

Технология. Arduino

255 школа

Методическое пособие

Ярмолинский
Арсений Маркович

Ярмолинская
Марита Вонбеневна

2025

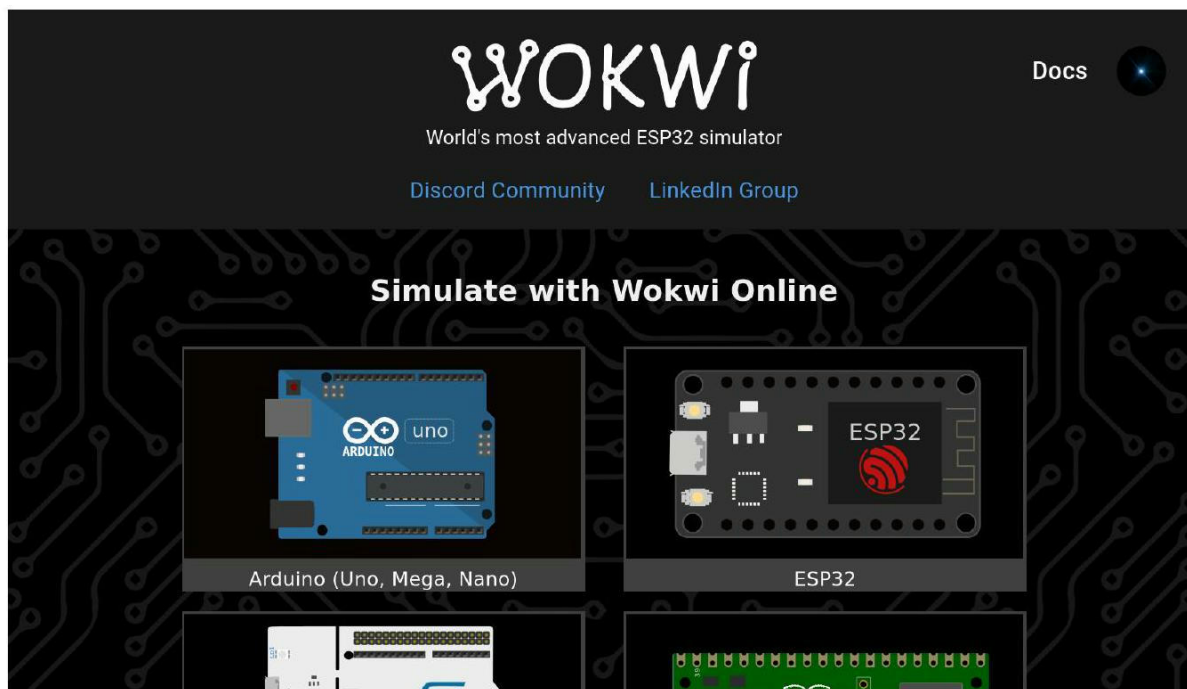
Содержание

Начало работы в WokWi	2
Создание проекта	2
Привет, мир!	8
Плавный маяк	9
Регулируемая лампочка	10
Автоматический ночник	11
Пульсар	12
Эквалайзер	14
Кнопки	16
Механизированный флаг	18
Автоматические ворота	20
Калькулятор	22
Тестовое задание	24

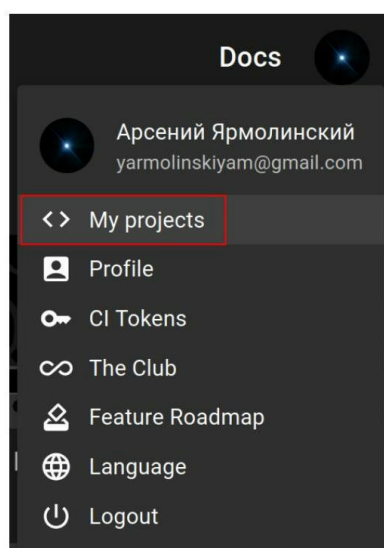
Начало работы в WokWi

Создание проекта

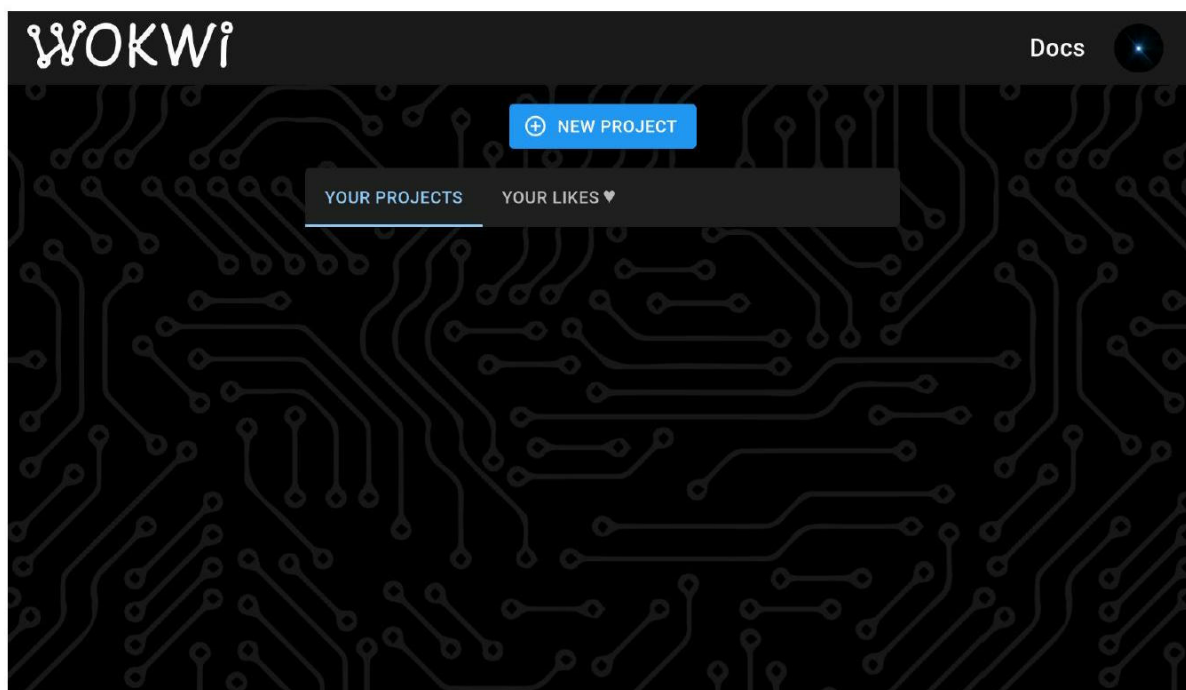
Вид главной страницы Wokwi:



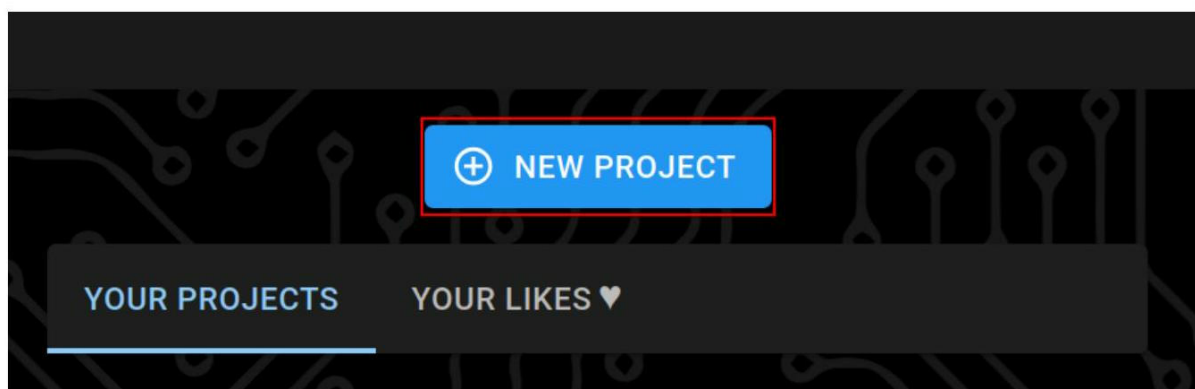
Откройте выпадающее меню профиля и нажмите кнопку «Мои проекты»



