

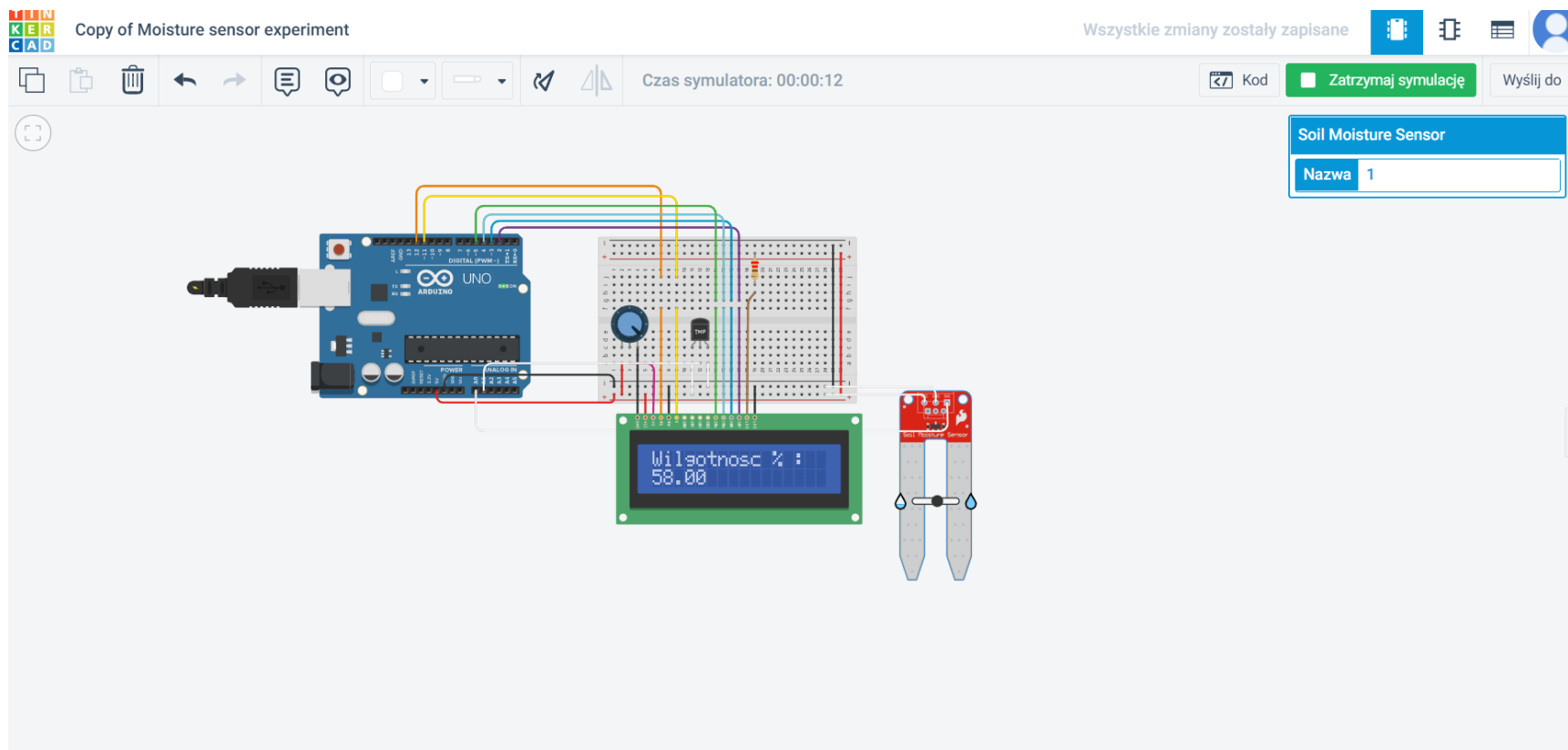
Pomiar wilgotności, temperatury, natężenia światła

