

Технология. Arduino

8 класс 255 школа

Методическое пособие

Ярмолинский Арсений Маркович

10 сентября 2024 г.

Содержание

| | | |
|----------|------------------------------|-----------|
| 1 | Начало работы в WokWi | 2 |
| 1.1 | Создание проекта | 2 |
| 2 | План | 8 |
| 3 | Привет, мир! | 9 |
| 4 | Плавный маяк | 10 |
| 5 | Регулируемая лампочка | 12 |
| 6 | Автоматический ночник | 13 |
| 7 | Пульсар | 15 |
| 8 | Эквалайзер | 17 |

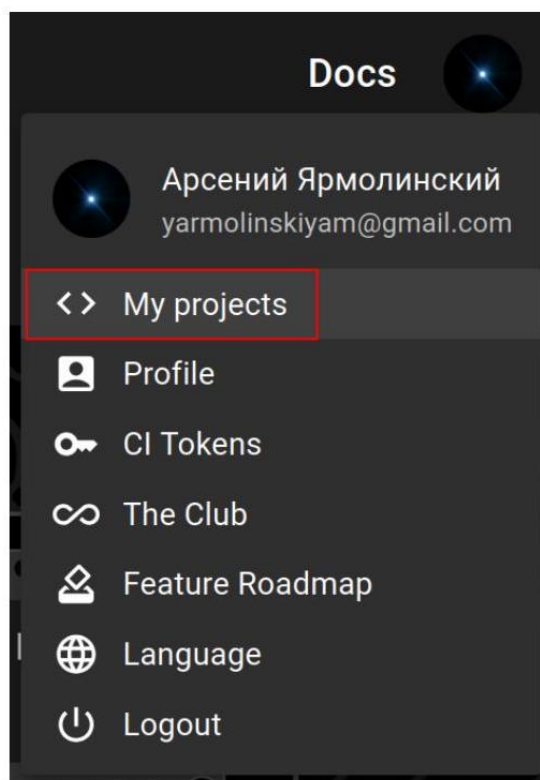
1 Начало работы в WokWi

1.1 Создание проекта

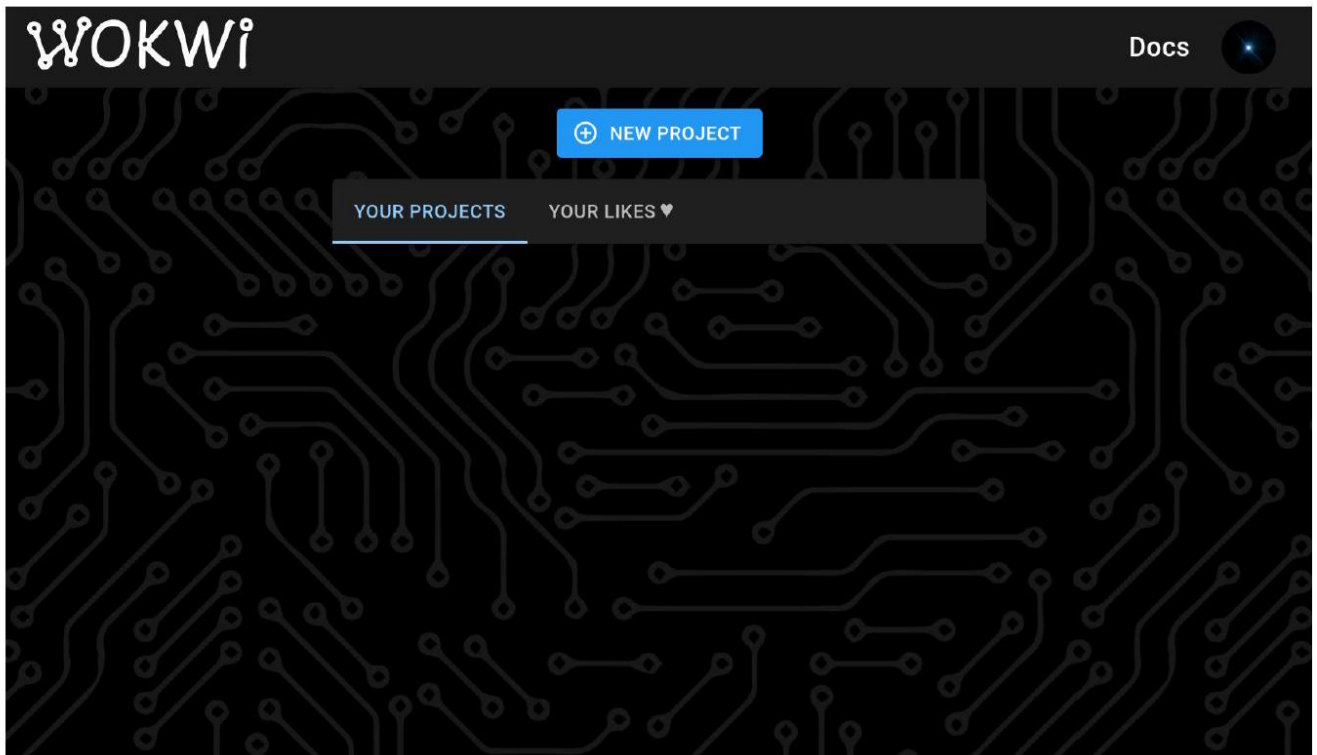
1. Вид главной страницы Wokwi:



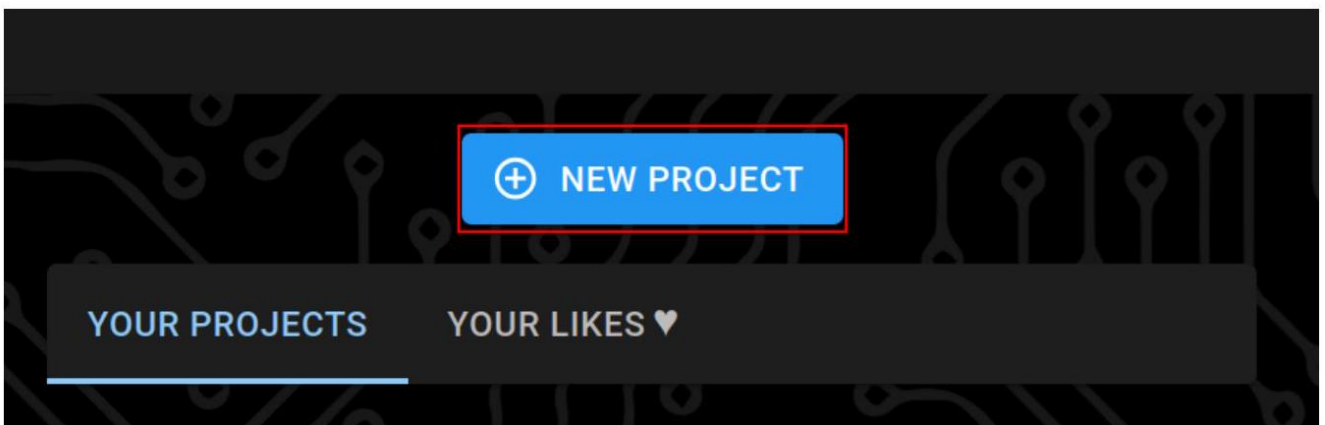
2. Откройте выпадающее меню профиля и нажмите кнопку "Мои проекты"