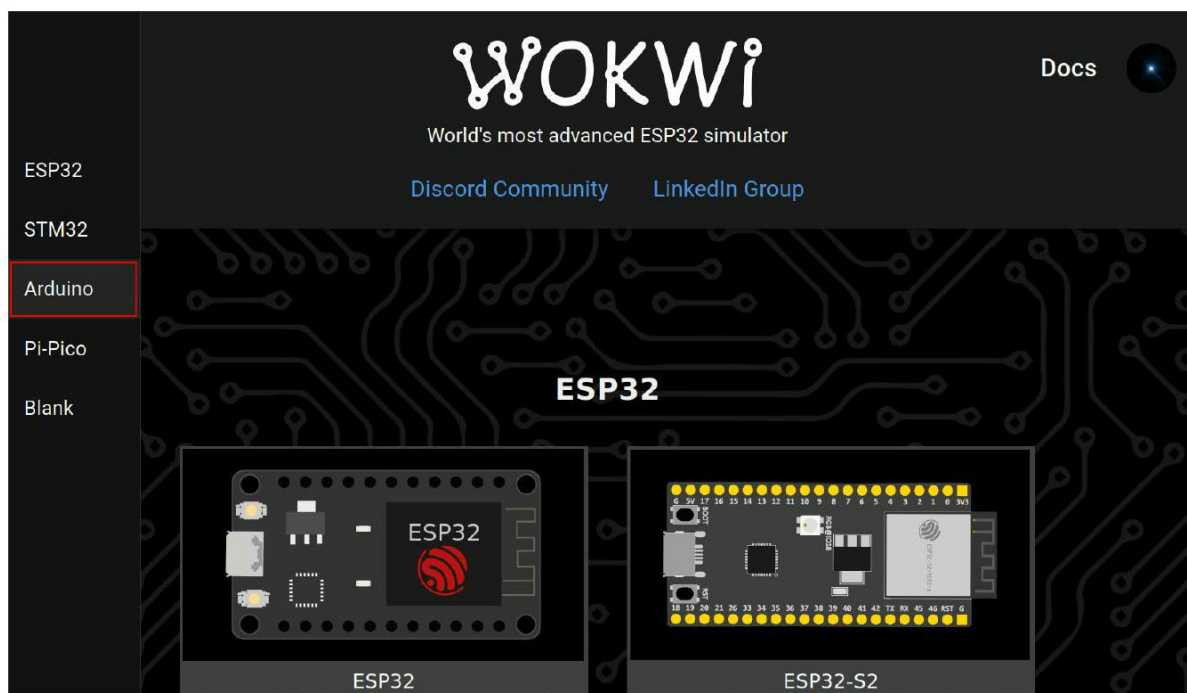
Страница проектов



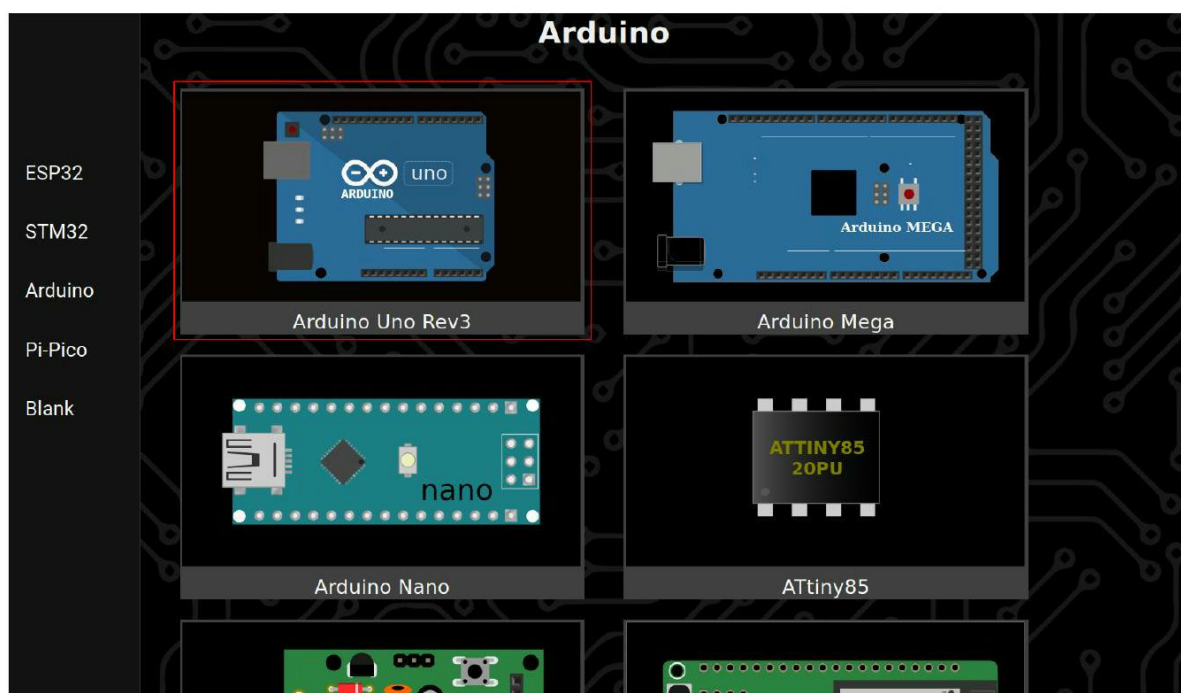
Создайте новый проект



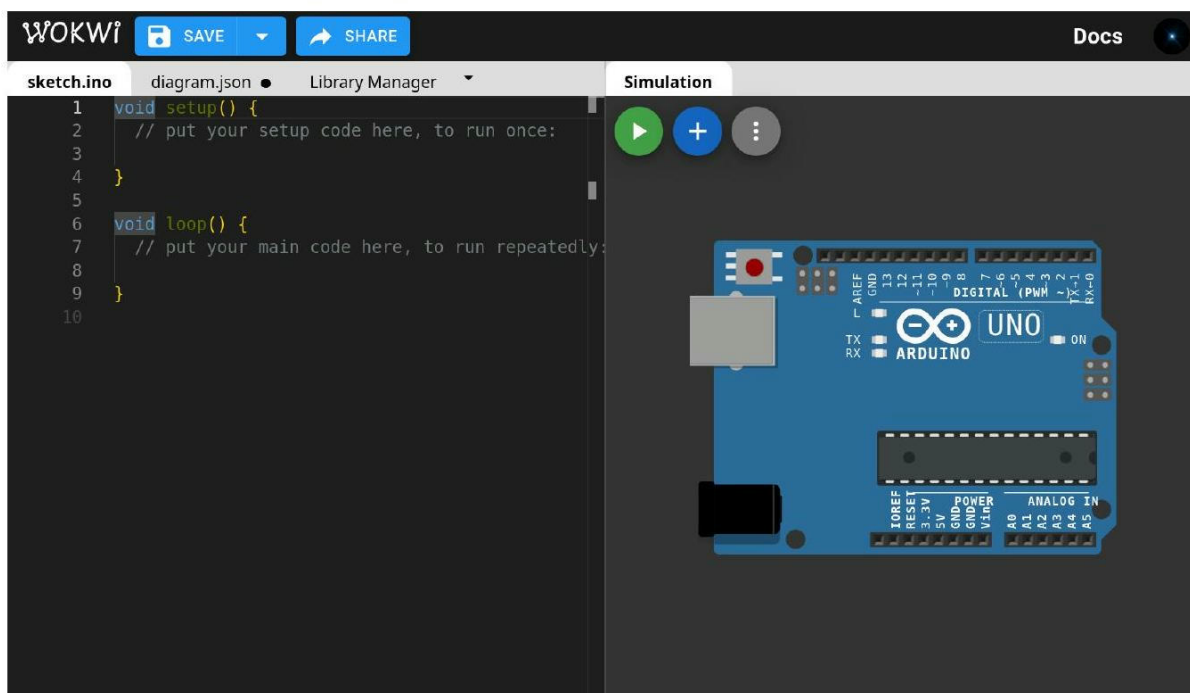
Выберите в меню слева семейство контроллеров Arduino