Aplikacje systemów wbudowanych

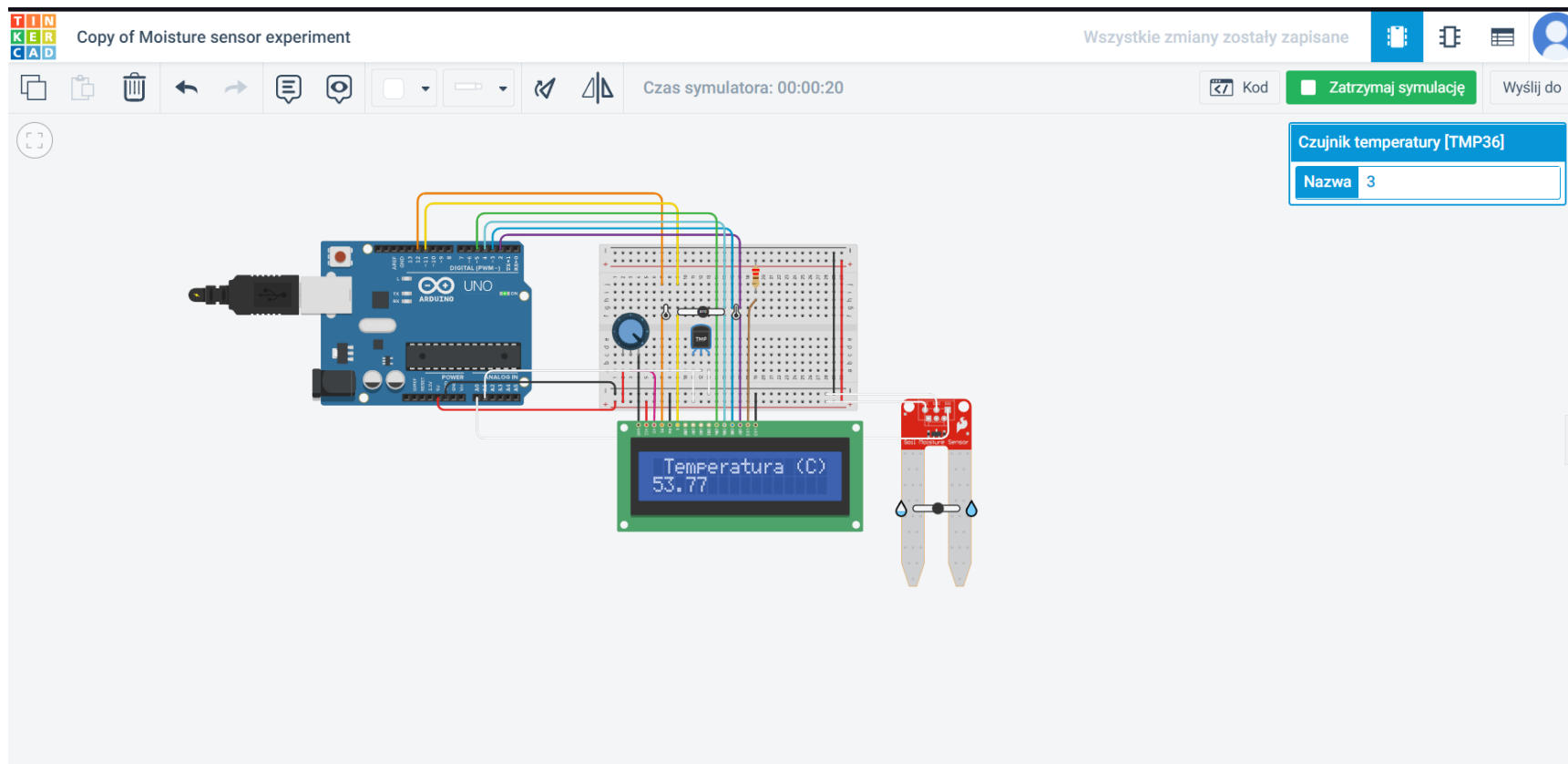
Zadania do wykonania

- 1. Odczytaj wartości temperatury i wilgotności. Pamiętaj o przeskalowaniu i odpowiednich jednostkach. (1 pkt.)
- 2. Stwórz regulator temperatury, który w zależności od zmierzonej wartości będzie uruchamiać chłodzenie (niebieska dioda) lub grzanie (dioda czerwona). Zakres zmian ustal samodzielnie. (2 pkt.)
- 3. Stwórz program, który w zależności od zmierzonego natężenia światła za pomocą fotorezystora, będzie zapalał stopniowo diody (bargraf). (2 pkt.)
- 4. Stwórz program, który dzięki zebranych pomiarom, będzie informował jaką mamy obecnie pogodę, np. „słoneczny ciepły dzień” lub „duże zachmurzenie”. Przygotuj min. 4 różne konfigurację. (2 pkt.)

Zadanie 1 - zdjęcie



Zadanie 1 - zdjęcie



Zadanie 1 - kod

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
    lcd.begin(16,2);
}

void loop()
{
    wilgotnosc();
    delay(5000);
    temperatura();
    delay(5000);
}

void wilgotnosc()
{
    int hum;

    hum=analogRead(A0);
    hum=map(hum,0,890,5,1000);
    lcd.setCursor(0,0);
    lcd.print("Wilgotnosc % : ");
    lcd.setCursor(0,1);
    lcd.print((float)(hum/10.0));
    delay(200);
}

void temperatura()
{
    int val=analogRead(A1);
    float cel=(float)val*500.0/1024.0-60.0;
    lcd.setCursor(0, 0);
    lcd.print(" Temperatura (C): ");
    lcd.setCursor(0, 1);
    lcd.print(cel);