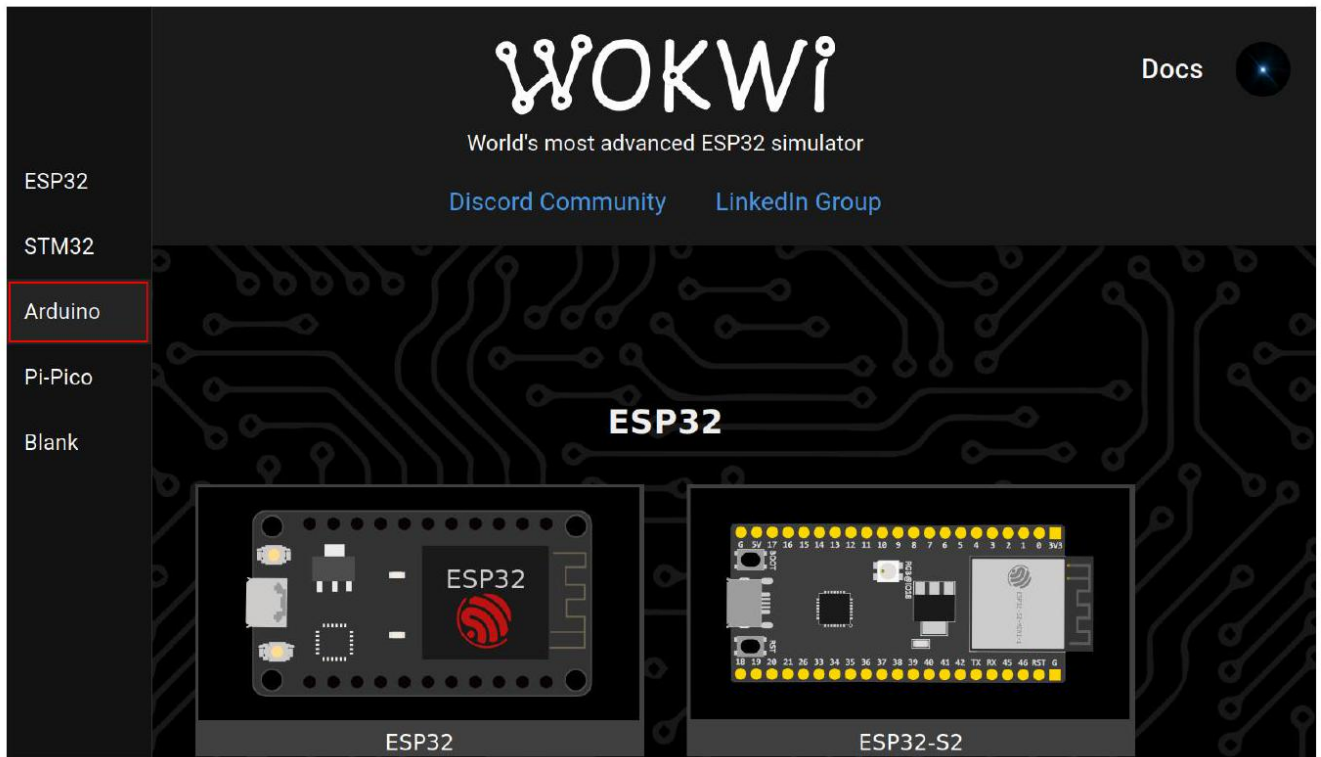
3. Страница проектов



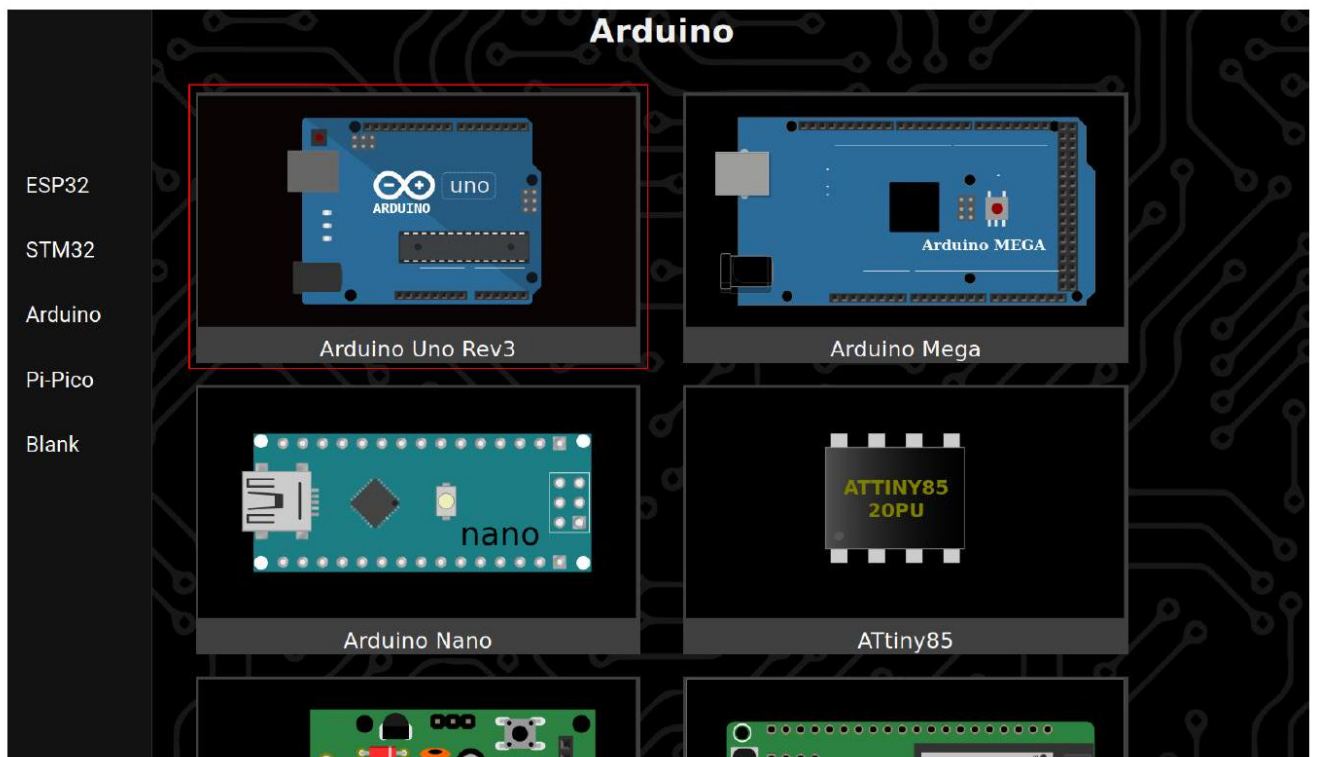