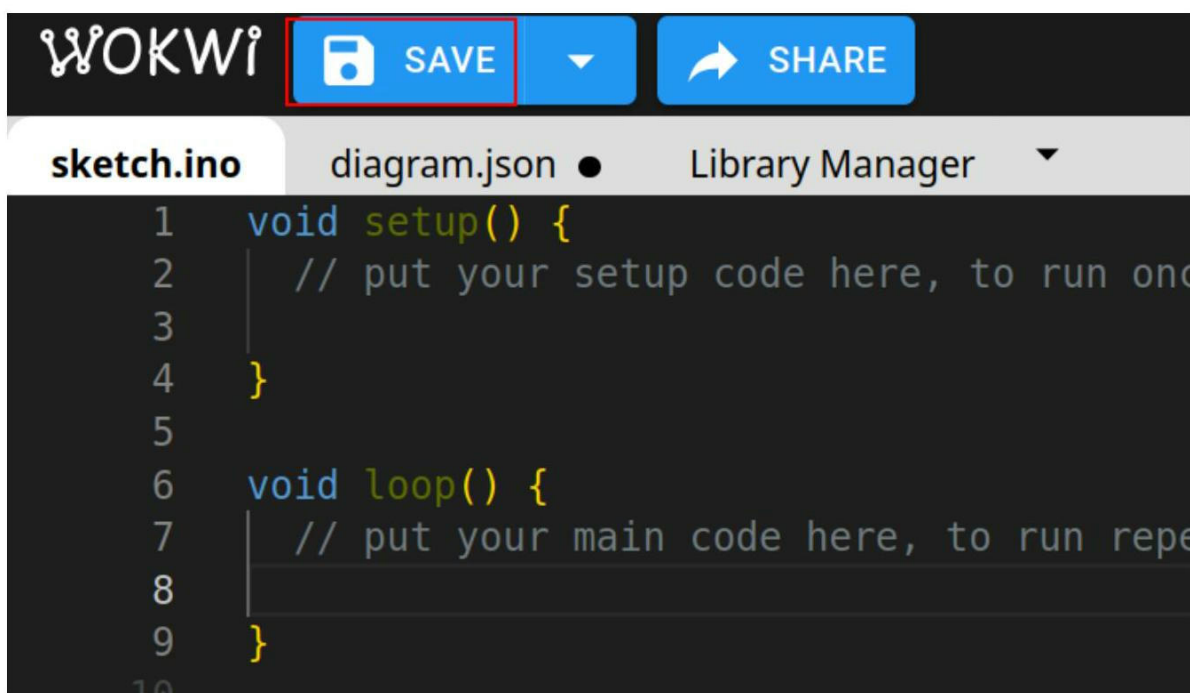
Выберите контроллер Arduino Uno Rev3



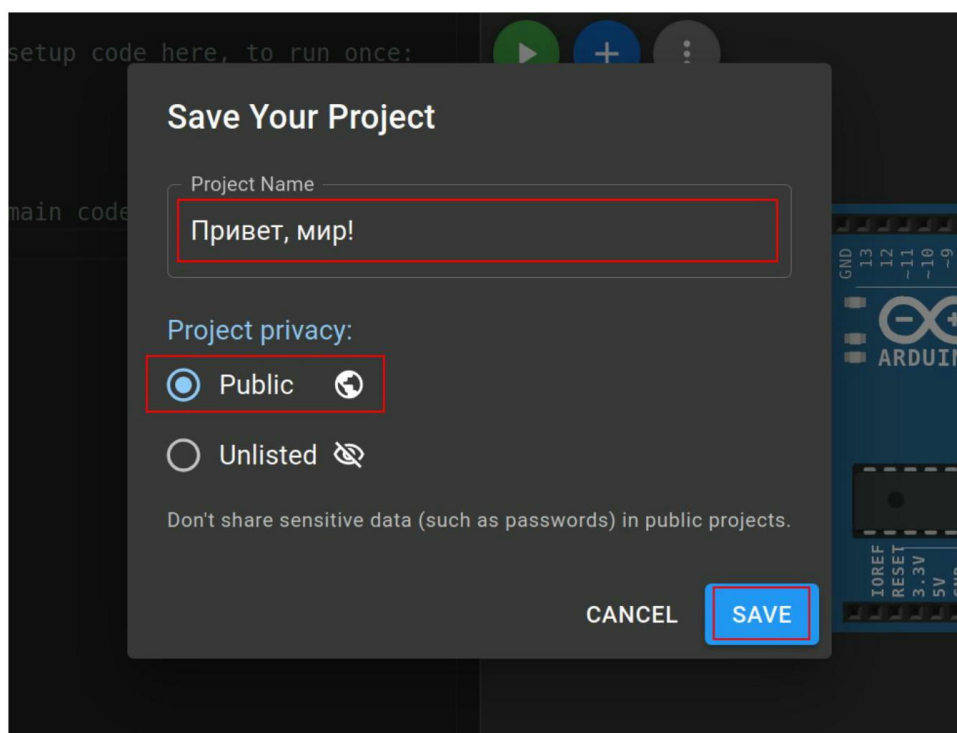
Созданный пустой проект!



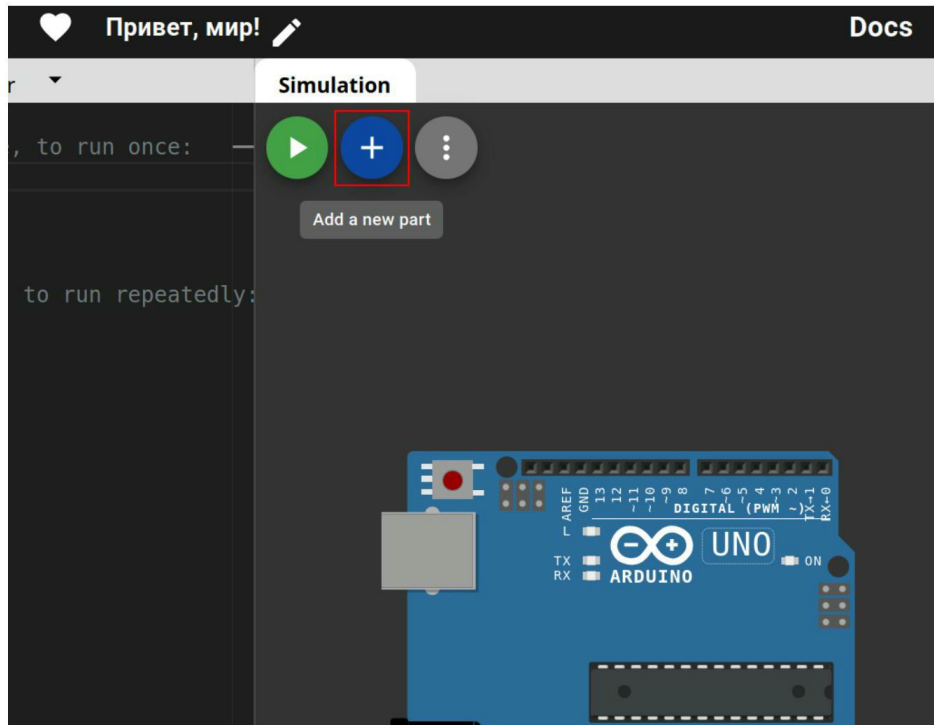
После внесения изменений сохраните проект с помощью кнопки SAVE



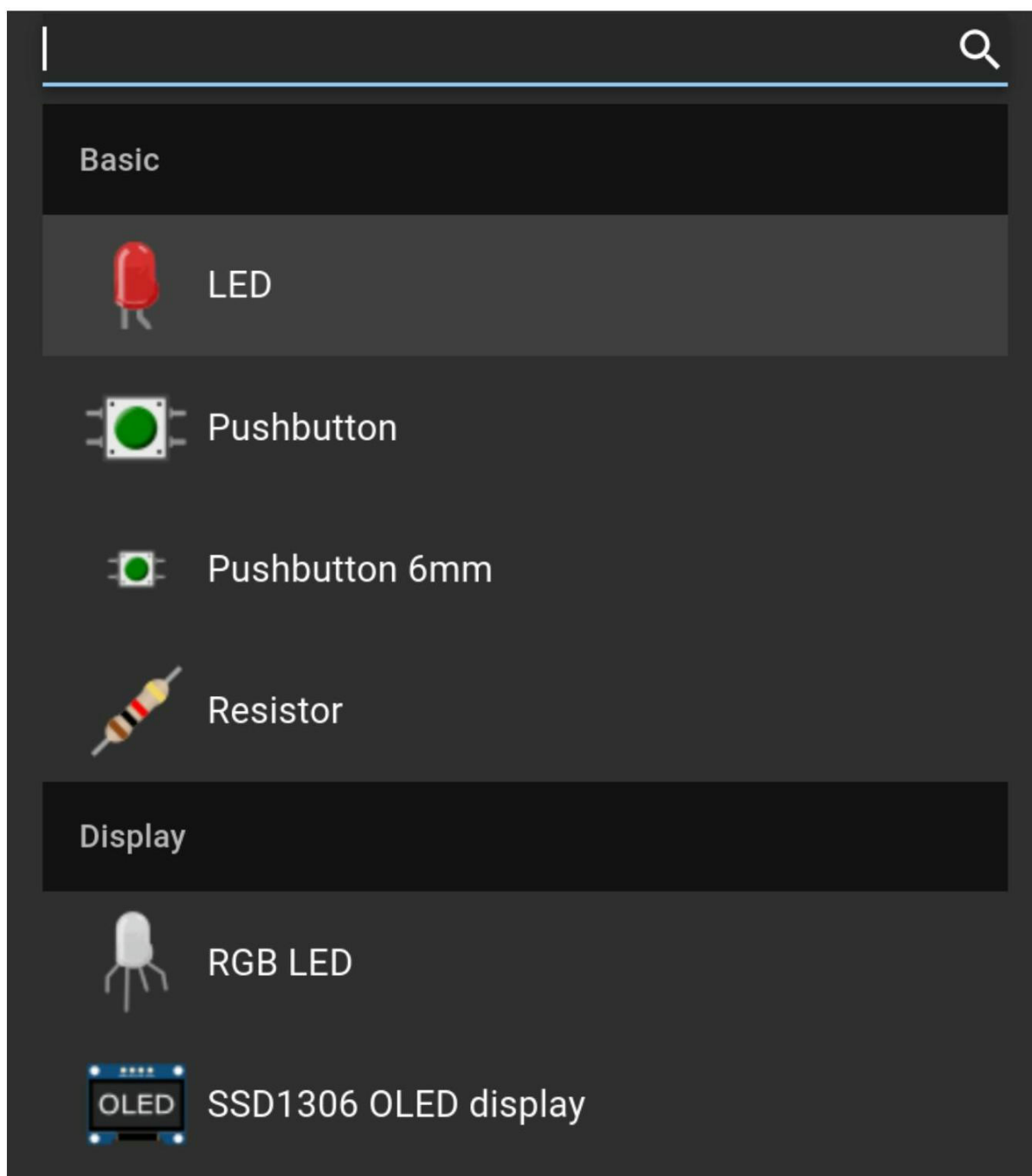
Введите название проекта и убедитесь, что он сохраняется как Public



