




Arduino Project Hub


A011, A013

Arduino Project Hub

- ▶ Arduino Project Hub, jak nazwa wskazuje jest społecznością tworzącą różne ciekawe projekty opierające się na Arduino i Genuino. Na tej platformie można w prosty sposób podzielić się swoimi najnowszymi projektami ze znajomymi i całą resztą społeczności w formie poradnika.

Arduino Project Hub




 PROJECT HUB


ADD PROJECT

SEARCH PROJECTS


SIGN IN




MoveThisWay Heavy Payload Assistant
Project showcase by Team Four Guys (EC...)
2,314 VIEWS 7 COMMENTS 19 RESPECTS



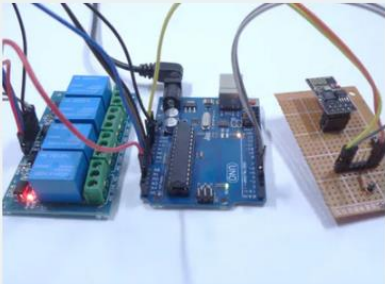
Architectural Model of a Bus Stop with Automatic Sunshade
Project tutorial by addicttux
2,078 VIEWS 8 COMMENTS 22 RESPECTS




Railway Interlocking System
Project showcase by Steve Massikker
5,662 VIEWS 12 COMMENTS 26 RESPECTS



Color Sorting Machine
Project tutorial by Little_french_kev
1,096 VIEWS 0 COMMENTS 20 RESPECTS