4. Создайте новый проект



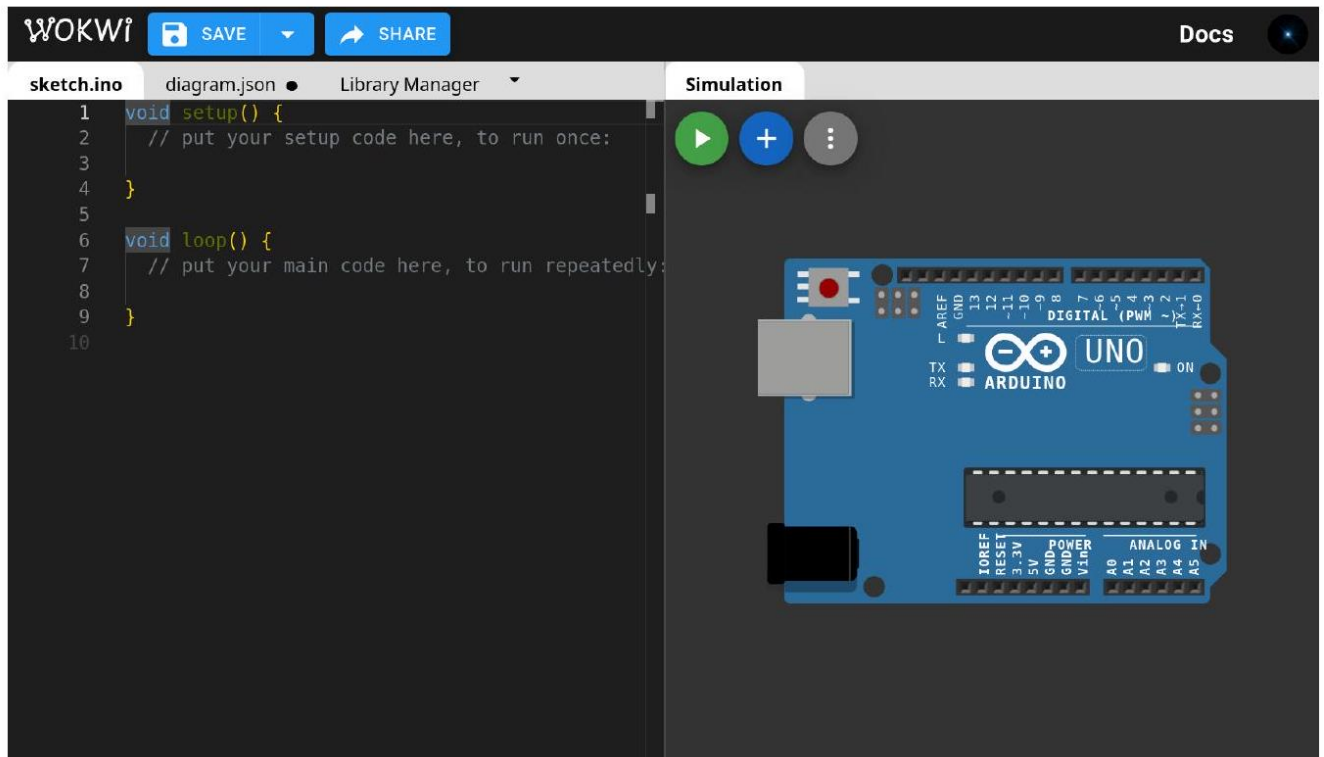
5. Выберите в меню слева семейство контроллеров Arduino



