and strip off both ends of each wire. Connect one end of each to the bottom part of the binding post. Curl the other end of each wire. The curly ends are the one which will dive into the liquid.



Program the arduino board

To have a functioning water probe, you ll need to upload a specific program onto the arduino uno board.
Here is the sketch you need to upload:

```
/*
Water Conductivity Monitor
Sketch for an Arduino gadget that measures the electrical
conductivity of water.
This example code is based on example code that is in the public domain.
*/
const float ArduinoVoltage = 5.00; // CHANGE THIS FOR 3.3v Arduinos
const float ArduinoResolution = ArduinoVoltage / 1024;
const float resistorValue = 10000.0;
int threshold = 3;
int inputPin = A0;
int ouputPin = A5;
void setup()
{
Serial.begin(9600);
pinMode(ouputPin, OUTPUT);
pinMode(inputPin, INPUT);
}
void loop()
{
int analogValue=0;
int oldAnalogValue=1000;
float returnVoltage=0.0;
float resistance=0.0;
double Siemens;
float TDS=0.0;
while(((oldAnalogValue-analogValue)>threshold) || (oldAnalogValue<50))
{
oldAnalogValue = analogValue;
digitalWrite( ouputPin, HIGH );
delay(10); // allow ringing to stop
analogValue = analogRead( inputPin );
digitalWrite( ouputPin, LOW );
}
Serial.print("Return voltage = ");
returnVoltage = analogValue *ArduinoResolution;
Serial.print(returnVoltage);
Serial.println(" volts");
Serial.print("That works out to a resistance of ");
resistance = ((5.00 * resistorValue) / returnVoltage) - resistorValue;
Serial.print(resistance);
Serial.println(" Ohms.");
Serial.print("Which works out to a conductivity of ");
Siemens = 1.0/(resistance/1000000);
Serial.print(Siemens);
Serial.println(" microSiemens.");
Serial.print("Total Dissolved Solids are on the order of ");
TDS = 500 * (Siemens/1000);
Serial.print(TDS);
Serial.println(" PPM.");
if (returnVoltage>4.9) Serial.println("Are you sure this isn't metal?");
delay(5000);
}
```

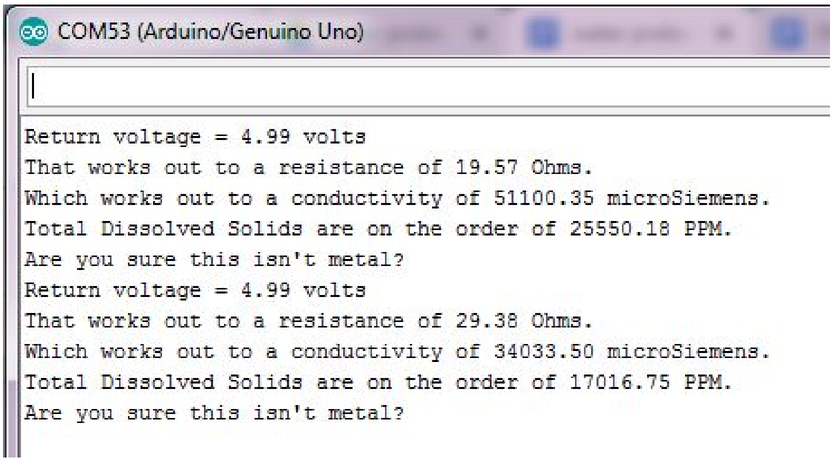
The complete code s also available here.

Using the water probe

After you've uploaded the code, dip the two curly ends of the water probe into a liquid and open the serial monitor.

You should be getting readings from the probe, which give you a rough idea of the resistance of the liquid, hence its conductivity.

You can easily test whether your probe is working properly, by just connecting the two curly ends to a piece of metal.
If the serial monitor returns the following message: “Are you sure this isn’t metal?”, you can be sure that the probe is giving you accurate readings.

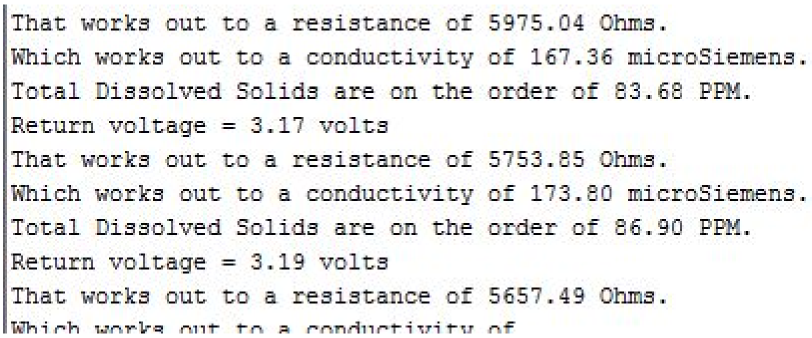


```
Return voltage = 4.99 volts
That works out to a resistance of 19.57 Ohms.
Which works out to a conductivity of 51100.35 microSiemens.
Total Dissolved Solids are on the order of 25550.18 PPM.
Are you sure this isn't metal?
Return voltage = 4.99 volts
That works out to a resistance of 29.38 Ohms.
Which works out to a conductivity of 34033.50 microSiemens.
Total Dissolved Solids are on the order of 17016.75 PPM.
Are you sure this isn't metal?
```

For tap water, you should be getting a conductivity of about 60 microSiemens.

Now try to add some washing up liquid to the water and see what readings you get.

This time, the conductivity of the liquid raises up to about 170 microSiemens.



```
That works out to a resistance of 5975.04 Ohms.
Which works out to a conductivity of 167.36 microSiemens.
Total Dissolved Solids are on the order of 83.68 PPM.
Return voltage = 3.17 volts
That works out to a resistance of 5753.85 Ohms.
Which works out to a conductivity of 173.80 microSiemens.
Total Dissolved Solids are on the order of 86.90 PPM.
Return voltage = 3.19 volts
That works out to a resistance of 5657.49 Ohms.
Which works out to a conductivity of
```

Water pollution

There is a straightforward connexion between water conductivity and water pollution. Since conductivity is an indication of the amount of foreign substances dissolved in water, it follows that the more conductive a liquid is, the more polluted it also is.

The consequences of water pollution are negative in many ways. One example is related to the concept surface tension.

Because of their polarity, water molecules are strongly attracted to one another, which gives water a high **surface tension**. The molecules at the surface of the water “stick together” to form a type of ‘skin’ on the water, strong enough to support very light objects. Insects that walk on water are taking advantage of this surface tension. Surface tension causes water to clump in drops rather than spreading out in a thin layer. It also allows water to move through plant roots and stems and the smallest blood vessels in your body – as one molecule moves up the tree root or through the capillary, it ‘pulls’ the others with it.

However, when foreign substances (ex. washing up liquid) are dissolved into water , this alters the surface tension of water altogether, causing a number of issues.