Для добавления новой детали нажмите кнопку добавления



И выберите нужную деталь из выпадающего меню



Привет, мир!

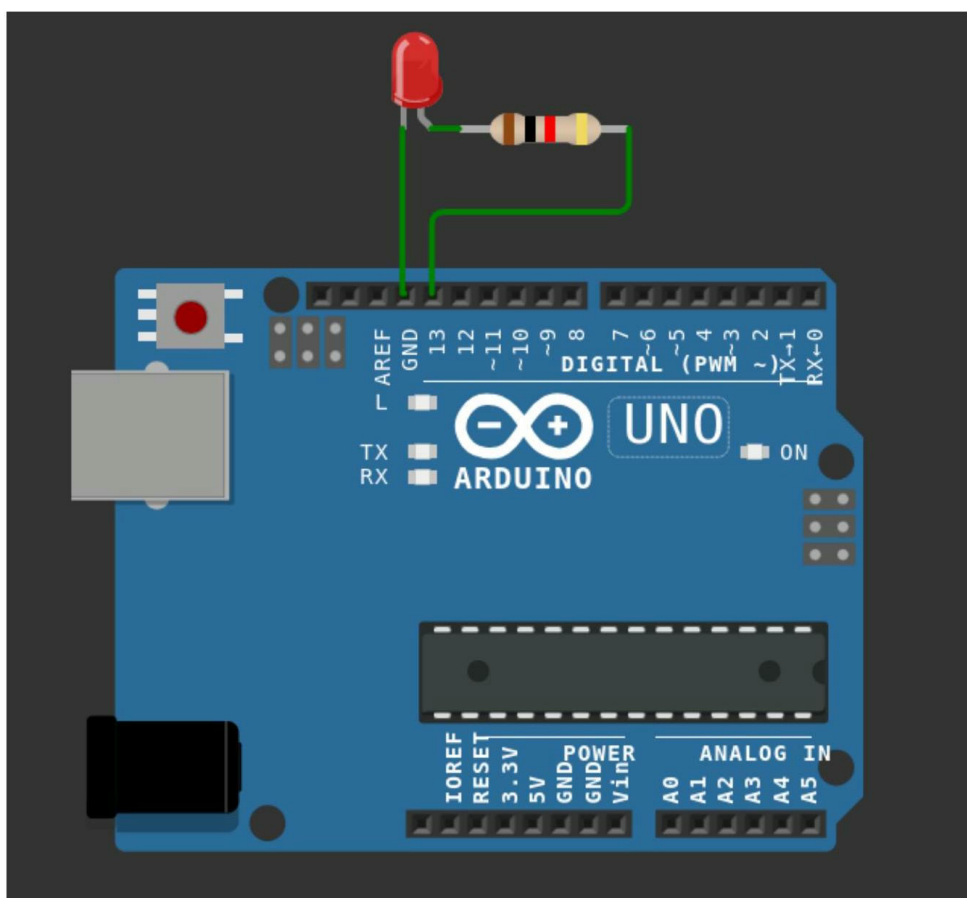


Рис. 1. Привет, мир!

```
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()
```

Плавный маяк

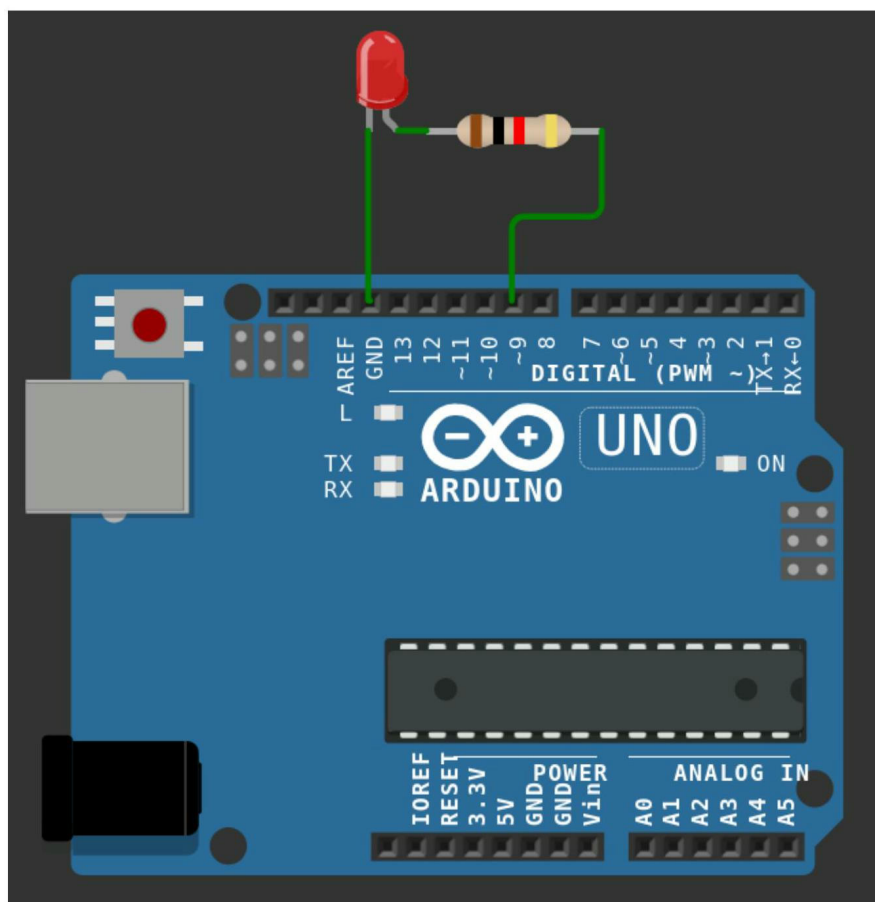


Рис. 2. Плавный маяк

```
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 }
```