    delay(1000);
}
```

Zadanie 2

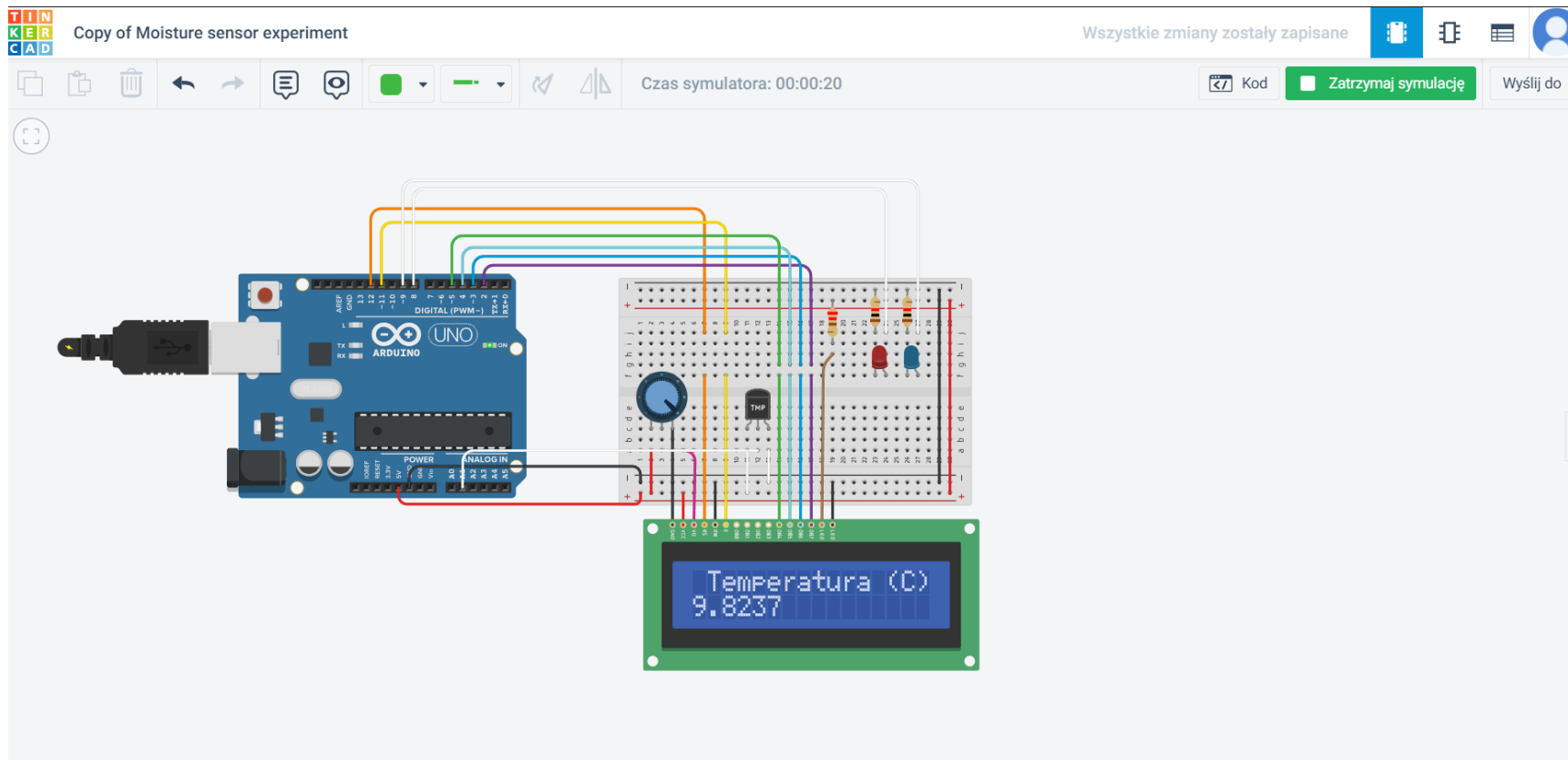
Zakres temperatur:

Grzanie: temperatura ≤ 18

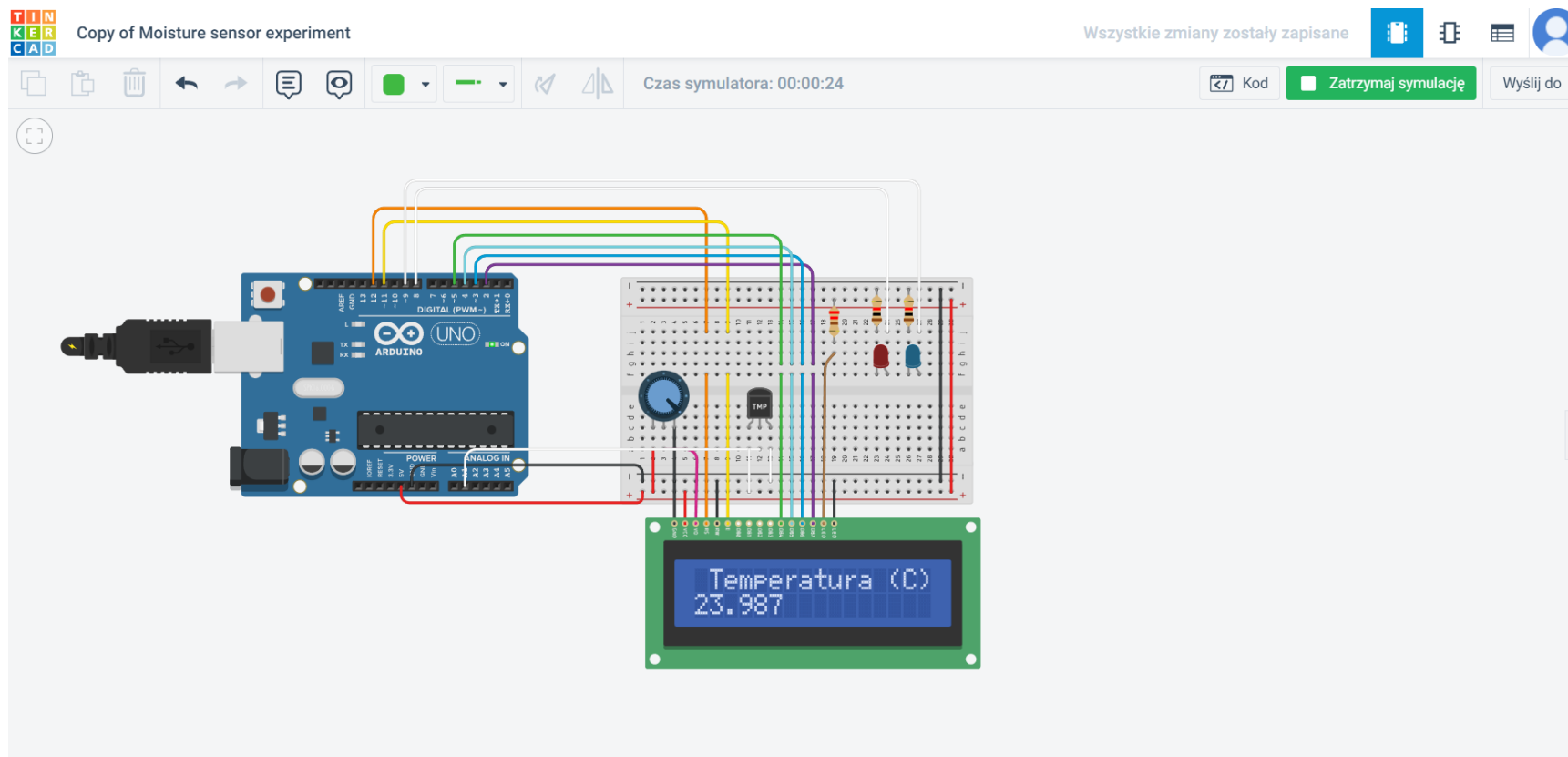
Idealna: temperatura od 19 do 25

Chłodzenie: temperatura > 26

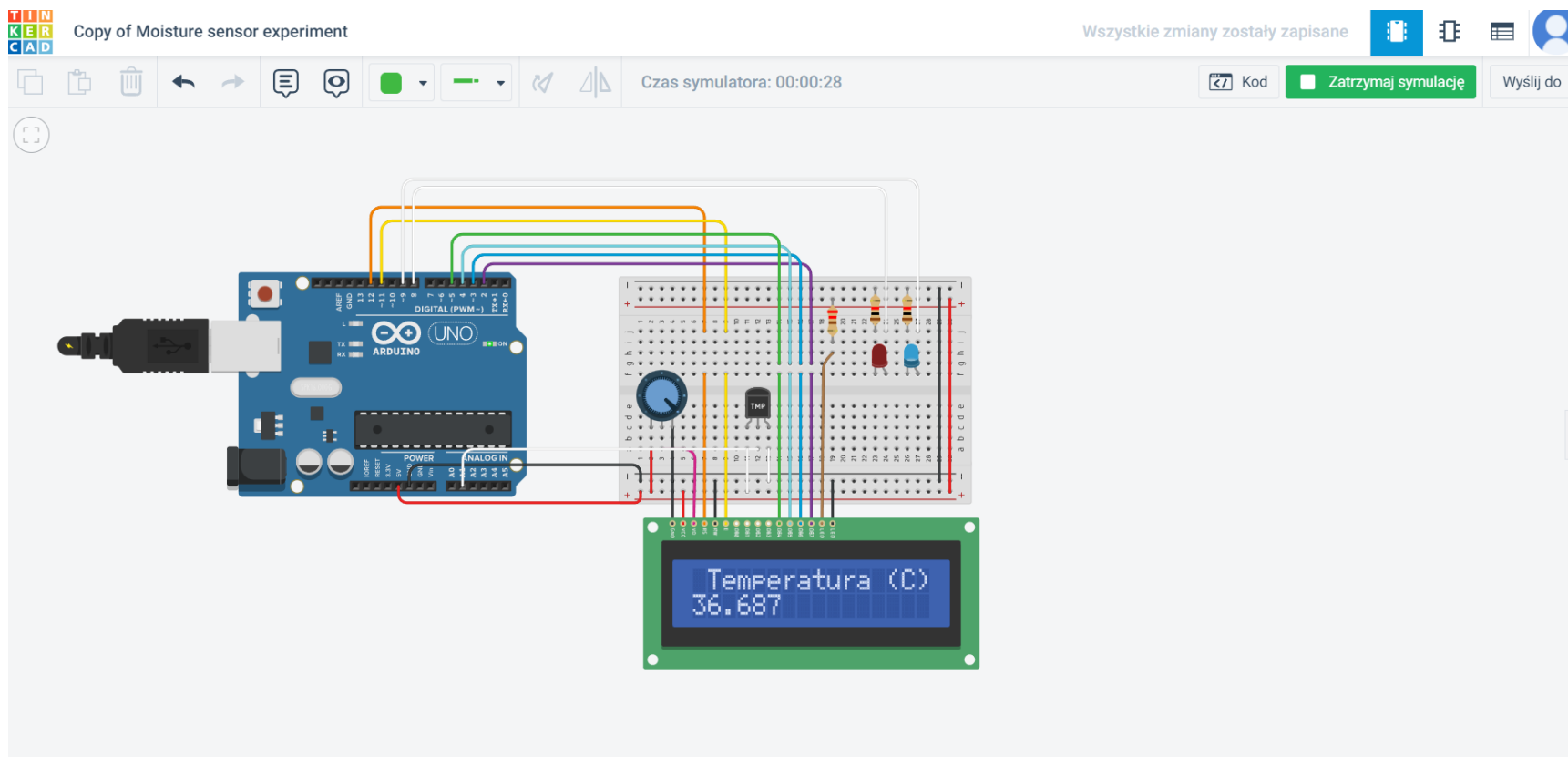
Zadanie 2 - zdjęcie



Zadanie 2 - zdjęcie



Zadanie 2 - zdjęcie



Zadanie 2 - kod

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int ledRED = 8;
int ledBLUE = 9;

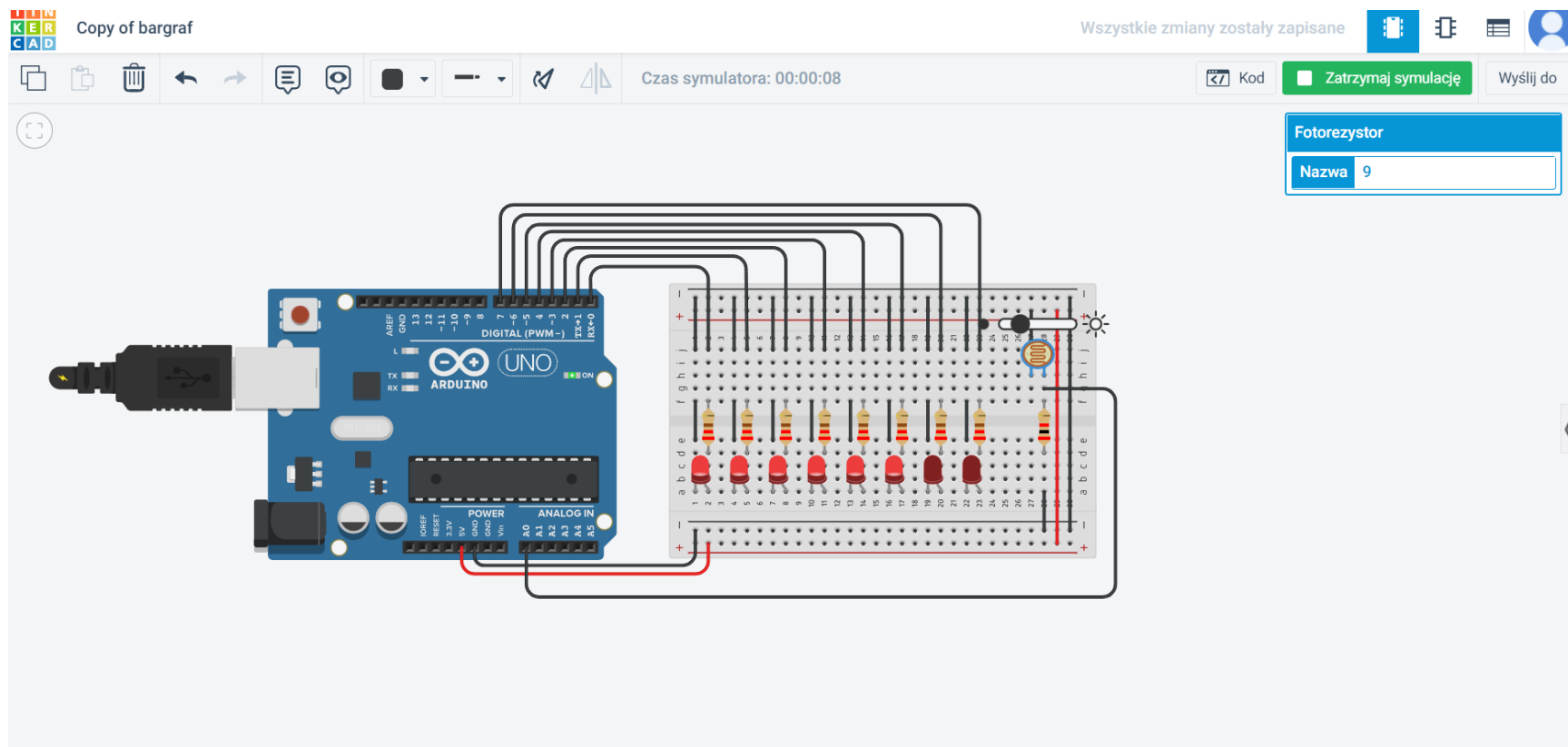
void setup()
{
  lcd.begin(16,2);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
}

void loop()
{
  temperatura();
}

void temperatura()
{
  int val=analogRead(A1);
  float cel=(float)val*500.0/1024.0-60.0;
  lcd.setCursor(0, 0);
  lcd.print(" Temperatura (C): ");
  lcd.setCursor(0, 1);
  lcd.print(cel);

  if(cel <= 18){
    digitalWrite(9, LOW);
    digitalWrite(8, HIGH);
  }
  if(cel > 19 || cel < 25){
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
  }
  if(cel > 26){
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
  }
}
```