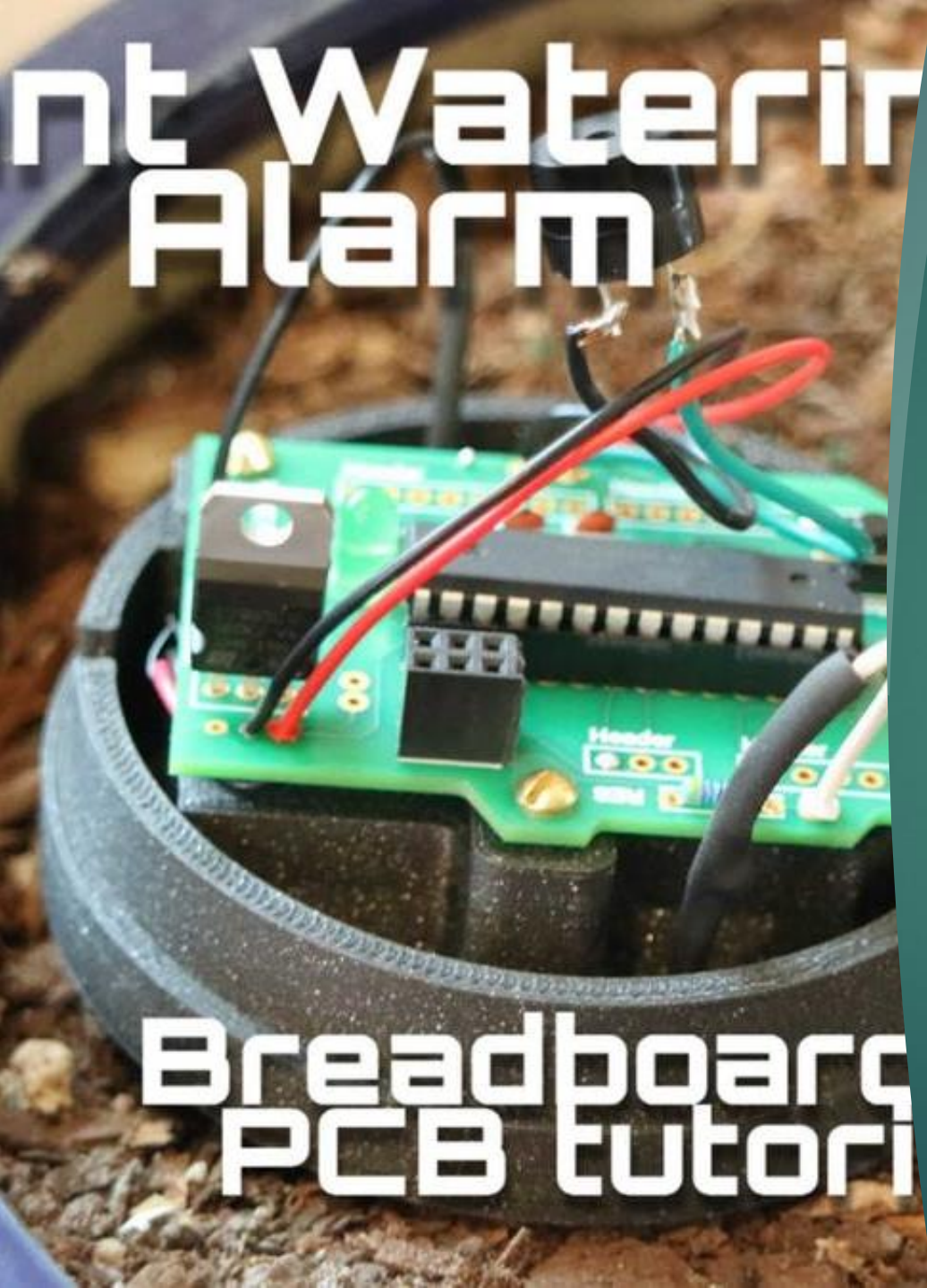


Home Automation Using ESP8266 Wi-Fi Module Without
by Muchika
1,178 VIEWS 0 COMMENTS 6 RESPECTS



Arduino Nano 33 IoT Debugging
Project tutorial by visualmicro
2,352 VIEWS 2 COMMENTS 11 RESPECTS

nt Waterin Alarm



































Breadboard
PCB tutorial

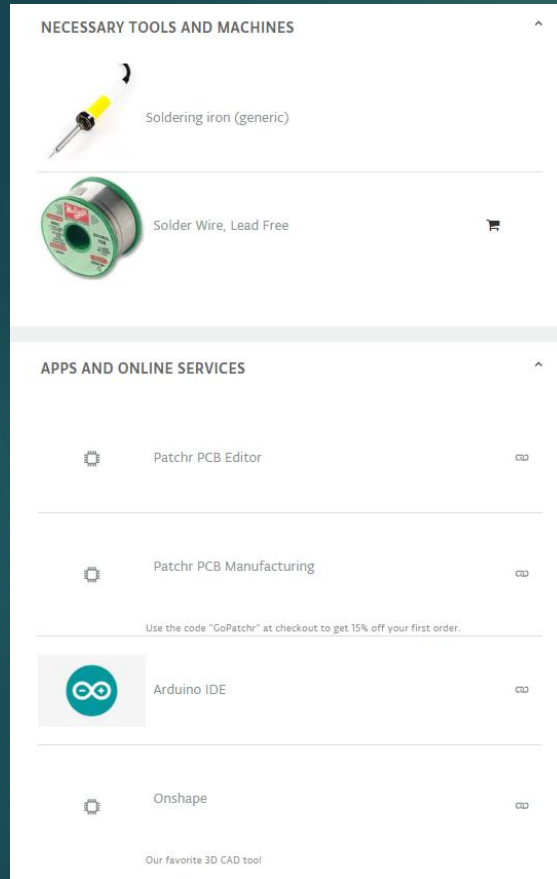
- ▶ Dla przykładu takiego poradnika z Arduino Project Hub wzięliśmy "Thirst alert plant alarm". Jest to dosyć ciekawy, a zarazem prosty projekt, który może być świetnym początkiem dla kogoś, kto by chciał pobawić się z elektroniką.
- ▶ Podstawowe założenie tego projektu jest takie, jeżeli ziemia będzie sucha, puści dźwięk który ma być przypomnieniem o podlaniu.

1. Potrzebne komponenty

Strona projektu zaczyna się od listy części, które będą nam potrzebne do zbudowania projektu.