Регулируемая лампочка

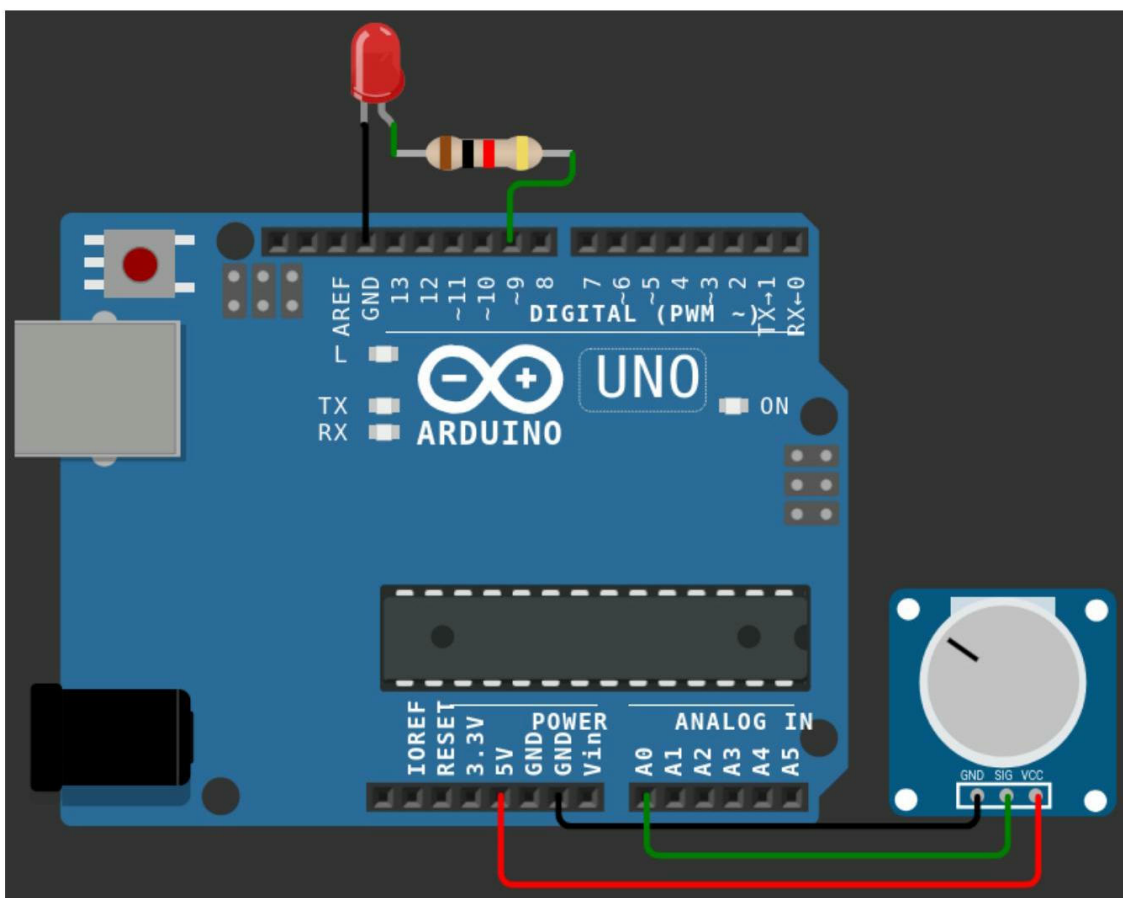


Рис. 3. Регулируемая лампочка

```
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 }
```

Автоматический ночник

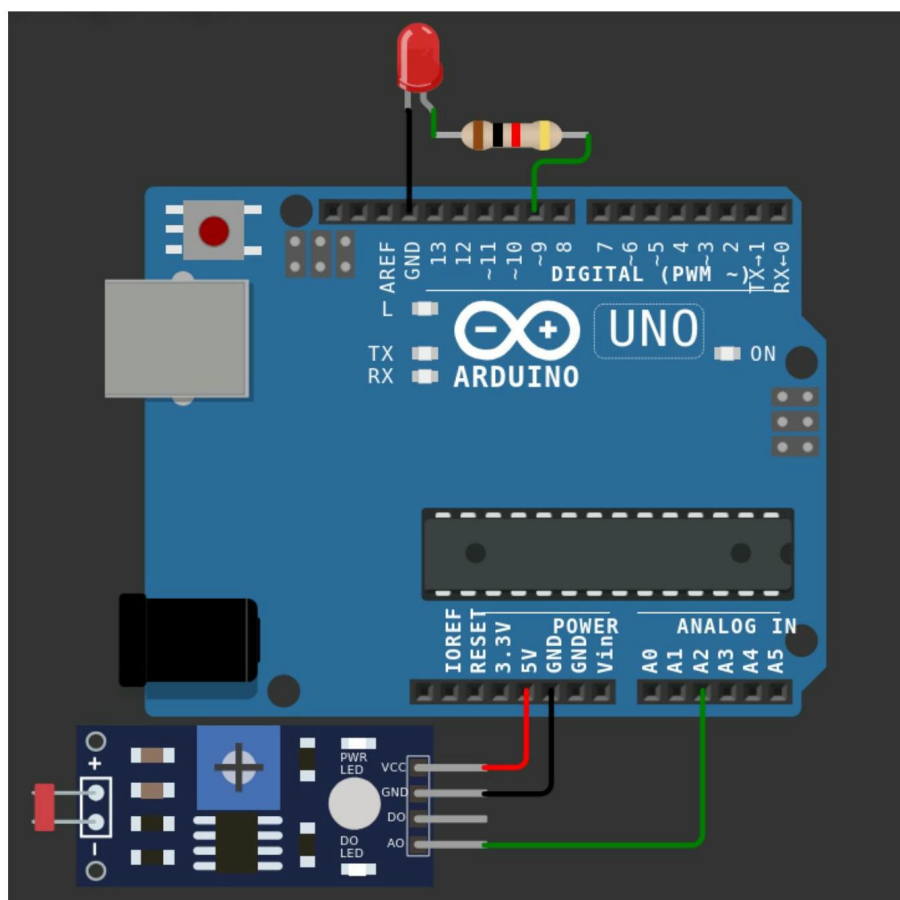


Рис. 4. Автоматический ночник

```
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 }
```