Zadanie 3 - zdjęcie



Zadanie 3 - kod

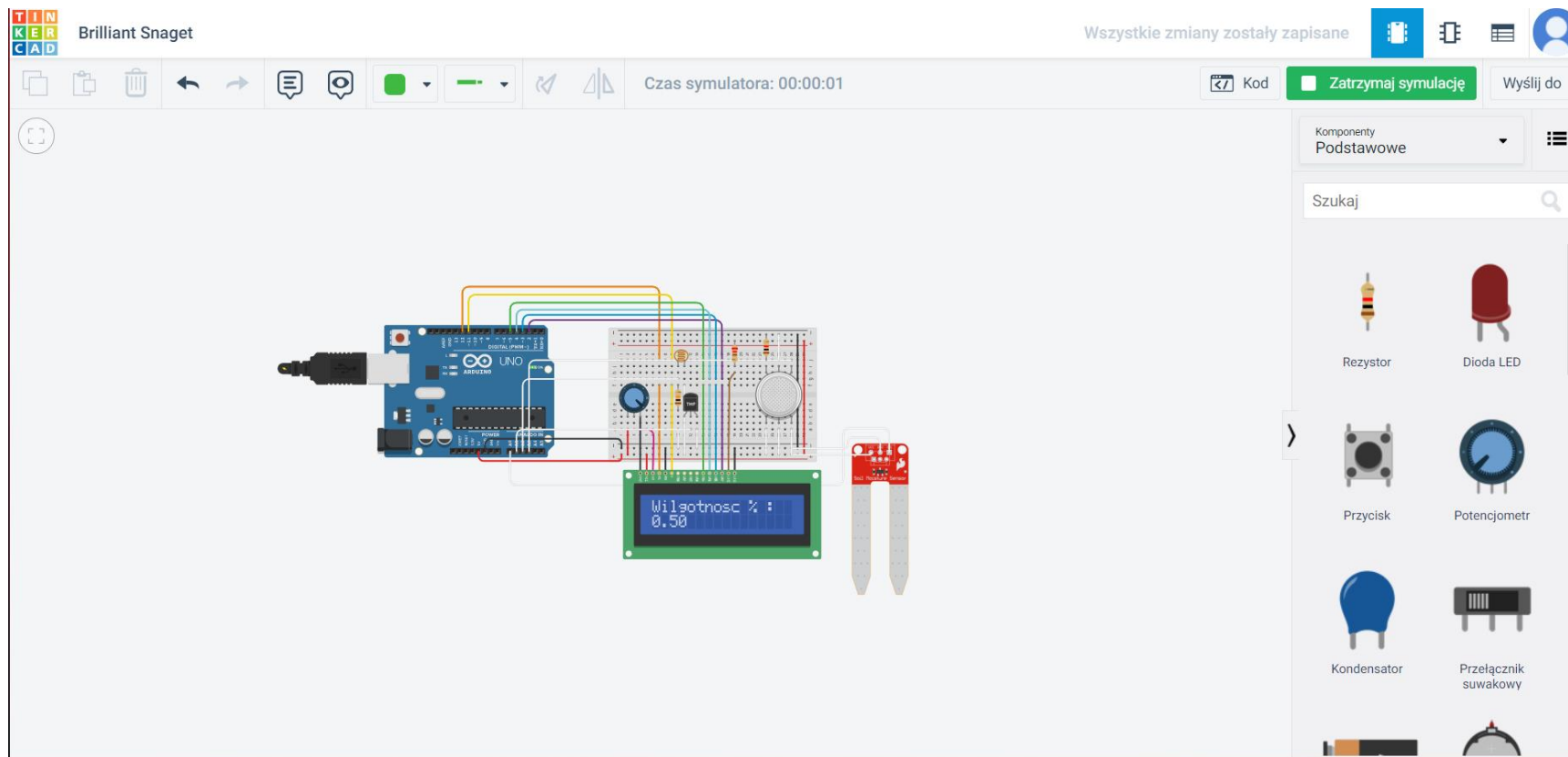
```
#include <avr/io.h>
```

```
uint8_t values[9] = {0x00, 0x01, 0x03, 0x07, 0x0F,  
                     0x1F, 0x3F, 0x7F, 0xFF};
```