One experiment you can run at home will help illustrate surface tension and the consequences of polluting water.

Take a paper clip and delicately lower it onto a bowl full of water. The paper clip should then stay on the surface and float.

If, however, a single drop of washing up liquid or other chemical is introduced in the bowl of water, this will cause the paper clip to immediately sink.

The analogy here is between the paper clip and those insects that take advantage of the surface tension of water to walk on it. As foreign substances are introduced in a water reservoir (be this a lake, a stream, etc.) the surface tension is altered, and these insects will no longer be able to float on the surface. Ultimately this impacts on their lifecycle.

You can watch a video of this experiment here.

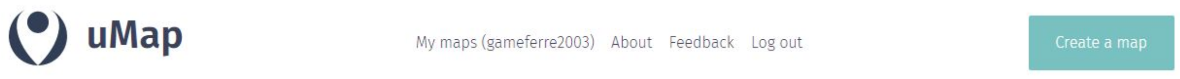
Plotting data on a shared online map

You will be sharing the data recorded by your water probe on UMap, which allows users to create their own maps embedding the data of their choice.

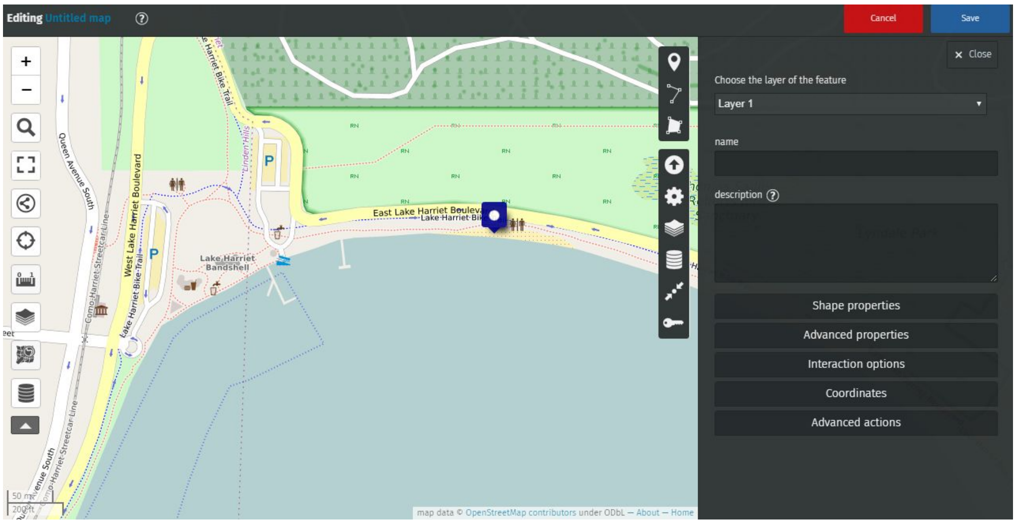
First, head to https://umap.openstreetmap.fr/fr/

You ll need to create an account to be able to edit your own maps.

Once you ve successfully logged in, hit the “Create a map” button.



You can now draw markers and enter the description you wish to publish.



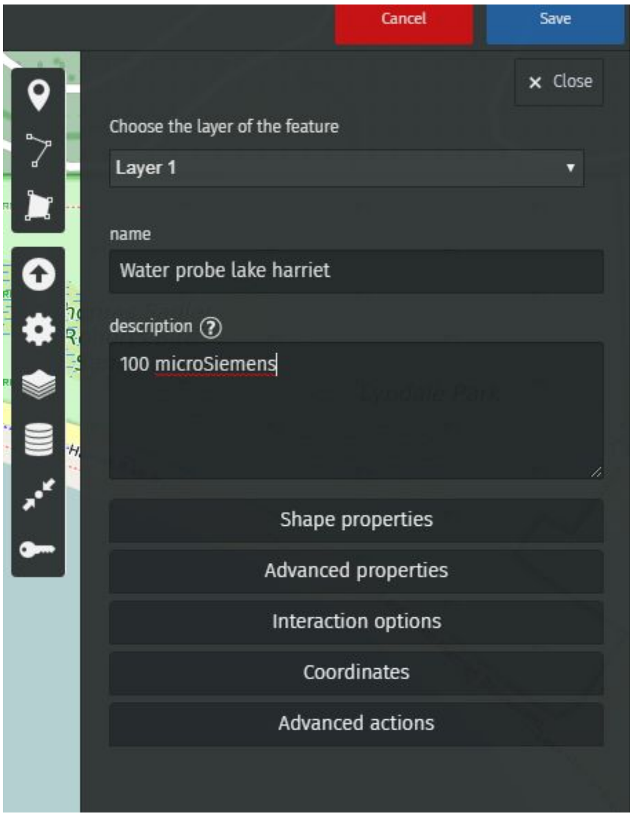
Choose the layer of the feature: Layer 1

name

description

For example, let's enter the value recorded by our water probe at Lake Harriet, in Minneapolis.

We've recorded 100 microsiemens, which by the way means that it is safe to bathe in Lake Harriet!



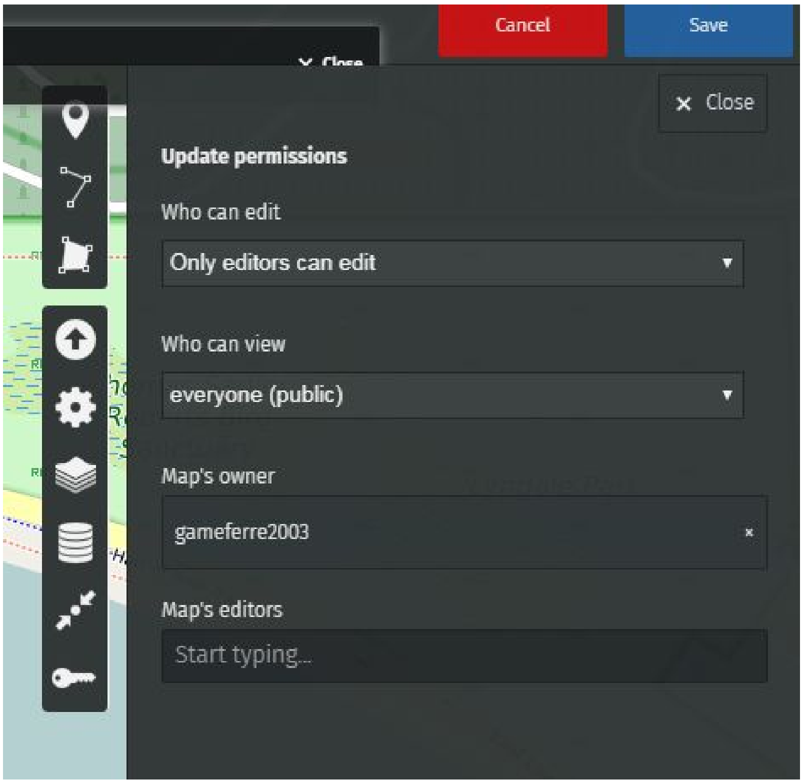
Choose the layer of the feature: Layer 1

name: Water probe lake harriet

description: 100 microSiemens

To share this map with anyone, you just need to provide them with the appropriate link.