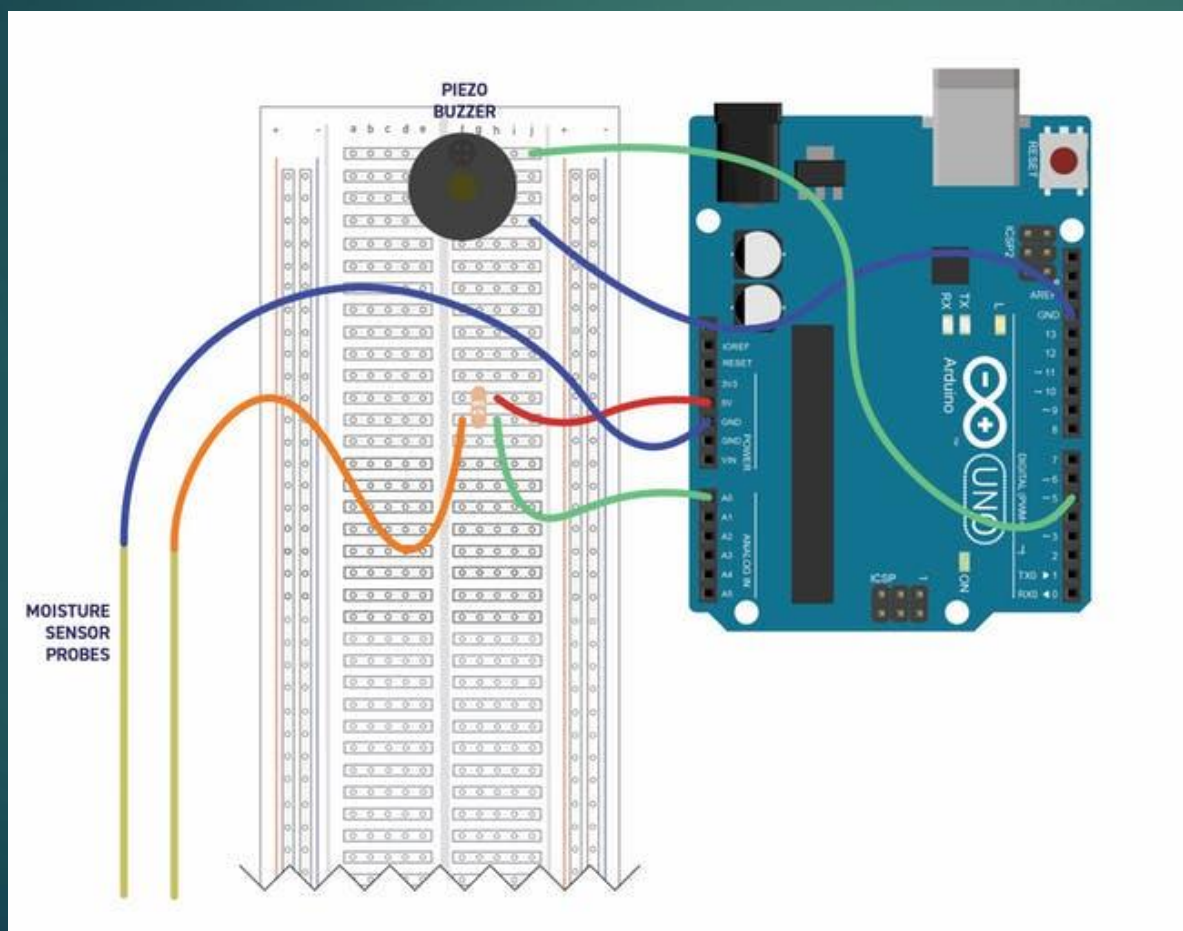
	Microchip ATmega328	x	1			Slide Switch	x	1	
	16 MHz Crystal	x	1			C&K Switches PTS 645 Series Switch	x	1	
	Ceramic Disc Capacitor, 20 pF	x	2			Buzzer, Piezo	x	1	
	Linear Regulator (7805)	x	1			9V battery (generic)	x	1	
	LED (generic) Use whatever colors you want with this one!	x	2			9V Battery Clip	x	1	
	Resistor 10k ohm	x	1			SparkFun Soil Moisture Sensor (with Screw Terminals)	x	1	
	Resistor 220 ohm	x	1			Arduino UNO & Genuino UNO	x	1	
	Through Hole Resistor, 470 ohm	x	1			3D Printed Case	x	1	