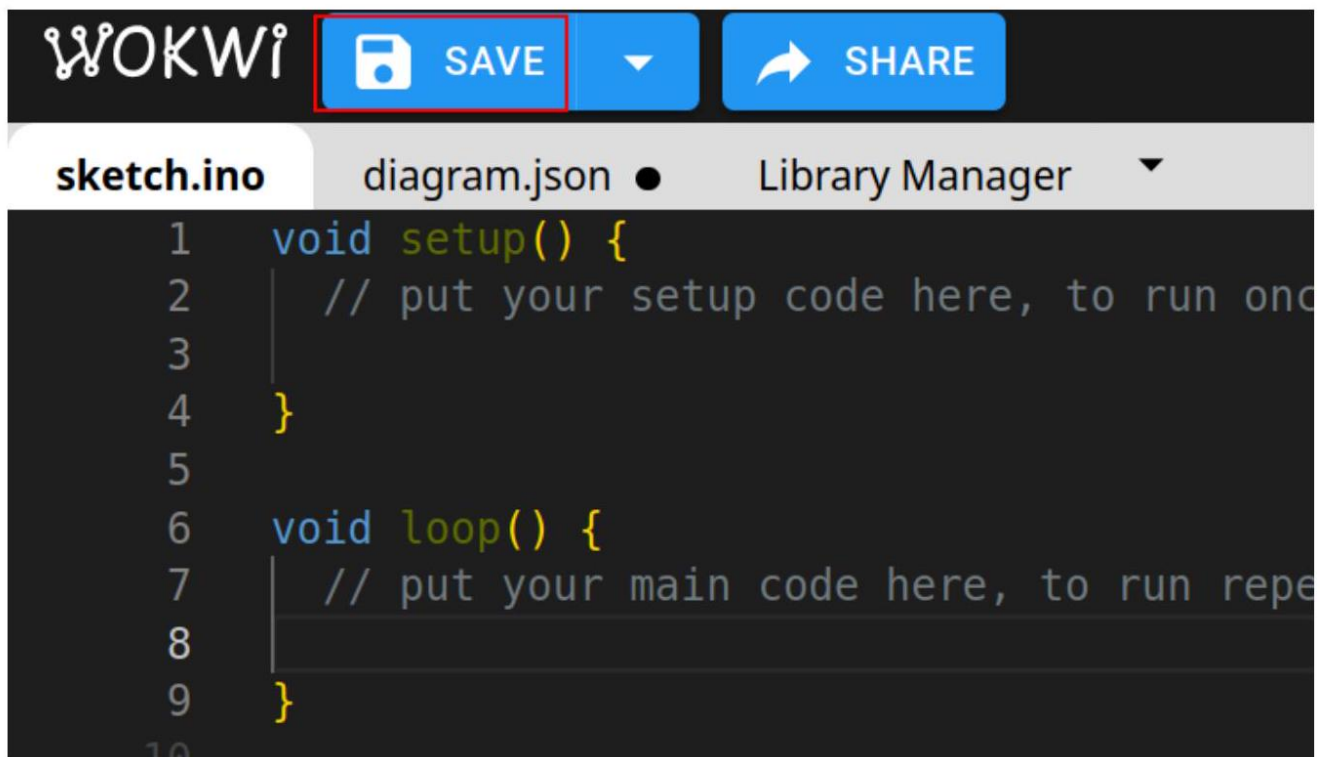
6. Выберите контроллер Arduino Uno Rev3



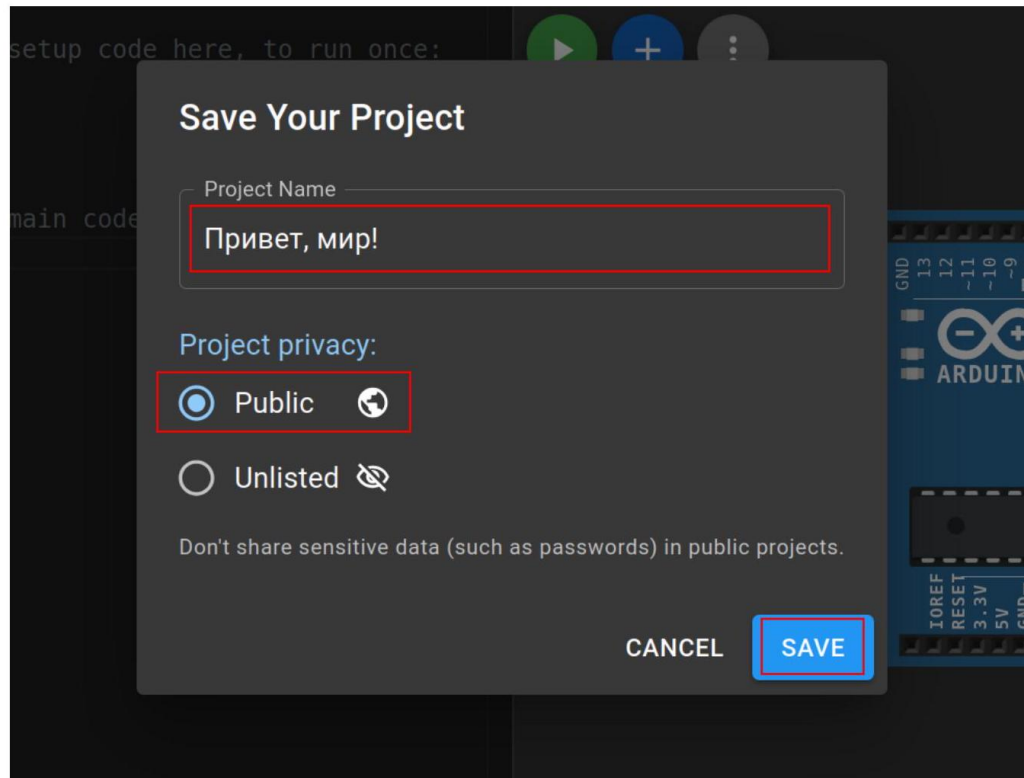
7. Созданный пустой проект!



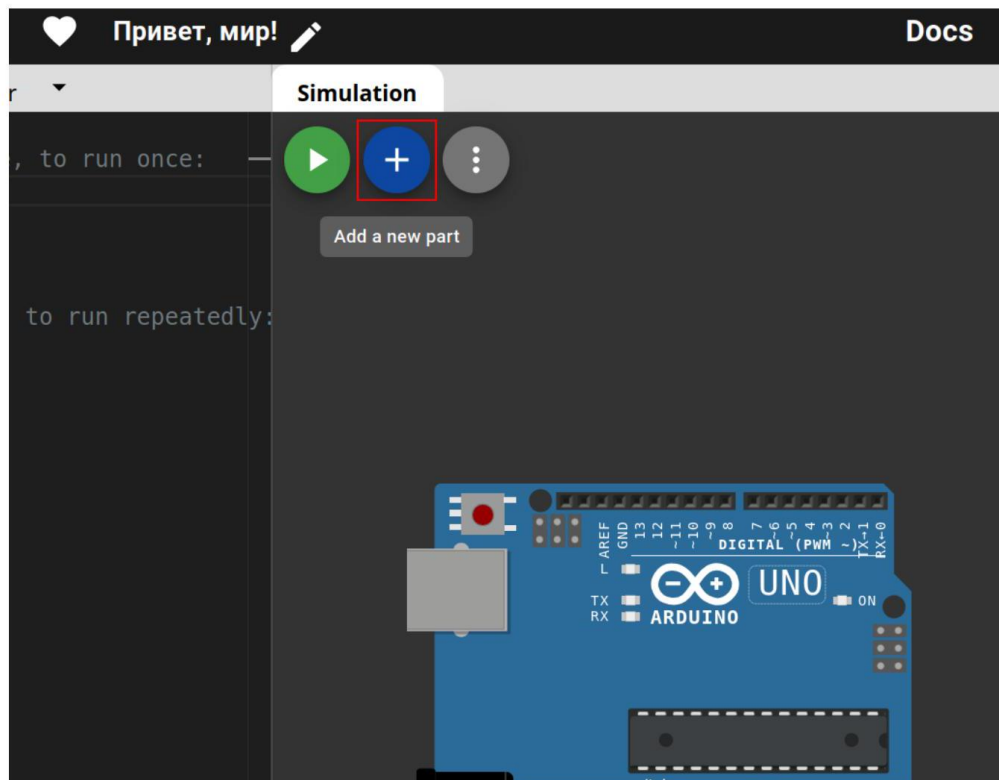
8. После внесения изменений сохраните проект с помощью кнопки SAVE