Go to Update permissions and editors



Update permissions

Who can edit: Only editors can edit

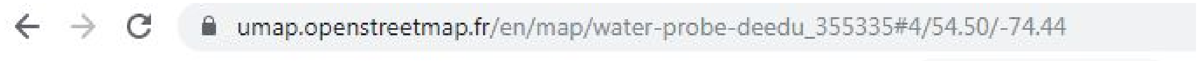
Who can view: everyone (public)

Map's owner: gameferre2003

Map's editors: Start typing...

You can choose who can view and who can edit this map.

To enable editors to edit the map, copy the link of your map (the web address of your map) and share it with whoever you want.



umap.openstreetmap.fr/en/map/water-probe-deedu_355335#4/54.50/-74.44

Launching a citizen science project

Citizen science is the involvement of the public in scientific research – whether community-driven research or global investigations.

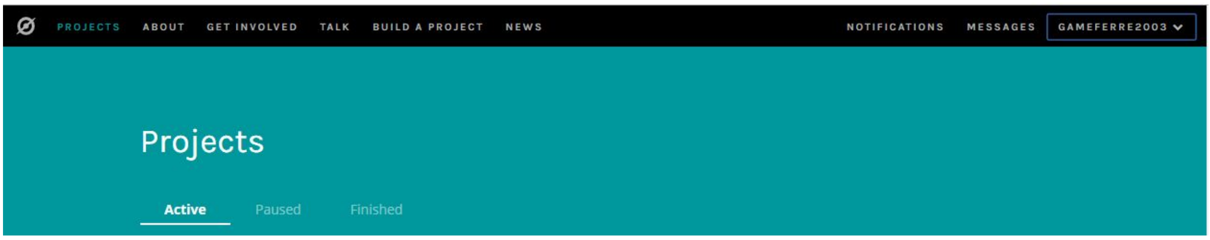
Citizen scientists can help trained scientists gather data from all over the world — even from space. They can provide new ideas and new ways of thinking.

Kids often make great citizen scientists because they tend to be curious and good at following precise directions. Sometimes they're even better at these things than adults.

You will be creating a project on zooniverse.com, anyone will be able to view and contribute to the project.

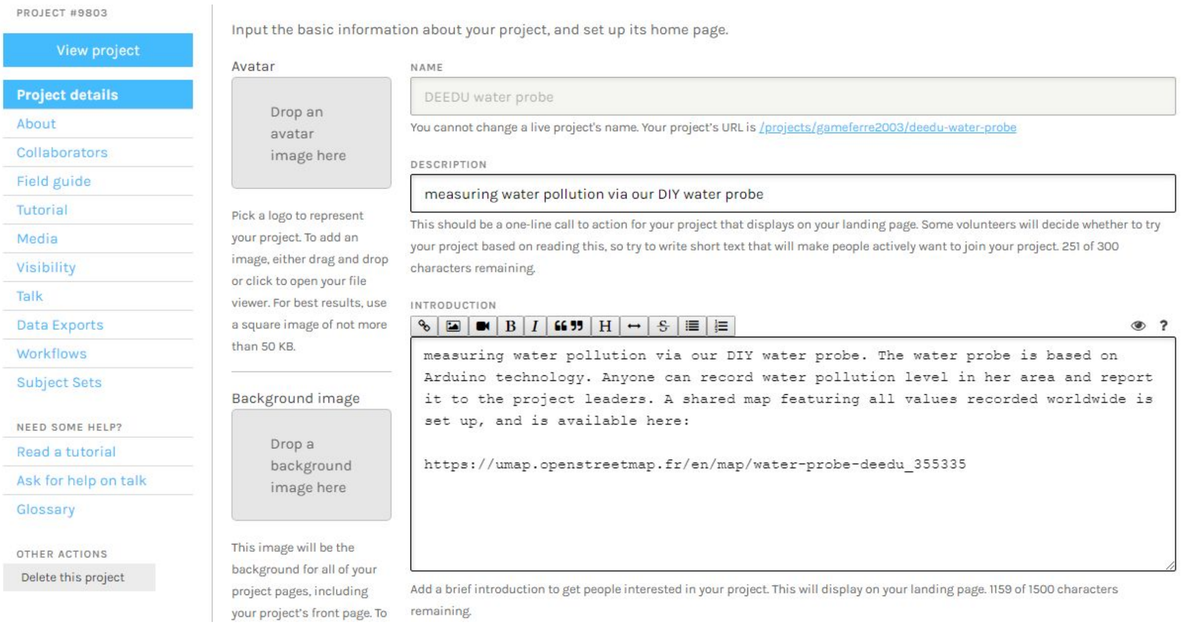
First, head to zooniverse.org and create an account.

Once you’ve successfully logged in, create a new project by clicking on build a project



From the dashboard, you can edit your project and enter as much information as you wish.

Don’t forget to embed a link to the shared map in the description of the project.



Input the basic information about your project, and set up its home page.

Avatar

Pick a logo to represent your project. To add an image, either drag and drop or click to open your file viewer. For best results, use a square image of not more than 50 KB.

Background image

This image will be the background for all of your project pages, including your project's front page. To

NAME

DEEDU water probe

You cannot change a live project's name. Your project's URL is /projects/gameferre2003/deedu-water-probe

DESCRIPTION

measuring water pollution via our DIY water probe

This should be a one-line call to action for your project that displays on your landing page. Some volunteers will decide whether to try your project based on reading this, so try to write short text that will make people actively want to join your project. 251 of 300 characters remaining.

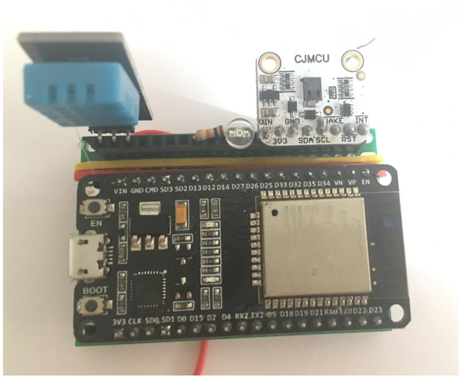
INTRODUCTION

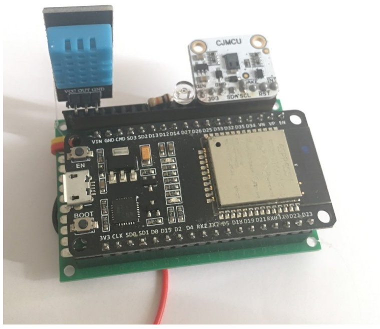
```
measuring water pollution via our DIY water probe. The water probe is based on
Arduino technology. Anyone can record water pollution level in her area and report
it to the project leaders. A shared map featuring all values recorded worldwide is
set up, and is available here:

https://umap.openstreetmap.fr/en/map/water-probe-deedu_355335
```