Пульсар

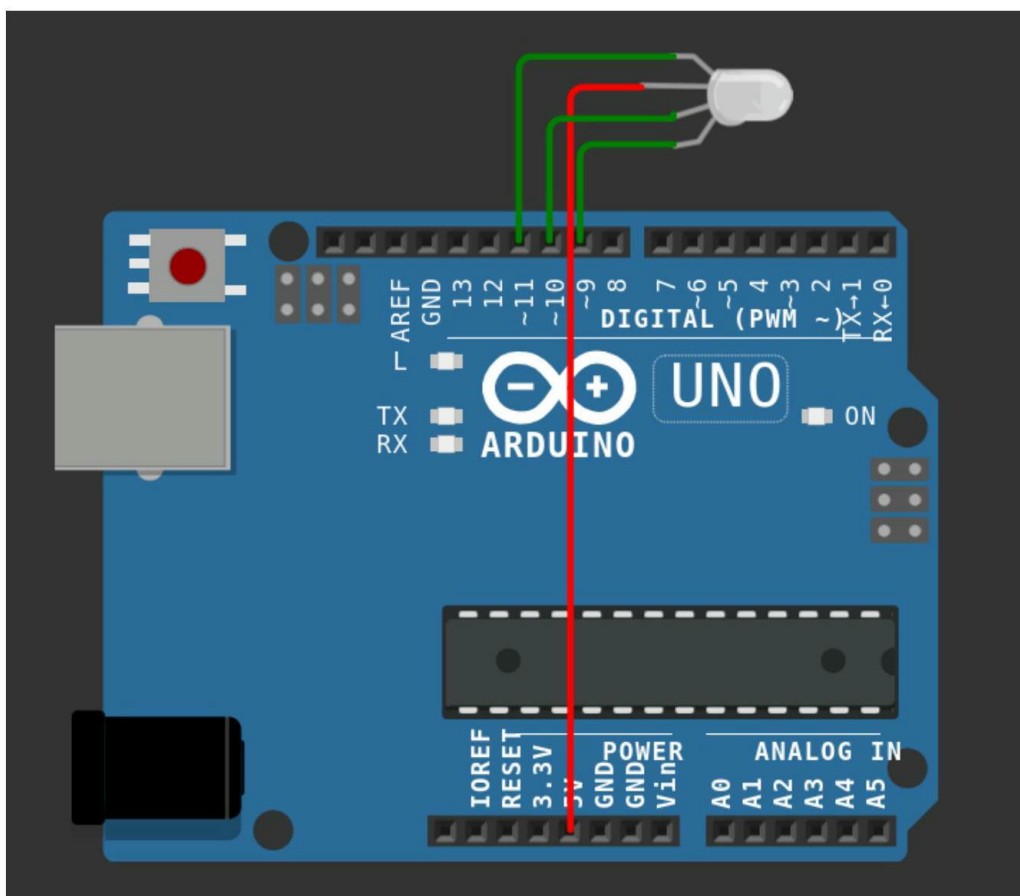


Рис. 5. Пульсар

```
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 }
```

Эквалайзер

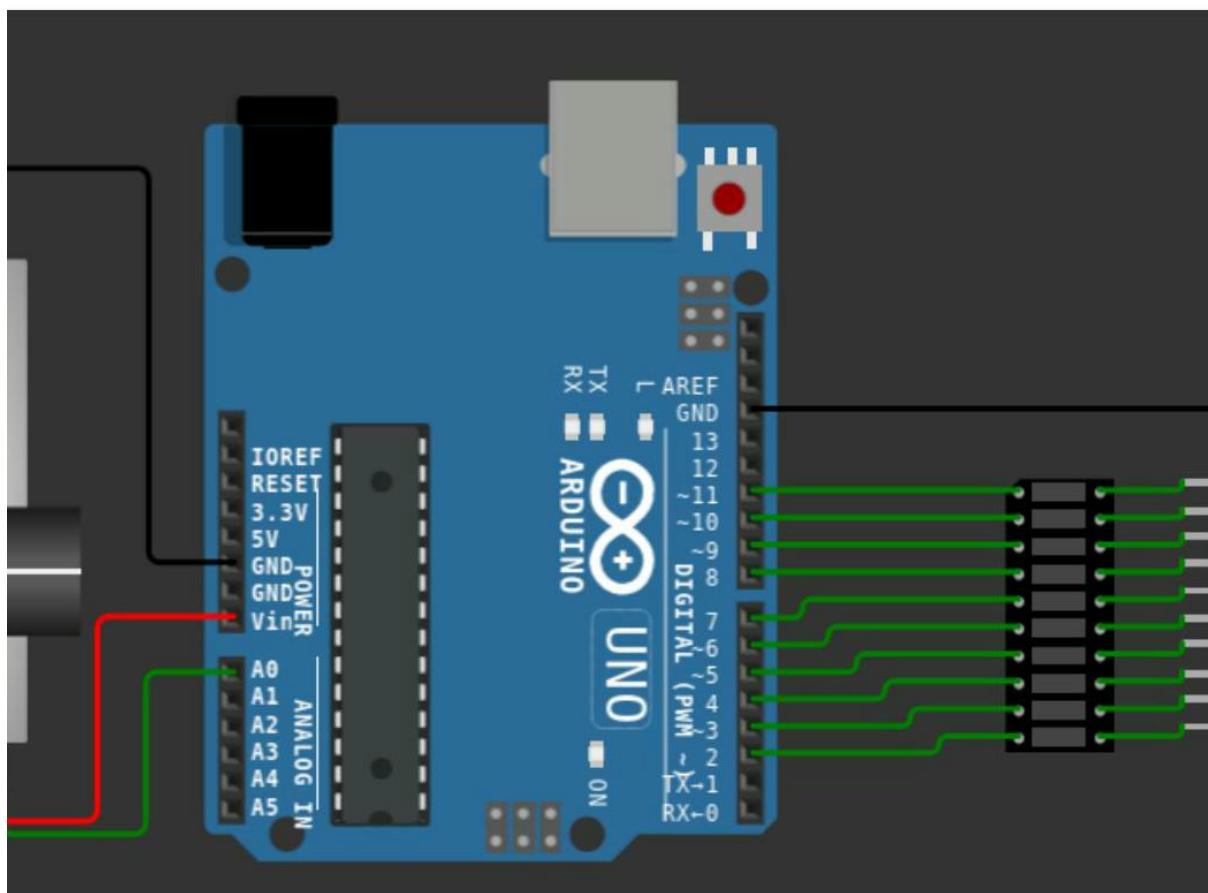


Рис. 6. Эквалайзер

```
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
6    uses
7    10
8    LEDs, you can use any number by changing the LED count and the pins
9    in the
10   array.
11   This method can be used to control any series of digital outputs
12   that
13   depends
14   on an analog input.
15   The circuit:
16   - LEDs from pins 2 through 11 to ground
17   created 4 Sep 2010
18   by Tom Igoe
19   This example code is in the public domain.
20   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 are attached
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  }

```

Кнопки

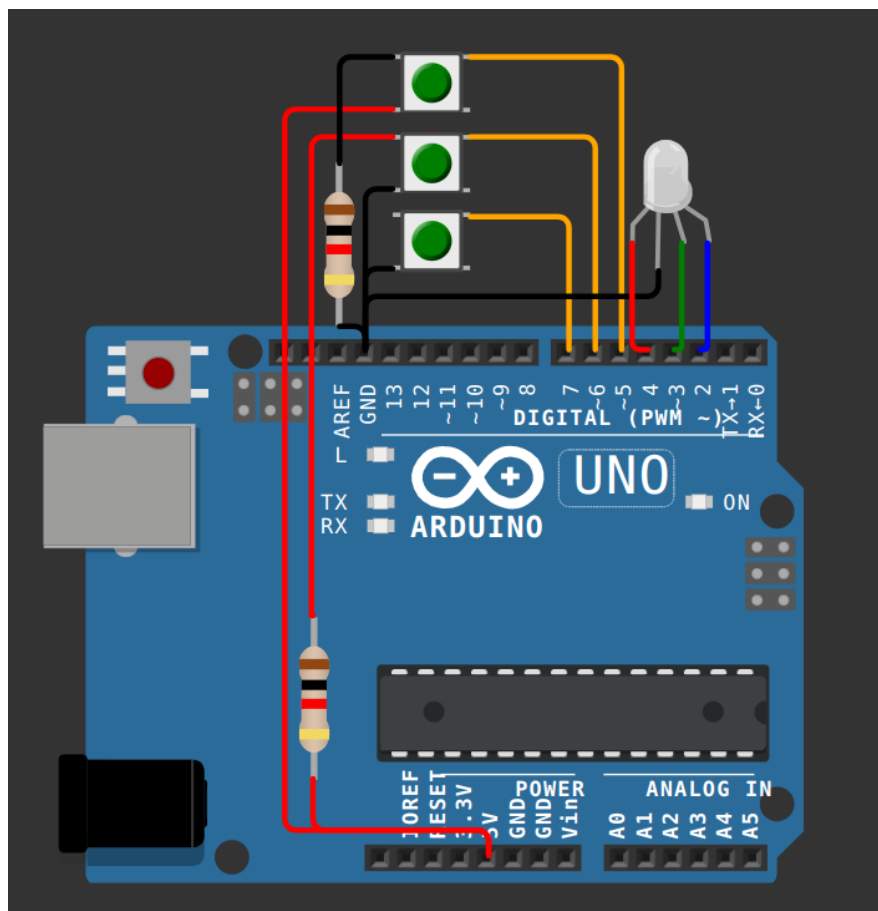


Рис. 7. Кнопки

```
1 #define BUTTON_EXT_PULLDOWN_PIN 5
2 #define BUTTON_EXT_PULLUP_PIN 6
3 #define BUTTON_INP_PULLUP_PIN 7
4
5 #define LED_R_PIN 4
6 #define LED_G_PIN 3
7 #define LED_B_PIN 2
8
9 void setup()
10 {
11     pinMode(BUTTON_EXT_PULLDOWN_PIN, INPUT);
12     pinMode(BUTTON_EXT_PULLUP_PIN, INPUT);
13     pinMode(BUTTON_INP_PULLUP_PIN, INPUT_PULLUP);
14
15     pinMode(LED_R_PIN, OUTPUT);
16     pinMode(LED_G_PIN, OUTPUT);
17     pinMode(LED_B_PIN, OUTPUT);
18 }
19
20 void loop()
```

```
21 {
22     bool button_extpulldown = digitalRead(BUTTON_EXT_PULLDOWN_PIN);
23     bool button_extpullup = digitalRead(BUTTON_EXT_PULLUP_PIN);
24     bool button_inpullup = digitalRead(BUTTON_INP_PULLUP_PIN);
25
26     digitalWrite(LED_R_PIN, button_extpulldown);
27     digitalWrite(LED_G_PIN, button_extpullup);
28     digitalWrite(LED_B_PIN, button_inpullup);
29
30     delay(10);
31 }
32
```