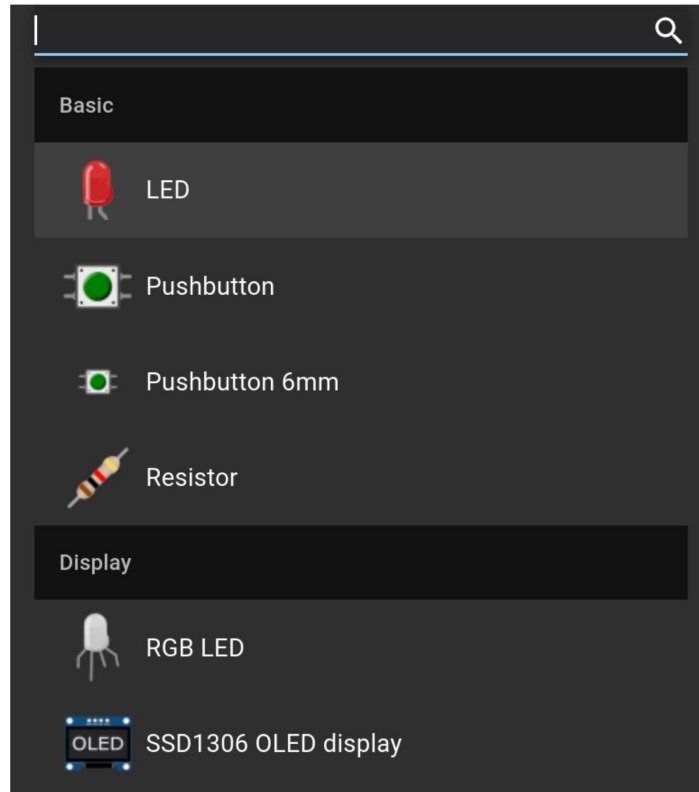
9. Введите название проекта и убедитесь, что он сохраняется как Public



10. Для добавления новой детали нажмите кнопку добавления



11. И выберите нужную деталь из выпадающего меню



2 План

1. Мигание светодиодом. `delay()`, `digitalWrite()`
2. Маячок с нарастающей яркостью. ШИМ, `analogWrite()`, Идентификаторы
3. Регулируемая лампочка. `analogRead()`, `int`, Переменные
4. Автоматический ночник. `bool`, `if()`
5. Пульс. `mod`
6. Бегущий огонь. `for()`
7. Вентилятор. Силовая нагрузка. Мотор
8. Световой телеграф. подключение кнопок, `digitalRead()`
9. Лампочка с кнопкой. Защита отдребезга
10. Регулируемая лампочка. Функции
11. Кнопочные ковбои. Массивы
12. Секундомер. Семисегментный индикатор, `bitRead()`
13. Счетчик нажатий. Сдвиговый регистр, `shiftOut()`
14. Комнатный термометр. Датчик температуры. Математические выражения
15. Метеостанция. `Serial.print()`, `Serial.println()`
16. Пантограф. `Servo`
17. USB лампочка. `Serial.read()`

3 Привет, мир!

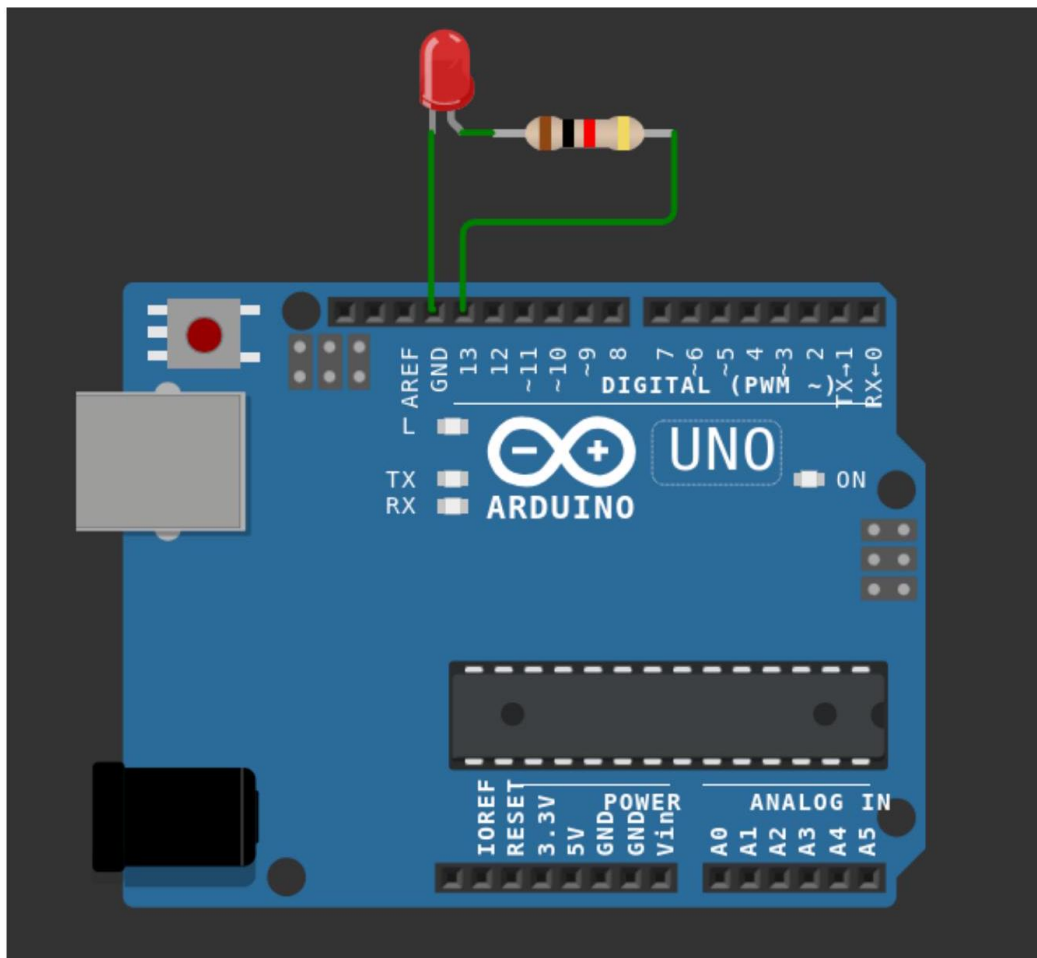


Рис. 1: Привет, мир!

```
1 void setup()
2 {
3     pinMode(13, OUTPUT); // Настраиваем 13 выход Ардуино на выход
4 } // Команды в void setup() выполнятся один раз при включении микроконтроллера
5 void loop()
6 {
7     digitalWrite(13, HIGH); // Подаем высокое напряжение на пине 13
8     delay(1000); // Ждем 1 секунду (1000 миллисекунд)
9     digitalWrite(13, LOW); // Подаем низкое напряжение на пин 13
10    delay(1000); // Снова ждем секунду
11 } // Эта последовательность действий будет повторяться бесконечно,
12    // поскольку написана в void loop()
```