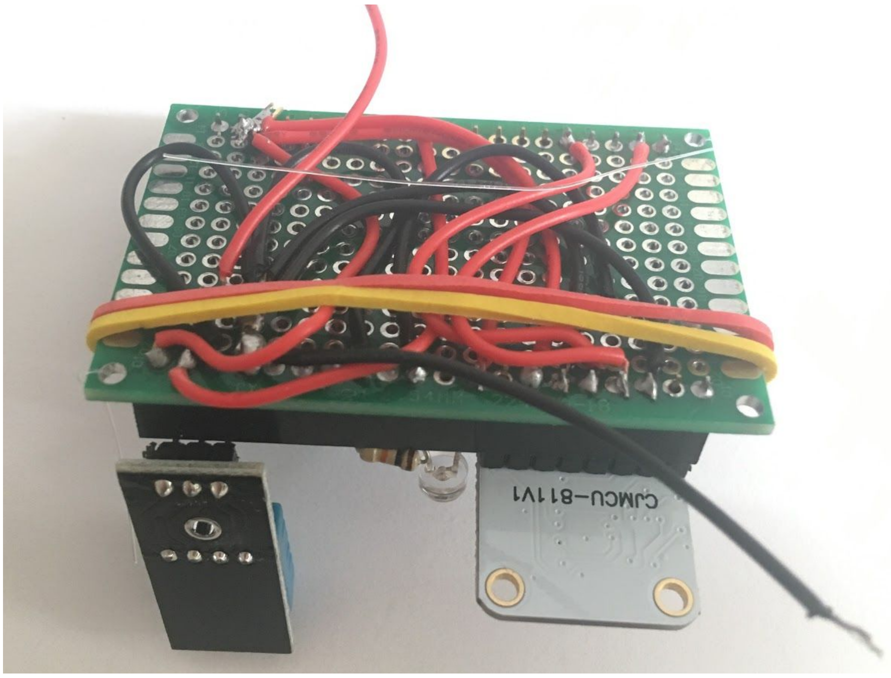
Add a brief introduction to get people interested in your project. This will display on your landing page. 1159 of 1500 characters remaining.

Connected weather station - device tutorial

In this tutorial you will learn how to set up a weather station based on ESP32, and how to monitor its readings remotely, via the Blynk app as well as via a website.







Setting up the weather station

A timelapse of the assembling process is available here.

List of parts

1x ESP32 board + usb power cable
1x DHT11 sensor
1x light sensor
1x 10 Kohm resistor
1x CJMCU CCS811 sensor

several jumper wires
a number of breadboards or a PCB (if you decide to do some soldering)
female headers (if you decide to do the soldering).

Wiring up the components

Connect the sensors as follows:

Light sensor

One end to 3V
the other end to the 10kohm resistor which in turn is connected to GND. the same end of the LDR is also connected to pin D34 on the ESP32

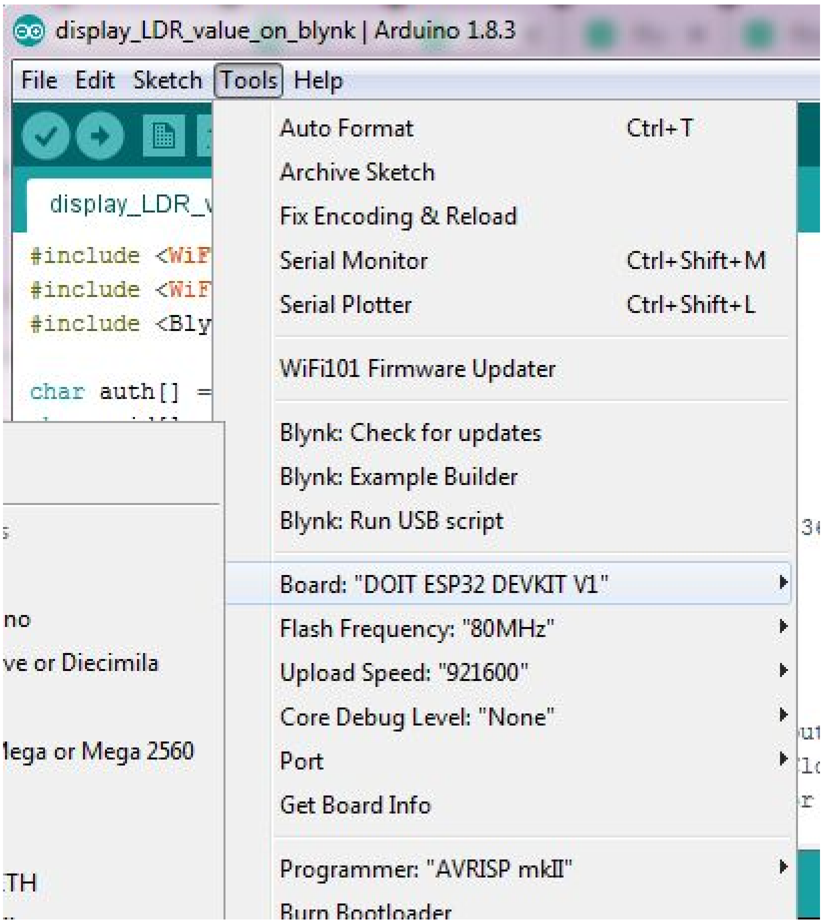
CJMCU CCS811
3V → 3V on the ESP32 board
GND → GND
SDA → D21 pin on ESP32
SCL → D22 pin on ESP32
WAKE → GND