Механизированный флаг

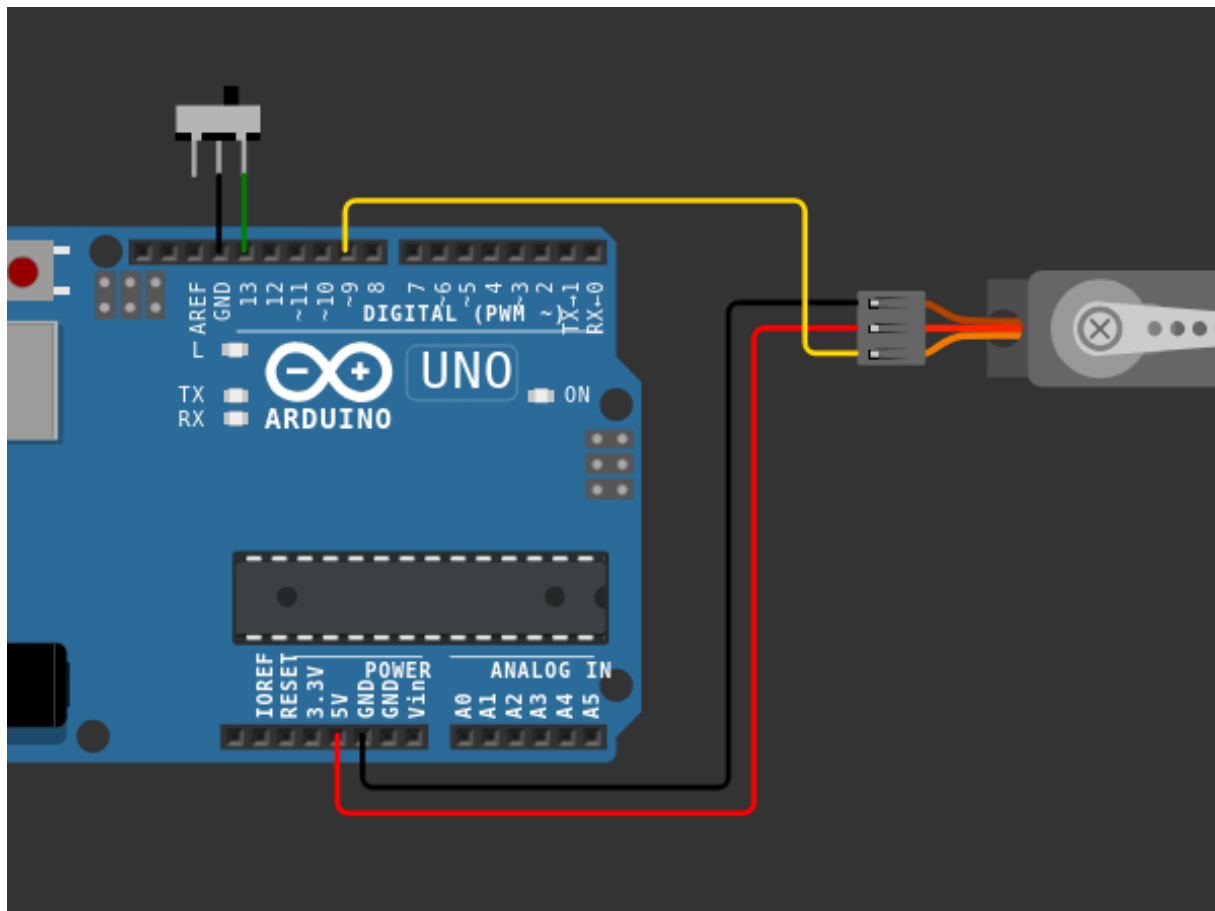


Рис. 8. Механизированный флаг

```
1  #include <Servo.h>
2
3  #define SERVO_PIN 9
4  #define SWITCH_PIN 13
5
6  Servo servo;
7
8  void setup()
9  {
10     servo.attach(SERVO_PIN);
11     pinMode(SWITCH_PIN, INPUT_PULLUP);
12 }
13
14 void loop()
15 {
16     if(digitalRead(SWITCH_PIN))
17     {
18         servo.write(0);
19     }
20     else
```

```
21  {  
22    servo.write(90);  
23  }  
24  delay(100);  
25 }  
26
```

Автоматические ворота

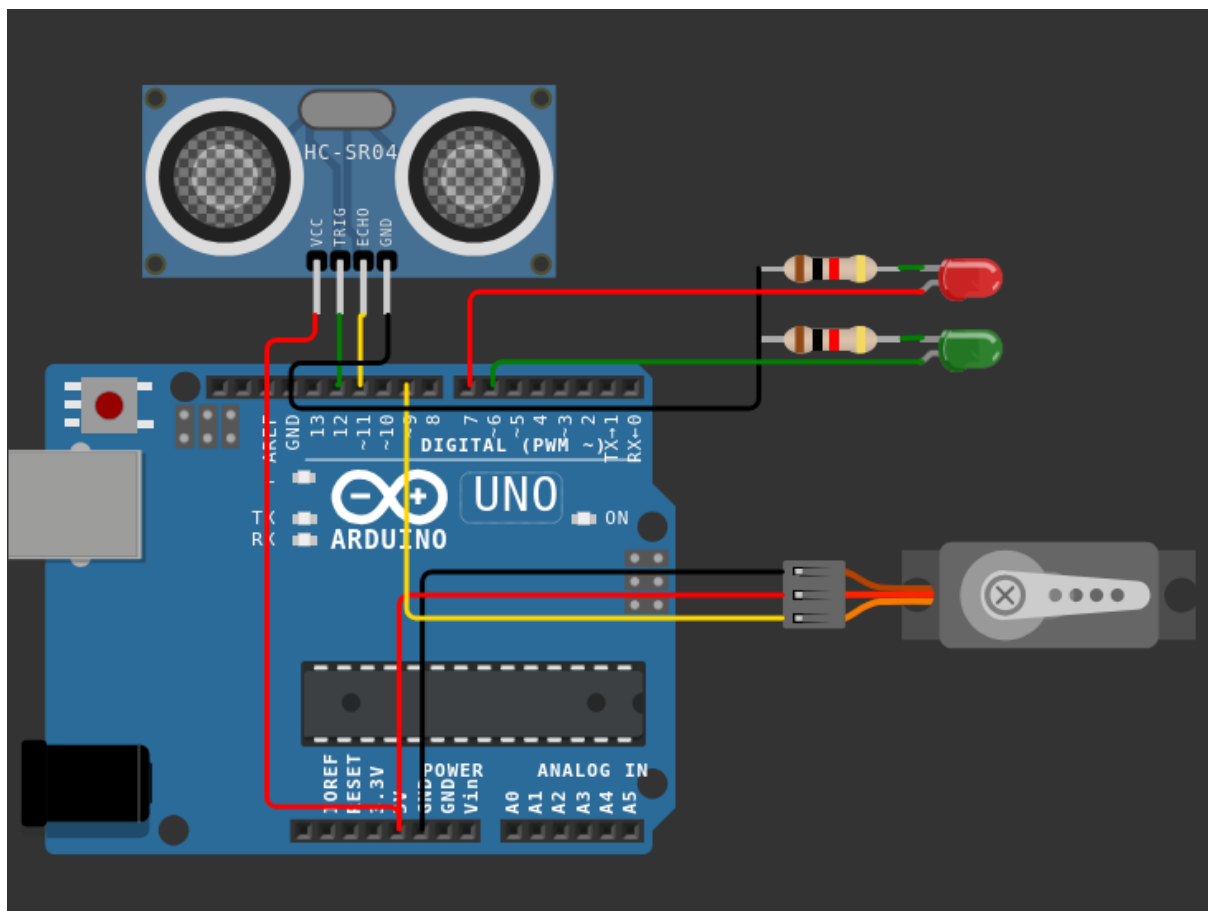


Рис. 9. Автоматические ворота

```
1  #include <Servo.h>
2
3  #define SONAR_ECHO 11
4  #define SONAR_TRIG 12
5
6  #define SERVO_PIN 9
7
8  #define RED_PIN 7
9  #define GREEN_PIN 6
10
11 #define ALARM_DIST 200 // [cm]
12
13 Servo servo;
14
15 void setup()
16 {
17     pinMode(SONAR_ECHO, INPUT);
18     pinMode(SONAR_TRIG, OUTPUT);
19
20     servo.attach(SERVO_PIN);
```

```

21
22   pinMode(REDA_PIN, OUTPUT);
23   pinMode(GREEN_PIN, OUTPUT);
24 }
25
26 int getDistanceCM()
27 {
28     // https://docs.wokwi.com/parts/wokwi-hc-sr04
29
30     // Start a new measurement:
31     digitalWrite(SONAR_TRIG, HIGH);
32     delayMicroseconds(10);
33     digitalWrite(SONAR_TRIG, LOW);
34
35     // Read the result:
36     int duration = pulseIn(SONAR_ECHO, HIGH);
37     int distanceCM = duration / 58;
38
39     return distanceCM;
40 }
41
42 void loop()
43 {
44     int distance = getDistanceCM();
45
46     bool red_light = false;
47     bool green_light = true;
48     bool gate_open = true;
49
50     if(distance < ALARM_DIST)
51     {
52         red_light = true;
53         green_light = false;
54         gate_open = false;
55     }
56
57     digitalWrite(REDA_PIN, red_light);
58     digitalWrite(GREEN_PIN, green_light);
59     if(gate_open)
60     {
61         servo.write(0);
62     }
63     else
64     {
65         servo.write(90);
66     }
67 }
68