4 Плавный маяк

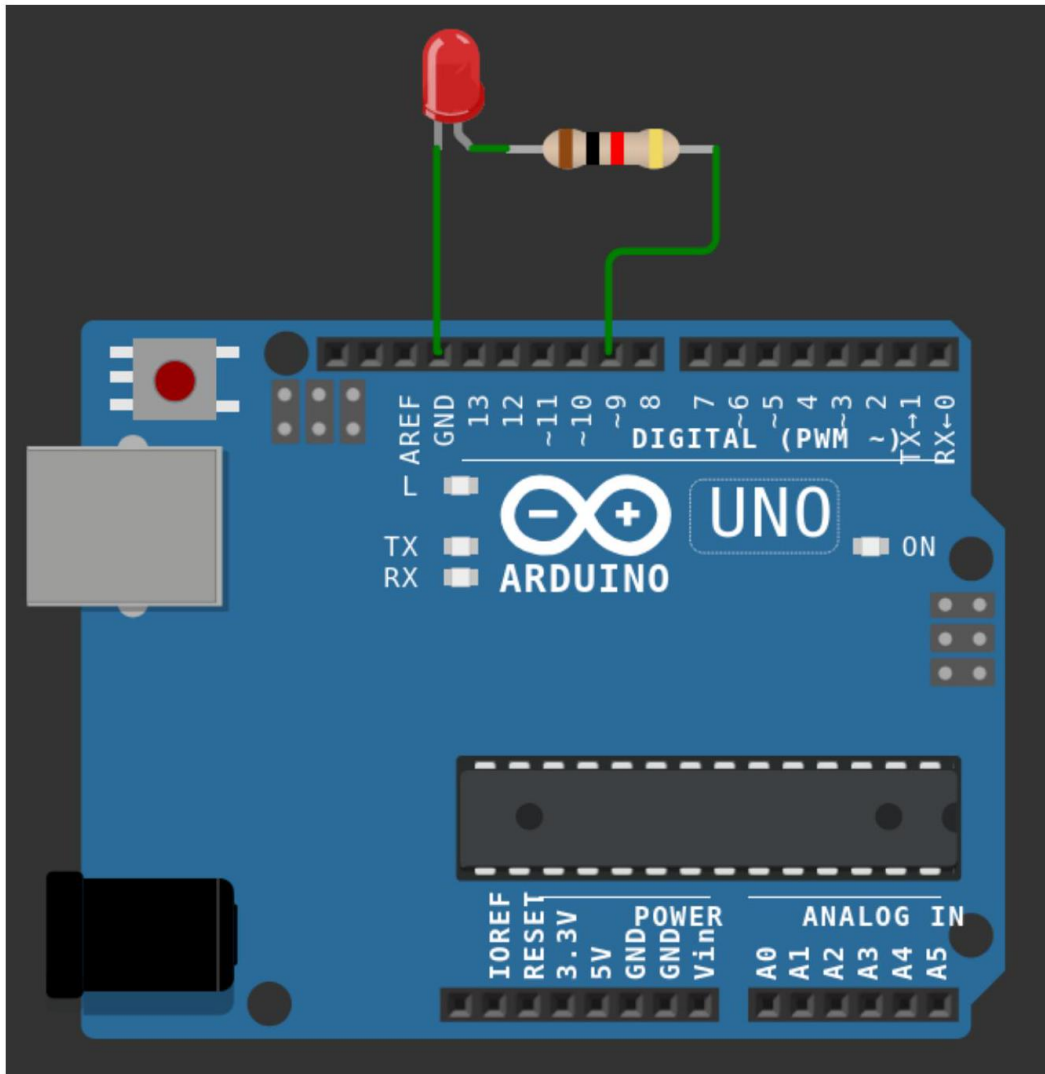


Рис. 2: Плавный маяк

```
1  #define LED_PIN 9
2  void setup()
3  {
4      pinMode(LED_PIN, OUTPUT);
5  }
6  void loop()
7  {
8      analogWrite(LED_PIN, 0);
9      delay(250);
10     analogWrite(LED_PIN, 30);
```

```
11     delay(250);  
12     analogWrite(LED_PIN, 100);  
13     delay(250);  
14     analogWrite(LED_PIN, 255);  
15     delay(250);  
16 }
```