DHT11
GND → GND on ESP32
VCC → 3V on ESP32
OUT → D34 on ESP32

Program the ESP32 board

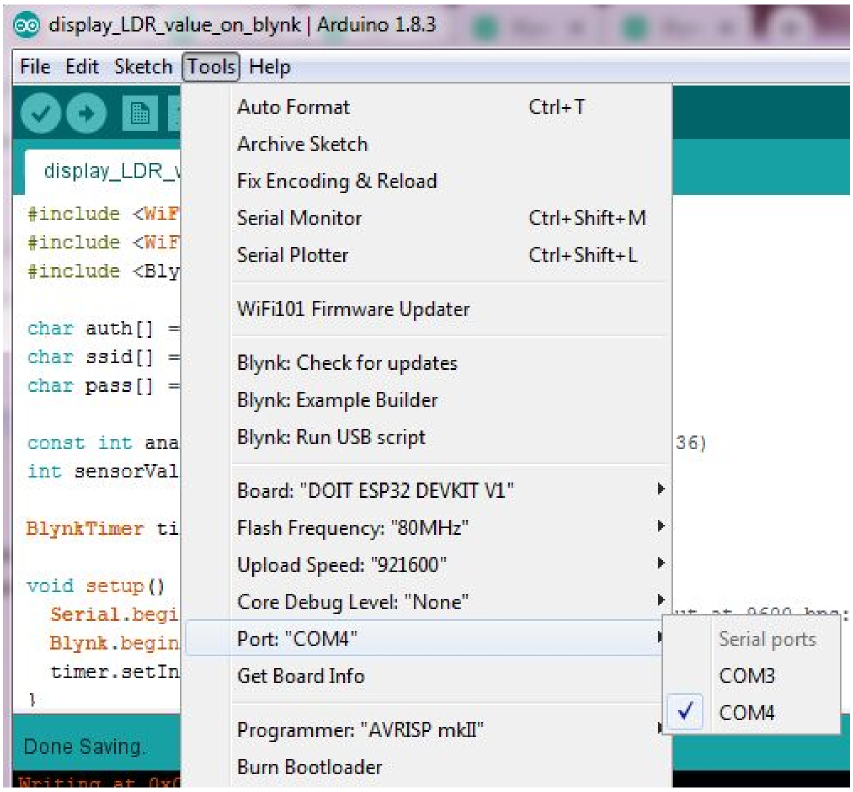
Launch Arduino IDE.

Select your ESP32 board from the Tools menu.



Board: "DOIT ESP32 DEVKIT V1"
Flash Frequency: "80MHz"
Upload Speed: "921600"
Core Debug Level: "None"
Programmer: "AVRISP mkII"

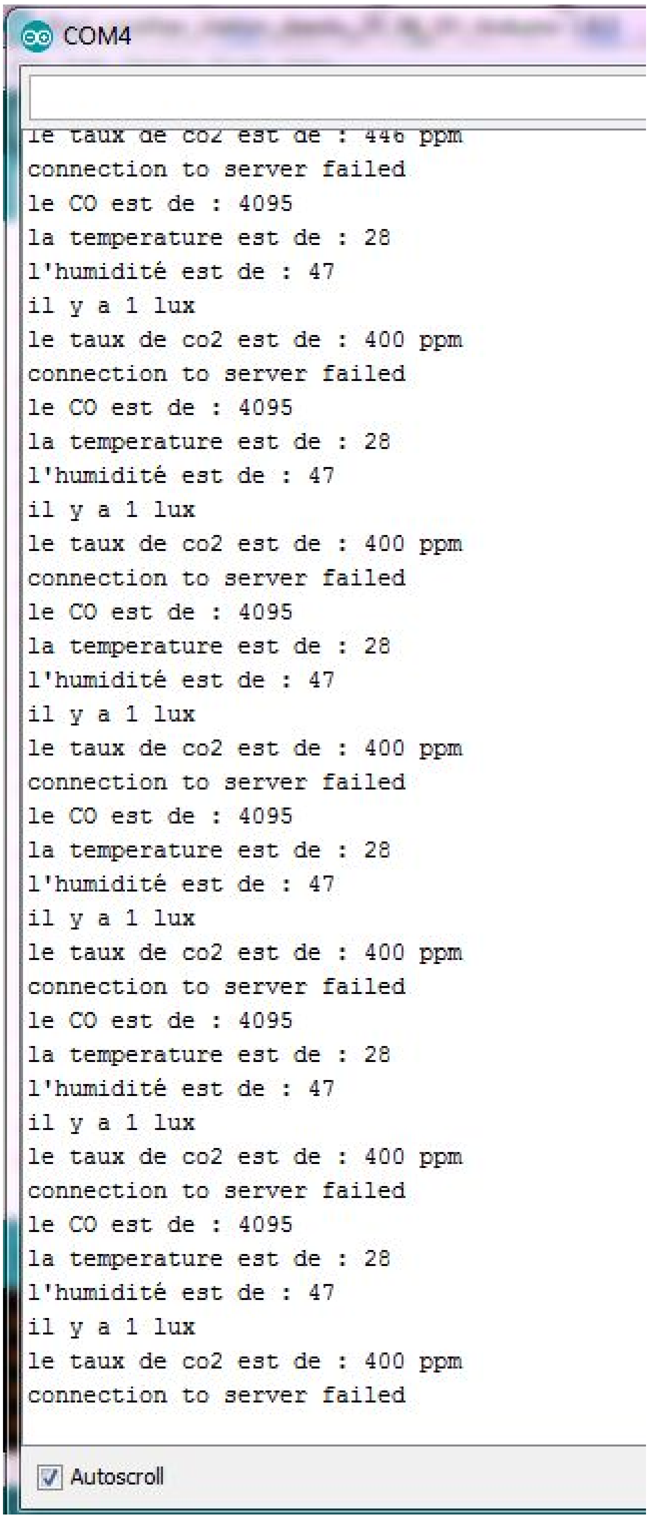
Make sure you've also selected the correct Port.



Board: "DOIT ESP32 DEVKIT V1"
Flash Frequency: "80MHz"
Upload Speed: "921600"
Core Debug Level: "None"
Port: "COM4"
Programmer: "AVRISP mkII"

Upload this code on to the board.

If you open the serial monitor (set the baud rate to 9600), you should be able to view the values recorded by the different sensors.



```
le taux de co2 est de : 446 ppm
connection to server failed
le CO est de : 4095
la temperature est de : 28
l'humidité est de : 47
il y a 1 lux
le taux de co2 est de : 400 ppm
connection to server failed
le CO est de : 4095
la temperature est de : 28
l'humidité est de : 47
il y a 1 lux
le taux de co2 est de : 400 ppm
connection to server failed
le CO est de : 4095
la temperature est de : 28
l'humidité est de : 47
il y a 1 lux
le taux de co2 est de : 400 ppm
connection to server failed
le CO est de : 4095
la temperature est de : 28
l'humidité est de : 47
il y a 1 lux
le taux de co2 est de : 400 ppm
connection to server failed
le CO est de : 4095
la temperature est de : 28
l'humidité est de : 47
il y a 1 lux
le taux de co2 est de : 400 ppm
connection to server failed
le CO est de : 4095
la temperature est de : 28
l'humidité est de : 47
il y a 1 lux
le taux de co2 est de : 400 ppm
connection to server failed
```

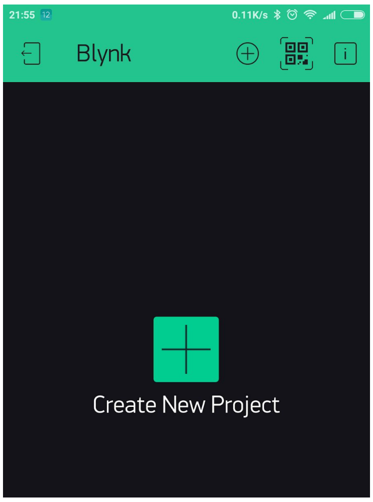
☑ Autoscroll

Monitor the weather station remotely via Blynk App

Blynk app enables us to monitor the values recorded by the weather station remotely, directly on our smartphone, regardless our distance from the weather station.
All we need is the blynk app and an internet connection. In this project you'll learn how to monitor the values recorded by the LDR sensor and the DHT11 sensor only.