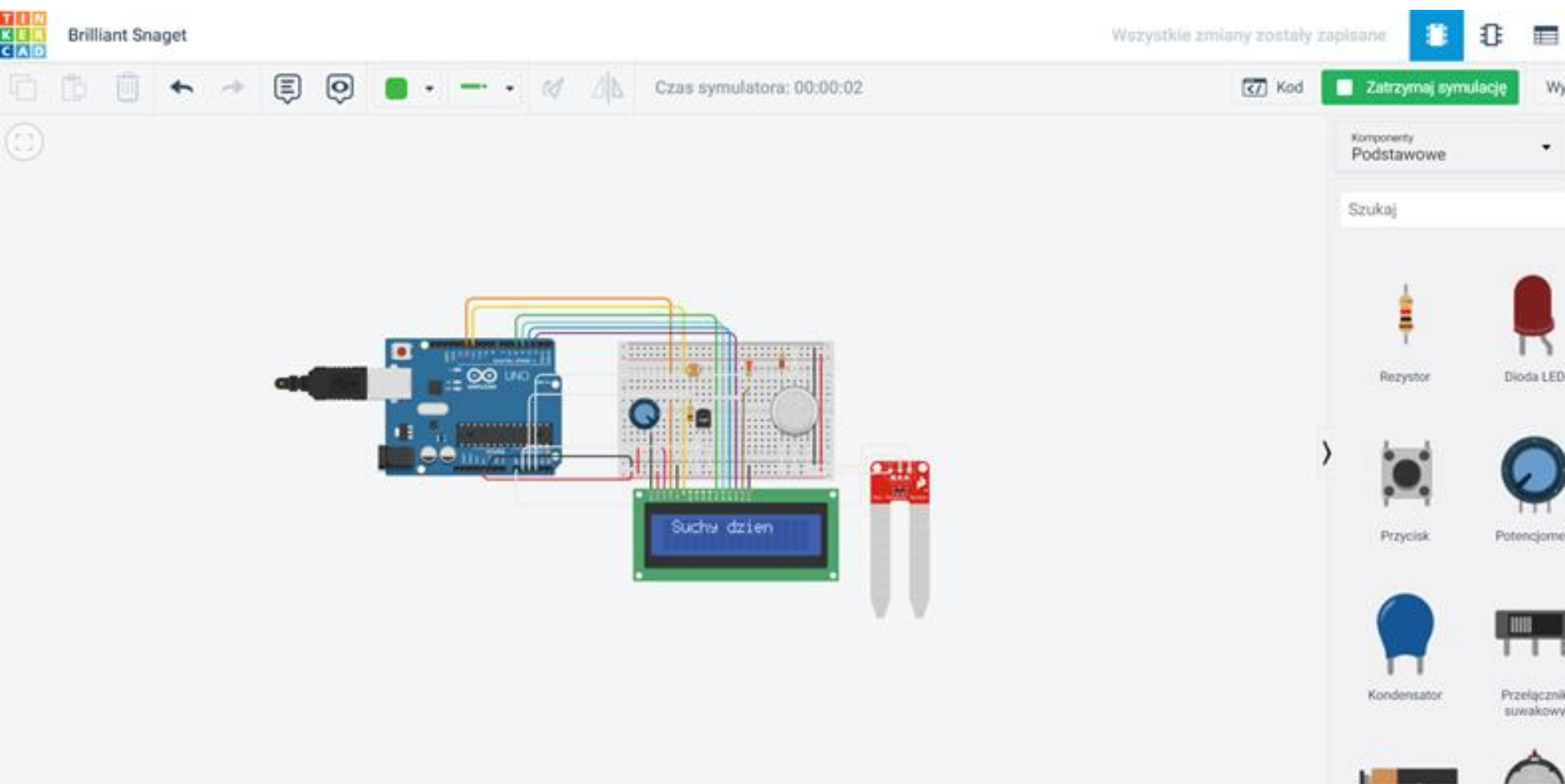
```
void setup()  
{  
    DDRD = 0xFF;  
    PORTD = 0x00;  
}
```

```
void loop()  
{  
    int index = map(analogRead(A0), 0, 818, 0, 9);  
    PORTD = values[index];  
}
```