5 Регулируемая лампочка

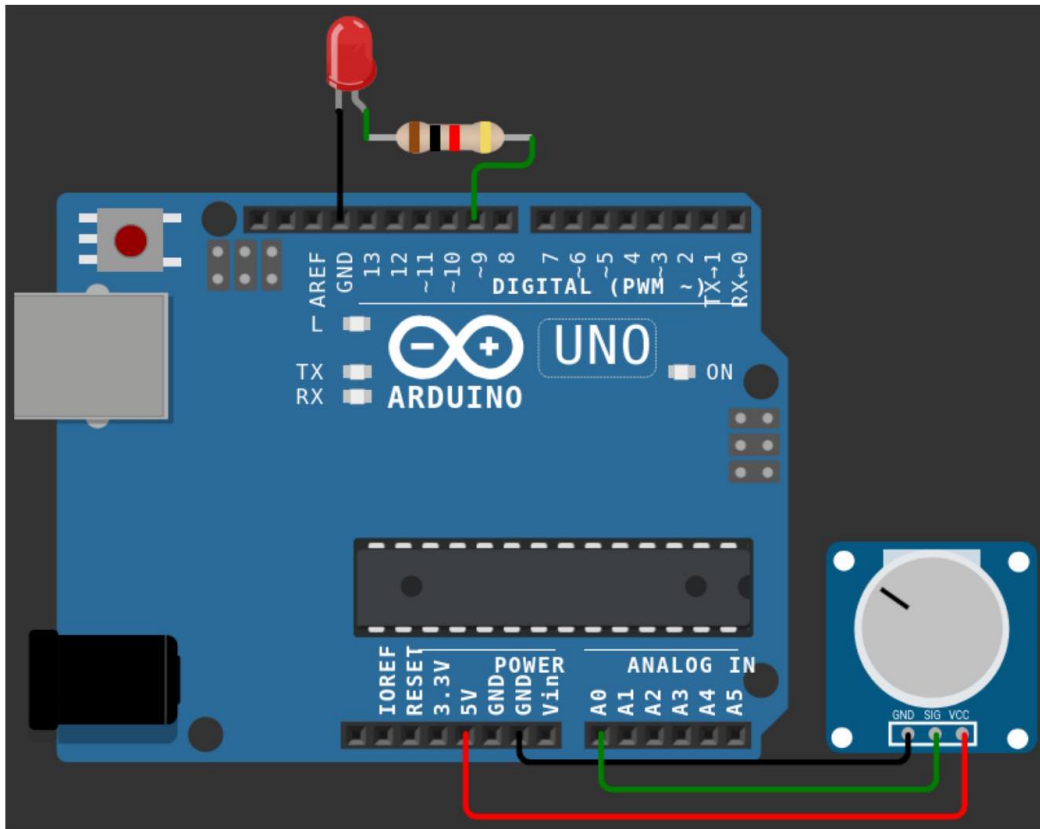


Рис. 3: Регулируемая лампочка

```
1  #define LED_PIN 9
2  #define POT_PIN A0
3  void setup()
4  {
5      pinMode(LED_PIN, OUTPUT);
6      pinMode(POT_PIN, INPUT); // Объявляем наш потенциометр как вход
7  }
8  void loop()
9  {
10     int rotation, brightness;
11     rotation = analogRead(POT_PIN);
12     brightness = rotation / 4;
13     analogWrite(LED_PIN, brightness);
14 }
```

6 Автоматический ночник

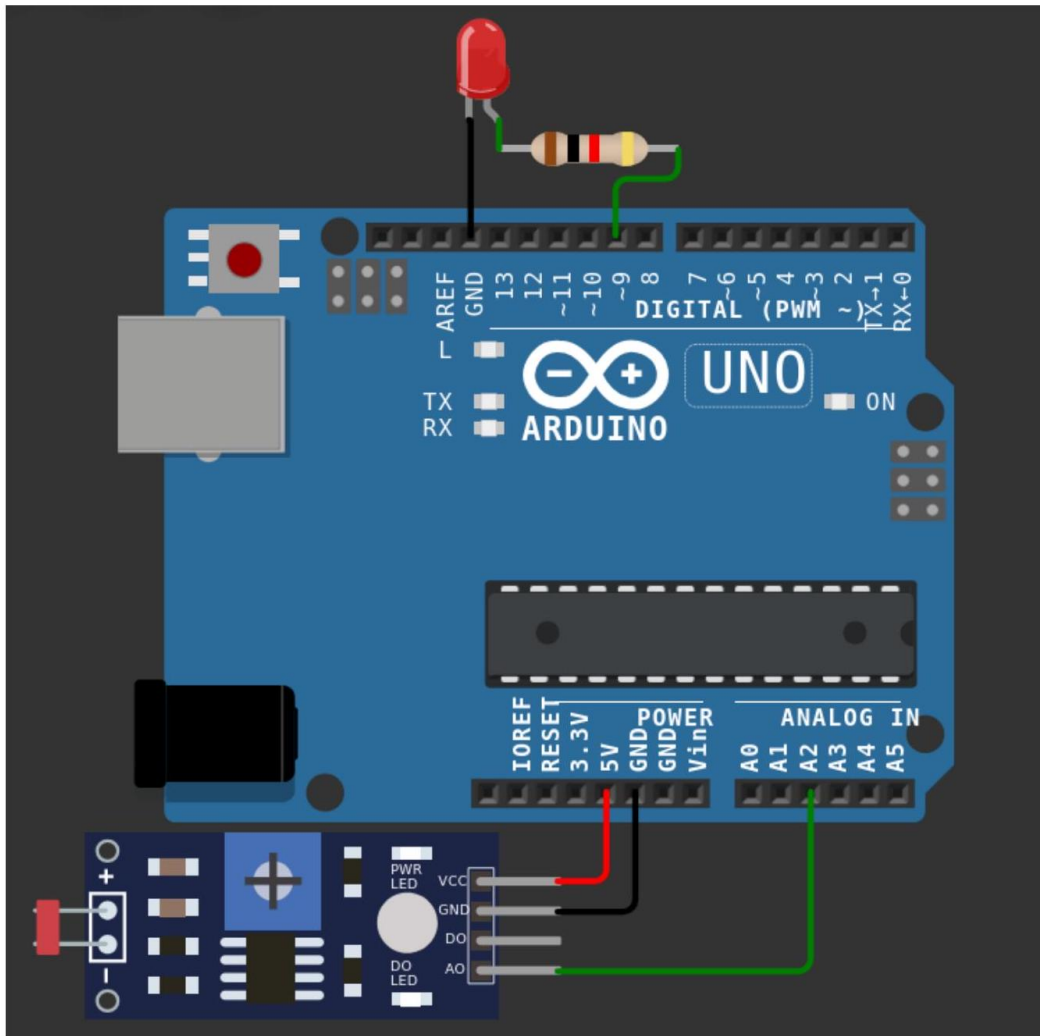


Рис. 4: Автоматический ночник

```
1  #define LED_PIN 9
2  #define LIGHT_PIN A2
3
4  void setup()
5  {
6      pinMode(LED_PIN, OUTPUT);
7      pinMode(LIGHT_PIN, INPUT);
8  }
9
10 void loop()
11 {
```

```
12     int light;
13     bool is_night;
14
15     light = analogRead(LIGHT_PIN);
16
17     is_night = light > 500;
18
19     analogWrite(LED_PIN, is_night);
20 }
```