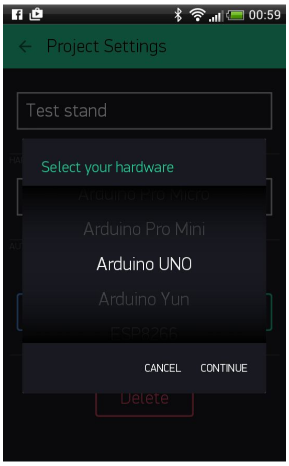
Create a new project in the Blynk app

After you've download the app and you've successfully logged into your account, start by creating a new project.



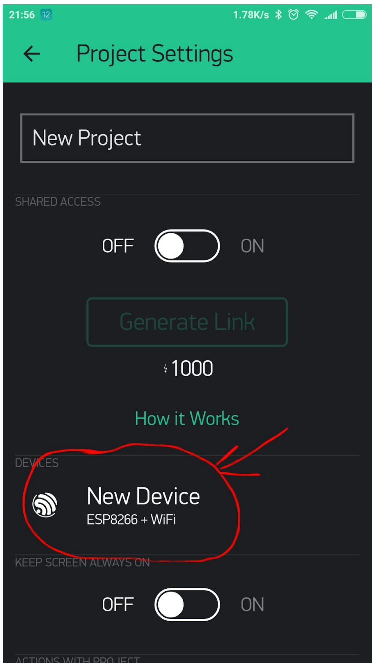
Choose Your Hardware

Select the hardware model you will use. If you are following this tutorial you ll probably be using an ESP32 board.



Auth Token

Auth Token is a unique identifier which is needed to connect your hardware to your smartphone. Every new project you create will have its own Auth Token. You'll get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device:



21:56 1.78K/s

Project Settings

New Project

SHARED ACCESS

OFF ON

Generate Link

1000

How it Works

DEVICES

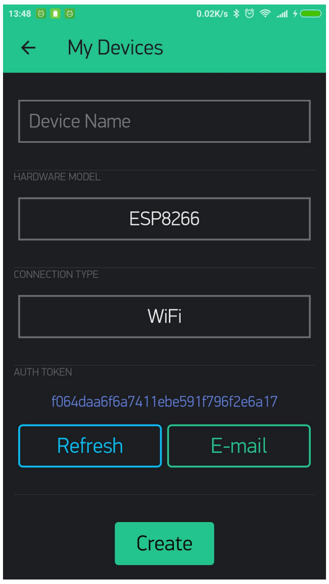
New Device
ESP8266 + WiFi

KEEP SCREEN ALWAYS ON

OFF ON

ACTIONS WITH PROJECT

And you'll see token :



13:48 0.02K/s

My Devices

Device Name

HARDWARE MODEL

ESP8266

CONNECTION TYPE

WiFi

AUTH TOKEN

f064daa6f6a7411ebe591f796f2e6a17

Refresh E-mail

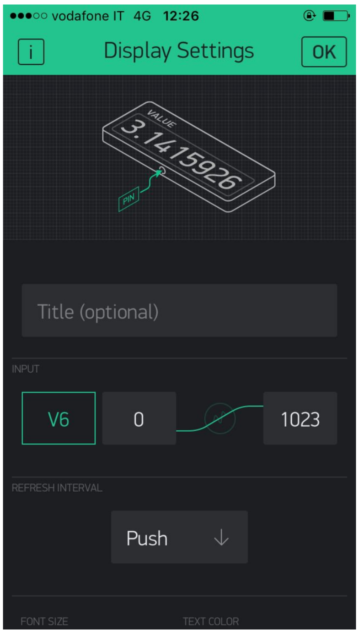
Create

Set up the Blynk app

You will need to create a project on blynk which is tailored for the purpose of monitoring the parameters recorded by the weather station.

Grab 3 value display widgets.

Configure them one by one. The first will receive V6 as input, the second V5 and the third V0. You ll notice that they re all set to push mode.



●●●○○ vodafone IT 4G 12:26

Display Settings

Title (optional)

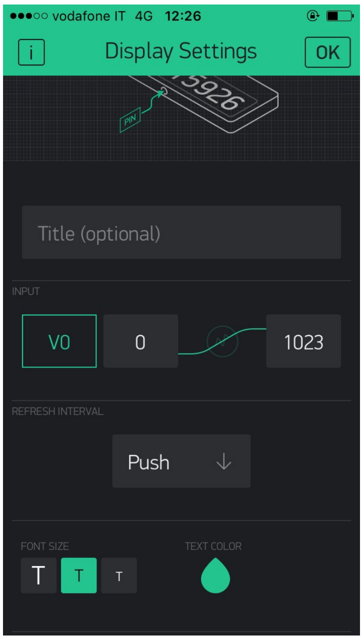
INPUT

V6 0 1023

REFRESH INTERVAL

Push

FONT SIZE TEXT COLOR



●●●○○ vodafone IT 4G 12:26

Display Settings

Title (optional)

INPUT

V0 0 1023

REFRESH INTERVAL

Push

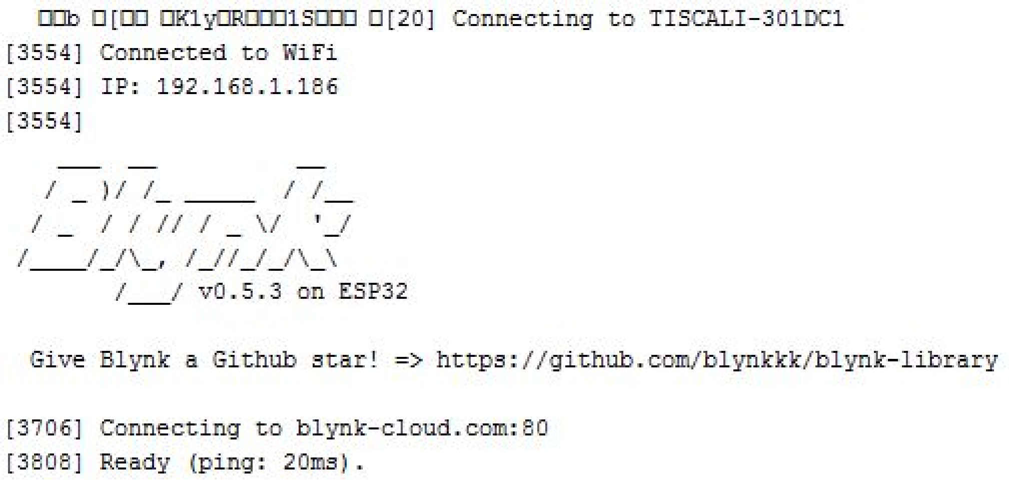
FONT SIZE

TEXT COLOR

Program the ESP32 board

Launch arduino IDE and open this program.