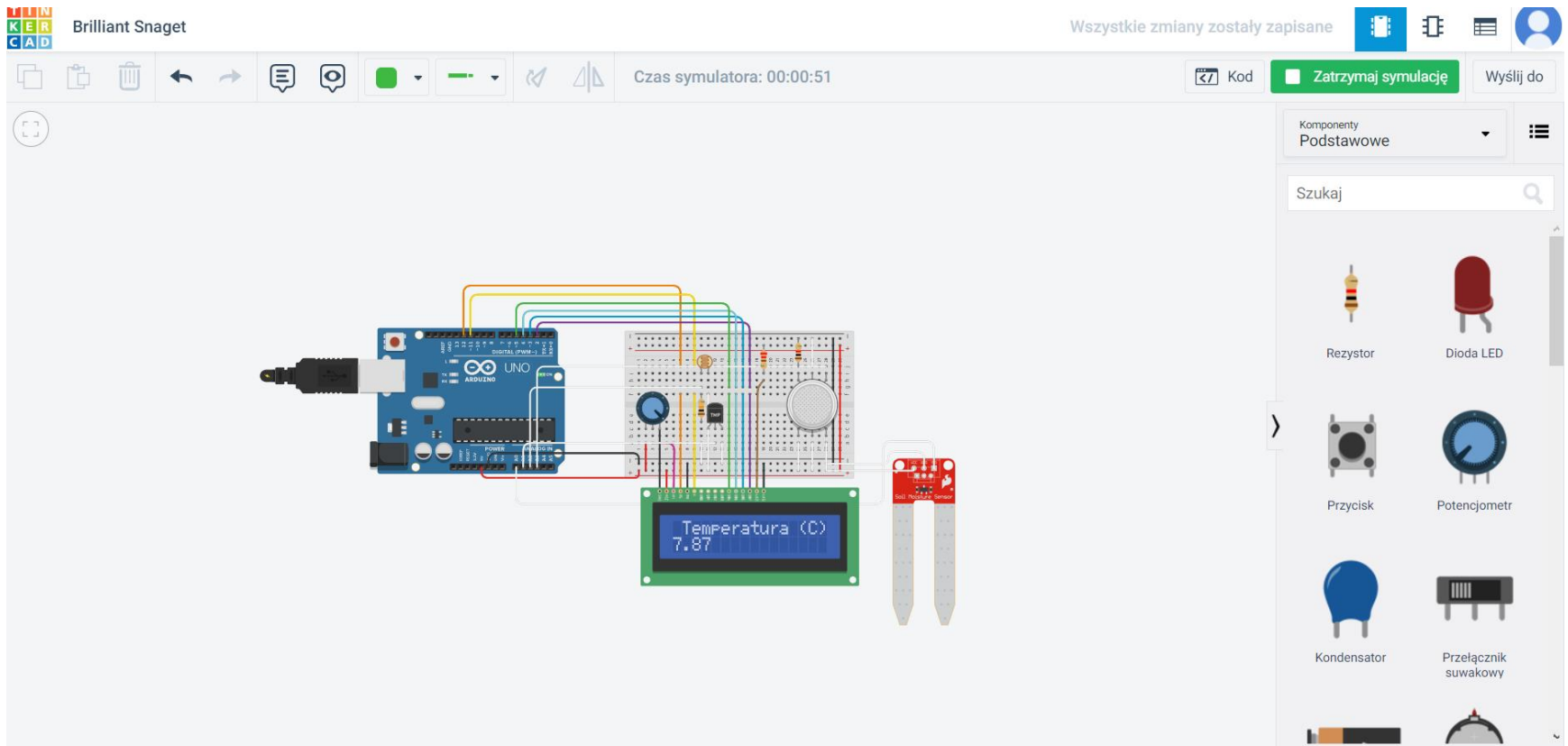
Zadanie 4 – zdjęcie wilgoć



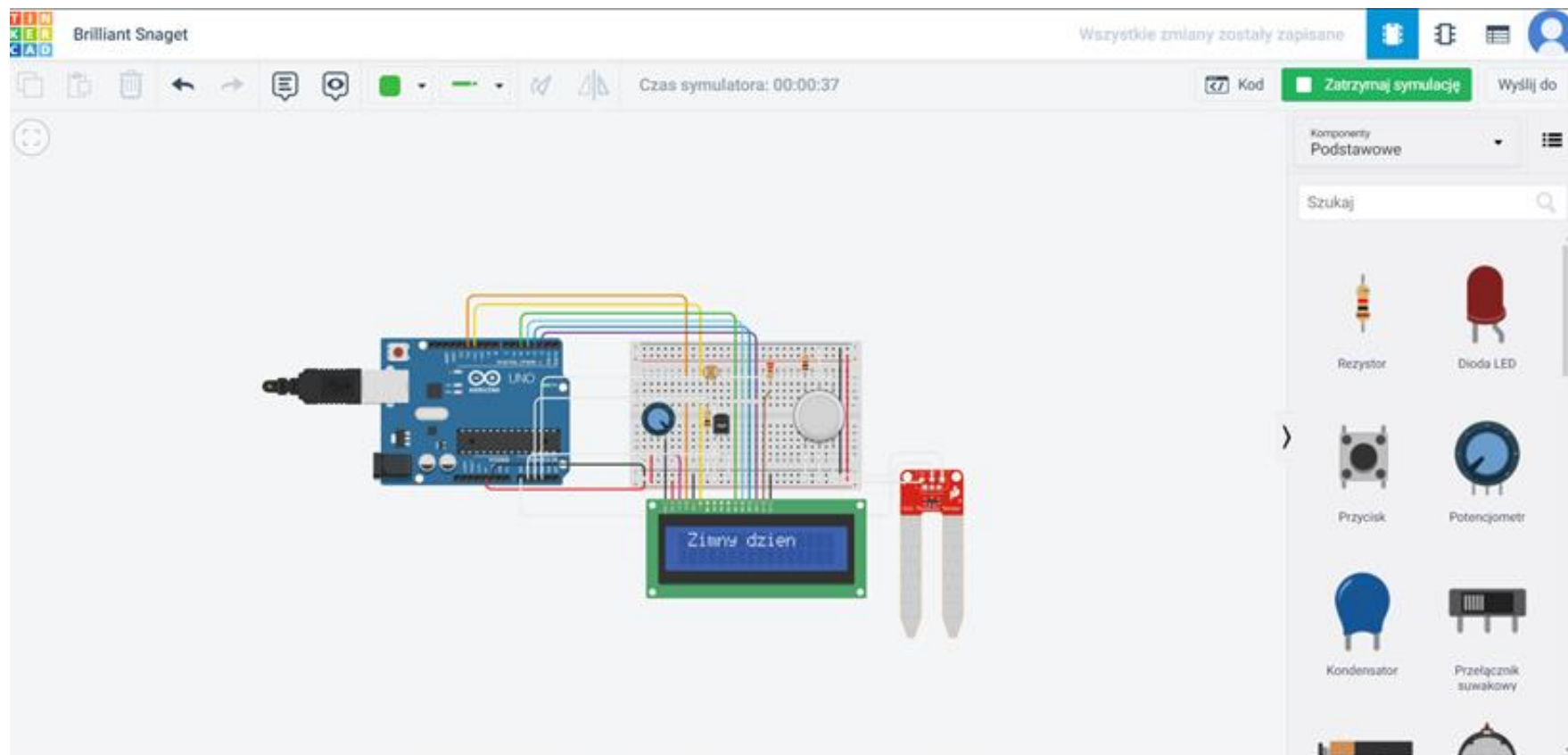
Zadanie 4 – zdjęcie wilgoć



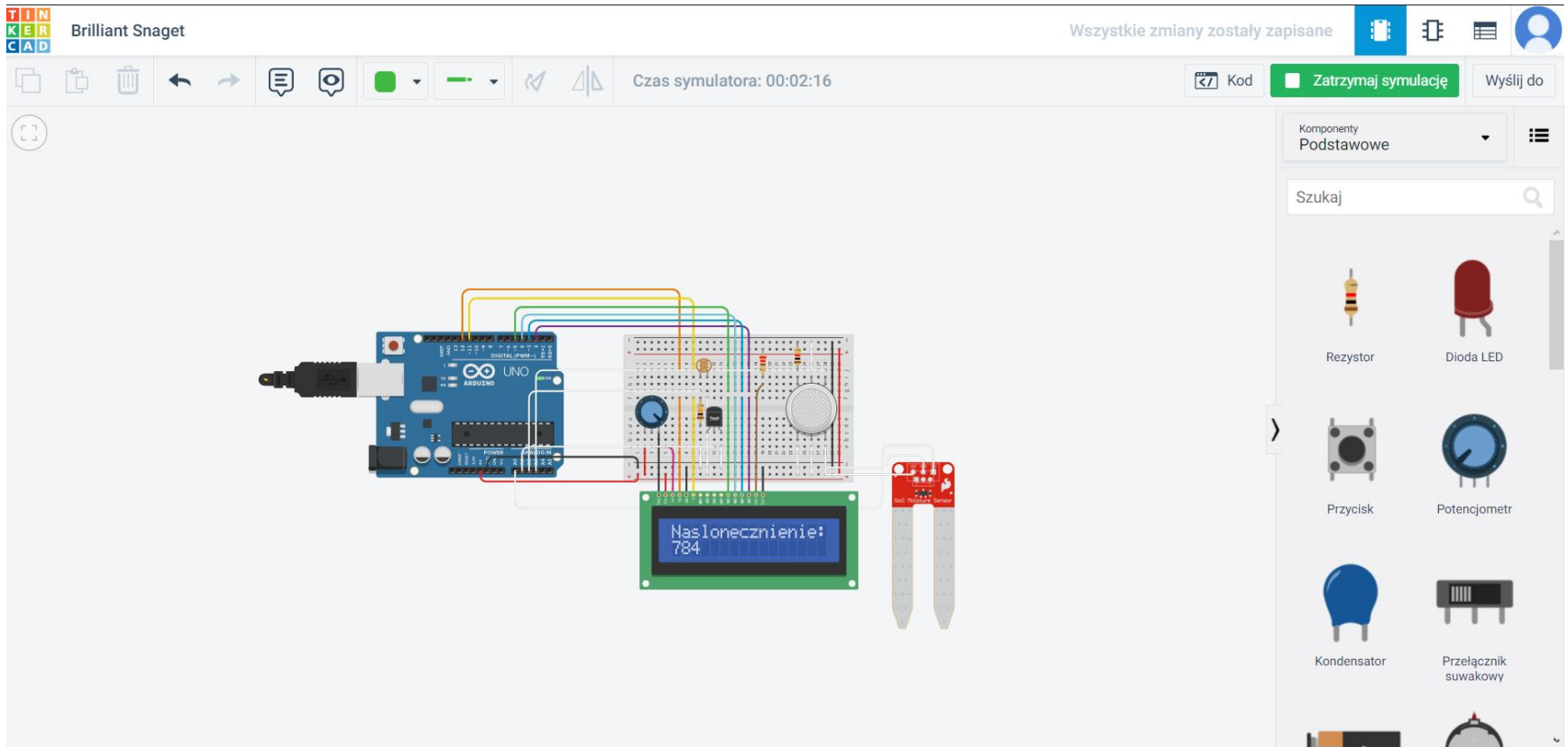
Zadanie 4 – zdjęcie temperatura



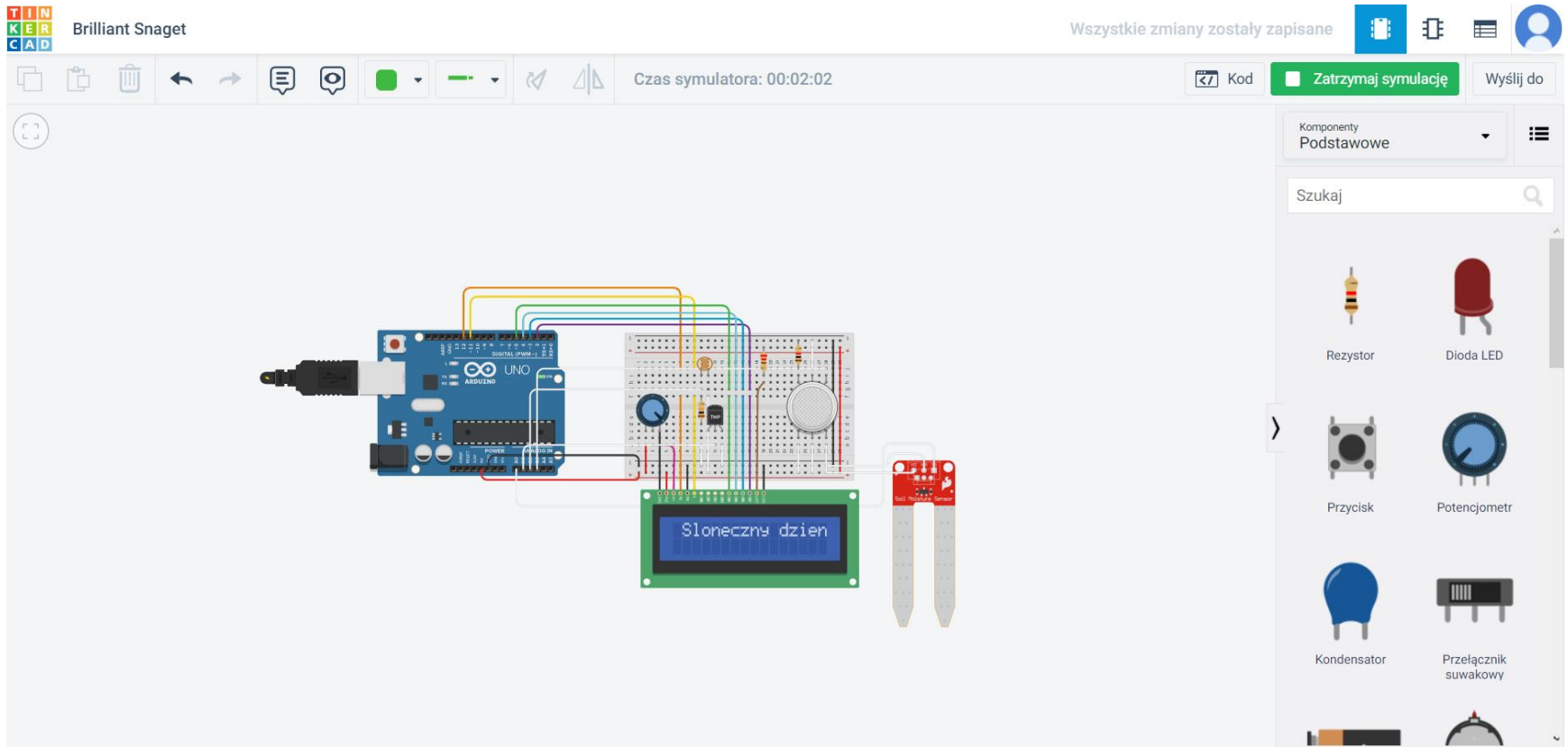
Zadanie 4 – zdjęcie temperatura



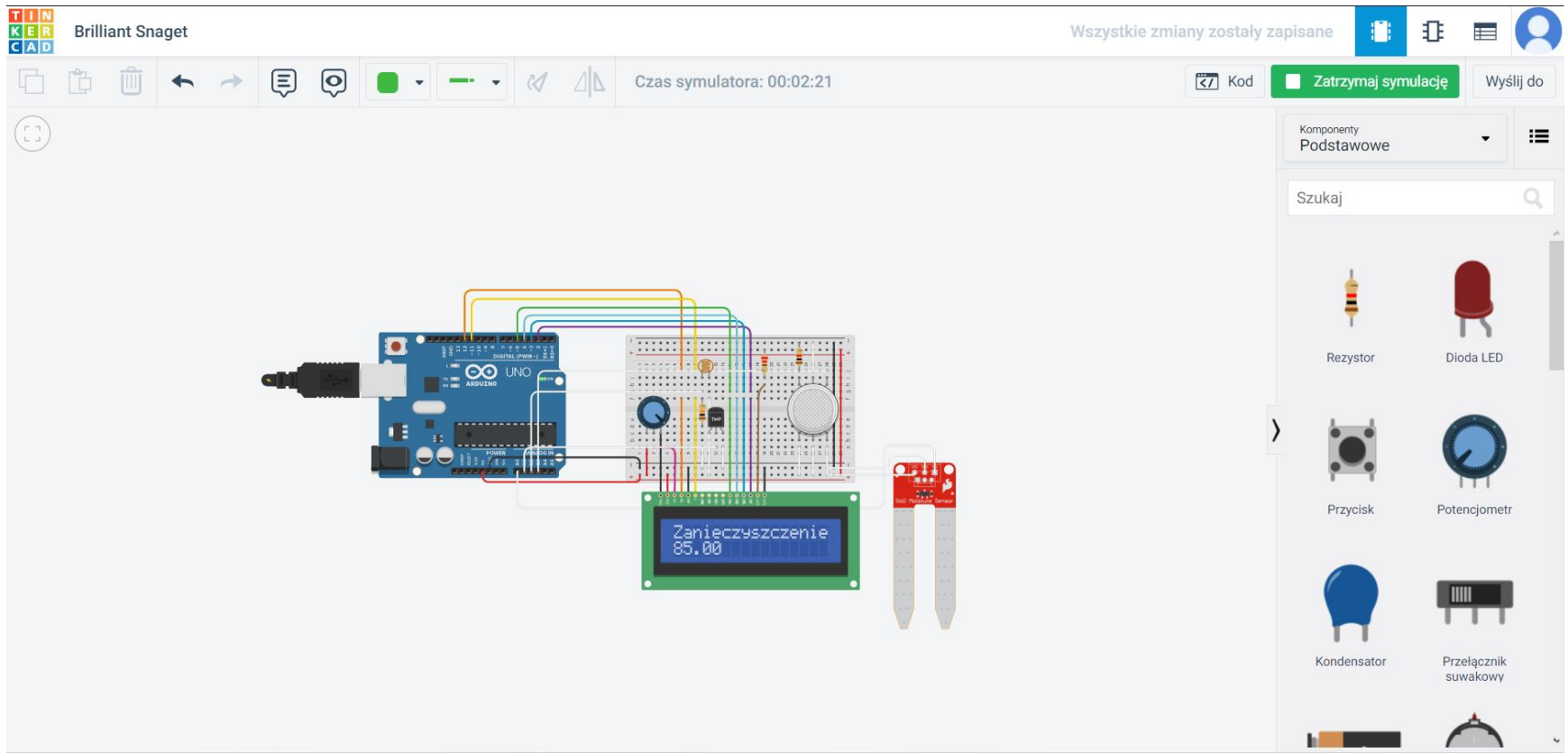
Zadanie 4 – zdjęcie nasłonecznienie



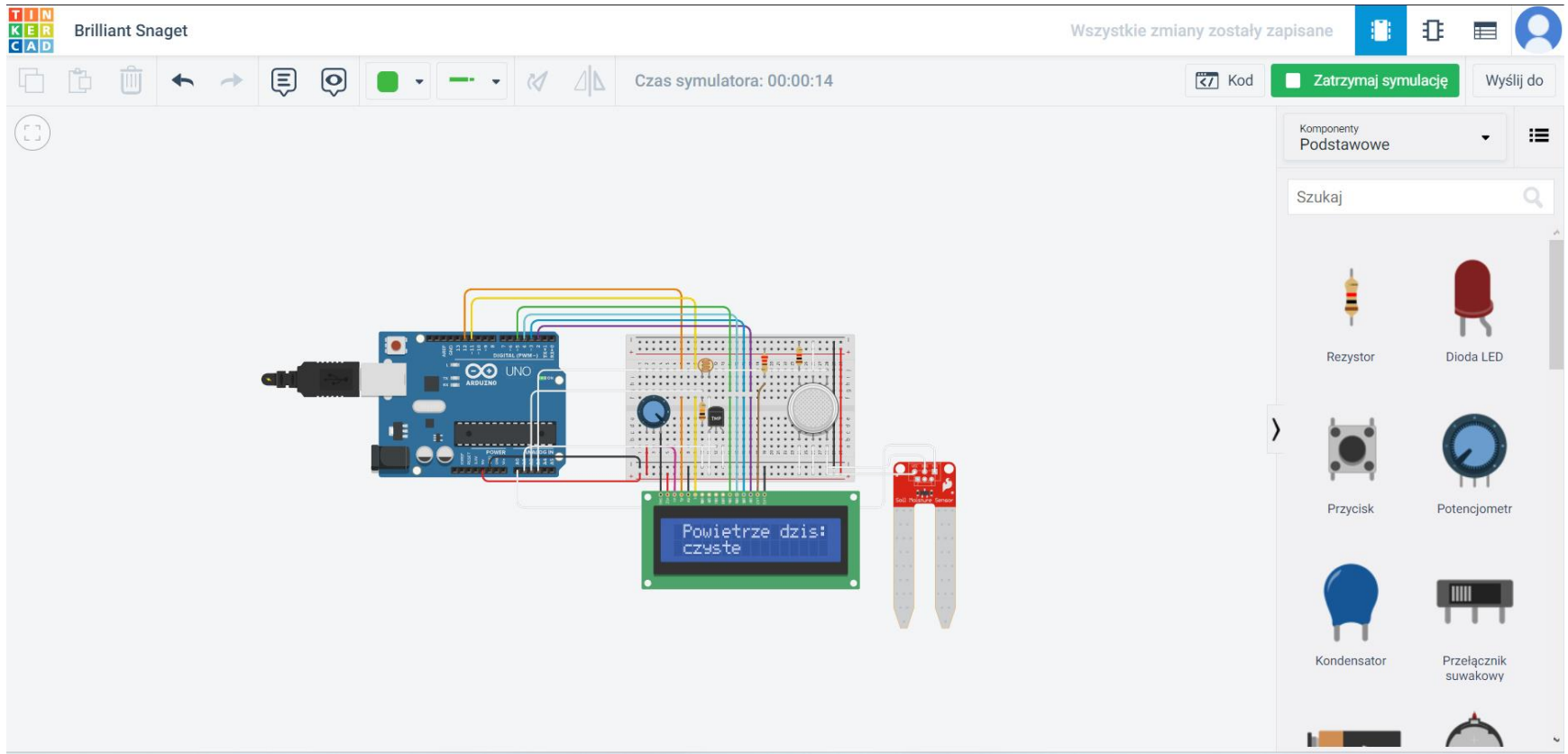
Zadanie 4 – zdjęcie nastonecznienie



Zadanie 4 – zdjęcie czystość powietrza



Zadanie 4 – zdjęcie czystość powietrza



Zadanie 4 - kod

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
  lcd.begin(16,2);
}

void loop()
{

  int hum;
  int kom_hum;

  hum=analogRead(A0);
  hum=map(hum,0,890,5,1000);
  lcd.setCursor(0,0);
  lcd.print("Wilgotnosc % : ");