7 Пульсар

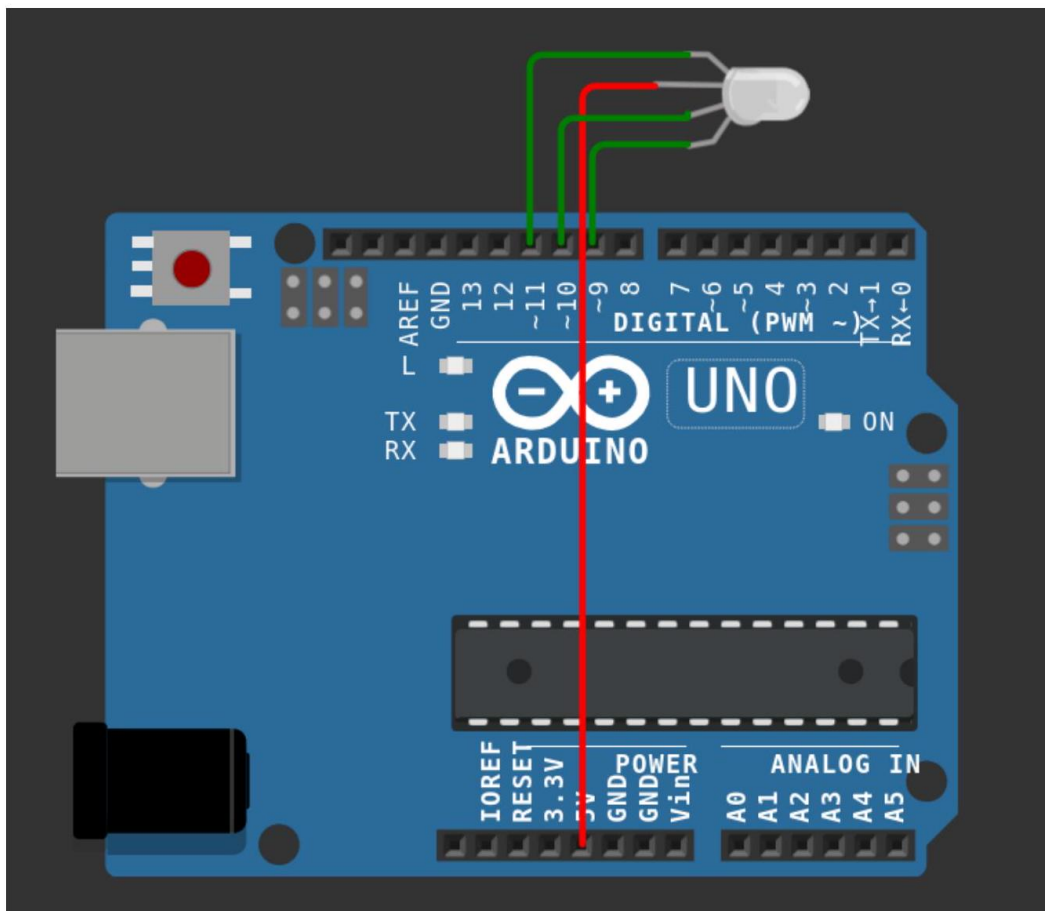


Рис. 5: Пульсар

```
1  #define RED_PIN 11
2  #define GRN_PIN 10
3  #define BLU_PIN 9
4
5  int brightness_R = 0;
6  int brightness_G = 86;
7  int brightness_B = 171;
8
9  void setup()
10 {
11     pinMode(RED_PIN, OUTPUT);
12     pinMode(GRN_PIN, OUTPUT);
13     pinMode(BLU_PIN, OUTPUT);
```

```
14 }
15
16 void loop()
17 {
18     brightness_R = (brightness_R + 1) % 256;
19     brightness_G = (brightness_G + 1) % 256;
20     brightness_B = (brightness_B + 1) % 256;
21
22     analogWrite(RED_PIN, brightness_R);
23     analogWrite(GRN_PIN, brightness_G);
24     analogWrite(BLU_PIN, brightness_B);
25
26     delay(10);
27 }
```


8 Эквалайзер

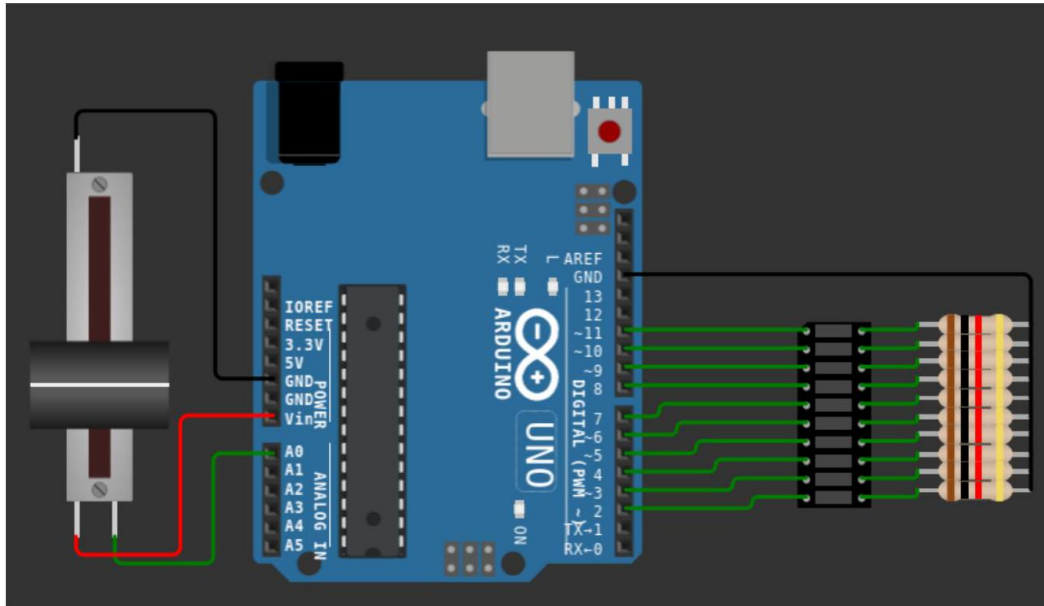


Рис. 6: Эквалайзер

```
1  /*
2   LED bar graph
3   https://wokwi.com/arduino/projects /309829489359061570
4   Turns on a series of LEDs based on the value of an analog sensor.
5   This is a simple way to make a bar graph display. Though this graph uses
6       10
7   LEDs, you can use any number by changing the LED count and the pins in the
8   array.
9   This method can be used to control any series of digital outputs that
10      depends
11   on an analog input.
12   The circuit:
13   - LEDs from pins 2 through 11 to ground
14   created 4 Sep 2010
15   by Tom Igoe
16   This example code is in the public domain.
17   https://www.arduino.cc/en/Tutorial/BuiltInExamples/BarGraph
18  */
19  // these constants won't change:
20  #define ANALOG_PIN A0 // the pin that the potentiometer is attached to
```

```

21 #define LED_COUNT 10 // the number of LEDs in the bar graph
22 int ledPins [] = {
23     2, 3, 4, 5, 6, 7, 8, 9, 10, 11}; // an array of pin numbers to which LEDs
24
25 void setup()
26 {
27     // loop over the pin array and set them all to output:
28     for (int thisLed = 0; thisLed < LED_COUNT; thisLed++)
29     {
30         pinMode(ledPins[thisLed], OUTPUT);
31     }
32 }
33 void loop()
34 {
35     // read the potentiometer:
36     int sensorReading = analogRead(ANALOG_PIN);
37     // map the result to a range from 0 to the number of LEDs:
38     int ledLevel = map(sensorReading, 0, 1023, 0, LED_COUNT);
39     // loop over the LED array:
40     for (int thisLed = 0; thisLed < LED_COUNT; thisLed++)
41     {
42         // if the array element's index is less than ledLevel,
43         // turn the pin for this element on:
44         if (thisLed < ledLevel)
45         {
46             digitalWrite(ledPins[thisLed], HIGH);
47         }
48         // turn off all pins higher than the ledLevel:
49         else
50         {
51             digitalWrite(ledPins[thisLed], LOW);
52         }
53     }
54 }

```