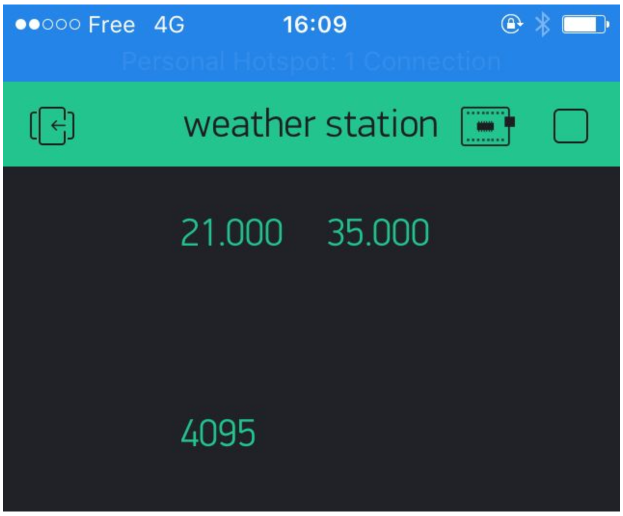
Select the board you re using from the Tools menu, as well as the correct Port.
Upload the code.
If upload is successful, you should be able to see a message from Blynk on the serial monitor.



```
  □□b □[□□ □K1y□R□□□1S□□□ □[20] Connecting to TISCALI-301DC1
[3554] Connected to WiFi
[3554] IP: 192.168.1.186
[3554]
    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/ v0.5.3 on ESP32

  Give Blynk a Github star! => https://github.com/blynkkk/blynk-library

[3706] Connecting to blynk-cloud.com:80
[3808] Ready (ping: 20ms).
```

Connected weather station - 9-12 yo activity tutorial

In this activity, participants will set up their weather station, sent it up in the air, and monitor the recordings (light, temperature, humidity) in real time via the Blynk app.



●●○○○ Free 4G 16:09

Personal Hotspot: 1 Connection

weather station

21.000 35.000

4095

List of parts

1x ESP32 weather station
1x helium bottle
1x nylon wire bobbin
several inflatable balloons.

Setting up the weather station

Follow the instructions on page 1-4 on this tutorial.

Monitoring the values recorded by the station via Blynk

Follow the instructions on page 4-8 on this tutorial.

Setting up the flying station

In order to send your weather station up in the air, first of all you ll need to engineer a system similar to that of a hot air balloon.

Create a case to accommodate the weather station. This can be made of cardboard or any other material, as long as the weight is not too significant. Beware that helium balloons can only lift very light masses.

Once you've secured the weather station to the case, attach the former to the helium balloons. You may need to use multiple balloons in order to be able to lift the weather station off the ground.

Don’t forget to tie some nylon wire to the flying system so that you ll be able to take it back to the ground at any time.

Watch the flying station in action here.

Connected weather station - 12+ activity tutorial

In this activity, participants will set up their weather station, sent it up in the air, and monitor the recordings (light, temperature, humidity) in real time via the Blynk app. On top of all this, you will learn how to publish the values recorded by your weather station on a shared online map.

List of parts

1x ESP32 weather station
1x helium bottle
1x nylon wire bobbin

Setting up the weather station

Follow the instructions on page 1-4 on this tutorial.

Monitoring the values recorded by the station via Blynk

Follow the instructions on page 4-8 on this tutorial.

Setting up the flying station

In order to send your weather station up in the air, first of all you ll need to engineer a system similar to that of a hot air balloon.

Create a case to accommodate the weather station. This can be made of cardboard or any other material, as long as the weight is not too significant. Beware that helium balloons can only lift very light masses.

Once you've secured the weather station to the case, attach the former to the helium balloons. You may need to use multiple balloons in order to be able to lift the weather station off the ground.

Don’t forget to tie some nylon wire to the flying system so that you ll be able to take it back to the ground at any time.

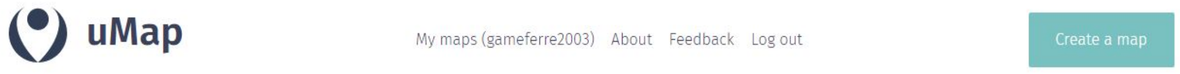
Plotting the values recorded by the station on UMap

You will be sharing the data recorded by your water probe on UMap, which allows users to create their own maps embedding the data of their choice.

First, head to https://umap.openstreetmap.fr/fr/

You ll need to create an account to be able to edit your own maps.

Once you have successfully logged in, hit the “Create a map” button.

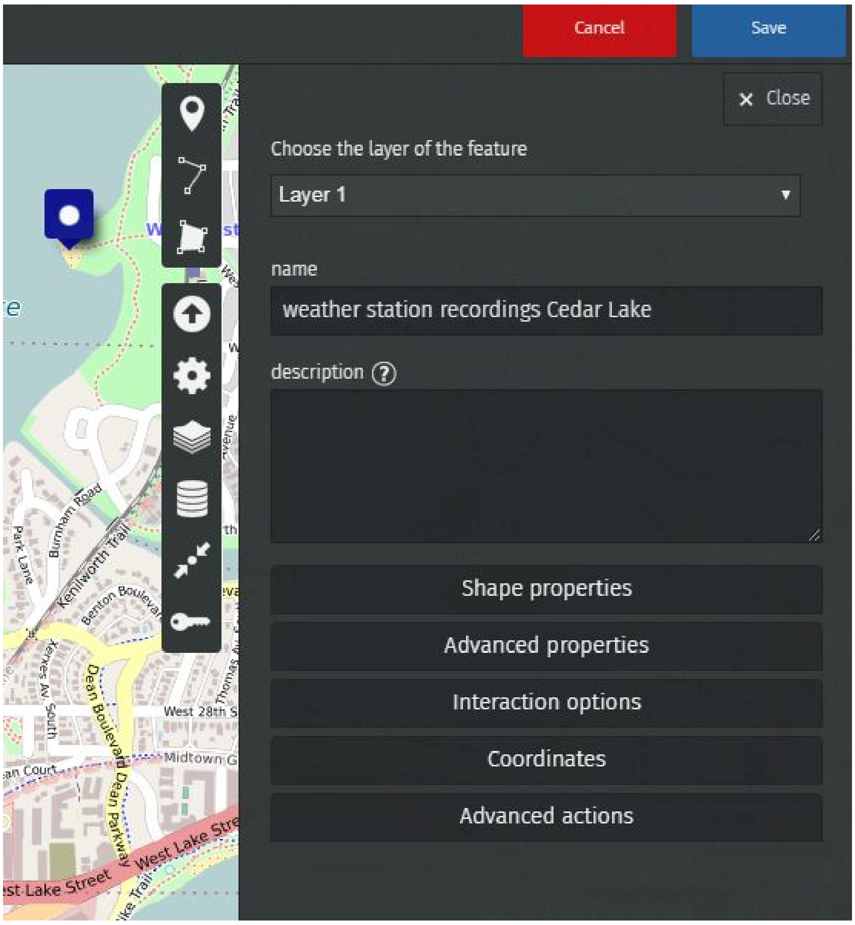


uMap

My maps (gameferre2003) About Feedback Log out

Create a map

You can now draw markers and enter any description you wish to publish.