  lcd.setCursor(0,1);
  lcd.print((float)(hum/10.0));

  delay(2000);

  if(hum >= 75){

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Wilgotny dzien ");
  }
  if(hum <= 5){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Suchy dzien ");
  }

  delay(2000);

  int val=analogRead(A1);
  float cel=(float)val*500.0/1024.0-60.0;
  lcd.setCursor(0, 0);
  lcd.print(" Temperatura (C): ");
  lcd.setCursor(0, 1);
  lcd.print(cel);

  delay(2000);

  if(cel >= 26){
```

```
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Goracy dzien ");
}
if(cel <= 12){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Zimny dzien ");
}

delay(2000);

int light_sensor=A2;
int light_reading;
light_reading = analogRead(light_sensor);

lcd.setCursor(0,0);
lcd.print("Naslonecznienie: ");
lcd.setCursor(0,1);
lcd.print(light_reading);

delay(2000);

if(light_reading >= 450){

lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Sloneczny dzien ");
}
if(light_reading <= 150){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Pochmurny dzien ");
}

delay(2000);

const int gaz = 0;
int gazpin = A3;

float sensorValue;
sensorValue = analogRead(gazpin);

lcd.setCursor(0,0);
lcd.print("Zanieczyszczenie: ");
lcd.setCursor(0,1);
lcd.print(sensorValue);

delay(2000);

if(sensorValue >= 150){

lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Powietrze dzis: ");
lcd.setCursor(0,1);
lcd.print(" brudne ");
}
if(sensorValue <= 100){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Powietrze dzis: ");
lcd.setCursor(0,1);
lcd.print(" czyste ");
}

delay(2000);

}
```