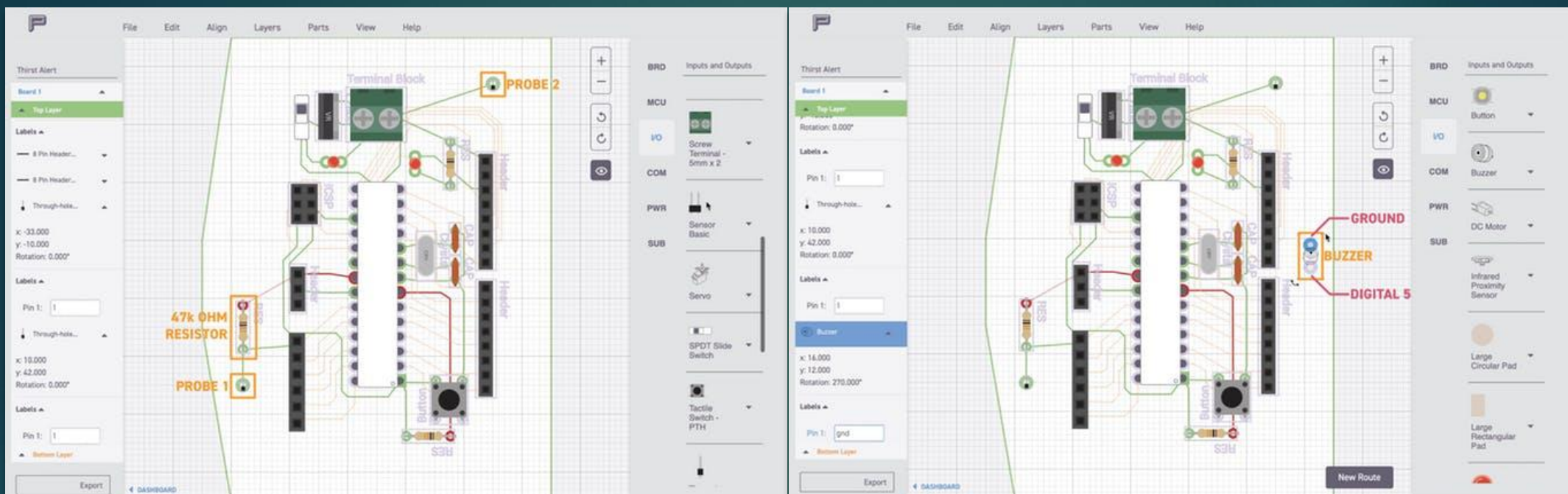
2. Narzędzia i aplikacje



- Z narzędzi potrzebujemy jedynie zestaw do lutowania.
- W zakładce aplikacji autor projektu dodał programy Patchr, dzięki nim możemy sobie stworzyć projekt i zamówić takie customowe PCB, jakie potrzebujemy, jakieś IDE do arduino i OnShape - do zrobienia projektu obudowy 3d.

Przy procesie budowania jest pokazana płytką prototypowa. Autor pokazał także jak zrobić własny czujnik wilgoci używając pręcików, przewodów i odpowiedniego opornika.





Projektowanie PCB

- Autor użył w oprogramowaniu gotowego szablonu Explorer'a Uno, który został następnie zmodyfikowany. Dzięki temu nie trzeba robić wszystkiego samemu od podstaw.

...i gotowy
projekt



3. Kod źródłowy do projektu

Zczytywana wartość z czujnika przekazywana jest płytce Arduino. Jeżeli wartość z czujnika będzie wyższa, niż 300 program zaczyna puszczać w pętli dźwięki o losowej przerwie między nimi.

```
9  #include "Volume.h" /* Developed by Connor Nishijima (2016) */
10
11  int sensorPin = A0; //set the pin for the moisture sensor
12  int sensorValue;
13  int limit = 300;
14
15  Volume vol;
16
17  void setup()
18  {
19    Serial.begin(9600);
20    vol.begin();
21    vol.setMasterVolume(1.00); //Set this between 0.00 and 1.00 dependent on how loud you want the project to be
22    vol.delay(500);
23  }
24
25  void loop()
26  {
27    sensorValue = analogRead(sensorPin);
28    Serial.println(sensorValue);
29
30    if (sensorValue > limit) {
31      Serial.println("**cricket**"); //if the sensor value is greater than 300 then play the cricket noise at a random interval between 1 and 30
32      chirp();
33      vol.delay(random(1000, 3000));
34    }
35    else {
36      vol.noTone();
37    }
38  }
39
40
41  void chirp() { //this part of the software was written by Connor Nishijima and creates the chirp tones
42    int freq = random(3700, 4000);
43    float volume = random(10, 100+1) / 100.00;
44
45    int chirpType = 1;
46
47    if (chirpType == 1) {
48      Serial.println("2");
49      int chirpCount = random(1,3);
50      chirp(chirpCount, freq, volume);
51    }
52  }
```

```
54  void chirp(byte times, int freq, float volume) {
55    float fadeOut = 1.00;
56    while (times > 0) {
57      float mv = 0;
58      while (mv < 1.00) {
59        int v = 0;
60        while (v < 255) {
61          vol.tone(freq, v * volume * mv * fadeOut);
62          v += 4;
63        }
64        v = 255;
65        while (v > 0) {
66          vol.tone(freq, v * volume * mv * fadeOut);
67          v -= 4;
68        }
69        vol.delay(20);
70        mv += 0.2;
71      }
72      mv = 1.00;
73      while (mv > 0.00) {
74        int v = 0;
75        while (v < 255) {
76          vol.tone(freq, v * volume * mv * fadeOut);
77          v += 4;
78        }
79        v = 255;
80        while (v > 0) {
81          vol.tone(freq, v * volume * mv * fadeOut);
82          v -= 4;
83        }
84        vol.delay(20);
85        mv -= 0.2;
86      }
87      vol.noTone();
88      fadeOut -= 0.75;
89      times--;
90    }
91  }
```

Dziękujemy za uwagę

[HTTPS://CREATE.ARDUINO.CC/
PROJECTHUB](https://create.arduino.cc/projecthub)