Cancel Save

× Close

Choose the layer of the feature

Layer 1 ▾

name

weather station recordings Cedar Lake

description ?

Shape properties

Advanced properties

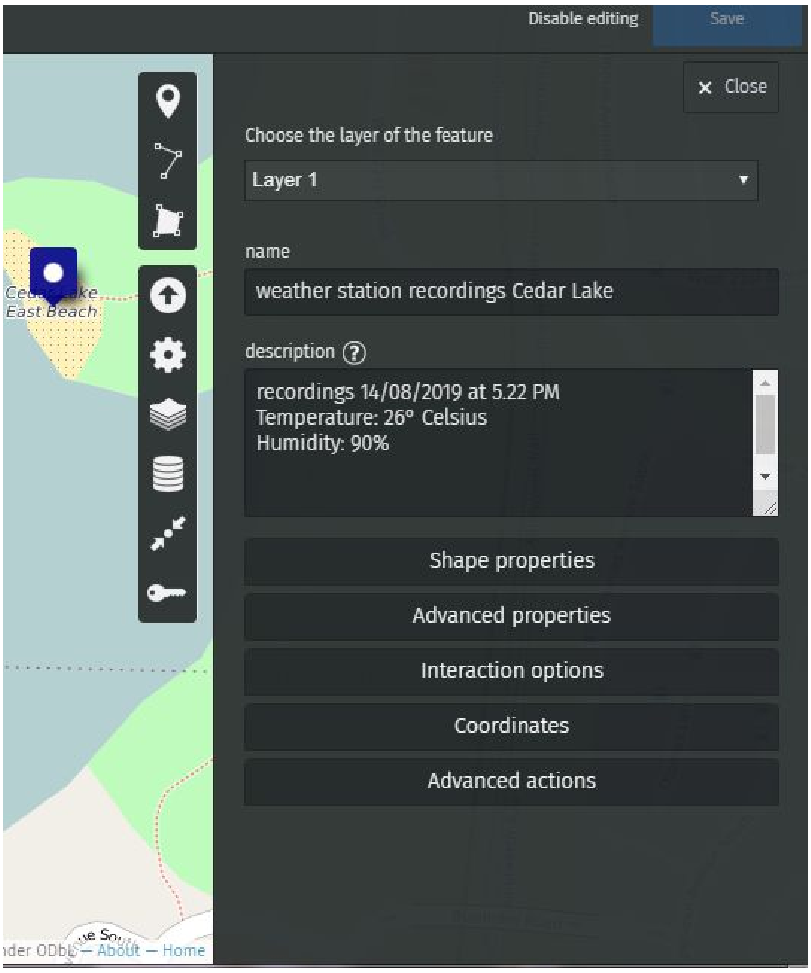
Interaction options

Coordinates

Advanced actions

Let’s add a marker featuring the recordings of the weather station above Cedar Lake, in Minneapolis.

Today the temperature in the air above Cedar Lake is 26° Celsius and the humidity is 90%. Indeed, it is raining!



Choose the layer of the feature: Layer 1

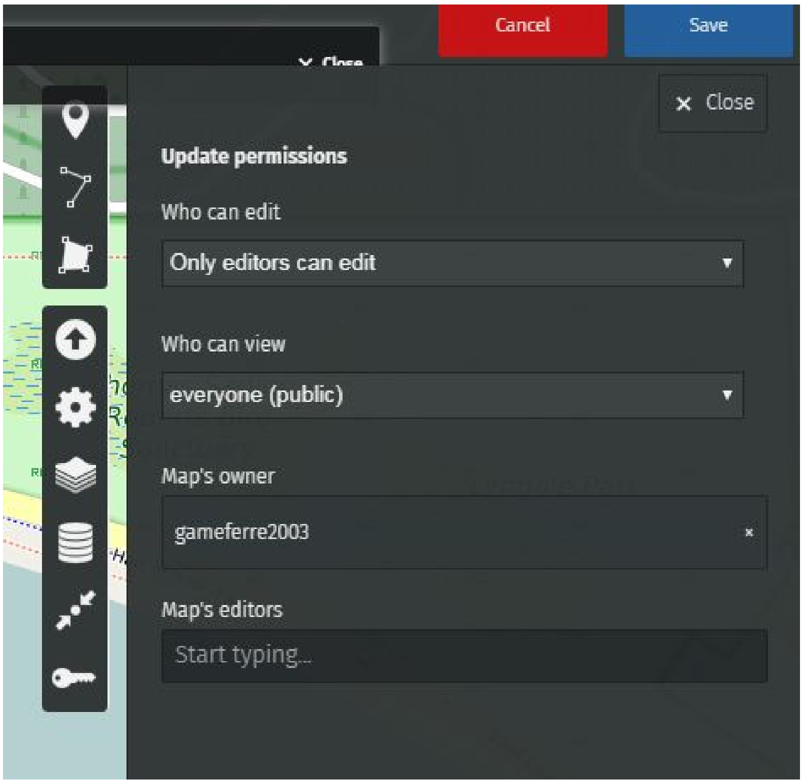
name: weather station recordings Cedar Lake

description:
recordings 14/08/2019 at 5.22 PM
Temperature: 26° Celsius
Humidity: 90%

Once you’re done, hit the Save button.

To share this map with anyone, you just need to provide them with the appropriate link.

Go to Update permissions and editors.



Update permissions

Who can edit: Only editors can edit

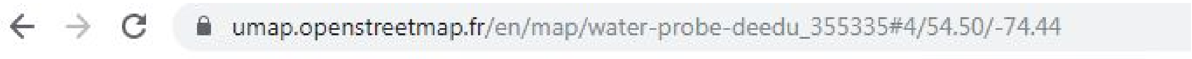
Who can view: everyone (public)

Map's owner: gameferre2003

Map's editors: Start typing...

You can choose who can view and who can edit this map.

To enable editors to edit the map, copy the link of your map (the web address of your map) and share it with whoever you want.



umap.openstreetmap.fr/en/map/water-probe-deedu_355335#4/54.50/-74.44

Digital Environmental Education (DEEDU) is a european partnership that produced educational resources for the youth work sector, on the topic of environmental science and tech.

The consortium is composed by HorizonLab (Italy), Educentrum (Czech Republic), Pralnia (Poland) and Digijeunes (France), the project leader.

The project is co-funded by the Erasmus + Programme of the European Union. Project n°: 2018-1-FR02-KA205-014144.

Find out more on deeduproject.eu

EDU centrum

DIGIJEUNES

Co-funded by the
Erasmus+ Programme
of the European Union