```

Калькулятор

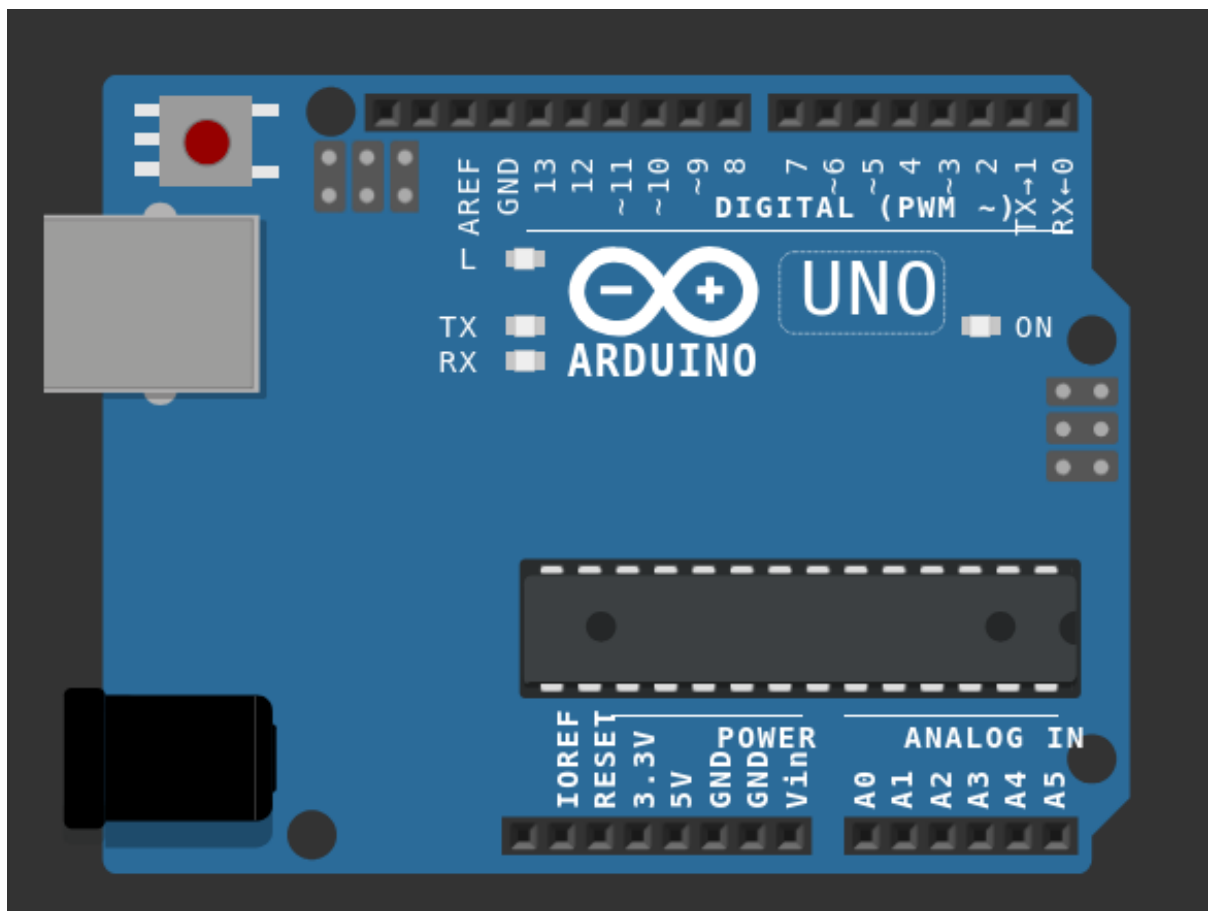


Рис. 10. Калькулятор

```
1 void setup()
2 {
3   Serial.begin(115200);
4   Serial.setTimeout(100000); // 100 sec
5 }
6
7 void loop()
8 {
9   Serial.print("Введите первое число: ");
10
11   float a = Serial.parseFloat();
12
13   Serial.println(a);
14
15   Serial.print("Введите второе число: ");
16
17   float b = Serial.parseFloat();
18
19   Serial.println(b);
20
```

```
21  float sum = a + b;
22
23  Serial.println("Сумма равна: " + String(sum));
24  Serial.println("Спасибо за использование калькулятора!");
25  Serial.println();
26  delay(2000);
27  }
28
```

Тестовое задание

Текст перед картинкой

- abc

- 1.
- 2.
- 3.

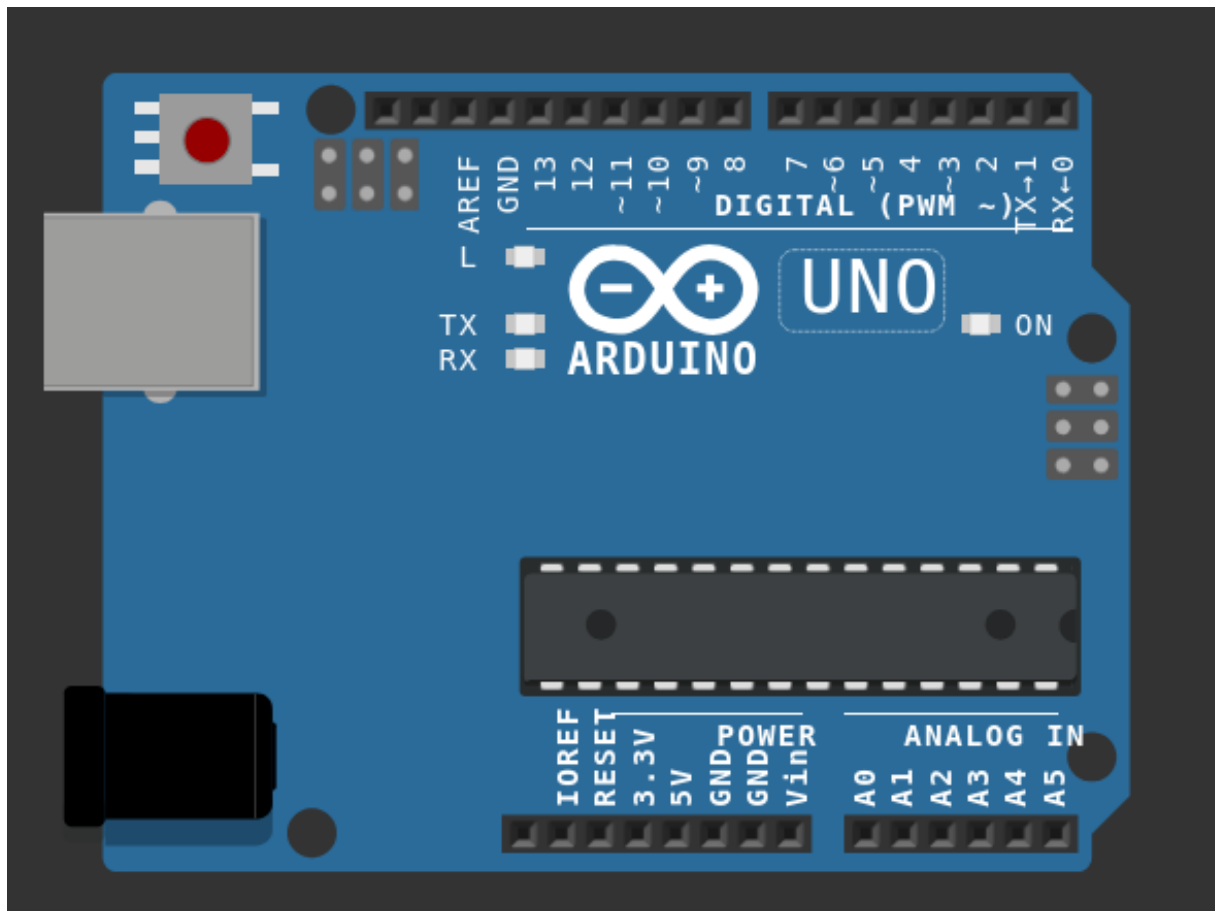


Рис. 11. Тестовое задание

Текст между картинкой и листингом

```
1 void setup()
2 {
3   Serial.begin(115200);
4   Serial.setTimeout(100000); // 100 sec
5 }
6
7 void loop()
8 {
9   Serial.print("Введите первое число: ");
10
```

```
11  float a = Serial.parseFloat();
12
13  Serial.println(a);
14
15  Serial.print("Введите второе число: ");
16
17  float b = Serial.parseFloat();
18
19  Serial.println(b);
20
21  float sum = a + b;
22
23  Serial.println("Сумма равна: " + String(sum));
24  Serial.println("Спасибо за использование калькулятора!");
25  Serial.println();
26  delay(2000);
27  }
28
```

Текст после листинга